Higher-order Inference in Conditional Random Fields using Submodular Functions

D.Phil Thesis

Optimization for Vision and Learning Group
Department of Engineering Science
University of Oxford



Pankaj Pansari
St Catherine's College

Supervised by
Professor Pawan Mudigonda

HILARY TERM, 2019

Pankaj Pansari
St Catherine's College

Doctor of Philosophy
Hilary Term, 2019

Higher-order Inference in Conditional Random Fields using Submodular Functions

Abstract

Higher-order and dense conditional random fields (CRFs) are expressive graphical models which have been very successful in low-level computer vision applications such as semantic segmentation, and stereo matching. These models are able to capture long-range interactions and higher-order image statistics much better than pairwise CRFs. This expressive power comes at a price though - inference problems in these models are computationally very demanding. This is a particular challenge in computer vision, where fast inference is important and the problem involves millions of pixels.

In this thesis, we look at how submodular functions can help us designing efficient inference methods for higher-order and dense CRFs. Submodular functions are special discrete functions that have important properties from an optimisation perspective, and are closely related to convex functions. We use submodularity in a two-fold manner: (a) to design efficient MAP inference algorithm for a robust higher-order model that generalises the widely-used truncated convex models, and (b) to glean insights into a recently proposed variational inference algorithm which give us a principled approach for applying it efficiently to higher-order and dense CRFs.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy. The work in this thesis was carried out under the guidance of the co-authors of the included, published papers.

Pankaj Pansari
St Catherine's College
January, 2019

Acknowledgements

I would like to express appreciation to my supervisor, Prof. Pawan Mudigonda, for having been a tremendous mentor to me. Thanks a lot for encouraging my research, for always being there when needed, and being supportive in ways big and small.

To Mummy, Papa and Meenu: I owe it all to you. I am grateful for your unconditional love and support.

Prof. Chris Russell was generous with his time, advice, and moral support. It was a wonderful experience to have collaborated with you. I also learned a lot from Brooks Paige at the Alan Turing Institute. My experience at Oxford was enriched by the excellent members of the OVAL group: I learned a lot from discussions with Rudy, Leon, and Alban.

I'll cherish for long the friendships I made at Oxford. Thank you Claudius and Puneet for your wonderful companionship, and motivation. It was a lot of fun to have spent time in the company of Saumya, Sid, Jaya John John, Diane, Ajanthan, Tuan-Anh and Vivek: the ultimate frisbee games and hiking trips were memorable. Thanks to Brijesh, Saurabh and Niharika for staying in touch over the years and always being ready to give me an ear.

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Contributions and Outline 3

1.3 Publications 5

2 Preliminaries **7**

2.1 Graphical Models 8

2.1.1 Probability Theory 9

2.1.2 Bayesian and Markov Networks 14

2.1.3 Conditional Random Fields 21

2.2 Inference 23

2.3 Computational Complexity and NP 26

2.3.1 Computational Problems and Algorithms 26

2.3.2 Polynomial-time Reductions 29

2.3.3 Efficient Certification and Class NP 31

2.4 Convex Optimisation and Duality 34

2.4.1 The Lagrange Dual Function 35

2.4.2 The Lagrange dual problem 39

2.4.3 Max-min characterisation of duality 41

2.4.4 Optimality conditions 42

2.5 Multi-label Submodular functions 45
2.5.1 Submodular set functions 45
2.5.2 Labelling problems of second order 46

3 Related Work **49**
3.1 CRF Modeling for Computer Vision 49
3.2 Inference in CRFs 52
3.2.1 Submodular Functions 52
3.2.2 Marginal Inference 53
3.2.3 MAP Inference 56

4 Truncated Max-of-Convex Models **61**

5 Worst-case Optimal Submodular Extensions for Marginal Estimation **101**

6 LP-based Submodular Extensions for Marginal Estimation **113**

7 Conclusion **141**

8 Errata **143**

CHAPTER 1

Introduction

Many low-level computer vision problems involve labeling each pixel with a discrete label such that the final labeling is in accordance with the observed data and has piecewise smoothness. For instance, in semantic segmentation we seek to label each pixel with an object class, and in stereo matching we use discretised disparity values as labels. Conditional random fields (CRFs) have been widely used for these structured prediction problems [49, 91]. The conditioning here is on the input image (pair of images for stereo matching). Given a CRF, the vision problem may involve either doing (a) *MAP inference* or *energy minimisation*: finding a most like assignment, or (b) *marginal inference*: finding the probability of each pixel taking each of the labels, while marginalising out the random variables representing all other pixels.

There has been a growing interest in higher-order and dense pairwise CRFs since they are much more expressive as compared to sparse pairwise CRFs [2, 19, 98]. However, exact MAP and marginal inference tend to be computationally intractable for these more expressive models for most interesting choices of smoothness priors, such as those that preserve discontinuities. Also, these problems often involve millions of pixels. This motivates the use of approximate

inference methods that are fast and scale well to large sized problems.

Submodularity has turned out to be an important tool for the design of these inference algorithms [39, 52, 94]. Submodular functions are special discrete functions that have important properties from an optimisation perspective, and are closely related to convex functions [4]. This thesis looks at how submodular functions make both MAP and variational inference computationally tractable in dense and higher-order CRFs.

1.1 Motivation

Truncated convex models (TCM) are a special case of pairwise CRFs that have been widely used for low-level vision applications. However, pairwise models significantly limit the expressive power of CRFs and there is often a need for higher-order models in vision to capture useful image statistics. To this end, we propose a higher-order model called truncated max-of-convex model (TMCM) which generalises TCM. TMCM encourages smooth labelings, is discontinuity preserving, and robust to inaccurate clique definitions. Energy minimisation for TMCM is NP-hard. We design a fast approximate minimisation algorithm based on graph-cuts, a special class of submodular functions.

While submodular functions have been well-studied for MAP inference problems, their use for marginal inference has been very limited. An exception is the work of Zhang et al. [110] who construct a submodular extension for the class of Potts energy functions, and use its extended polytope for variational inference. However, their submodular extension was designed in an ad-hoc manner. This had two implications. First, the choice of an extension is important for the accuracy of the method and it was unclear if a better submodular extension exists for Potts model. Second, due to a lack of a principled approach such extensions were not known for more general pairwise and higher-order models.

As a result, the method had limited applicability. Even for Potts model, the method was computationally intractable for dense. In this thesis, we redress these shortcomings.

1.2 Contributions and Outline

The thesis is organised as follows: this introduction forms chapter 1, where we set up the context and motivation for our work. Chapter 2 provides brief overview of computational and mathematical concepts that form the background of our research work. We review related literature in chapter 3 under two broad headings: (a) MAP and variational inference in CRFs, and (b) submodular optimisation in the context of CRF inference. The next three chapters (chapters 4, 5, 6) describe in detail our contributions:

Truncated max-of-convex models: In chapter 4, we propose TMCM which generalises TCM to higher-order CRFs. The unary potentials in our model do not have any restrictions. We define clique potentials as proportional to the sum of the m largest truncated convex distances computed over disjoint pairs of random variables in the clique. We use the term disjoint pairs to imply that the label of no random variable is used more than once to compute the value of the clique potential. The use of convex distance encourages smooth labelings, truncation preserves discontinuity, and the use of $m > 1$ provides robustness to inaccurate clique definitions.

We design a fast algorithm for approximate energy minimisation in TMCM. Note that energy minimisation is NP-hard for TMCM. Our method is based on the majorization-minimization paradigm [90] and on range-expansion [61], the latter having been fruitfully used for energy minimisation in TCM. We design a graph-representable submodular upper-bound to the non-submodular objective

at every iteration, and then minimise this submodular function via a fast *st*-MINCUT solver [9]. Our method provides strong approximation multiplicative bounds for linear and quadratic convex functions.

We test our algorithm on two real-world applications: (a) image denoising and inpainting, and (b) stereo matching. Comparison with baselines is possible only for $m = 1$ and truncated max-of-linear model. In this setting, our method gives lower energy value solution as compared to the baselines. For the sake of completeness, we also show results on synthetic data and using truncated max-of-quadratic model.

Worst-case optimal submodular extensions: In chapter 5, we establish a connection between the variational inference method of Zhang et al. [110] and LP relaxation of the corresponding MAP inference problem. Specifically, we consider a temperature-annealed version of the variational algorithm, and show how it transforms to an LP relaxation of MAP in the limit of the temperature going to 0. We then make use of the tightest LP relaxation for MAP problem in Potts model [42] to prove the worst-case optimality of the submodular extension used in [110]. Furthermore, our results give us a principled way to derive the worst-case optimal extension for the more general hierarchical Potts model.

An important consequence of our results is that we are able to operationalise our method for dense CRF models with Gaussian potentials [53]. We show an equivalence between computing conditional gradient in the variational method and subgradient of the related LP relaxation. This relation enables us to make use of an efficient Gaussian filtering algorithm [2] for fast computation of conditional gradient. Ours is the first efficient method to obtain upper-bounds on the log-partition functions for dense CRFs. We carry out experiments on dense CRFs to perform stereo matching using a standard dataset.

LP-based submodular extensions: The CRF models considered in chapter 5 were pairwise models. Often, these models are not expressive enough to capture image statistics in vision applications, motivating the use of higher-order models. In chapter 6, we extend the variational inference method to a higher-order diversity model. In this model, the unary terms can be arbitrary. Clique potentials are proportional to the number of different labels used in labeling the clique. Our model promotes parsimony, that is, encourages the use of as few diverse labels as possible. As in TMCM, clique weights can be derived using image statistics of the clique, such as intensity variance.

We derive a submodular extension for the higher-order diverse model using an accurate LP relaxation of the corresponding MAP problem. The tightness property of this relaxation has not been well-studied, hence we cannot claim worst-case optimality of our extension as we did in chapter 5. However, the resemblance in form of this LP relaxation to the worst-case optimal Potts relaxation suggests the accuracy of our approach. We use our model and inference method for two applications: (a) semantic segmentation on the MSRC-21 dataset [89] and (b) stereo matching.

Finally, we detail avenues for future research on these topics in chapter 7.

1.3 Publications

The work presented in this thesis has been published as the following papers:

- Chapter 4: P. Pansari and M. P. Kumar. Truncated max-of-convex models. In *CVPR* 2017.
- Chapter 5: P. Pansari, C. Russell and M. P. Kumar. Worst-case optimal submodular extensions for marginal estimation. In *AISTATS* 2018.
- Chapter 6: P. Pansari, C. Russell and M. P. Kumar. LP-based submod-

ular extensions for marginal estimation. *Computer Vision and Image Understanding* 2019.

A note on presentation. This thesis is presented as an *integrated thesis* [1], where the publications are reproduced in the format they were submitted for publication at the respective venues.

[1] https://www.admin.ox.ac.uk/edc/policiesandguidance/policyonresearchdegrees/annexbintegratedthesesguidancefordivisionalboards/

CHAPTER 2

Preliminaries

We explain the computational and mathematical concepts that form the background of our research work. In section 2.1, we introduce graphical models, their representation capacities and their association with probability distributions. To this end, we also briefly discuss relevant concepts from probability theory. Doing so will provide us the context to work with high-order CRFs and their associated energy functions. In this work, we solve MAP (chapter 4) and marginal (chapters 5 and 6) inference problems in CRFs. By showing how these inference methods derive from statistical decision theory, section 2.2 helps us appreciate that we are essentially minimising specific risk functions when we perform MAP and marginal inference. We give a short overview of complexity classes P, NP, NP-complete, and NP-hard for computational problems in section 2.3. This will enable us to understand the nature of the intractable MAP inference problem of TMCM in chapter 4 and the need for an algorithm yielding approximate solutions. In chapters 5 and 6, we need to maximise a linear function with non-negative coefficients on the submodular polyhedron. We are able to do so efficiently in closed-form thanks to convex duality. To appreciate this better, duality theory in convex optimisation and its min-max characterisation is discussed in section

2.4. Since the special properties of submodular function play a central role in this thesis to solve inference problems in high-order CRFs, we end with a short description of submodular function, its multi-label version, and its application in labelling problems of second-order 2.5.

2.1 Graphical Models

Our task is to reason probabilistically about the values of one or more variables, possibly given observations about some others. In order to do so using principled probabilistic reasoning, we need to construct a *joint distribution* over the space of possible assignments to the set of random variables. This type of model allows us to answer a broad range of interesting queries.

Probabilistic graphical models use a graph-based representation as the basis for compactly encoding a complex distribution over a high-dimensional space. There is a dual perspective that one can use to interpret the structure of this graph. From one perspective, the graph is a compact representation of a set of *independencies* that hold in the distribution; these properties take the form "$\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$", denoted by $(\mathbf{X} \perp \mathbf{Y}|\mathbf{Z})$, for some subsets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. The other perspective is that the graph defines a skeleton for compactly representing a high-dimensional distribution: rather than encode the probability of every possible assignment to all of the variables in our domain, we can "break up" the distribution into smaller *factors*, each over a much smaller space of possibilities. We can then define the overall joint distribution as a product of these factors.

It turns out that these two perspectives - the graph as a representation of a set of independencies, and the graph as a skeleton for factorising a distribution - are, in a deep sense, equivalent. The independence properties of the distribution are precisely what allow it to be represented compactly in a factorised form.

Conversely, a particular factorisation of the distribution guarantees that certain independencies hold.

After reviewing relevant concepts from probability theory, we describe two families of graphical representation of distributions. One, called *Bayesian networks*, uses a directed graph. The second, called *Markov networks*, uses an undirected graph. Both representations provide the duality of independencies and factorisation, but they differ in the set of independencies they can encode and in the factorisation of the distribution that they induce. We end this section with a discussion of *Conditional random fields*, a subclass of Markov networks used widely in computer vision. We draw upon the material in [49, 50, 60] for this overview of graphical models.

2.1.1 Probability Theory

Graphical models specify complex probability distributions. In this subsection, we briefly review basic concepts from probability theory.

Event spaces

We define *events* by assuming that there is an agreed upon *space* of possible outcomes, which we denote by Ω. In addition, we assume that there is a set of *measurable events* $\mathcal{S}$ to which we are willing to assign probabilities. Formally, each element $\alpha \in \mathcal{S}$ is a subset of Ω.

Probability theory requires that the event space satisfy three basic properties:

- It contains the *empty event* $\emptyset$, and the trivial event Ω.
- It is closed under union. That is, if $\alpha, \beta \in \mathcal{S}$, then so is $\alpha \cup \beta$.
- It is closed under complementation. That is, if $\alpha \in \mathcal{S}$, then so is $\Omega - \alpha$.

Probability distributions

Definition 1. *A probability distribution P over $(\Omega, \mathcal{S})$ is a mapping from events in $\mathcal{S}$ to real values that satisfies the following conditions:*

- $P(\alpha) \geq 0$ *for all* $\alpha \in \mathcal{S}$.
- $P(\Omega) = 1$.
- *If* $\alpha, \beta \in \mathcal{S}$ *and* $\alpha \cap \beta = \emptyset$, *then* $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$.

Random variables

Our discussion of probability distributions deals with events. We can consider any event from the set of measurable events. The description of events is in terms of sets of outcomes. In many cases, however, it is more natural to consider *attributes* of the outcome. The machinery for discussing attributes and their values in different outcomes are *random variables*. Formally, a random variable is defined by a function that associates with each outcome in Ω a value. For example, *Grade* in an exam can be a random variable, and it is defined by a function f_{Grade} that maps each person in Ω to their grade (say, one of A, B, or C). The event $Grade = A$ is a shorthand for the event $\{\omega \in \Omega : f_{Grade}(\omega) = A\}$. We are going to use uppercase letter X to denote random variable, lowercase letter x to denote value of a random variable, boldface uppercase letter $\mathbf{X}$ to denote a vector of random variables, and boldface lowercase letter $\mathbf{x}$ to denote an assignment to a vector of random variables.

Conditional Probability

The *conditional probability* of an event β given another event α has occurred is defined as

$$P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)}. \tag{2.1}$$

The conditional probability given an event (say α) satisfies the properties of definition 1, and thus it is a probability distribution by its own right. Hence, we can think of the conditioning operation as taking one distribution and returning another over the same probability space.

Chain rule and Bayes' rule

From the definition of the conditional distribution, we immediately see that

$$P(\alpha \cap \beta) = P(\alpha)P(\beta|\alpha). \tag{2.2}$$

This equality is known as the *chain rule* of conditional probabilities. More generally, if $\alpha_1, \ldots, \alpha_k$ are events, then we can write

$$P(\alpha_1 \cap \cdots \cap \alpha_k) = P(\alpha_1)P(\alpha_2|\alpha_1) \cdots P(\alpha_k|\alpha_1 \cup \cdots \cup \alpha_{k-1}). \tag{2.3}$$

In other words, we can express the probability of a combination of several events in terms of the probability of the first, the probability of the second given the first, and so on. It is important to notice that we can expand this expression using any order of events - the result will remain the same.

Another immediate consequence of the definition of conditional probability is *Bayes' rule*

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}. \tag{2.4}$$

Bayes' rule is important in that it allows us to compute the conditional probability $P(\alpha|\beta)$ from the "inverse" conditional probability $P(\beta|\alpha)$.

Independence

We usually expect the conditional probability $P(\alpha|\beta)$ to be different from $P(\alpha)$ since learning that event β has occurred changes our probability of α. However,

in some situations equality can occur, so that $P(\alpha|\beta) = P(\alpha)$. That is, learning that β occurs did not change our probability of α.

Definition 2. *We say that an event α is independent of event β in P, denoted by $P \models (\alpha \perp \beta)$, if $P(\alpha|\beta) = P(\alpha)$ or if $P(\beta) = 0$.*

We can also provide an alternative definition for the concept of independence:

Proposition 1. *A distribution P satisfies $(\alpha \perp \beta)$ if and only if $P(\alpha \cap \beta) = P(\alpha)P(\beta)$.*

As an immediate consequence of this alternative definition, we see that independence is a symmetric notion. That is, $(\alpha \perp \beta)$ implies $(\beta \perp \alpha)$.

Conditional independence

While independence is a useful property, it is not often that we encounter two independent events. A more common situation is when two events are independent given an additional event.

Definition 3. *We say that an event α is conditionally independent of event β given event γ in P, denoted by $P \models (\alpha \perp \beta|\gamma)$, if $P(\alpha|\beta \cap \gamma) = P(\alpha|\gamma)$ or if $P(\beta \cap \gamma) = 0$.*

An alternative definition of conditional independence is the following:

Proposition 2. *P satisfies $(\alpha \perp \beta|\gamma)$ if and only if $P(\alpha \cap \beta|\gamma) = P(\alpha|\gamma)P(\beta|\gamma)$.*

Querying a distribution

We are going to be very interested in using a joint probability distribution over multiple random variables to answer queries of interest.

1. **Probability queries** The most common query type is the *probability query*. Such a query consists of two parts:

- the *evidence*: a subset $\mathbf{E}$ of random variables in the model, and an instantiation $\mathbf{e}$ to these variables;
- the *query variables*: a subset $\mathbf{Y}$ of random variables in the model.

Our task is to compute

$$P(\mathbf{Y}|\mathbf{E} = \mathbf{e}), \tag{2.5}$$

that is, the *posterior probability distribution* over the values $\mathbf{y}$ of $\mathbf{Y}$, conditioned on the fact that $\mathbf{E} = \mathbf{e}$. This expression can also be viewed as the marginal over $\mathbf{Y}$ in the distribution we obtain by conditioning on $\mathbf{e}$.

2. **MAP queries** A second important type of task is that of finding a high-probability joint assignment to some subset of variables. The simplest example of this type of task is the *MAP* query (short for *maximum a-posteriori* query), whose aim is to find the *MAP assignment* - the most likely assignment to all the (non-evidence) variables. More precisely, if we let $\mathbf{W} = \mathcal{X} - \mathbf{E}$, our task is to find the most likely assignment to the variables in $\mathbf{W}$ given the evidence $\mathbf{E} = \mathbf{e}$:

$$MAP(\mathbf{W}|\mathbf{e}) = \arg\max_{\mathbf{w}} P(\mathbf{w}, \mathbf{e}). \tag{2.6}$$

We note that the assignment where each variable individually picks its most likely value can be quite different from the most likely joint assignment to all variables simultaneously.

Example 1. *Let us consider a two node chain $X \to Y$ where X and Y are both binary-valued. Let us assume that:*

x^0	x^1
0.6	*0.4*

X	y^0	y^1
x^0	*0.5*	*0.5*
x^1	*0.9*	*0.1*

We see that MAP(A) = x^0 since $P(x^0) > P(x^1)$. However, the most

probable joint assignment MAP(A, B) = (x^1, y^0) with probability 0.4 × 0.9 = 0.36. The most likely assignment containing x^0 has a smaller value of $0.6 \times 0.5 = 0.3$. *Thus, we see that* $\arg\max_{A,B} P(A, B) \neq (\arg\max_A P(A), \arg\max_B P(B))$.

3. **Marginal MAP queries** Given an assignment to the evidence variables $\mathbf{E} = \mathbf{e}$, let $\mathbf{Y}$ be the query variables and $\mathbf{Z} = \mathcal{X} - \mathbf{Y} - \mathbf{E}$, the *marginal MAP* task is to compute

$$MAP(\mathbf{Y}|\mathbf{e}) = \arg\max_{\mathbf{Y}} \sum_{\mathbf{Z}} P(\mathbf{Y}, \mathbf{Z}|\mathbf{e}). \tag{2.7}$$

Thus, marginal MAP queries contain both summations and maximisations; in a way, it contains elements of both a conditional probability query and a MAP query.

2.1.2 Bayesian and Markov Networks

Our discussion of probability theory enabled us to lay the groundwork for studying graphical models. In this subsection, we discuss in detail the two types of graphical models and the independencies they represent. Since our work in this thesis is more focused on Markov network, we give only a very brief description of Bayesian network.

Bayesian Networks

The core of the Bayesian network (BN) representation is a directed acyclic graph (DAG) $\mathcal{G}$. The nodes of $\mathcal{G}$ are the random variables in our domain and the edges correspond, intuitively, to the direct influence of one node on another. One way to view this graph is as a data structure that provides the skeleton for representing the joint distribution compactly in a factorised way.

Let $\mathcal{G}$ be a BN graph over the variables $X_1, \ldots, X_n$. Each random variable X_i in the network has an associated *conditional probability distribution (CPD)* or *local probabilistic model*. The CPD for X_i, given its parent in the graph (denoted by $\mathbf{Pa}_{X_i}$), is $P(X_i|\mathbf{Pa}_{X_i})$.

Definition 4. *Let $\mathcal{G}$ be a Bayesian network graph over the variables $X_1, \ldots, X_n$. We say that a distribution $P_{\mathcal{B}}$ factorises according to $\mathcal{G}$ if $P_{\mathcal{B}}$ can be expressed as a product*

$$P_{\mathcal{B}}(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\mathbf{Pa}_{X_i}). \tag{2.8}$$

Formally, a Bayesian network is a pair $(\mathcal{G}, \theta_{\mathcal{G}})$ where $P_{\mathcal{B}}$ factorises over $\mathcal{G}$, and where $P_{\mathcal{B}}$ is specified as set of CPDs associated with $\mathcal{G}$'s nodes, denotes by $\theta_{\mathcal{G}}$.

The equation above is called the *chain rule for Bayesian networks.* It gives us a method for determining the probability of any complete assignment to the set of random variables: any entry in the joint can be computed as a product of factors, one for each variable. Each factor represents a conditional probability of the variable given its parent in the network. Figure 2.1 shows an example of a Bayesian network and the resulting factorisation of the probability distribution.

Markov Networks

The second common class of probabilistic graphical models is called a *Markov network* or *Markov random field* (MRF). The models are based on undirected graphical models. These models are useful in modelling a variety of phenomena where one cannot naturally ascribe a directionality to the interaction between variables. Furthermore, the undirected models also offer a different and often simpler perspective than directed models, both in terms of the independence structures and the inference task.

As in a Bayesian network, the nodes in the undirected graph of a Markov

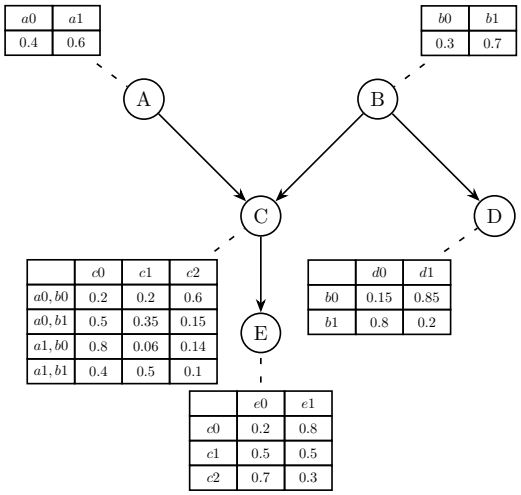


Figure 2.1: An example of a Bayesian network. The distribution can be represented as a product of conditional probability distributions specified by the tables. The form of these distributions is described by the edges in the graph. In this case, the joint probability distribution factorises as $P(A, B, C, D, E) = P(E|C)P(C|A, B)P(D|B)P(A)P(B)$.

network graph $\mathcal{H}$ represent the variables, and the edges correspond to some notion of direct probabilistic interaction between the neighbouring variables.

The remaining question is how to parametrize this undirected graph. The graph structure represents the qualitative properties of the distribution. To represent the distribution, we need to associate the graph structure with a set of parameters, in the same way that CPDs were used to parametrize the directed graph structure. However, the parametrization of Markov networks is not as intuitive as that of Bayesian networks, as the factors do not correspond either to probabilities or to conditional probabilities.

The most general parametrization is a *factor*:

Definition 5. *Let $\boldsymbol{D}$ be a set of random variables. We define a factor to be a function from $Val(\mathbf{D})$ to $\mathbb{R}^+$, where $Val(\mathbf{D})$ denotes an assignment to $\boldsymbol{D}$.*

Definition 6. *Let $\mathcal{H}$ be a Markov network structure. A distribution $P_{\mathcal{H}}$ factorizes*

over $\mathcal{H}$ if it is associated with

- *a set of subsets $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_m$, where each $\mathbf{D}_i$ is a complete subgraph of $\mathcal{H}$;*
- *factors $\phi_1[\mathbf{D}_1], \phi_2[\mathbf{D}_2], \ldots, \phi_m[\mathbf{D}_m]$,*

such that

$$P_{\mathcal{H}}(X_1, \ldots, X_n) = \frac{1}{Z} P'_{\mathcal{H}}(X_1, \ldots, X_n), \tag{2.9}$$

where

$$P'_{\mathcal{H}}(X_1, \ldots, X_n) = \phi_1[\mathbf{D}_1] \times \phi_2[\mathbf{D}_2] \times \cdots \times \phi_m[\mathbf{D}_m] \tag{2.10}$$

is an unnormalized measure and

$$Z = \sum_{X_1, \ldots, X_n} P'_{\mathcal{H}}(X_1, \ldots, X_n) \tag{2.11}$$

is a normalizing constant called the partition function. A distribution P that factorises over $\mathcal{H}$ is also called a Gibbs distribution over $\mathcal{H}$. (The naming convention has roots in statistical physics.)

We note that this definition is quite similar to the factorisation definition for Bayesian networks: there, we decomposed the distribution as a product of CPDs. In the case of Markov networks, the only constraint on the parameters in the factor is non-negativity.

As every complete subgraph is a subset of some clique, we can simplify the parametrization by introducing factors only for cliques, rather than for subcliques. More precisely, let $\mathbf{C}_1, \ldots, \mathbf{C}_k$ be the cliques in $\mathcal{H}$. We can parametrize P using a set of factors $\phi_1[\mathcal{C}_1], \ldots, \phi_2[\mathcal{C}_k]$. These factors are called *clique potentials* (in the context of the Markov network $\mathcal{H}$). We emphasize that the clique potentials do not represent the marginal probabilities of the variables in their scope. Figure 2.2 shows the cliques in a simple Markov network, the conditional

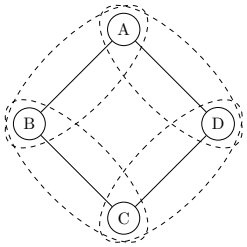


Figure 2.2: In this simple Markov network, the cliques are the pairs {A, B}, {B, C}, {C, D}, {D, A}. The conditional independencies implied by this network are $(B \perp D|A, C)$ and $(A \perp C|B, D)$. The probability distribution for this network factorises as $P(A, B, C, D) = \phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(D, A)/Z$, where the partition function is $Z = \sum_{A,B,C,D} \phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(D, A)$.

independencies implied by the network, and the resulting factorisation of the joint probability distribution.

It is often useful to consider a slightly different way of specifying potentials, by using a logarithmic transformation. We can rewrite a factor $\phi[\mathbf{D}]$ as

$$\phi[\mathbf{D}] = \exp(-\epsilon[\mathbf{D}]), \tag{2.12}$$

where $\epsilon[\mathbf{D}] = -ln\phi[\mathbf{D}]$ is called an *energy function*. The use of the word "energy" derives from statistical physics, where the probability of a physical state (e.g., a configuration of a set of electrons), depends inversely on its energy.

In this logarithmic representation, we have

$$P_{\mathcal{H}}(X_1, \ldots, X_n) \propto \exp\left[-\sum_{i=1}^{m} \epsilon_i[\mathbf{D}_i]\right]. \tag{2.13}$$

The above representation ensures that the probability distribution is positive.

A subclass of MRFs that arises in many contexts is that of *pairwise Markov networks*, representing distributions where all of the factors are over single variables or pairs of variables. More precisely, a pairwise Markov network over

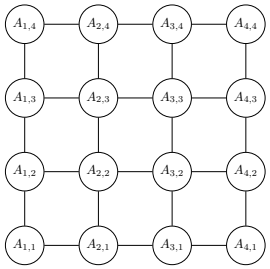


Figure 2.3: A pairwise Markov network structured as a grid.

a graph $\mathcal{H}$ is associated with a set of *node potentials* $\phi[\mathbf{X}_i] : i = 1, \ldots, n$ and a set of *edge potentials* $\phi[\mathbf{X}_i, \mathbf{X}_j] : (X_i, X_j) \in \mathcal{H}$. The overall distribution is the normalized product of all of the potentials (both node and edge). Figure 2.3 shows a pairwise Markov network structured as a grid, which is especially useful in computer vision applications.

Independencies in Markov Networks

The graph structure in MRF can be viewed as encoding two sets of independence assumptions: the local Markov properties and the global Markov properties.

The local Markov properties are associated with each node in the graph and are based on the intuition that we can block all influences on a node by conditioning on its immediate neighbours.

Definition 7. *Let $\mathcal{H}$ be an undirected graph. Then for each node $X \in \mathcal{X}$, the Markov blanket of X, denoted by $\mathcal{N}_{\mathcal{H}}(X)$, is the set of neighbours of X in the graph (those that share an edge with X). We define the local Markov independencies associated with $\mathcal{H}$ to be*

$$\mathcal{I}_l(\mathcal{H}) = \{(X \perp \mathcal{X} - \{X\} - \mathcal{N}_{\mathcal{H}}(X) | \mathcal{N}_{\mathcal{H}}(X)) : X \in \mathcal{X}\}. \tag{2.14}$$

In other words, the Markov assumptions state that X is independent of the rest

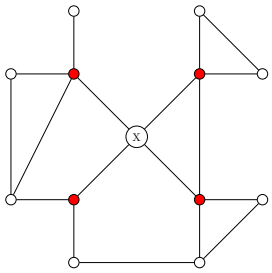


Figure 2.4: In a Markov network, a node X is independent from the rest of the graph given its Markov blanket (that is, its neighbours, shown here in red.)

of the nodes in the graph given its immediate neighbours.

Figure 2.4 illustrates the concept of a Markov blanket. To define the global Markov properties, we begin by defining active paths in undirected graphs.

Definition 8. *Let $\mathcal{H}$ be a Markov network structure, and $X_1 - - \cdots X_k$ be a path in $\mathcal{H}$. Let $\mathbf{E} \subseteq \mathcal{X}$ be a set of observed variables. The path $X_1 - - \cdots X_k$ is active given $\mathbf{E}$ if none of the X_i's, $i = 1, \ldots, k$, is in $\mathbf{E}$.*

Using this notation, we define a notion of *separation* in the undirected graph.

Definition 9. *We say that a set of nodes $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{H}$, denoted by $\text{sep}_{\mathcal{H}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$, if there is no active path between any node $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ given $\mathbf{Z}$. We define the global Markov assumptions associated with $\mathcal{H}$ to be*

$$\mathcal{I}_g(\mathcal{H}) = \{(\mathbf{X} \perp \mathbf{Y}|\mathbf{Z}) : \operatorname*{sep}_{\mathcal{H}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})\}. \tag{2.15}$$

Figure 2.5 illustrates the concept of separation in a Markov network. We can make a connection between the local Markov properties and the global Markov properties. Our first theorem asserts that a Gibbs distribution satisfies the global independencies associated with the graph.

Theorem 1. *Let P be a distribution over $\mathcal{X}$, and $\mathcal{H}$ a Markov network structure over $\mathcal{X}$. If P is a Gibbs distribution over $\mathcal{H}$, then all the local Markov properties*

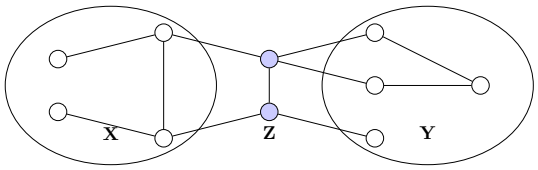


Figure 2.5: The set of nodes **Z** (shown in blue) separates the set of nodes **X** and **Y** in the above graph.

associated with $\mathcal{H}$ hold in P.

The other direction, which goes from the global independence properties of a distribution to its factorization, is known as *Hammersley-Clifford theorem*. This direction only holds under the additional assumption that P is a positive distribution - that is, every possible joint instantiation has probability greater than 0 in P.

Theorem 2. ***Hammersley-Clifford theorem****: Let P be a positive distribution over $\mathcal{X}$, and $\mathcal{H}$ a Markov network graph over $\mathcal{X}$. If all of the independence constraints implied by $\mathcal{H}$ hold in P, then P is a Gibbs distribution over $\mathcal{H}$.*

This result shows that, for positive distributions, the global Markov property implies that the distribution factorizes according to the network structure. Thus, for this class of distributions, we have that a distribution P factorizes over a Markov network $\mathcal{H}$ if and only if all of the independencies implied by $\mathcal{H}$ hold in P. The positivity assumption is necessary for this result to hold.

2.1.3 Conditional Random Fields

So far, we have described the Markov network representation as encoding a joint distribution over $\mathcal{X}$. The same undirected graph representation and parametrization can also be used to encode a *conditional distribution* $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{Y}$ is a set of *target variables* and $\mathbf{X}$ is a (disjoint) set of *observed variables*. This representation is generally called a *conditional random field* (CRF).

Formally, a CRF is an undirected graph whose nodes correspond to $\mathbf{Y} \cup \mathbf{X}$. This graph is parametrized in the same way as an ordinary Markov network, as a set of factors $\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)$. However, rather than encoding the distribution $P(\mathbf{Y}, \mathbf{X})$, we view it as representing the conditional distribution $P(\mathbf{Y}|\mathbf{X})$. To have the network structure and parametrization correspond naturally to a conditional distribution, we want to avoid representing a probabilistic model over $\mathbf{X}$. We therefore disallow potentials that involve only variables in $\mathbf{X}$. Figure 2.6 illustrates an example of a simple chain-structured CRF.

Definition 10. *A conditional random field is an undirected graph $\mathcal{H}$ whose nodes correspond to $\mathbf{X} \cup \mathbf{Y}$; the network is annotated with a set of factors $\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)$ such that each $\mathbf{D}_i \not\subseteq \mathbf{X}$. The network encodes a conditional distribution as follows:*

$$\begin{aligned} P(\mathbf{Y}|\mathbf{X}) &= \frac{1}{Z(\mathbf{X})}\tilde{P}(\mathbf{Y}|\mathbf{X}) \\ \tilde{P}(\mathbf{Y}|\mathbf{X}) &= \prod_{i=1}^{m} \phi_i(\mathbf{D}_i) \\ Z(\mathbf{X}) &= \sum_{\mathbf{Y}} \tilde{P}(\mathbf{Y}|\mathbf{X}). \end{aligned} \tag{2.16}$$

Two variables in $\mathcal{H}$ are connected by an (undirected) edge whenever they appear together in the scope of some factor.

The fact that we avoid encoding the distribution over the variables in $\mathbf{X}$ is one of the main strengths of the CRF representation. This flexibility allows us to incorporate into the model a rich set of observed variables whose dependencies may be quite complex. It also permits us to use domain knowledge in order to define a rich set of features characterizing our domain without worrying about modelling their joint distribution.

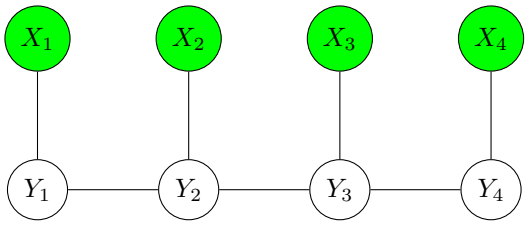


Figure 2.6: Linear chain-structured CRF, where the feature variables $\mathbf{X}$(observed) are denoted by green colour. $\mathbf{Y}$ is the set of output variables.

2.2 Inference

The goal of probabilistic modelling in computer vision is to make predictions about unobserved properties for a given data instance, this process being called *inference*. We would like these predictions to be *as good as possible*, which shows the necessity for a measure of *prediction quality*. In this section, we give a brief overview of inference from the viewpoint of *statistical decision theory*. Our discussion here is based on [76].

We denote the set of output variables with $\mathbf{Y}$ and the overall *output domain* by $\mathcal{Y}$. The output domain is the product of individual variable domains $\mathcal{Y}_i$, so that we have $\mathcal{Y} := \times_{i \in \mathbf{Y}} \mathcal{Y}_i$. In many practical models have $\mathcal{Y}_i := \mathcal{L}$, a single set $\mathcal{L}$ of labels. Let $d(\mathbf{X}, \mathbf{Y})$ denote the probability distribution of the data for the problem we are trying to solve, which we factor into the conditional probability distribution $d(\mathbf{Y}|\mathbf{X})$, and a data prior $d(\mathbf{X})$. Furthermore, let $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ be a *loss function* where $\Delta(\mathbf{y}, \mathbf{y}')$ specifies the cost of predicting $\mathbf{y}'$ for a sample when the correct label is $\mathbf{y}$. For any sample $\mathbf{x} \in \mathcal{X}$ and prediction rule $f : \mathcal{X} \to \mathcal{Y}$ we can measure the quality of predicting the label $f(\mathbf{x})$ by the *expected loss* of

this decision:

$$\mathcal{R}_f^{\Delta}(\mathbf{x}) = \mathbb{E}_{\mathbf{y}\sim d(\mathbf{Y}|\mathbf{X})}\Delta(\mathbf{y}, f(\mathbf{x})) \tag{2.17}$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}} d(\mathbf{y}|\mathbf{x})\Delta(\mathbf{y}, f(\mathbf{x})). \tag{2.18}$$

Assuming that the parametrized distribution $p(\mathbf{y}|\mathbf{x};\mathbf{w})$ reflects $d(\mathbf{y}|\mathbf{x})$ well, we obtain the following approximation

$$\mathcal{R}_f^{\Delta}(\mathbf{x}) = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{Y}|\mathbf{X})}\Delta(\mathbf{y}, f(\mathbf{x})) \tag{2.19}$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x};\mathbf{w})\Delta(\mathbf{y}, f(\mathbf{x})). \tag{2.20}$$

The two important inference problems frequently encountered in computer vision applications, MAP inference and probabilistic inference, can be interpreted as the optimal decision rules for two specific loss functions.

Arguably, the most common loss function for classification tasks is the $0/1$ loss, $\Delta_{0/1}(\mathbf{y},\mathbf{y}') = [\mathbf{y} \neq \mathbf{y}']$, that is, $\Delta_{0/1}(\mathbf{y},\mathbf{y}') = 0$ for $\mathbf{y} = \mathbf{y}'$, and $\Delta_{0/1}(\mathbf{y},\mathbf{y}') = 1$ otherwise. Computing its expected loss we obtain

$$\mathcal{R}_f^{0/1}(\mathbf{x}) = 1 - p(\mathbf{Y} = f(\mathbf{x})|\mathbf{x};\mathbf{w}). \tag{2.21}$$

This expression is minimized by choosing $f(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x};\mathbf{w})$ for every $\mathbf{x} \in \mathcal{X}$, hence the optimal prediction rule in this case is MAP inference.

Definition 11. *CRF MAP inference: Given a CRF graph, parametrization, and weight vector* $\mathbf{w}$*, and given the observation* $\mathbf{x}$*, we need to find the state* $\mathbf{y}^* \in \mathcal{Y}$ *of maximum probability,*

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{Y} = \mathbf{y}|\mathbf{x};\mathbf{w}) = \arg\min_{\mathbf{y}\in\mathcal{Y}} E(\mathbf{y}|\mathbf{x};\mathbf{w}). \tag{2.22}$$

Another popular choice of loss function for structured prediction tasks is the *Hamming loss*: $\Delta(\mathbf{y}, \mathbf{y}') = \frac{1}{|\mathbf{Y}|} \sum_{i \in \mathbf{Y}} [y_i \neq y'_i]$. It is sometimes more intuitive than the 0/1-loss. For example, in pixel-wise image segmentation, the Hamming loss is proportional to the number of misclassified pixels, whereas the 0/1-loss assigns the same cost to every labelling that is not pixel-by-pixel identical to the intended one. For the Hamming loss, the expected loss takes the form

$$\mathcal{R}_f^H(\mathbf{x}) = 1 - \frac{1}{|\mathbf{Y}|} \sum_{i \in \mathbf{Y}} p(Y_i = f(\mathbf{x})_i | \mathbf{x}; \mathbf{w}), \tag{2.23}$$

which is minimized by predicting with $f(\mathbf{x})_i = \arg\max_{y_i \in \mathcal{Y}_i} p(Y_i = y_i | \mathbf{x}; \mathbf{w})$, that is, by maximizing the marginal distribution of each node. To evaluate this prediction rule, we rely on probabilistic inference.

Definition 12. *CRF Probabilistic inference: Given a CRF graph, parametrization, and weight vector* $\mathbf{w}$*, and given the observation* $\mathbf{x}$*, we need to find the value of the log-partition function*

$$\log Z(\mathbf{x}; \mathbf{w}) = \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp(-E(\mathbf{y} | \mathbf{x}; \mathbf{w})), \tag{2.24}$$

and marginal distribution for each variable $p(Y_i = y_i | \mathbf{x}, \mathbf{w}), \forall\, Y_i \in \mathbf{Y}, \forall\, y_i \in \mathcal{Y}_i$.

When using the Hamming loss in conjunction with a distribution $p(\mathbf{y}|\mathbf{x}; \mathbf{w})$, we use the per-variable marginal distribution $p(\mathbf{Y}_i = y_i | \mathbf{x}; \mathbf{w})$ to make a single joint prediction $\mathbf{y}$ for all variables. This produces a small loss in expectation, but might produce a labelling $\mathbf{y}$ that does not have a low energy. Furthermore, if we allow infinite energy values in the factors - in effect making some configurations infeasible - then a labelling produced from variable marginals may even have infinite energy.

Comparing the two inference problems, the MAP inference provides us with the mode of $p(\mathbf{y}|\mathbf{x}, \mathbf{w})$, whereas the result μ of the probabilistic inference describes

the marginal distributions of $p(\mathbf{y}|\mathbf{x}, \mathbf{w})$ for each factor. Both inference problems are known to be NP-hard for general graphs and factors [87] but can be tractable in special cases.

2.3 Computational Complexity and NP

We encounter discrete computational problems in diverse fields - among them optimisation, artificial intelligence, combinatorics, logic. Due to their prevalence, we are very interested in finding efficient algorithms for solving them. On the one hand, this search has been unfruitful for many fundamental problems and on the other, researchers have been unable to prove the non-existence of efficient procedures. In the face of this formal ambiguity, a lot of progress has been made in characterising the complexity of many problems relative to each other and some complexity classes have been proposed. Once we prove that a new problem belongs to one of these classes, we obtain valuable information about the practical likelihood of finding an efficient algorithm for it.

We begin this section with a discussion of two major types of computational problems - decision and optimisation. Then, a useful tool, polynomial-time reduction, for studying the relative computational complexity is introduced. We end with brief overviews of the complexity classes P, NP, NP-complete, and NP-hard. Our treatment of this subject here closely follows that in [43].

2.3.1 Computational Problems and Algorithms

The two types of fundamental computational problems are:

1. *Decision problem* - a problem with a yes/no answer. Some examples of decision problems are:

    - *SAT* Problem: We are given a set X of n boolean variables $x_1, x_2, \ldots,$

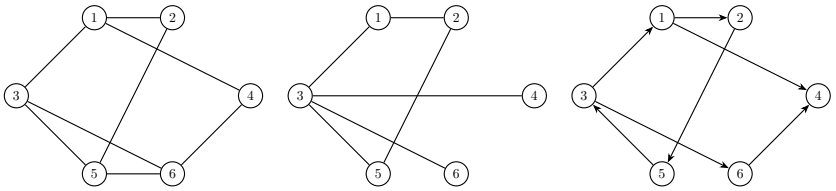


Figure 2.7: (a) An undirected graph in which Hamiltonian cycle/path exist. The node sequence (1, 2, 5, 3, 6, 4, 1) is a Hamiltonian cycle. The node sequence (1, 2, 5, 3, 6, 4) is a Hamiltonian path. (b) An undirected graph in which a Hamiltonian cycle does not exist. Although one can traverse the cycle (1, 2, 5, 3), it is not a Hamiltonian cycle because it does not include all nodes. (c) A directed graph in which a Hamiltonian cycle does not exist.

x_n. A *clause* is a disjunction of distinct terms

$$t_1 \vee t_2 \vee \cdots \vee t_l$$

where each $\{t_i \in x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n\}$. Given a collection of clauses $C_1, C_2, \ldots, C_k$, each containing l terms, does there exist an assignment of boolean values to variables in X, such that all clauses are simultaneously satisfied (that is, evaluate to 1)? When $l = 3$, we have a special case of the SAT problem called 3-SAT.

- Hamiltonian Cycle Problem: Given a graph $G = (V, E)$, directed or undirected, does there exist a cycle that visits each node exactly once? We note that a Hamiltonian cycle contains all nodes of the graph. Figure 2.7 shows an undirected graph with a Hamiltonian cycle. It also gives examples of an undirected and a directed graph in which no Hamiltonian cycle exists.
- Hamiltonian Path Problem: Given a graph $G = (V, E)$, directed or undirected, does there exist a path that visits each node exactly once?

2. *Optimisation problem* - a problem that maximises/minimises an objec-

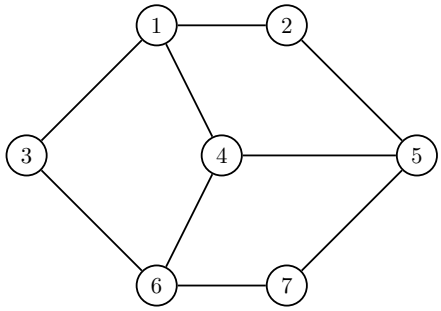


Figure 2.8: A graph in which the node set {1, 5, 6} forms the minimum vertex cover and the node set {2, 3, 4, 7} forms the maximum independent set.

tive function based on some constraints. Some examples of optimisation problems are:

- Travelling Salesman Problem: We are given a list of n cities $v_1, v_2, \ldots,$ v_n. The distance from city v_i to city v_j is given by a non-negative number $d(v_i, v_j)$. In the general case, the distances may be asymmetric, that is $d(v_i, v_j) \neq d(v_j, v_i)$, and may not satisfy the triangle inequality. In what sequence $v_{i_1}, v_{i_2}, \ldots, v_{i_n}$ should one visit the cities so as to minimise the total distance $\sum_j d(v_{i_j}, v_{i_{j+1}}) + d(v_{i_n}, v_{i_1})$?
- Minimum Vertex Cover: Given a graph $G = (V, E)$, a *vertex cover* is a set of vertices $S \subseteq V$ such that every edge $e \in E$ has at least one end in S. What is the vertex cover of smallest size? Figure 2.8 shows the minimum vertex cover of an example graph.
- Maximum Independent Set: Given a graph $G = (V, E)$, an *independent set* is a set of vertices $S \subseteq V$ such that no two vertices in S are joined by an edge. What is the independent set of largest size? Figure 2.8 shows the maximum independent set of an example graph.

Sometimes we need to construct decision versions of optimisation problems. These are posed in terms of existence of solutions with given upper (for minimi-

sation problems) or lower (for maximisation problems) bounds. For example, the decision versions of the above three optimisation problems are:

- Travelling Salesman Decision: Given a set of distances on n cities, and a bound D, does there exist a sequence in which one can visit the cities such that the tour length is at most D?
- Vertex Cover: Given a graph G and a number k, does G contain a vertex cover of size at most k?
- Independent Set: Given a graph G and a number k, does G contain an independent set of size at least k?

The complexity classes NP and NP-complete, which we shall discuss in the following subsections, will be applicable only to decision problems, whereas the class NP-hard applies to computational problems in general (including decision problems).

The input to a computational problem will be encoded as a finite binary string s. The length of a string s is denoted by $|s|$. A decision problem X will be identified with the *set* of strings on which the answer is "yes". An algorithm A for a decision problem takes in an input string s and returns the value "yes" or "no" - we denote the value returned by $A(s)$. A is said to have *solved* the problem X if for all strings s, $A(s) = yes$ if and only if $s \in X$. We say that A has a *polynomial running time* if there is a polynomial function $p(.)$ so that for every string s, A halts on s in at most $\mathcal{O}(p(|s|))$ steps.

2.3.2 Polynomial-time Reductions

The notion of reducibility helps us to quantify the relative complexities of problems. For instance, we know that finding shortest paths between all pairs of vertices is a more general problem than finding the shortest pair from source to

terminal in a graph. We could easily use the algorithm which solves the former, general problem to solve the latter. We want to formally express statements like, "Problem X is at least as hard as problem Y". We do this using the notion of *reduction*: we argue that, if we had a "black box" capable of solving X, then we could also solve Y.

Let us assume that we have a black box that can solve instances of a problem X - if the input for the problem X is written down, the black box will return the correct yes/no answer. We say that "Y is polynomial-time reducible to X", written as $Y \leq_P X$ if arbitrary instances of problem Y can be solved using a polynomial number of standard computational steps plus a polynomial number of calls to the black box that solves problem X.

An important consequence of our definition of $\leq_P$ is this: suppose $Y \leq_P X$ and that there exists a polynomial-time algorithm to solve X. Then we can replace our black box with this polynomial-time algorithm for X. Since $Y \leq_P X$, this implies that the algorithm for Y now involves a polynomial number of steps plus a polynomial number of calls to a subroutine that runs in polynomial time - effectively, it has become a polynomial-time algorithm. Hence,

Proposition 3. *Suppose $Y \leq_P X$. If X can be solved in polynomial time, then Y can be solved in polynomial time.*

However, to establish the computational intractability of various problems, the contrapositive form of proposition 3 is useful:

Proposition 4. *Suppose $Y \leq_P X$. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.*

We use proposition 4 in the following way to establish the hardness of a given new problem X: if we have a problem Y that is known to be hard and we show that $Y \leq_P X$, then the hardness has "spread" to X; X must be hard or else it

could be used to solve Y. Some of the previously discussed problems follow this relation:

- Independent Set $\leq_P$ Vertex Cover
- Vertex Cover $\leq_P$ Independent Set
- 3-SAT $\leq_P$ Independent Set

The proofs of the above relations can be found in [43].

2.3.3 Efficient Certification and Class NP

Efficient Certification

Finding a solution to a decision problem and *checking* a proposed solution are very different tasks. For many hard problems such as *Independent Set* or *3-SAT*, we do not know of a polynomial-time algorithm to find solutions; however, one can easily check a proposed solution in polynomial-time. For instance, given a proposed solution S for *Independent Set*, we need only check that there does not exist an edge between any pair of vertices in S. Given a proposed *3-SAT* truth assignment, we check if all the clauses are satisfied. However, not all problems admit efficient certification. As an example, it is hard to design a certifier to prove that a given *3-SAT* problem is *not* satisfiable; one cannot easily come up with an "evidence" that can be checked in polynomial time to convince us that the problem is unsatisfiable.

To formalize this, we now introduce the idea of a checking algorithm that takes in the input string s, plus a separate certificate string t that contains the evidence that s is a yes instance of X. We say that B is an *efficient certifier* for a problem X if the following two properties are satisfied:

1. B is a polynomial-time algorithm that takes in two inputs s and t.

2. There is a polynomial function p so that for every string s, $s \in X$ if and only if there exists a string t such that $|t| \leq p(|s|)$ and $B(s,t) = yes$.

Hence, B does not actually try to decide whether an input s belongs to X on its own. Rather, it efficiently evaluates proposed proofs t that s belongs to X - provided they are not too long - and it is a correct algorithm in the weak sense that s belongs to X if and only if there exists a proof that will convince it.

Classes P and NP

Class P is the set of all problems X for which there exists an algorithm A with a polynomial running time that solves X. A few examples of problems in this class are: sorting an array, finding minimum st-cut in graphs, finding closest pair of points in a plane, and finding minimum spanning-tree in graphs.

Class NP is the set of problems for which there exists an efficient certifier. [1] We can immediately observe the following:

Proposition 5. *$P \subseteq NP$.*

Proof. Consider a problem $X \in$ P, with a polynomial-time algorithm A to solve it. To show that $X \in$ NP, we need to find an efficient certifier B for X.

We design B as follows: given an instance s (and the evidence t that is ignored), the certifier B simply returns $A(s)$. Clearly B has polynomial running time since A does, hence it is efficient. If a string $s \in X$, then for every t of length at most $p(|s|)$, we have $B(s,t) = \texttt{yes}$. On the other hand, if $s \notin X$, then for every t of length at most $p(|s|)$, we have $B(s,t) = \texttt{no}$. □

All of the decision problems introduced in subsection 2.3.1 belong to NP. We mention, for some of these problems, how an efficient certifier can make use of a certificate string t:

[1] The act of searching for a string t that will cause an efficient certifier to accept the input s is often viewed as non-deterministic search over the space of possible proofs t; for this reason, NP was named as an acronym for "non-deterministic polynomial time".

- For the 3-SAT Problem, the certificate t is an assignment of truth values to the variables; the certifier B evaluates the given set of clauses with respect to this assignment.

- For the Independent Set Problem, the certificate t is the identity of a set of at least k vertices; the certifier B checks that, for these vertices, no edge joins any pair of them.

- For the Hamiltonian Cycle Problem, the certificate t is a sequence of vertices forming the Hamiltonian Cycle; the certifier B checks that this sequence forms a valid cycle - the graph can be traversed in this cycle - and that all vertices are present exactly once in the sequence, except the start and end vertices which are the same.

However, it is not known if there is a problem in NP that does not belong to P - in other words, whether P = NP. The general consensus is that P $\neq$ NP, but there is not a lot of hard technical evidence for it. A huge amount of effort has gone into failed attempts at designing polynomial time algorithms for hard problems in NP. Besides, it is believed unlikely that there could be general transformation from the task of *checking* a solution to the much harder task of actually *finding* a solution.

Class NP-complete

In the absence of progress on the P = NP question, people have turned to a related but more approachable question: What are the hardest problems in NP? Polynomial time reducibility gives us a way of addressing this question and gaining insight into the structure of NP.

Arguably the most natural way to define a "hardest" problem X is via the following two properties: (i) $X \in NP$, and (ii) for all $Y \in NP$, $Y \leq_P X$. In

other words, we require that every problem in NP can be reduced to X. Such a problem X is called an *NP-complete* problem.

It follows from the definition, that an NP-complete problem X can be solved in polynomial-time if and only if P = NP. A consequence of this fact is that if there is *any* problem in NP that cannot be solved in polynomial time, then no NP-complete problem can be solved in polynomial time. This fact helps to further reinforce our use of the term *hardest*.

Class NP-hard

NP-hardness is the defining property of a class of problems that are informally "at least as hard as the hardest problems in NP". Precisely, a problem X is *NP-hard* if there exists some NP-complete problem Y that can be reduced to it in polynomial time, that is, $Y \leq_P X$.

The class NP-hard is applicable for both decision and optimisation problems. The optimisation problems Travelling Salesman, Minimum Vertex Cover, Maximum Independent Set, introduced in subsection 2.3.1, are some examples. NP-hard problems do not have to be elements of the class NP; decision problems that do not have efficient certifiers are also included in this category. For instance, the decision problem of determining whether a given SAT instance is *not* satisfiable is not in NP, but belongs to NP-hard.

Figure 2.9 pictorially shows the relationship among the complexity classes that have been discussed in this section.

2.4 Convex Optimisation and Duality

In this section, we provide brief overview of relevant concepts from convex optimisation: we introduce the Lagrangian, the Langrangian dual function, and the Lagrange dual problem. We examine closely the relation between the primal

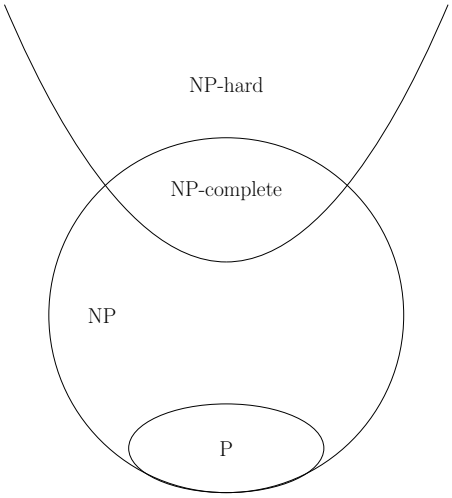


Figure 2.9: A diagram representing relations among complexity classes P, NP, NP-complete, and NP-hard under the assumption P $\neq$ NP.

and dual problems. Our treatment of the subject derives from [8]. We note that in this section we overload the notation for vector $\mathbf{x}$; while in sections 2.1 and 2.2, we used $\mathbf{x}$ to denote a labelling, here we use it to denote the vector of optimisation variables.

2.4.1 The Lagrange Dual Function

The Lagrangian

Let us consider an optimisation problem in the standard form:

$$\begin{aligned} &\text{minimize} && f_0(\mathbf{x}) \\ &\text{subject to} && f_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ &&& h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p, \end{aligned} \tag{2.25}$$

with variable $\mathbf{x} \in \mathbb{R}^n$. We assume its domain $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\ f_i \bigcap \cap_{i=0}^{p} \mathbf{dom}\ h_i$ is non-empty, and denote the optimal value of problem (2.25) by p^*. We do not

assume the problem (2.25) is convex.

The basic idea in Langrangian duality is to take the constraints in problem (2.25) into account by augmenting the objective function with a weighted sum of the constraint functions. We define the *Lagrangian* $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ associated with the problem (2.25) as

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}), \tag{2.26}$$

with **dom** $L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$. We refer to λ_i as the *Lagrange multiplier* associated with the i-th inequality constraint $f_i(\mathbf{x}) \leq 0$; similarly we refer to ν_i as the Lagrange multiplier associated with the i-th equality constraint $h_i(\mathbf{x}) = 0$. The vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are called the dual variables or Lagrange multiplier vectors associated with the problem (2.25).

The Lagrange dual function

We define the *Lagrange dual function* (or just *dual function*) $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ as the minimum value of the Lagrangian over $\mathbf{x}$: for $\boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\nu} \in \mathbb{R}^p$,

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{x \in \mathcal{D}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \right). \tag{2.27}$$

When the Lagrangian is unbounded below in $\mathbf{x}$, the dual function takes on the value $-\infty$. Since the dual function is the point wise infimum of a family of affine functions of $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, it is concave, even when the problem (2.25) is not convex.

Lower bounds on optimal value

The dual function yields lower bounds on the optimal value p^* of the problem (2.25): For any $\boldsymbol{\lambda} \succeq \mathbf{0}$ and any $\boldsymbol{\nu}$ we have

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^* \tag{2.28}$$

This important property is easily verified. Suppose $\tilde{\mathbf{x}}$ is a feasible point for the problem (2.25), that is, $f_i(\tilde{\mathbf{x}}) \leq 0$ and $h_i(\tilde{\mathbf{x}}) = 0$, and $\boldsymbol{\lambda} \succeq \mathbf{0}$. Then we have

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^{p} \nu_i h_i(\tilde{\mathbf{x}}) \leq 0, \tag{2.29}$$

since each term in the first sum is non-positive, and each term in the second sum is zero, and therefore

$$L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\tilde{\mathbf{x}}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^{p} \boldsymbol{\nu}_i h_i(\tilde{\mathbf{x}}) \leq f_0(\tilde{\mathbf{x}}). \tag{2.30}$$

Hence

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \leq L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\tilde{\mathbf{x}}). \tag{2.31}$$

Since $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\tilde{\mathbf{x}})$ holds for every feasible point $\tilde{\mathbf{x}}$, the inequality (2.28) follows. The inequality (2.28) holds, but is vacuous, when $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = -\infty$. The dual function gives a non-trivial lower bound on p^* only when $\boldsymbol{\lambda} \succeq \mathbf{0}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu}) \in$ **dom** g, that is, $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \geq -\infty$. We refer to the pair $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ with $\boldsymbol{\lambda} \succeq \mathbf{0}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu}) \in$ **dom** g as *dual feasible*.

Linear Approximation Interpretation

The Lagrangian and lower bound property can be given a simple interpretation, based on a linear approximation of the indicator functions of the sets $\{0\}$ and

-$\mathbb{R}_+$. We first rewrite the original problem (2.25) as an unconstrained problem,

$$\min f_0(\mathbf{x}) + \sum_{i=1}^{m} I_-(f_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x})), \tag{2.32}$$

where $I_- : \mathbb{R} \to \mathbb{R}$ is the indicator function for the non-positive real,

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0, \end{cases} \tag{2.33}$$

and similarly I_0 is the indicator function of $\{0\}$. In the formulation (2.32), we can think of functions $I_-(u)$ and $I_0(u)$ as "brick wall" or "infinitely" hard displeasure functions for constraints being violated.

Now we replace the function $I_-(u)$ with the linear function $\lambda_i \cdot u$, where $\lambda_i \geq 0$, and the function $I_0(u)$ with $\nu_i \cdot u$. The objective becomes the Lagrangian function $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$, and the dual function value $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is the optimal value of the problem

$$\min L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i}^{p} \nu_i h_i(\mathbf{x}). \tag{2.34}$$

In this formulation, we use a linear or "soft" displeasure function for constraint violation in place of I_- and I_0. For an inequality constraint, our displeasure is zero when $f_i(\mathbf{x}) = 0$, and is positive when $f_i(\mathbf{x}) > 0$ (assuming $\lambda_i > 0$); our displeasure grows as the constraint becomes "more violated".

Clearly the approximation of the indicator function $I_-(u)$ with a linear function $\lambda_i \cdot u$ is rather poor. But the linear function is at least an underestimator of $I_-(u)$. Since $\lambda_i \cdot u \leq I_-(u)$ and $\nu_i \cdot u \leq I_0(u)$ for all u, we see that the dual function yields a lower bound on the optimal value of the original problem.

2.4.2 The Lagrange dual problem

For each pair $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ with $\boldsymbol{\lambda} \succeq \mathbf{0}$, the Lagrange dual function gives us a lower bound on the optimal value p^* of the optimisation problem (2.25). Thus we have a lower bound that depends on some parameters $\boldsymbol{\lambda}, \boldsymbol{\nu}$. We are interested in the best lower bound that can be obtained from the Lagrange dual function.

This leads to the optimisation problem

$$\begin{array}{ll} \text{maximize} & g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} & \boldsymbol{\lambda} \succeq \mathbf{0} \end{array} \tag{2.35}$$

This problem is called the *Lagrange dual problem* associated with problem (2.25). In this context the original problem (2.25) is sometimes called the *primal problem*. Hence, the term *dual feasible* is used for the pair $\boldsymbol{\lambda}, \boldsymbol{\nu}$ with $\boldsymbol{\lambda} \succeq \mathbf{0}$ and $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) > -\infty$ since $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is feasible for the dual problem (2.35). We refer to $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ as *dual optimal* or *optimal Lagrange multipliers* if they are optimal for the problem (2.35).

The Lagrange dual problem (2.35) is a convex optimisation problem, since the objective to be maximized is concave and the constraint is convex. This is the case whether or not the primal problem (2.25) is convex.

Weak duality

The optimal value of the Lagrange dual problem, which we denote by d^*, is, by definition, the best lower bound on p^* that can be obtained from the Lagrange dual function. In particular, we have the simple but important inequality

$$d^* \leq p^*, \tag{2.36}$$

which holds even if the original problem is not convex. This property is called *weak duality*. The weak duality inequality (2.36) holds when d^* and p^* are infinite. We refer to the difference $p^* - d^*$ as the *optimal duality gap* of the original problem, since it gives the gap between the optimal primal problem and the best (that is, greatest) lower bound on it that can be obtained from the Lagrange dual function. The optimal duality gap is always non-negative. The bound (2.36) can sometimes be used to find a lower bound on the optimal value of a problem that is difficult to solve, since the dual problem is always convex, and in many cases can be solved efficiently, to find d^*.

Strong duality and Slater's constraint qualification

If the equality

$$d^* = p^* \tag{2.37}$$

holds, that is, the optimal duality gap is zero, then we say that *strong duality* holds. This means that the best bound that can be obtained from the Lagrange dual function is tight.

Strong duality does not, in general, hold. But if the primal problem (2.25) is convex, that is, of the form

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & A\mathbf{x} = \mathbf{b}, \end{array} \tag{2.38}$$

with $f_0, \ldots, f_m$ convex, we usually (but not always) have strong duality. Conditions called *constraint qualifications* establish conditions on the problem, beyond convexity, under which strong duality holds.

One simple constraint qualification is *Slater's condition*: there exists an

$\mathbf{x} \in$ relint $\mathcal{D}$ such that

$$f_i(\mathbf{x}) < 0, \quad i = 1, \ldots, m, \quad A\mathbf{x} = \mathbf{b}. \tag{2.39}$$

Such a point is sometimes called *strictly feasible*, since the inequality constraints hold with strict inequalities. Slater's theorem states that strong duality holds, if Slater's condition holds (and the problem is convex).

2.4.3 Max-min characterisation of duality

It is possible to express the primal and dual optimisation problems in a form that is more symmetric. To simplify the discussion we assume here that there are no equality constraints; the results are easily extended to cover them.

First we note that

$$\sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) \right) \tag{2.40}$$

$$= \begin{cases} f_0(\mathbf{x}) & f_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m \\ \infty & \text{otherwise.} \end{cases} \tag{2.41}$$

We can express the optimal value of the primal problem as

$$p^* = \inf_{\mathbf{x}} \sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}). \tag{2.42}$$

By the definition of the dual function, we also have

$$d^* = \sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}). \tag{2.43}$$

Thus weak duality can be expressed as the inequality

$$\sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) \leq \inf_{\mathbf{x}} \sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}), \tag{2.44}$$

and strong duality as the equality

$$\sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \inf_{\mathbf{x}} \sup_{\boldsymbol{\lambda} \succeq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}), \tag{2.45}$$

Strong duality means that the order of the minimisation over $\mathbf{x}$ and the maximisation over $\boldsymbol{\lambda} \succeq \mathbf{0}$ can be switched without affecting the result.

In fact, the inequality (2.44) does not depend on any properties of L: we have

$$\sup_{\mathbf{z} \in \mathcal{Z}} \inf_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}, \mathbf{z}) \leq \inf_{\mathbf{w} \in \mathcal{W}} \sup_{\mathbf{z} \in \mathcal{Z}} f(\mathbf{w}, \mathbf{z}) \tag{2.46}$$

for any $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ (and any $\mathcal{W} \subseteq \mathbb{R}^n$ and $\mathcal{Z} \subseteq \mathbb{R}^m$). This general inequality is called the *max-min inequality*. When equality holds, that is,

$$\sup_{\mathbf{z} \in \mathcal{Z}} \inf_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}, \mathbf{z}) = \inf_{\mathbf{w} \in \mathcal{W}} \sup_{\mathbf{z} \in \mathcal{Z}} f(\mathbf{w}, \mathbf{z}) \tag{2.47}$$

we say that f (and $\mathcal{W}$ and $\mathcal{Z}$) satisfy the *strong max-min property* or the *saddle-point* property. Of course, the strong max-min property holds only in special cases, for example, when $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the Lagrangian of a problem for which strong duality is achieved, $\mathcal{W} = \mathbb{R}^n$, and $\mathcal{Z} = \mathbb{R}^m_+$.

2.4.4 Optimality conditions

Optimality certificate

We do not assume problem (2.25) to be convex, unless otherwise stated. If we can find a dual feasible $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, we establish a lower bound on the optimal

value of the primal problem: $p^* \geq g(\boldsymbol{\lambda}, \boldsymbol{\nu})$. Thus a dual feasible point $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ provides a *proof* or *certificate* that $p^* \geq g(\boldsymbol{\lambda}, \boldsymbol{\nu})$. Strong duality means there exist arbitrarily good certificates.

Dual feasible points allow us to bound how suboptimal a given feasible point is, without knowing the exact value of p^*. Indeed, if $\mathbf{x}$ is primal feasible and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is dual feasible, then

$$f_0(\mathbf{x}) - p^* \leq f_0(\mathbf{x}) - g(\boldsymbol{\lambda}, \boldsymbol{\nu}). \tag{2.48}$$

In particular, this establishes that $\mathbf{x}$ is ϵ-suboptimal, with $\epsilon = f_0(\mathbf{x}) - g(\boldsymbol{\lambda}, \boldsymbol{\nu})$. It also establishes that $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is ϵ-suboptimal for the dual problem. We refer to the gap between primal and dual objectives,

$$f_0(\mathbf{x}) - g(\boldsymbol{\lambda}, \boldsymbol{\nu}), \tag{2.49}$$

as the *duality gap* associated with the primal feasible point $\mathbf{x}$ and dual feasible point $(\boldsymbol{\lambda}, \boldsymbol{\nu})$. A primal dual feasible pair $\mathbf{x}, (\boldsymbol{\lambda}, \boldsymbol{\nu})$ localizes the optimal value of the primal and dual problems to an interval:

$$p^* \in [g(\boldsymbol{\lambda}, \boldsymbol{\nu}), f_0(\mathbf{x})], \quad d^* \in [f_0(\mathbf{x}), g(\boldsymbol{\lambda}, \boldsymbol{\nu})], \tag{2.50}$$

the width of which is the duality gap.

If the duality gap of the primal dual feasible pair $\mathbf{x}, (\boldsymbol{\lambda}, \boldsymbol{\nu})$ is zero, that is, $f_0(\mathbf{x}) = g(\boldsymbol{\lambda}, \boldsymbol{\nu})$, then $\mathbf{x}$ is primal optimal and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is dual optimal. We can think of $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ as a certificate that proves that $\mathbf{x}$ is optimal. Similarly, we can think of $\mathbf{x}$ as a certificate that proves $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is dual optimal.

Complementary slackness

Suppose that the primal and dual optimal values are attained and equal (so, in particular, strong duality holds). Let $\mathbf{x}^*$ be a primal optimal and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ be a dual optimal point. This means that

$$\begin{aligned} f_0(\mathbf{x}^*) &= g(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) \\ &= \inf_{\mathbf{x}\in\mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i^* h_i(\mathbf{x}) \right) \\ &\leq f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}^*) + \sum_{i=1}^{p} \nu_i^* h_i(\mathbf{x}^*) \\ &\leq f_0(\mathbf{x}^*). \end{aligned}$$

The last inequality follows from $\boldsymbol{\lambda}^* \geq 0, f_i(\mathbf{x}^*) \leq 0$, $i = 1, \ldots, m$, and $h_i(\mathbf{x}^*) = 0, i = 1, \ldots, p$. We conclude that $\mathbf{x}^*$ is a minimizer of $L(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ over $\mathbf{x}$.

Another important conclusion is that

$$\sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}^*) = 0. \tag{2.51}$$

Since each term in this sum is non-positive, we conclude that

$$\lambda_i^* f_i(\mathbf{x}^*) = 0, \quad i = 1, \ldots, m. \tag{2.52}$$

This condition is known as *complementary slackness*; it holds for any primal optimal $\mathbf{x}^*$ and any dual optimal $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ (when strong duality holds). It means that the i-th optimal Lagrange multiplier is zero unless the i-th constraint is active at the optimum.

2.5 Multi-label Submodular functions

Submodular functions form a special class of discrete functions and efficient algorithms exist for their minimisation. This fact is often used to design approximate algorithms for intractable discrete optimisation problems. Moreover, submodular functions have important connections with convex optimisation theory [4]. In this section, we formally define submodular functions and the related set-functions - modular and super-modular. We then discuss labelling problems and see how the notion of submodularity applies to them.

2.5.1 Submodular set functions

Let us consider a set $V = \{1, \ldots, p\}$, $p > 0$ and its power set 2^V (that is, set of all subsets) which is of cardinality 2^p. We also consider a real-valued set-function $F : 2^V \to \mathbb{R}$ such that $F(\varnothing) = 0$.

Definition 13. ***(Submodular function)*** *A set-function $F : 2^V \to \mathbb{R}$ is submodular if and only if, for all subsets $A, B \subseteq V$, we have: $F(A) + F(B) \geq F(A \cup B) + F(A \cap B)$.*

A set-function F is *supermodular* if its opposite, that is, $A \mapsto -F(A)$ is submodular. A set-function that is both submodular and supermodular is called a *modular* function.

Definition 14. ***(Modular function)*** *A set-function $F : 2^V \to \mathbb{R}$ such that $F(\varnothing) = 0$ is modular if and only if there exists $\mathbf{s} \in \mathbb{R}^p$ such that $F(A) = \sum_{k \in A} s_k$.*

From now on we denote by the vector $\mathbf{s}$ the modular set-function defined as $\mathbf{s}(A) = \sum_{k \in A} s_k = \mathbf{s}^T \mathbf{1}_A$, where $\mathbf{1}_A \in \mathbb{R}^p$ is the indicator vector of the set A. Modular functions essentially play for sets the same role as linear functions do for continuous spaces.

2.5.2 Labelling problems of second order

We discuss the concept of energy functions for labelling problems, and its relation to the notion of submodularity. Our discussion is based on the work of [85]. A labelling problem is defined by means of the following components:

$G = (\mathcal{X}, \mathcal{E})$	an directed graph with set of nodes $\mathcal{X}$ and the set of edges $\mathcal{E} \subseteq \mathcal{X} \times \mathcal{X}$. Nodes and edges are denoted by $a \in \mathcal{X}$ and $e = (a, b) \in \mathcal{E}$ respectively;
$\mathcal{L}$	a finite set of labels;
$(W, \oplus, \otimes)$	a semiring with a set of elements W and operations $\oplus$ and $\otimes$;
$\theta_a : \mathcal{L} \to W$	unary node potential for node X_a, that assigns a value from W to each label in that node;
$\theta_{ab} : \mathcal{L} \times \mathcal{L} \to W$	edge potential for edge $(a, b) \in \mathcal{E}$, that assigns a value from W to each pair of labels in that edge;

Let $\mathbf{x} : \mathcal{X} \to \mathcal{L}$ be a labelling that maps the set of nodes into the set of labels; $\mathbf{x}_a$ denotes the label of the node X_a. The general computational task for labelling problems is to find the quantity

$$\bigoplus_{\mathbf{x} \in \mathcal{L}} \left[\bigotimes_{X_a \in \mathcal{X}} \theta_a(\mathbf{x}_a) \otimes \bigotimes_{(a,b) \in \mathcal{E}} \theta_{ab}(\mathbf{x}_a, \mathbf{x}_b) \right]. \tag{2.53}$$

In computer vision, we are mainly interested in these special cases (with respect to the semiring):

- **M**inSum: $(\mathbb{R}, \min, +)$. MinSum problems, also known as "Energy minimisation", are very important in machine learning. The task is an optimisation

problem of the form

$$\min_{\mathbf{x}\in\mathcal{L}} \left[\sum_{X_a\in\mathcal{X}} \theta_a\left(\mathbf{x}_a\right) + \sum_{(a,b)\in\mathcal{E}} \theta_{ab}\left(\mathbf{x}_a,\mathbf{x}_b\right) \right]. \tag{2.54}$$

For MinSum problem, we say that we are minimising the energy E (the sum of unary and edge potentials) over the set of all labels. MinSum problems can be interpreted as MAP decisions for corresponding CRFs defined by the graph and the unary and edge potentials.

- **S**umProd: $(\mathbb{R},\ +\ ,\ \times)$. When we are interested in obtaining marginal a-posteriori probabilities, we solve SumProd problem of the form

$$\sum_{\mathbf{x}\in\mathcal{L}} \left[\prod_{X_a\in\mathcal{X}} \theta_a\left(\mathbf{x}_a\right) \cdot \prod_{(a,b)\in\mathcal{E}} \theta_{ab}\left(\mathbf{x}_a,\mathbf{x}_b\right) \right]. \tag{2.55}$$

Submodularity for MinSum problems of second order

Let the set of labels $\mathcal{L}$ be a completely ordered set. It means that for each pair of labels l_1 and l_2 we define a relation "above/below", that is, $l_2 \geq l_1$ means "the label l_2 is located above or is equal to the label l_1". Let l_1 and l_2 be two labels in a node X_a, so that $l_2 \geq l_1$. Similarly let l'_1 and l'_2 be two labels in another node X_b, so that $l'_2 \geq l'_1$. For an energy function E to be submodular, we let the unary potentials be arbitrary, and impose a condition only on the edge potentials. An edge potential θ_{ab} is submodular if

$$\theta_{ab}(l_1,l'_1) + \theta_{ab}(l_2,l'_2) \leq \theta_{ab}(l_1,l'_2) + \theta_{ab}(l_2,l'_1) \tag{2.56}$$

holds for each such four-tuple $l_2 \geq l_1, l'_2 \geq l'_1$. The energy function E is called submodular if all edge potentials are submodular.

We can derive an equivalent definition of submodular energy functions. Let

$l_2 = l_1 + 1$. The previous definition is then equivalent to the following one. An edge potential θ_{ab} is submodular if

$$\theta_{ab}(l, l') + \theta_{ab}(l+1, l'+1) \leq \theta_{ab}(l, l') + \theta_{ab}(l+1, l'+1) \tag{2.57}$$

holds for each $l, l' \in \mathcal{L}$ (leaving out the "highest" labels which have no successor $l + 1$). Obviously equation (2.57) follows from equation (2.56) directly. In fact, the reverse statement also holds true, that is, equation (2.56) follows from equation (2.57). To see this, let us consider the quantities

$$\alpha(l_1, l_2, l'_1, l'_2) = \theta_{ab}(l_1, l'_1) + \theta_{ab}(l_2, l'_2) - \theta_{ab}(l_1, l'_2) - \theta_{ab}(l_2, l'_1) \tag{2.58}$$

for each four-tuple of such labels. These quantities fulfil

$$\alpha(l_1, l_2, l'_1, l'_2) = \sum_{l_3=l_1}^{l_2-1} \sum_{l'_3=l'_1}^{l'_2-1} \alpha(l_3, l_3+1, l'_3, l'_3+1) \tag{2.59}$$

for each four-tuple of such labels. If a function θ_{ab} is submodular according to the second definition equation (2.57), all addends in the above equation (2.59) are non-positive, hence the function θ_{ab} is submodular according to the first definition equation (2.56) as well.

The notion of submodularity is particular to the order introduced for the set of labels. An energy function which is submodular with respect to one ordering is in general not submodular with respect to another ordering. Hence, one cannot speak of submodular energy functions without specifying the label ordering.

CHAPTER 3

Related Work

In this chapter, we review relevant literature on CRF inference and submodular functions. In section 3.1, we introduce modeling of CRFs for computer vision applications. Section 3.2 deals with inference problems in CRFs and the role of submodular functions therein. Though we have separated modeling and inference of CRFs in this review, it is important to note that they are inextricably linked. Many of the efficient inference algorithms that we discuss are specialised for particular class of models, and the development of these models was often informed by the question of whether an existing efficient inference method could be generalised to a more general model.

3.1 CRF Modeling for Computer Vision

Depending on the order of interaction among variables, CRF models are usually classified into *pairwise models* and *higher-order models*. We describe these models below.

Pairwise CRFs: In *pairwise CRFs*, the energy function is defined on cliques of size two. The energy function in this case is a sum of *unary* and *pairwise*

potentials. Pairwise CRFs have been popular due to their simplicity and computational efficiency [103]. Pairwise interactions are the fundamental type of interactions required for modeling many vision applications.

Grid-like structures are the most typical graphs used for pairwise CRFs since they provide a natural representation for images. In these grids, traditionally the pixels have been assumed to be connected in 4-neighbourhood or 8-neighbourhood manner. Later, *dense CRFs* were proposed where each pixel is connected to every other pixel in the image [53]. Dense CRFs, being more expressive, perform significantly better than CRFs based on sparse connections for vision applications. Though inference in these dense models is computationally expensive, efficient methods have been proposed in literature that work well in practice [2, 19, 59].

The form of pairwise potentials determines the representation capability and inference complexity of pairwise CRFs. The most commonly used pairwise potentials have been Potts model, hierarchical Potts model and truncated convex models (TCM). Potts model imposes the same penalty for unequal label assignment to neighbouring variables, regardless of label dissimilarity. It is suitable for use in applications where an ordering among labels does not exist, such as segmentation [24, 33, 84]. However, for other vision applications such as image denoising, stereo correspondence, and optical flow, there is a natural ordering among the label set, and Potts model is unable to make use of such an ordering.

Two models that work well for applications where label ordering exists are the hierarchical Potts and truncated convex models. Hierarchical Potts is a generalisation of Potts model that specifies the distance between labels using an r-HST tree [42]. Truncated convex models [91] define pairwise potentials using truncated convex distance functions. For dense CRFs, these pairwise potentials are augmented with Gaussian kernels which modulate the strength of pairwise connections based on pixel spatial location and intensity values. These potentials

enable the use of fast Gaussian filtering methods for inference [53].

Pairwise CRFs are limited in their representation capacity, since they restrict the order of potentials to be at most two. Some image statistics can only be inferred by looking at larger image segments in unison [46]. Since pairwise models cannot capture such useful image properties, their accuracies for many vision applications tends to be limited. This motivated the development of higher-order CRFs and we discuss these next.

Higher-order CRFs: Higher-order CRFs have potential functions defined on cliques of size greater than two. These higher-order models can capture richer interactions among random variables as compared to pairwise models.

Early works introduced higher-order potentials based on label consistency in image regions. These regions (also called superpixels) are obtained by unsupervised segmentation algorithms [13]. One popular example is the P^n Potts model [44], which encourages label consistency over a set of random variables. It enforces label consistency rigidly, and this can be a problem if superpixels are defined inaccurately or in case of a conflict between overlapping superpixels. To deal with this, the robust P^n Potts model was proposed [45], which takes into account the number of random variables that disagree with the majority label of a clique. However, both P^n Potts and its robust version are unsuitable for applications where labeling ordering exists, such as in image denoising.

Global models form another class that have higher-order potentials acting on all variables in the model. Hence, the potential cost depends on the labeling of all pixels in the image. An instance of this class is the global connectivity prior [7, 97], which enforces the topological constraint that pixels of a segmentation should be connected. Another type of global model involves potential costs based on label statistics. For example, Ladicky et al. [65] proposed higher-order potentials based on object co-occurrence statistics - a measure of which labels

are likely to appear together in the final labeling. Delong et al. [17, 18] proposed a clique potential that is proportional to the number of labels used in the final labeling. However, their model provides additive bounds that are not invariant to reparameterizations of the energy function. This theoretical limitation is addressed by Dokania and Kumar [17, 18] who proposed a higher-order model based on the concept of parsimony. Here, the clique potentials are defined as being proportional to a diversity function of the set of unique labels present in the clique labeling. Parsimonious labeling extends pairwise metric labeling to higher-order models, and can be thought of as complementary to TMCM that generalises TCM.

3.2 Inference in CRFs

Efficient inference is critical for CRFs in computer vision since millions of variables can be involved. There are two inference problems in CRFs - (a) marginal inference where we compute the marginal probability for each pixel and each putative label, and (b) MAP inference, also known as energy minimisation, where we find the most likely labeling, corresponding to the mode of the conditional distribution. We review the literature for both of these problems below.

3.2.1 Submodular Functions

Submodular functions are a special class of set functions that have found important applications in diverse fields such as economics, game theory, and operations research. They can be minimised exactly in polynomial time. Maximisation of submodular functions is NP-hard but there exists algorithms with constant factor approximations. The wide applicability of submodular functions in diverse domains is perhaps due to the fact that they exhibit an intuitive diminishing returns property - that is, the marginal benefit of adding an element is greater

for a smaller set than a larger set. Submodular functions have close connections with convex functions. In some respects, submodular functions are also linked to concave functions.

Besides their application in CRF inference, submodular functions have been widely used for other tasks in machine learning such as learning the structure of graphical models [11, 73], clustering [74, 111], and subset selection [16, 67]. They have also been used as regularizers [5, 36] and in studying influence propagation in social networks [34, 41, 66].

3.2.2 Marginal Inference

Exact marginal inference in graphs with cycles (frequently occurring in vision where an image is considered as a grid graph) is $\#P$-hard [40]. Hence, approximate inference methods have been proposed. Two classes of approximate inference algorithms are popular: *Monte Carlo* methods and *variational* methods.

Monte Carlo (MC) algorithms are stochastic methods that can approximately produce samples from a distribution of interest [83]. The most important class of Monte Carlo methods for complex CRFs is *Markov Chain Monte Carlo* (MCMC) [75]. Two specific MCMC algorithms are widely used: Gibbs sampling [32, 80] and Metropolis-Hastings [69]. MCMC methods are unbiased, meaning that they are guaranteed to sample from the given distribution given sufficient computation time. However, doing so often requires a long time and it is hard to know in practice when that point has been reached. Moreover, to reach a good solution in a reasonable amount of time, MCMC methods require making some important choices of the sampling technique - for instance, selecting a good proposal in Metropolis-Hastings. Making such good choices is more of an art. Since the focus of our work is on variational methods, we do not provide a detailed literature review for Monte Carlo methods.

Variational Inference: The central idea of variational methods is to pose inference as an optimisation problem. Given an intractable probability distribution p, variational methods attempt to optimise over a class of tractable distributions Q in order to find $q \in Q$ that is most similar to p according to a chosen measure. This problem is also called *I-projection* of p onto the family Q. One can then query q rather than p to obtain an approximate solution.

When KL-divergence is used as the measure of similarity, most formulations end up maximising a lower bound $\mathcal{L}$ on the log-partition function for any family of Q. This lower bound $\mathcal{L}$ is referred to as the evidence lower bound (ELBO). The most popular approach, called *naive mean-field*, is to consider Q as the family of fully-factorised distributions [102]. The resulting optimisation problem of maximising the ELBO is amenable to coordinate ascent algorithm, which consists of making sequential updates to the marginals. To make mean-field scalable for real-world vision problems, strategies based on parallel updates have been proposed [6, 59]. To deal with the high computational cost of mean-field for dense CRFs, Koltun and P. Krahenbuhl [53] used a fast Gaussian filtering method to speed up mean-field updates for the Gaussian Potts model. Vineet et al. [99] later extended this approach to higher-order models with P^n Potts potential as well as the cooccurence potential. Often, the naive assumption of fully-factorised distribution is too strong, since it cannot model dependencies among variables. To address this, *structured mean-field* approximates p by a structured distribution, such as combination of chain or tree models [49].

In the mean-field-based approaches, we restricted the domain Q to marginals with specific structures. There are two classes of variational inference algorithms that are instead based on relaxing the marginal polytopes in addition to approximating the objective function - (a) loopy belief-propagation, (LBP) and (b) tree reweighted message-passing (TRW).

The general idea behind LBP is to run belief propagation, a dynamic programming method that gives exact marginals for tree-structured models [78], on graphs with loops. We do this in spite of the fact that the presence of loops does not guarantee convergence. Theoretically, LBP approximates the ELBO by a Bethe free-energy approximation that is easier to compute [108]. The resulting maximisation problem is non-concave and can be solved by the concave-convex procedures [109]. LBP, when it converges, finds a local maximum.

TRW, on the other hand, works with an approximate objective that is an upper-bound on the log-partition function. It approximates p with a mixture of spanning tree-based distributions [101]. The resulting problem is concave and can be solved by projected gradient ascent. Both LBP and TRW have been usually applied for sparse pairwise CRFs and are still intractable for dense CRFs with Gaussian potentials, and higher-order models .

Submodular functions have formed the basis of some variational inference methods. There is a special polyhedron associated with a submodular function over which a linear program can be solved efficiently using a greedy procedure. This is in spite of the fact that the polyhedron is defined by an exponential number of constraints. This property has been used by Djolonga et al.[20] who proposed a variational method for submodular energies of pairwise CRFs. The method was shown to be equivalent to minimisation of the inclusive Rényi divergence [22]. Later, the method was extended for multi-label pairwise CRFs in [110]. Submodular functions have also been used for variational inference in a more general class of models called *cooperative graphical models* where the edge costs, instead of being additive, are submodular [21].

3.2.3 MAP Inference

There are three main classes of MAP inference methods for CRFs: (a) message passing, (b) dual methods, and (c) graph cuts-based approaches. Since many dual methods are also message passing-based, we add the caveat that by message passing methods, we refer to those that are not directly derived from convex relaxations. Our work focuses on graph cuts-based methods, so we give only a brief review of message passing and dual methods.

Message Passing: Belief propagation can be used for MAP inference as well in addition to marginal inference. The MAP inference version is known as the *max-product algorithm*. Loopy belief propagation has been widely used for MAP inference in vision applications [24, 29, 104]. The message passing scheme can be either parallel or sequential. Tappen et al. [92] showed empirically that the sequential scheme converged much faster than the parallel update, with both methods giving similar accuracies. There has been a significant amount of research work on improving the efficiency of message passing by exploiting specific types of graph structures or energy functions and providing convergence guarantees [51, 54, 63, 79].

Running belief propagation on factor graphs provides a simple way to deal with MAP inference in higher-order CRFs. However, the computation complexity increases exponentially with the size of the largest clique. Several methods have been proposed to accelerate belief propagation for special families of higher-order models such as linear constraint potentials and cardinality-based potentials [25, 70, 81, 93].

Dual Methods: MAP inference can be posed as integer linear programming (ILP). The ILP is then relaxed to yield convex optimisation problems for which efficient algorithms exist. Several convex relaxations have been studies for MAP

inference [62]: (a) linear programming (LP), (b) quadratic programming (QP), and (c) second order cone programming (SOCP).

LP is perhaps the most common relaxation [88]. The LP solution is usually fractional and is converted to an integer solution using rounding schemes. The LP approach provides tight multiplicative bounds for certain classes of energy functions such as Potts and truncated linear models using specific rounding schemes [12, 42]. Werner [106, 107] extended LP-based relaxations to higher-order models based on *min-sum diffusion*.

It is generally infeasible to solve the LP in the primal form since the number of variables involved is too large for state-of-the-art interior point solvers. To alleviate this difficulty, several efficient algorithms have been proposed to approximately solve the Lagrangian dual of the LP [38, 51, 54, 58, 100, 105]. Approaches based on the dual formulation have also been extended to higher-order models [55]. An exception is the recent work of Ajanthan et al. [2] who directly solved the primal form of the LP for dense CRFs using a modified version of the fast Gaussian filtering method.

QP [82] and SOCP [72] are have more flexibility in the form of objective function and type of constraints, but do not provide as good an approximation as LP. Kumar et al. [62] showed that QP and SOCP are in fact equivalent.

Graph Cuts: Some classes of energy functions can be represented as *st*-graphs [27, 37, 52], such that there is a one-to-one correspondence between the energy of each putative labeling and the capacity of the corresponding cut in the graph. Pairwise Ising and convex models are examples of such models. The task of computing minimum energy is then equivalent to finding a minimum *st*-cut in such a graph. There is a duality between minimum cut and maximum flow problems. Hence, efficient maximum flow algorithms based on Ford-Fulkerson style augmenting path or Goldberg-Tarjan style push-relabel [14] can be used

for MAP inference in these models. By means of an encoding which represents a labeling as a set, energy functions can be viewed as set functions [77]. Graph-representable set functions are called *cut-functions*.

Cut functions are special classes of submodular functions. While there exists strongly polynomial time algorithms for minimising general submodular functions [15, 86], these are still computationally impractical for vision applications. For instance, the minimum-norm point method has $\mathcal{O}(n^7)$ runtime complexity with $\mathcal{O}(n^5)$ oracle calls [71]. Cut functions, on the other hand, can be minimised by max-flow-based algorithms in $\mathcal{O}(n^{3.3})$ time and in fact, much faster in practice by specialised solvers for vision CRFs such as that of Boykov and Kolmogorov [9], and GRIDCUT [1]. Given the importance of cut functions, the question of which energies can be represented and minimised using graph cuts has been well-studied [28, 30, 52, 112, 113]. Unfortunately, most energy functions typically used for vision applications, such as Potts model or TCM, are not graph-representable. Approximation algorithms using graph cuts have been proposed for these models.

Move-making algorithms are an important class of approximation algorithms. The main idea behind these is to define a set of possible moves based on the initial CRF configuration, accept the best move as the initial solution for next iteration, doing so iteratively until convergence. For example, α-expansion and $\alpha\beta$-swap were proposed which generalise binary graph cuts to deal with CRFs with metric or semi-metric energies [9, 10]. Range-move methods [64, 96] further improve the optimum solution for CRFs with truncated convex potentials. These methods explore a large search space by working with a range of consecutive labels at each iteration, instead of one or two labels as in α-expansion and $\alpha\beta$-swap. The range-move-based algorithms have been shown to have the same multiplicative bounds as the standard LP relaxation, but are faster.

There has been substantial work on increasing the efficiency of these graph

cuts-based algorithms. Veksler [95] proposed a dynamic programming-based method to speed up α-expansion. *Dynamic graph cuts*, a dynamic max-flow method, was studied in [47, 48] to reuse the flow by setting the initial condition for an iteration to the solution of the previous one. In [56, 57], a primal-dual scheme called *FastPD* based on LP relaxation was proposed for doing MAP inference by combining solutions from both the primal and dual formulations, leading to speedups.

There are two strands of work relating to graph cut-based MAP inference for higher-order CRFs. The first approach, based on order reduction, deals with the problem by a two-stage procedure: the higher-order model is first reduced to a pairwise one with the same solution, and then standard graph cuts algorithms for pairwise models are applied [3, 26, 28, 31, 38, 52]. The advantage of these methods is that they are applicable for general higher-order models. However, the resulting algorithms are exponential in the size of the cliques, which prevents their use in applications that require very high-order cliques (with hundreds or even thousands of random variables).

The other approach has been to propose specific classes of higher-order potentials and design efficient graph cuts-based algorithms applicable to a particular class. These models are often able to better capture the characteristics of the problem and the inference method tends to run faster as compared to order reduction procedures. It is this approach that we adopt in our work. One early work was the P^n Potts model [44], which encourages label consistency over a set of random variables. This work was extended in [45], which introduced robustness in the P^n Potts model by taking into account the number of random variables that disagreed with the majority label of a clique. Both the P^n Potts model and its robust version lend themselves to efficient optimization via the expansion algorithm [10], which solves one st-MINCUT problem at each iteration.

The expansion algorithm provides multiplicative bounds [35] which measure the quality of the estimated labeling with respect to the optimal one. Our work generalizes both the models and the corresponding inference methods. The expansion algorithm has also been used for label costs-based higher-order model of Delong et al. [17, 18], parsimonious labeling model of Dokania and Kumar [23], and cooccurrence-based clique potential of Ladicky et al. [65].

CHAPTER 4

Truncated Max-of-Convex Models

This work was presented as a poster at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. Here, we provide an extended version of the paper with additional results and proofs.

arXiv:1512.07815v2 [cs.CV] 3 Dec 2016

Truncated Max-of-Convex Models

Pankaj Pansari
University of Oxford
pankaj@robots.ox.ac.uk

M. Pawan Kumar
University of Oxford
pawan@robots.ox.ac.uk

Abstract

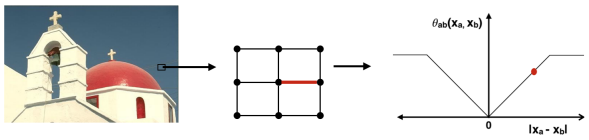
Truncated convex models (TCM) are a special case of pairwise random fields that have been widely used in computer vision. However, by restricting the order of the potentials to be at most two, they fail to capture useful image statistics. We propose a natural generalization of TCM to high-order random fields, which we call *truncated max-of-convex models* (TMCM). The energy function of TMCM consists of two types of potentials: (i) unary potential, which has no restriction on its form; and (ii) clique potential, which is the sum of the m largest truncated convex distances over all label pairs in a clique. The use of a convex distance function encourages smoothness, while truncation allows for discontinuities in the labeling. By using $m > 1$, TMCM provides robustness towards errors in the definition of the cliques. In order to minimize the energy function of a TMCM over all possible labelings, we design an efficient st-MINCUT based range expansion algorithm. We prove the accuracy of our algorithm by establishing strong multiplicative bounds for several special cases of interest. Using standard real data sets, we demonstrate the benefit of our high-order TMCM over pairwise TCM, as well as the benefit of our range expansion algorithm over other st-MINCUT based approaches.

1 Introduction

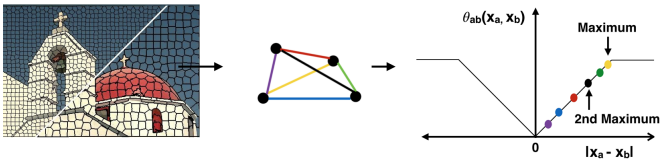
Truncated convex models (TCM) are a special case of pairwise random fields that have been widely used for low-level vision applications. A TCM is defined over a set of random variables, each of which can be assigned a value from a finite, discrete and ordered label set. In addition, a TCM also specifies a neighborhood relationship over the random variables. An assignment of values to all the variables is referred to as a labeling. In order to quantitatively distinguish the labelings, a TCM specifies an energy function that consists of unary and pairwise potentials.

Given an input, the output is obtained by minimizing the energy function of a TCM over all possible labelings. While this is an NP-hard problem, several approximate algorithms have been proposed in the literature [3, 4, 12, 17, 20, 21, 24, 27, 30], which provide accurate solutions in practice [28].

Since we cannot reasonably expect to improve the optimization of TCM, any failure cases must be addressed by modifying the model itself to better capture image statistics. To this end, we propose to address one of the main deficiencies of TCM, namely, the restriction to potentials of order at most two. Specifically, we propose a natural generalization of TCM to high-order random fields, which we refer to as *truncated max-of-convex models* (TMCM). Similar to TCM, our model places no restrictions on the unary potentials. Furthermore, unlike TCM, it allows us to define clique potentials over an arbitrary number of random variables. The value of the clique potential is proportional to the sum of the m largest truncated convex distances computed over disjoint pairs of random variables in the clique. Here, disjoint pairs imply that the label of no random variable is used more than once to compute the value of the clique potential. Figure 1 demonstrates how TMCM differs from TCM. The exact mathematical form of the TMCM energy function will be presented in section 4. The term m is a positive integer that is less than or equal to half the number of variables in the smallest clique. Importantly, the constant of proportionality for each clique potential can depend on the input



(a) Truncated convex model



(b) Truncated max-of-convex model

Figure 1: *TMCM as generalization of TCM. In (a), given an image, TCM considers pairwise 4-neighborhood relationships and uses truncated convex distance function for pairwise potential. In (b), TMCM considers superpixels as cliques. The clique potential for m =2 is the sum of the first and second maximum over all the pairwise truncated convex distances such that no variable is used more than once.*

corresponding to all the random variables in the clique. This can help capture more interesting image statistics, which in turn can lead to a more desirable output. For example, in image denoising, the clique weights can depend on the variance of intensity over a superpixel (group of pixels with similar semantic and perceptual characteristics).

In order to enable the use of TMCMs in practice, we require an efficient and accurate energy minimization algorithm that can compute the output for a given input. To this end, we extend the range expansion algorithm for TCM to deal with arbitrary sized clique potentials. Our algorithm retains the desirable property of iteratively solving an st-MINCUT problem over an appropriate directed graph (where the number of vertices and arcs grows linearly with the number of random variables and cliques, and quadratically with the number of labels). As the st-MINCUT problem lends itself to several fast algorithms [2], this makes our overall approach computationally efficient. Furthermore, we provide strong theoretical guarantees on the quality of the solution for several special cases of interest, which establishes its accuracy. Our multiplicative bounds are better than the baselines for cases where comparison is possible. Using standard real data sets, we show the benefit of high-order TMCM over pairwise TCM, as well as the advantage of our range expansion algorithm over other st-MINCUT based approaches.

2 Related Work

Pairwise TCM offer a natural framework to capture low-level cues for vision problems such as image denoising, stereo correspondence, segmentation and optical flow [28]. However, the restriction to pairwise potentials limits their representational power.

For the past few years, there has been a growing interest in high-order models. Though other inference algorithms such as message passing are possible [9, 19, 22, 26, 29, 31], in this work our focus is on models that admit efficient st-MINCUT based solutions and provide strong theoretical guarantees on the quality of the solution. One early work was the P^n Potts model [15], which encourages label consistency over a set of random variables. This work was extended in [16], which introduced robustness in the P^n Potts model by taking into account the number of random variables that disagreed with the majority label of a clique. Both the P^n Potts model and its robust version lend themselves to efficient optimization via the expansion algorithm [3], which solves one st-MINCUT problem at each iteration. The expansion algorithm provides multiplicative bounds [11], which measure the quality of the estimated labeling with respect to the optimal one. Our work generalizes both the models, as well as the corresponding expansion algorithm. Specifically, when the truncation factor of our models is set to 1, we recover the robust P^n model. Furthermore, a suitable setting of the range expansion algorithm (setting the interval length to 1) recovers the expansion algorithm.

Jegelka and Bilmes [14] introduced a nonsubmodular high-order model which is based on edge cooperation and is optimizable using st-MINCUT, but the algorithm has weak approximation bounds.

Delong et al. [6, 7] proposed a clique potential based on label costs that can also be handled via the expansion algorithm. However, unlike the robust P^n Potts model, their model provides additive bounds that are not invariant to reparameterizations of the energy function. This theoretical limitation is addressed by the recent work of Dokania and Kumar [8] on parsimonious labeling. Here, the clique potentials are defined as being proportional to a diversity function of the set of unique labels present in the clique labeling. Our work can be thought of as being complementary to parsimonious labeling. Specifically, while parsimonious labeling is an extension of pairwise metric labeling to high-order models, our work is an extension of TCM. The only metric that also results in a TCM is the truncated linear distance. As our experiments will demonstrate, our specialized range expansion algorithm provides significantly better results for truncated max-of-linear models compared to the hierarchical st-MINCUT approach of [8].

We note that there have been several works that deal with more general high-order potentials and design st-MINCUT style solutions for them. For example, Fix et al. [10] use the submodular max-flow algorithm [18], while Arora et al. [1] use generic cuts. However, the resulting algorithms are exponential in the size of the cliques, which prevents their use in applications that require very high-order cliques (with hundreds or even thousands of random variables). A notable exception to this is the work of Ladicky et al. [25], who proposed a co-occurrence based clique potential whose only requirement is that it should increase monotonically with the set of unique labels present in the clique labeling. However, the use of such a general clique potential still results in an inaccurate energy minimization algorithm, as will be seen in our experimental comparison.

3 Truncated Convex Models

A TCM is a random field defined by a set of discrete random variables $\mathbf{X} = \{X_a, a \in \mathcal{V}\}$, and a neighborhood relationship $\mathcal{E}$ over them (that is, X_a and X_b are neighboring random variables if $(a, b) \in \mathcal{E}$). Each random variable can take a value from a finite label set $\mathbf{L}$, which is assumed to be ordered to enable the use of convex distance functions. Without loss of generality, we define $\mathcal{V} = \{1, 2, \cdots, n\}$ and $\mathbf{L} = \{1, 2, \cdots, h\}$.

An assignment of values to all the random variables $\mathbf{x} \in \mathbf{L}^n$ is referred to as a labeling. To quantitatively distinguish the h^n possible labelings, a TCM defines an energy function that consists of two types of potentials. First, the unary potentials $\theta_a(x_a)$ that depends on the label x_a of one random variable X_a. Second, the pairwise potentials $\theta_{ab}(x_a, x_b)$ that depends on the labels x_a and x_b of a pair of neighboring random variables (X_a, X_b). There are no restrictions on the form of the unary potentials. However, the pairwise potentials are defined using a truncated convex distance function over the label set.

To provide a formal specification of the pairwise potentials, we require some definitions. We denote a convex distance function by $d : \mathbb{Z} \to \mathbb{R}$ (where $\mathbb{Z}$ is the set of integers and $\mathbb{R}$ is the set of real numbers). Recall that a convex distance function satisfies the following properties: (i) $d(y) \geq 0$ for all $y \in \mathbb{Z}$ and $d(0) = 0$; (ii) $d(y) = d(-y)$ for all $y \in \mathbb{Z}$; and (iii) $d(y + 1) - 2d(y) + d(y - 1) \geq 0$ for all $y \in \mathbb{Z}$. Note that the above properties also imply that $d(y) \geq d(z)$ if $|y| \geq |z|$, for all $y, z \in \mathbb{Z}$. Examples of convex distance functions include the linear (that is, $d(y) = |y|$) and the quadratic distance function (that is, $d(y) = y^2$) distance.

Given two labels $l_i, l_j \in \mathbf{L}$, we use a convex function $d(\cdot)$ to measure the distance between them as $d(l_i - l_j)$, thereby encouraging smooth labelings. In order to prevent the overpenalization of the discontinuities in an image, it is common practice to truncate the convex distance function [3, 23, 30]. Formally, a truncated convex function is defined as $\min\{d(\cdot), M\}$, where M is the truncation factor. We now define the pairwise potential as $\theta_{ab}(x_a, x_b) = \omega_{ab} \min\{d(x_a - x_b), M\}$, where ω_{ab} is a (data-dependent) non-negative constant of proportionality.

Hence, a TCM specifies an energy function $E(\cdot)$ over the labelings $\mathbf{x} \in \mathbf{L}^n$ as follows:

$$E(\mathbf{x}) = \sum_{a \in \mathcal{V}} \theta_a(x_a) + \sum_{(a,b) \in \mathcal{E}} \omega_{ab} \min\{d(x_a - x_b), M\}. \tag{1}$$

The unary potentials are arbitrary, the edge weights ω_{ab} are non-negative, $d(\cdot)$ is a convex function and $M \geq 0$ is the truncation factor. Given an input (which provides the values of the unary potentials and the edge weights), the desired output is obtained by solving the following optimization problem:

$\min_{\mathbf{x}\in\mathbf{L}^n} E(\mathbf{x})$. While this optimization problem is NP-hard, we can obtain an accurate approximate solution by using the efficient range expansion algorithm [23], as well as several other approaches based on st-MINCUT [3, 12, 21, 30] and linear programming [4, 17, 20].

4 Truncated Max-of-Convex Models

We now present a natural generalization of TCM to high-order random fields, which define potentials over random variables that form a clique (where all the random variables in a clique are neighbors of each other). Importantly, we do not place any restriction on the size of the clique.

Truncated Max-of-Convex Potentials. Consider a high-order clique consisting of the random variables $\mathbf{X_c} = \{X_a, a \in \mathbf{c} \subseteq \mathcal{V}\}$. We denote a labeling of the clique as $\mathbf{x_c} \in \mathbf{L}^c$, where we have used the shorthand $c = |\mathbf{c}|$ to denote the size of the clique. In order to specify the value of the clique potential for the labeling $\mathbf{x_c}$ we require a sorted list of the (not necessarily unique) labels present in $\mathbf{x_c}$. We denote this sorted list by $\mathbf{p}(\mathbf{x_c})$ and access its i-th element as $p_i(\mathbf{x_c})$. For example, consider a clique consisting of random variables $\mathbf{X_c} = \{X_1, X_2, X_3, X_4, X_5, X_6\}$. If the number of labels is $h = 10$, then one of the putative labelings of the clique is $\mathbf{x_c} = \{3, 2, 1, 4, 1, 3\}$ (that is, X_1 takes the value 3, X_2 takes the value 2 and so on). For this labeling, $\mathbf{p}(\mathbf{x_c}) = \{1, 1, 2, 3, 3, 4\}$. The value of $p_1(\mathbf{x_c})$ and $p_2(\mathbf{x_c})$ is 1, the value of $p_3(\mathbf{x_c})$ is 2 and so on. Given a convex function $d(\cdot)$, a truncation factor M and an integer $m \in [0, \lfloor c/2 \rfloor]$, the clique potential $\theta_{\mathbf{c}}(\cdot)$ is defined as

$$\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \sum_{i=1}^{m} \min\{d(p_i(\mathbf{x_c}) - p_{c-i+1}(\mathbf{x_c})), M\}. \tag{2}$$

Here, $\omega_{\mathbf{c}} \geq 0$ is the clique weight that does not depend on the labeling. However, it can be chosen based on the observed data - for instance, we may want to assign small weights to cliques with large variance of intensity or disparity. The term inside the summation is the truncated value of the i-th largest distance between the labels of all pairs of random variables within the clique, subject to the constraint that the label of no random variable is used more than once in the computation of the clique potential value. In other words, our clique potential is proportional to the sum of the truncation of the m largest convex distance functions over disjoint pairs of random variables.

Given an input, the desired output is obtained by solving the following optimization problem: $\min_{\mathbf{x}\in\mathbf{L}^n} E(\mathbf{x})$. Note that TMCM is a generalization of the P^n Potts model [15] (m = 1, M = 1) as well as its robust version [16] ($m > 1$, M = 1). Furthermore, it is complementary to the recently proposed parsimonious labeling, which generalizes metric labeling. Henceforth, we assume the unary potentials are non-negative. This assumption is not restrictive as we can always add a constant to the unary potentials of a random variable. This modification would only result in the energy of all labelings changing by the same constant. As we shall see, our algorithm as well as its theoretical guarantees are invariant to such changes in the energy function.

5 Advantages of using TMCM

Labeling	$m = 1$	$m = 2$	$m = 3$
(a) {1,1,1,1,2,2}	1	2	2
{1,2,3,4,5,6}	3	6	7
(b) {1,1,1,9,9,9}	3	6	9
{1,1,1,8,8,9}	3	6	9
(c) {1,1,1,1,1,7}	3	3	3
{1,1,1,2,3,4}	3	5	6

Table 1: *Clique potential value $\theta_{\mathbf{c}}(\mathbf{x_c})$ defined by a linear function with M = 3 and $\omega_{\mathbf{c}}$ = 1 for various values of m. Since clique size is 6, $0 \leq m \leq 3$. Pair (a) demonstrates why taking the largest convex distances favors smoothness; (b) demonstrates how truncation prevents overpenalization of discontinuities; (c) demonstrates how using $m > 1$ can provide some degree of robustness to errors in the definitions of the cliques.*

We know show how TMCM encourages smooth labelings, does not overpenalize discontinuities and is robust to erroneous clique definitions.

Smoothness. The truncated max-of-convex potentials encourage smooth labelings. In order to illustrate this, let us consider the example of a clique of six random variables $\mathbf{X_c}$ and a label set $\mathbf{L}$ of size 10. We can define a truncated convex distance using a linear function $d(y) = |y|$ and a truncation factor of $M = 3$. Consider pair (a) of labelings shown in table 1. If the labeling is {1, 1, 1, 1, 2, 2} and M = 3, m = 3, $\omega_{\mathbf{c}}$ = 1, $\theta_{\mathbf{c}}(\mathbf{x_c})$ = min(6-1, 3) + min(5 - 2, 6) + min(4 - 3, 3) = 3 + 3 + 1 = 7. Clearly, the first labeling of this pair is significantly smoother than the second, which is reflected in the value of the clique potential for all values of m. In contrast, if we were to consider the minimum distance among all pairs of labels, both the labelings will provide a clique potential value of 0.

Discontinuities. Similar to the pairwise case, the use of a truncation factor helps prevent overpenalization of discontinuities. For example, let us consider the pair (b) of labelings in table 1. In both cases, the six random variables appear to belong to two *groups*, one whose labels are low and one whose labels are high. Without a truncation, such a discontinuity would have been penalized heavily (for example, 8 for $m = 1$ for both the labelings). This in turn would discourage the clique to be assigned this labeling even though this type of discontinuity is expected to occur in natural images. However, with a truncation, the penalty is significantly less (for example, 3 for $m = 1$ for both the labelings), which can help preserve the discontinuities in the labeling obtained via energy minimization.

Robustness. In order to use a TMCM, we would be required to define the cliques. For example, given an image, we could use a bottom-up oversegmentation approach to obtain superpixels, and then set all the pixels in a superpixel to belong to a clique. However, oversegmentation can introduce errors since it has no notion of the specific vision application we are interested in modeling. To add robustness to errors in the clique definitions, we can set $m > 1$. For example, consider pair (c) of labelings in table 1. The first of these labelings contains a single random variable with a very high label, which could be due to the fact that the corresponding pixel has been incorrectly grouped in this superpixel. As can be seen from the values of the potential, the presence of such an erroneous pixel in the superpixel is not heavily penalized when we use $m > 1$. For example, when $m = 3$ the value of the clique potential for the first labeling (with an erroneous pixel) is the same as the second labeling (which is a fairly smooth labeling).

6 Optimization via Range Expansion

As TMCM is a generalization of TCM, it follows that the corresponding energy minimization problem is NP-hard. However, we show that the efficient and accurate range expansion algorithm can be extended to handle this more general class of energy functions.

Algorithm 1 shows the main steps of range expansion. The algorithm starts by assigning the random variables to an initial label (step 1). For example, all the random variables could be assigned to the label 1. Next, it selects an interval of consecutive labels of size at most h' (steps 3-4), where h' is specified as an input to the algorithm. We will see later in the section that the value of h' can be chosen to obtain the optimal worst case bound for specific instances of the TMCM. Next, it minimizes the energy over all the labelings that either allow a random variable to retain its current label, or choose a new label in the selected interval (step 5). If the energy of the new labeling is lower than that of the current labeling, then the solution is updated (steps 6-8). This process is repeated for all the intervals of consecutive labels of size at most h'. The entire algorithm stops when the energy cannot be reduced further for any choice of the interval.

The crux of the range expansion algorithm is problem (3), which needs to be solved for any given interval $\mathbf{I}$ and current labeling $\hat{\mathbf{x}}$. Unfortunately, this problem itself is NP-hard for TMCM. Indeed, when $h' = h$, problem (3) is equivalent to the original energy minimization problem. In order to operationalize the range expansion algorithm, we need to devise an approximate algorithm for problem (3). We achieve this in two steps. First, we obtain an overestimate of the energy function $E(\cdot)$, which we denote by $E'(\cdot)$. The energy function $E'(\cdot)$ is restricted to the labels in the interval $\mathbf{I}$ together with the labels specified by the current labeling $\hat{\mathbf{x}}$. Second, we minimize the overestimated energy $E'(\cdot)$ over all of its putative labelings by solving an equivalent st-MINCUT problem. We describe our two-step algorithm in the next two subsections in detail. Specifically, subsection 6.1 describes the exact form of the energy function $E'(\cdot)$, while subsection 6.2 describes the construction of the directed graph over which we solve the st-MINCUT problem to obtain the labeling $\mathbf{x}'$.

n	Number of random variables
h	Number of labels
$\mathcal{V}$	$\{1, 2, \ldots, n\}$
$\mathbf{L}$	$\{1, 2, \ldots, h\}$
$\mathbf{X}$	Set of random variables $\{X_a, a \in \mathcal{V}\}$
$\mathbf{X_c}$	Set of random variable belonging to a clique $\{X_a, a \in \mathbf{c} \subseteq \mathcal{V}\}$
$\mathbf{x_c}$	Labeling of clique $\mathbf{c}$
$\mathbf{p}(\mathbf{x_c})$	Sorted list of the labels present in $\mathbf{x_c}$
I_n	Interval of consecutive labels $[i_n + 1, j_n]$
h'	Length of interval, that is, $h' = j_n - i_n$
Γ_r	Set of intervals $\{[0, r], [r + 1, r + L], \ldots, [., h - 1]\}$
f	Labeling of the random field (v_a takes the label $l_{f(a)}$)
f^*	An optimal (MAP) labeling of the random field
$\theta_a(i)$	Unary potential of assigning label l_i to v_a
$\omega_{\mathbf{c}}$	Weight for clique $\mathbf{X_c}$
c	Size of clique $\mathbf{X_c}$
$d(.)$	Convex function used to define distance between two labels
M	Truncation factor
$\theta_{\mathbf{c}}(\mathbf{x_c})$	Clique potential of assigning label l_i to v_a
$E(f)$	Energy of the labeling f
$\mathcal{A}(f, I_n)$	$\{\mathbf{X_c} \in \mathcal{C}, f(v) \in I_n \forall v \in \mathbf{X_c}\}$
$\mathcal{B}(f, I_n)$	$\{\mathbf{X_c} \in \mathcal{C}, \exists X_v, X_w \in \mathbf{X_c} \vert f(v) \in I_n \wedge f(w) \notin I_n\}$

Table 2: Definitions of various symbols used in the supplementary document

Algorithm 1 The range expansion algorithm.

input Energy function $E(\cdot)$, initial labeling $\mathbf{x}^0$, interval length h'.

1: Initialize the output labeling $\hat{\mathbf{x}} = \mathbf{x}^0$.
2: **repeat**
3: **for all** $i_m \in [-h' + 2, h]$ **do**
4: Define an interval of labels $\mathbf{I} = \{f, \cdots, l\}$ where $f = \max\{i_m, 1\}$ and $l = \min\{i_m + h' - 1, h\}$.
5: Obtain a new labeling $\mathbf{x}'$ by solving the following optimization problem:

$$\begin{aligned} \mathbf{x}' = \quad & \underset{\mathbf{x}}{\operatorname{argmin}}\, E(\mathbf{x}), \\ & \text{s.t. } x_a \in \mathbf{I} \cup \{\hat{x}_a\}, \forall a \in \mathcal{V}. \end{aligned} \tag{3}$$

6: **if** $E(\hat{\mathbf{x}}) > E(\mathbf{x}')$ **then**
7: Update $\hat{\mathbf{x}} = \mathbf{x}'$.
8: **end if**
9: **end for**
10: **until** The labeling does not change for any value of i_m.

output The labeling $\hat{\mathbf{x}}$.

6.1 Overestimation of the Energy Function

Given an interval $\mathbf{I} = \{f, \cdots, l\}$ of consecutive labels, and the current labeling $\hat{\mathbf{x}}$, we define the new energy function $E'(\cdot)$ over the set of random variables $\mathbf{X}$. Unlike the original energy function, the label set corresponding to $E'(\cdot)$ is equal to $\mathbf{L}' = \{0, 1, \cdots, h'\}$, where $h' = l - f + 1$. The label 0 in the set $\mathbf{L}'$ corresponds to a random variable retaining its current label, while any other label $i \geq 1$ corresponds to a random variables taking the label $f + i - 1 \in \mathbf{I}$. A labeling of the energy function $E'(\cdot)$ is denoted by $\mathbf{y} \in (\mathbf{L}')^n$ in order to distinguish it from the labeling corresponding to the original energy function. We say that a labeling $\mathbf{x} \in \mathbf{L}^n$ corresponds to the labeling $\mathbf{y} \in (\mathbf{L}')^n$ if

$$x_a = \begin{cases} \hat{x}_a & \text{if } y_a = 0, \\ y_a + f - 1 & \text{otherwise.} \end{cases} \tag{4}$$

We define the unary potentials and the clique potentials of the energy function $E'(\cdot)$ as follows.

Unary Potentials. The unary potential of a random variable X_a (where $a \in \mathcal{V}$) being assigned a label $y_a \in \mathbf{L}'$ is given by the following equation:

$$\theta'_a(y_a) = \begin{cases} \theta_a(\hat{x}_a) + \kappa_a & \text{if } y_a = 0 \\ \theta_a(y_a + f - 1) + \kappa_a & \text{otherwise.} \end{cases} \tag{5}$$

In other words, if $y_a = 0$ then the unary potential corresponds to the random variable X_a retaining its current label $\hat{x}_a$, and if $y_a \neq 0$ then the unary potential corresponds to the random variable X_a being assigned the label $y_a + f - 1 \in \mathbf{I}$. The constant κ_a is added to the unary potentials to ensure that they are non-negative, which makes the description of the graph construction in the next subsection simpler.

Clique Potentials. In order to describe the high-order clique potentials of the new energy function we require a function $\delta_{a,b} : \mathbf{L}' \times \mathbf{L}' \to \mathbb{R}$ for each $(a, b) \in \mathcal{E}$, which is defined as follows:

$$\delta_{a,b}(y_a, y_b) = \begin{cases} \min\{d(\hat{x}_a - \hat{x}_b), M\} & \text{if } y_a = y_b = 0, \\ M + d(y_b - 1) & \text{if } y_a = 0, y_b \neq 0, \\ M + d(y_a - 1) & \text{if } y_a \neq 0, y_b = 0, \\ d(y_a - y_b) & \text{if } y_a \neq 0, y_b \neq 0. \end{cases} \tag{6}$$

Here, $d(\cdot)$ is the convex function and M is the truncation factor associated with the original energy function $E(\cdot)$.

Proposition 1. *$\delta_{a,b}(y_a, y_b)$ is submodular in the sense of label-set encoding used in [13]and is an overestimate of the truncated convex distance* $\min\{d(x_a - x_b), M\}$ *(Proof in Appendix A).*

Given a labeling $\mathbf{y_c} \in (\mathbf{L}')^c$ of a clique $\mathbf{c}$ of size c, we denote a sorted list of the labels in $\mathbf{y_c}$ as $\mathbf{p}(\mathbf{y_c})$. Furthermore, we denote the indices of the sorted list as $\mathbf{q}(\mathbf{y_c})$. In other words, the random variable corresponding to the i-th smallest label (that is, the i-th element of the list $\mathbf{p}(\mathbf{y_c})$, which is denoted by $p_i(\mathbf{y_c})$) is given by X_a where $a = q_i(\mathbf{y_c})$. To avoid clutter, we will drop the argument $\mathbf{y_c}$ from $\mathbf{p}$ and $\mathbf{q}$ whenever it is clear from context.

Using the above definitions, the high-order clique potential for the new energy $E'(\cdot)$ can be concisely specified as

$$\theta'_{\mathbf{c}}(\mathbf{y_c}) = \omega_{\mathbf{c}} \sum_{i=1}^{m} \delta_{q_i, q_{c-i+1}}(p_i, p_{c-i+1}). \tag{7}$$

Hence, the clique potentials in the energy function $E'(\cdot)$ are the sum of the m maximum submodular functions over disjoint pairs of random variables in the clique.

6.2 Graph Construction

Our problem is to minimize the energy function $E'(\cdot)$ over all possible labelings $\mathbf{y} \in (\mathbf{L}')^n$. To this end, we convert it into an equivalent st-MINCUT problem over a directed graph, which can be solved efficiently if all arc capacities are non-negative [2].

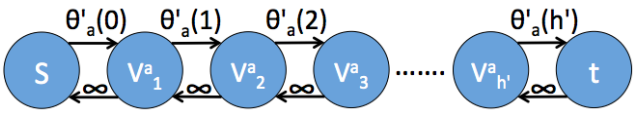


Figure 2: *Arcs and their capacities for representing the unary potentials for the random variable X_a. According to the labeling defined in equation (8), if $x_a = \hat{x}_a$, then the arc (s, V_1^a) will be cut, which will contribute exactly $\theta_a(\hat{x}_a)$ to the capacity of the cut. If $x_a = s + i - 1$ where $i \in \{1, \cdots, h' - 1\}$, then the arc (V_i^a, V_{i+1}^a) will be cut, which will contribute exactly $\theta_a(s + i - 1)$ to the capacity of the cut. If $x_a = l$, then the arc $(V_{h'}^a, t)$ will be cut, which will contribute exactly $\theta_a(l)$ to the capacity of the cut. The arcs with infinite capacity ensure that exactly one of the arcs from the set $(s, V_1^a) \cup \{(V_i^a, V_{i+1}^a), i = 1, \cdots, h' - 1\} \cup (V_{h'}^a, t)$ will be part of an st-cut with finite capacity, which will guarantee that we are able to obtain a valid labeling.*

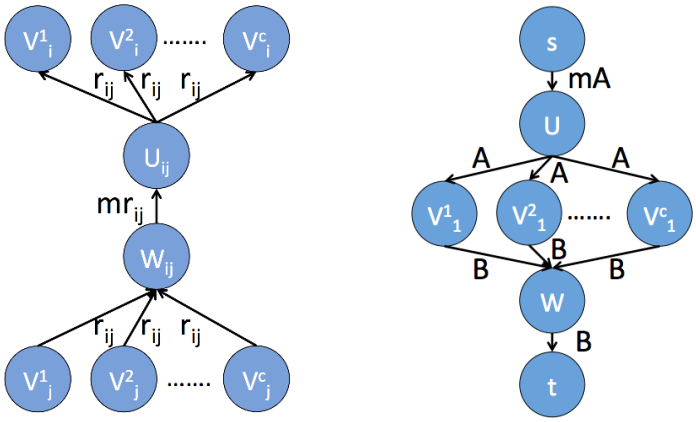


Figure 3: *Arcs used to represent the high-order potentials for the clique $\mathbf{X_c} = \{X_1, X_2, \cdots, X_c\}$.* **Left.** *The term r_{ij} is defined in equation (9). The arcs represent the sum of the m maximum convex distance functions over disjoint pairs of random variables when no random variable retains its old label. These arcs are specified only for $i \leq j$ and when either one or both of i and j are not equal to 1.* **Right.** *The terms A and B are defined in equation (10). The arcs represent an overestimation of the clique potential for the case where some or all the random variables retain their old label.*

We construct a directed graph over the set of vertices $\{s, t\} \cup \mathbf{V} \cup \mathbf{U} \cup \mathbf{W}$. The set of vertices $\mathbf{V}$ model the random variables $\mathbf{X}$. Specifically, for each random variable X_a we define $h' = l - f + 1$ vertices V_i^a where $i \in \{1, \cdots, h'\}$. The sets $\mathbf{U}$ and $\mathbf{W}$ represent *auxiliary* vertices, whose role in the graph construction will be explained later when we consider representing the high-order clique potentials. We also define a set of arcs over the vertices, where each arc has a non-negative capacity. We would like to assign arc capacities such that the st-cuts of the directed graph satisfy two properties. First, all the st-cuts with a finite capacity should include exactly one arc from the set $(s, V_1^a) \cup \{(V_i^a, V_{i+1}^a), i = 1, \cdots, h' - 1\} \cup (V_{h'}^a, t)$ for each random variable X_a. This property would allow us to define a labeling $\mathbf{y}$ such that

$$y_a = \begin{cases} 0 & \text{if the cut includes the arc } (s, V_1^a) \\ i & \text{if the cut includes the arc } (V_i^a, V_{i+1}^a) \\ h' & \text{if the cut includes the arc } (V_{h'}^a, t). \end{cases} \tag{8}$$

Second, we would like the energy of the labeling $\mathbf{y}$ defined above to be as close as possible to the capacity of the st-cut. This will allow us to obtain an optimal labeling with respect to the energy function $E'()$ by finding the st-MINCUT. We now specify the arcs and their capacities such that they satisfy the above two properties. We consider two cases: (i) arcs that represent the unary potentials; and (ii) arcs that represent the high-order clique potentials.

Representing Unary Potentials. We will represent the unary potential of X_a using the arcs specified in Figure 2. Since all the unary potentials are non-negative, it follows that the arc capacities in Figure 2 are also non-negative.

Representing Clique Potentials. Consider a set of random variables $\mathbf{X_c}$ that are used to define a high-order clique potential. Without loss of generality, we assume $\mathbf{X_c} = \{X_1, X_2, \cdots, X_c\}$. In order to represent the potential value for a putative labeling $\mathbf{x_c}$ of the clique, we introduce two types of arcs, which are depicted in Figure 3. For the arcs shown in Figure 3 (left), the capacities are specified using the term r_{ij} that is defined as follows:

$$r_{ij} = \begin{cases} \omega_{\mathbf{c}} \frac{\overline{d}(i,j)}{2} & \text{if } i = j \\ \omega_{\mathbf{c}} \overline{d}(i,j) & \text{otherwise.} \end{cases} \tag{9}$$

Here, the term $\overline{d}(i,j) = d(i-j+1) + d(i-j-1) - 2d(i-j) \geq 0$ since $d(\cdot)$ is convex, and $\omega_{\mathbf{c}} \geq 0$ by definition. It follows that $r_{ij} \geq 0$ for all $i, j \in \{1, \cdots, h'\}$. We have the following important lemma which will be helpful in the discussion of graph properties later:

Lemma 1. *Graph (a) of figure 3 exactly models clique potentials that are proportional to the sum of m maximum convex distance functions over all disjoint pairs of random variables of the clique. (Proof in Appendix B)*

For the arcs shown in Figure 3 (right), the capacities are specified using the terms A and B that are defined as follows:

$$A = \omega_{\mathbf{c}} M, B = \left(\omega_{\mathbf{c}} M - \frac{\theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}})}{m} \right). \tag{10}$$

Since $M \geq 0$, and $\theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \omega_{\mathbf{c}} m M$ due to truncation, it follows that $A, B \geq 0$.

6.2.1 Properties of the Graph

This part describes the properties of the above graph construction which will facilitate the analysis of our algorithm for the truncated max-of-linear models.

Property 1. *The cost of the $st-$cut exactly represents the sum of the unary potentials associated with the corresponding labeling f, that is, $\sum_{v_a \in \mathbf{v}} \theta_a(f(a))$.*

This property follows from the description of the graph construction for unary potentials.

Property 2. *For $c \in \mathcal{C}$, if for all $X_a \in \mathbf{X_c}$, $f(a) = f_n(a) \notin I_n$ (all variables in a clique retain their old labels), then the cost of the $st-$cut exactly represents the clique potential plus a constant $\kappa_{\mathbf{c}} = \omega_c m \cdot M - \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}})$.*

Graph (a) of figure 3 assigns 0 cost. In graph (b), the vertices $\{V_1^1, \ldots, V_1^c\} \in \mathbf{V_t}$ and hence, the $s - U$ arc of capacity mA is cut.

$$\begin{aligned} mA &= \omega_c \cdot m \cdot M \\ &= \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}}) + \omega_c m \cdot M - \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}}) \\ &= \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}}) + \kappa_{\mathbf{c}} \end{aligned}$$

Property 3. *If all variables in a clique move to a label in the current interval and $|l_c - l_1| \leq M$, then the cost of the $st-$cut exactly represents the clique potential plus a constant $\kappa_{\mathbf{c}} = \omega_c m \cdot M - \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}})$.*

As proved in lemma 1, graph (a) of figure 3 assigns exactly $\theta_{\mathbf{c}}(\mathbf{x_c})$ cost. In graph (b), the vertices $\{V_1^1, \ldots, V_1^c\} \in \mathbf{V_s}$ and hence, the $W - t$ arc of capacity B is cut, where

$$B = \omega_c \cdot m \cdot M - \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}}) = \kappa_{\mathbf{c}}$$

Property 4. *If all variables in a clique move to a label in the current interval and $|l_c - l_1| \geq M$, then the cost of the $st-$cut overestimates the clique potential, being*

$$\omega_c \sum_{i=1}^{m} d(p_i(\mathbf{x_c} - p_{c-i+1}(\mathbf{x_c})$$

plus a constant $\kappa_{\mathbf{c}} = \omega_c m \cdot M - \theta_{\mathbf{c}}(\hat{\mathbf{x}_{\mathbf{c}}})$.

Property 5. *If $k(< c)$ variables in a clique retains their old labels, then the cost of the $st-$cut overestimates the clique potential, being.*

$$\omega_c \sum_{i=1}^{k} d(p_{c-i+1}(\mathbf{x_c}) - i_n - 1) + \omega_c \cdot m \cdot M$$

plus a constant $\kappa_{\mathbf{c}} = \omega_c m \cdot M - \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}})$.

The following proposition follows from the properties of the graph:

Proposition 2. *Given a cut that partitions the vertices $\mathbf{V}$ into two disjoint sets $\mathbf{V}_s$ and $\mathbf{V}_t$, and the corresponding labeling $\mathbf{y}$ defined in equation (8), the capacity of the cut is equal to the energy $E'(\mathbf{y})$ up to a constant.*

Energy Minimization. The above proposition implies that the labeling $\mathbf{y}'$ corresponding to the st-MINCUT minimizes the energy $E'(\cdot)$ over all possible labelings $\mathbf{y} \in (\mathbf{L}')^n$. Since all the arc capacities in the graph are non-negative, the labeling $\mathbf{y}'$ can be computed efficiently by solving the st-MINCUT problem on the directed graph defined above. Once the labeling $\mathbf{y}'$ is computed, we find an approximate solution $\mathbf{x}'$ to problem (3) using equation (4).

6.3 Multiplicative Bounds

We obtain multiplicative bounds for our algorithm by making use of the fact that our algorithm terminates only when the energy cannot be reduced for any interval I_n. This implies that our algorithm has found the local minimum of the large neigbourhood defined by the intervals. We first obtain a lower bound on the reduction in energy for a given interval by making use of the graph properties. Since the final labeling f is a local minimum over the intervals, this lower bound is non-positive when the algorithm completes. This fact enables us to get an upper bound on the energy of the final labeling.

We first need to introduce some notation. Let f_n denote the labeling after the n-th iteration and $E(f_n)$ denote the corresponding energy. Also, f^* denotes optimal labeling of the MRF.

Let $r \in [0, h' - 1]$ be a uniformly distributed random integer and h' be the length of the interval. Using r we define the following set of intervals

$$\Gamma_r = \{[0, r], [r + 1, r + h'], [r + h' + 1, r + 2h'], ..., [., h - 1]\}$$

where h is the total number of labels.

Let $\mathbf{X}(f^*, I_n)$ contain all the random variables that take an optimal labeling in I_n, $\mathcal{A}(f, I_n)$ be the set of all cliques for which all variables take optimum label in the interval I_n and $\mathcal{B}(f, I_n)$ be the set of all cliques for which at least one, but not all, variable takes optimum label in the interval I_n (that is, at least one but not all variables retains old label).

In order to make the analysis more readable, the following shorthand notation is introduced:

- We denote $\omega_{\mathbf{c}} \max_{a,b \in \mathbf{X_c}} d(f^*(a) - f^*(b))$ as t_c^n
- We denote $\omega_{\mathbf{c}} \max_{a \in \mathbf{X_c}} d(f^*(a) - (i_n + 1)) + \omega_{\mathbf{c}} M$ as s_c^n

The following lemma establishes a lower bound in the reduction in energy for any given interval:

Lemma 2. *At an iteration of our algorithm, given the current labeling f_n and an interval $I_n = [i_n + 1, j_n]$, the new labeling f_{n+1} obtained by solving the st-mincut problem reduces the energy by at least the following:*

$$\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x_c})$$
$$- \left(\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right)$$

(Proof in Appendix C)

The following equation can be deduced from the above definitions:

$$\sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) = \sum_{I_n \in \mathcal{I}_r} \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) \tag{11}$$

since $f^*(a)$ belongs to exactly one interval in I_r for all X_a.

For the final labeling f of the range expansion algorithm, the term in lemma 2 should be non-positive for all intervals I_n because f is a local optimum. Hence,

$$\begin{aligned}
&\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x_c}) \\
&\leq \left(\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right), \forall I_n
\end{aligned}$$

We sum the above inequality over all $I_n \in \Gamma_r$. The summation of the LHS is at least $E(f)$. Also, using (28), the summation of the above inequality can be written as:

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \sum_{I_n \in \Gamma_r} \left(\sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right)$$

We now take the expectation of the above inequality over the uniformly distributed random integer $r \in [0, h' - 1]$. The LHS of the inequality and the first term on the RHS (that is, $\sum \theta_a(f^*(a))$) are constants with respect to r. Hence, we get

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \frac{1}{h'} \sum_r \sum_{I_n \in \Gamma_r} \left(\sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right) \tag{12}$$

For linear distance function $d(.)$, we have the following lemma:

Lemma 3. *When $d(.)$ is linear, that is, $d(x) = |x|$, the following inequality holds true:*

$$\begin{aligned}
&\frac{1}{h'} \sum_r \sum_{I_n \in \Gamma_r} \left(\sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right) \\
&\leq max \left\{ \frac{c}{2} \left(2 + \frac{h'}{M} \right), \left(2 + \frac{2M}{h'} \right) \right\} \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x_c})
\end{aligned} \tag{13}$$

where c is the largest clique in the random field (Proof in Appendix D).

Using lemma 3 in inequality 12, we obtain the multiplicative bound for max-of-linear models for $m = 1$:

Proposition 3. *The range expansion algorithm with $h' = M$ has a multiplicative bound of $O(C)$ for truncated max-of-linear model when $m = 1$. The term C equals the size of the largest clique. Hence, if $\mathbf{x}^*$ is a labeling with minimum energy and $\hat{\mathbf{x}}$ is the labeling estimated by range expansion algorithm then*

$$\sum_{a \in \mathcal{V}} \theta_a(\hat{x}_a) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \sum_{a \in \mathcal{V}} \theta_a(x_a^*) + O(C) \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}^*).$$

The above inequality holds for arbitrary set of unary potentials and non-negative clique weights (Proof in Appendix E).

We now state the generalization for Proposition 3 for any given m:

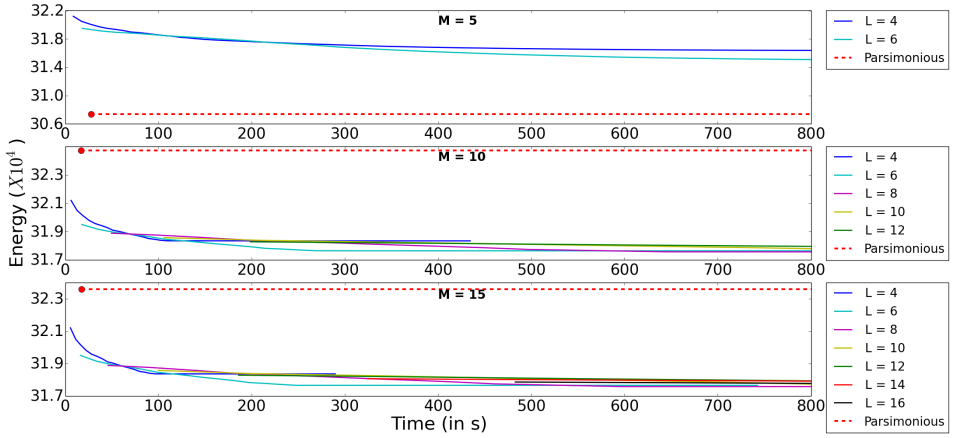


Figure 4: *Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 5$. We use truncation factors as M = 5, 10 and 15 and m = 1, and for each we vary interval lengths for our algorithm. Parsimonious labeling performs well for M = 5, but our approach outperforms for higher values of M. Red dot indicates convergence of parsimonious labeling and dotted line indicates extrapolation.*

Proposition 4. *The range expansion algorithm with $h' = M$ has a multiplicative bound of $O(m \cdot C)$ for truncated max-of-linear model for any general value of m. The term C equals the size of the largest clique. Hence, if $\mathbf{x}^*$ is a labeling with minimum energy and $\hat{\mathbf{x}}$ is the labeling estimated by range expansion algorithm then*

$$\sum_{a \in \mathcal{V}} \theta_a(\hat{x}_a) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \sum_{a \in \mathcal{V}} \theta_a(x^*_a) + O(C) \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}^*_{\mathbf{c}}).$$

The above inequality holds for arbitrary set of unary potentials and non-negative clique weights (Proof in Appendix F).

Note that for $m = 1$, the bound of the move making algorithm for parsimonious labeling (which is our baseline) is $\left(\frac{r}{r-1}\right)(C-1)O(\log h)\min(C, h)$ where C is the size of the largest clique and h is the number of labels [8]. Our algorithm gives a better bound of $O(C)$ and does not depend on the number of labels.

Similar to the case of max-of-linear models, by making use of Theorem 4 of [4], we obtain the bound for max-of-quadratic models for $m = 1$:

Proposition 5. *The range expansion algorithm with $h' = \sqrt{M}$ has a multiplicative bound of $O(C\sqrt{M})$ for the truncated max-of-quadratic model when $m = 1$.*

7 Experiments

To demonstrate the efficacy of our algorithm, we test it on the two problems of image inpainting and denoising, and stereo matching. The final labeling energy and convergence time are used as evaluation criteria. We used the parsimonious labeling algorithm of Dokania *et al.* [8] and the move-making algorithm for the co-occurrence based energy function of Ladicky *et al.* [25] as baselines. For comparison, we restrict ourselves to max-of-linear models and m = 1, as the available code for the baselines can only handle this special case. For completeness, we report the results of our range expansion algorithm for other cases of TMCM and on synthetic data as well.

7.1 Synthetic Data

Data. We generate lattices of size 100×100, where each lattice point represents a variable taking one of 20 labels. The cliques are defined as all 10×10 subwindows. The unary potentials are

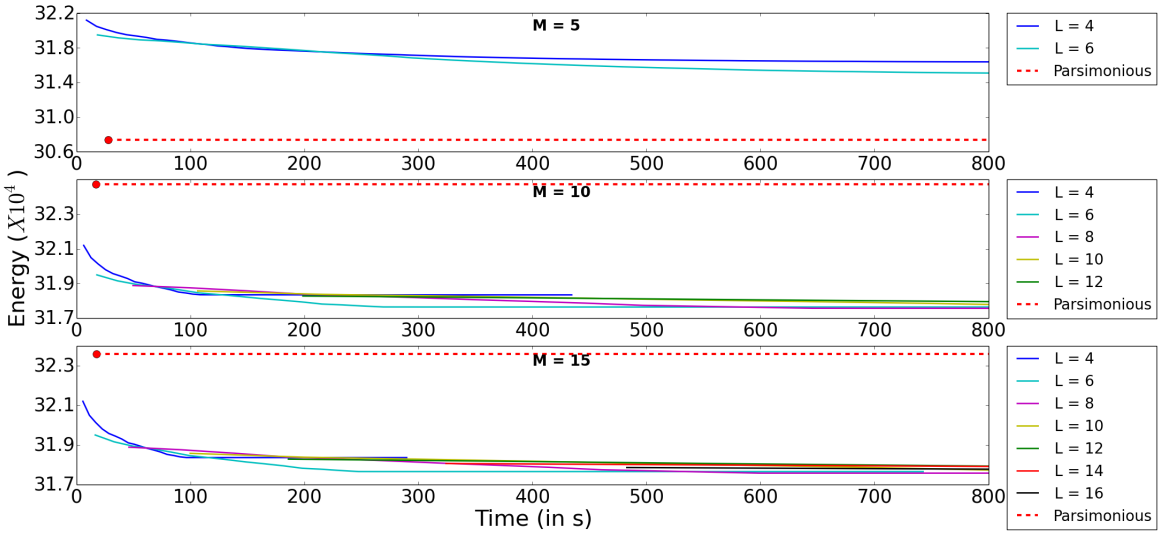


Figure 5: *Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 5$. We use truncation factors as M = 5, 10 and 15 and m = 1, and for each we vary interval lengths for our algorithm. This plot is the same as shown in the paper, but we include it here for the sake of comparison. Red dot indicates convergence of parsimonious labeling algorithm and dotted line indicates extrapolation.*

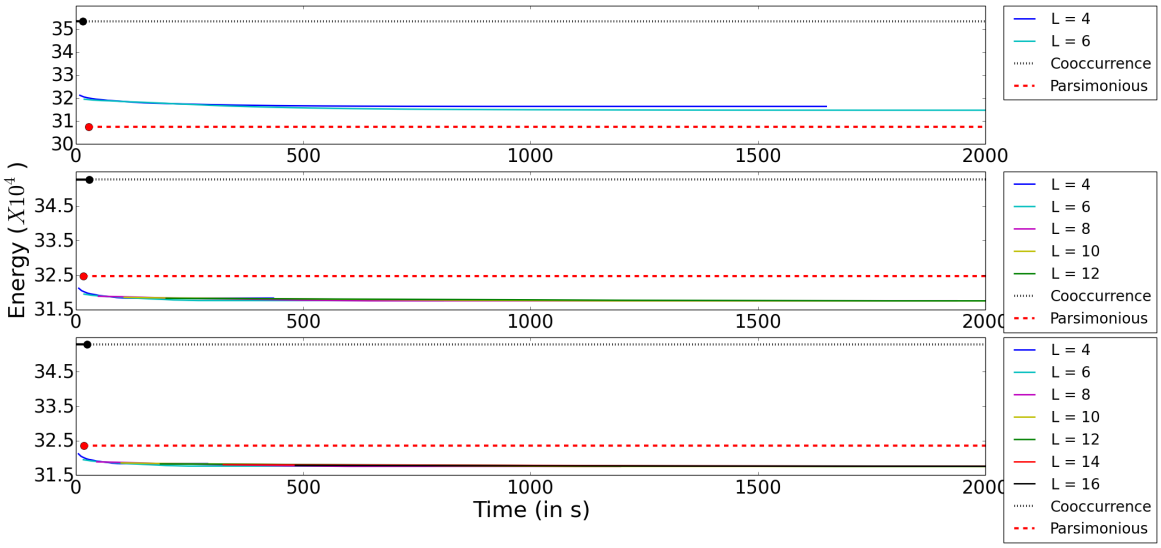


Figure 6: *Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 5$. We use truncation factors as M = 5, 10 and 15 and m = 1, and for each we vary interval lengths for our algorithm. This plot corresponds to the same experiment as mentioned in the paper, but with results for co-occurrence included. Red and black dots indicate convergence of respective algorithms and dotted line indicates extrapolation.*

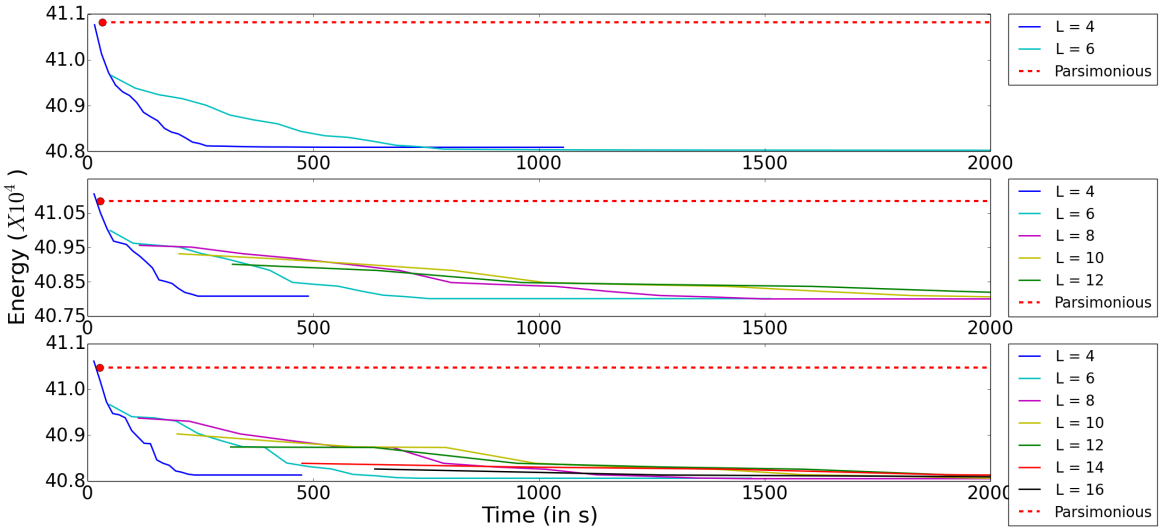


Figure 7: *Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 10$. We use truncation factors as M = 5, 10 and 15 and m = 1, and for each we vary interval lengths for our algorithm. Parsimonious labeling performs well for M = 5, but our approach outperforms for higher values of M. Red dot indicates convergence of parsimonious labeling and dotted line indicates extrapolation.*

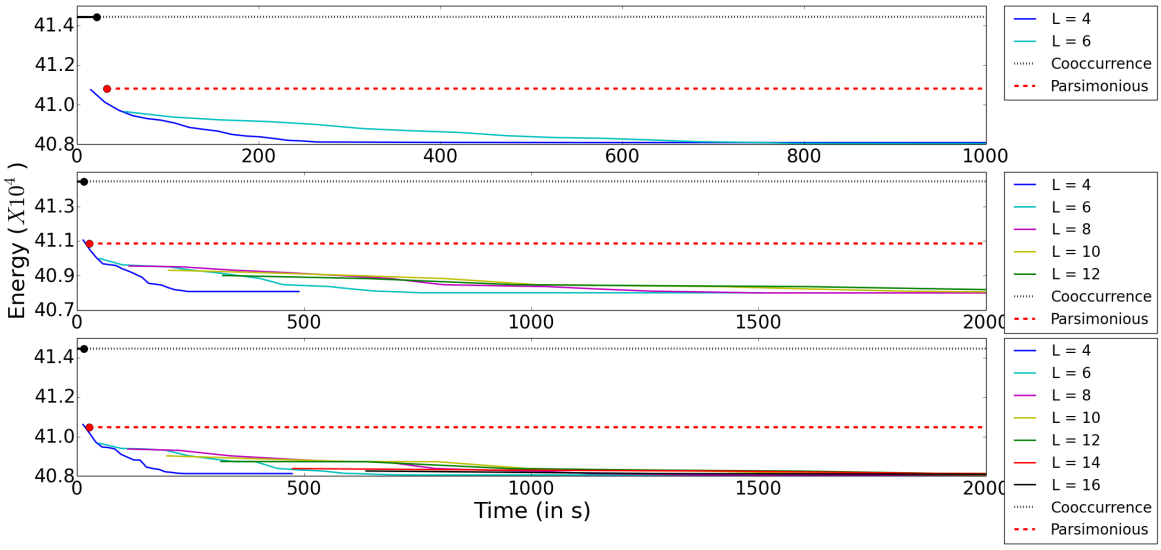


Figure 8: *Results for synthetic data using truncated linear distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 5$. We use truncation factors as M = 5, 10 and 15 and m = 1, and for each we vary interval lengths for our algorithm. This plot corresponds to the same experiment as in figure 7, but with results for co-occurrence included. Red and black dots indicate convergence of respective algorithms and dotted line indicates extrapolation.*

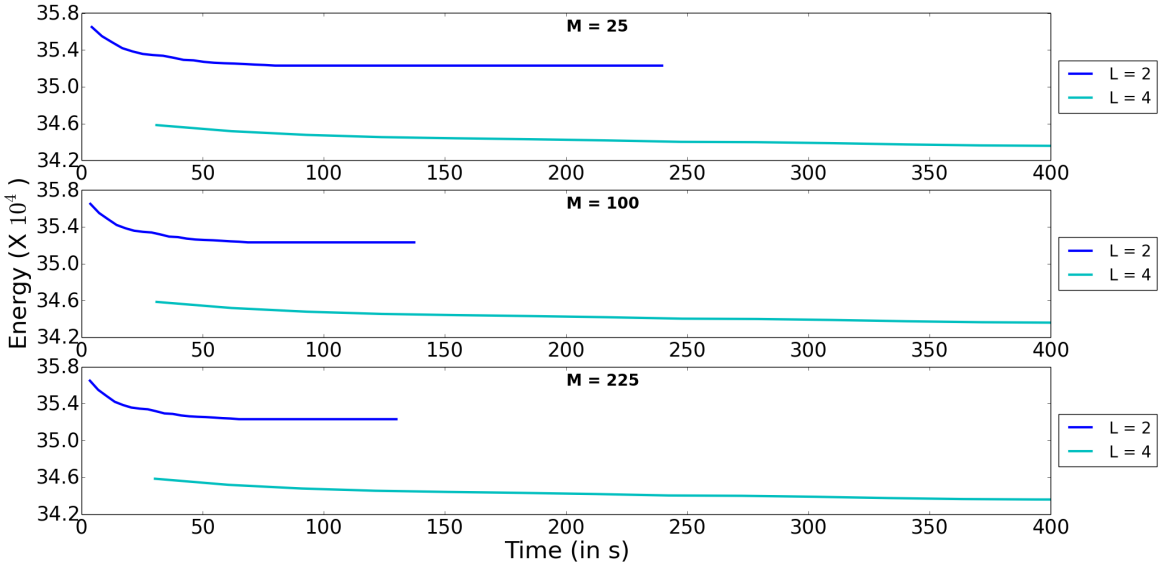


Figure 9: *Results for synthetic data using truncated quadratic distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 3$. We use M = 25, 100 and 225 and m = 1, and for each we vary interval lengths for our algorithm.*

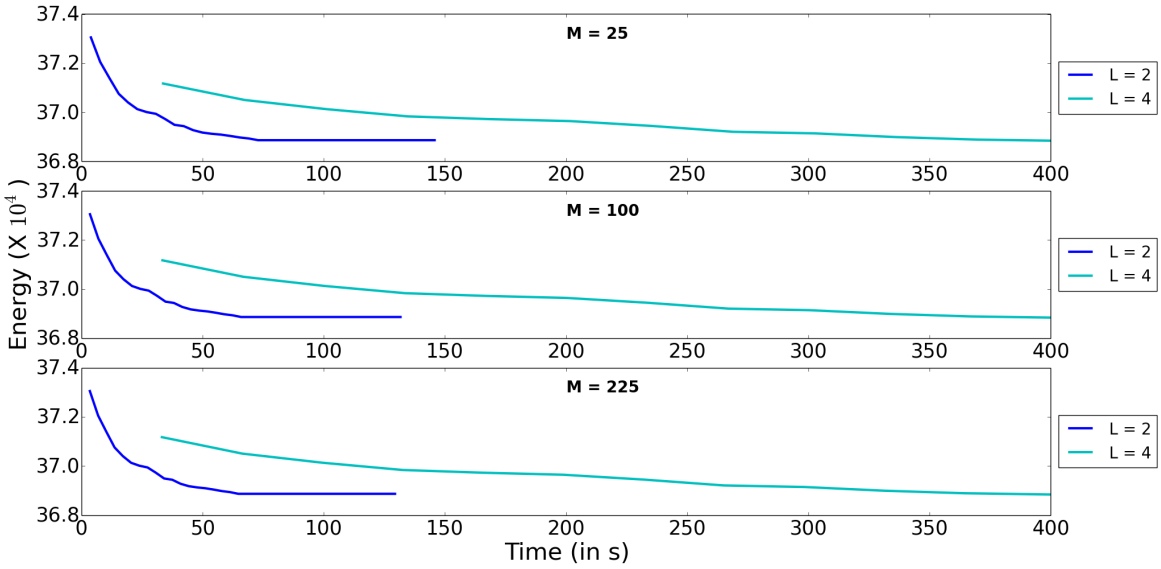


Figure 10: *Results for synthetic data using truncated quadratic distance function. The plots show the variation of energy versus time, averaged over 50 lattices using $\omega_c = 5$. We use M = 25, 100 and 225, and for each we vary interval lengths for our algorithm.*

uniformly sampled in the range [1, 100]. The truncation factors for linear case are $M \in \{5, 10, 15\}$ and for quadratic case $M \in \{25, 100, 225\}$. In each energy function, all cliques were assumed to have the same clique weight, being 5 and 10 for linear model and 3 and 5 for quadratic model. m varies in the range $\{1, 3, 5\}$.

Method. For each energy function obtained by a particular setting of the above parameters, we vary the interval length up to M + 1. We run [8] and [25] only for linear distance function and m = 1 and repeat the experiments for 50 randomly generated unaries for linear and quadratic cases.

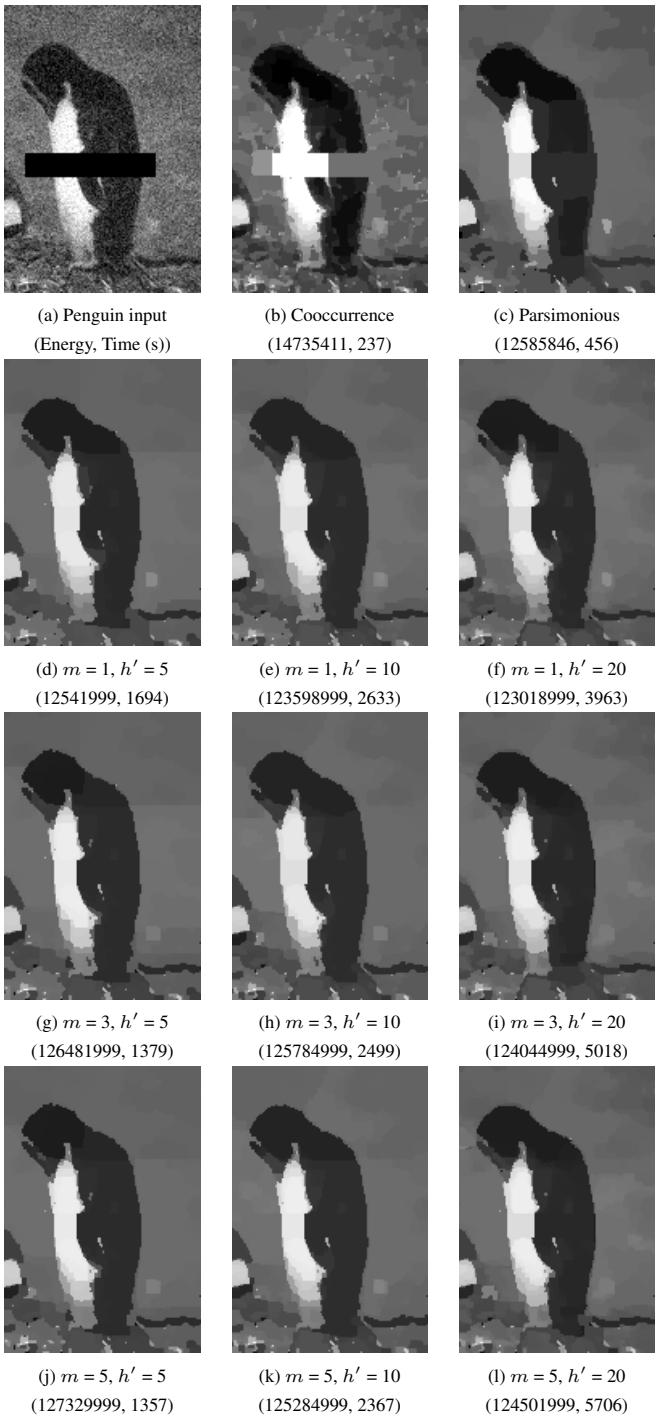
Results. The plots in Figure 6 and 5 show the average energy as a function of average time for our algorithm and the baselines using weight as 5 for max-of-linear models, with and without the results of [25] respectively. We see that the energy values of [25] are much higher as compared to our algorithm. Our algorithm gives lower energy solutions than parsimonious labeling method except when the truncation factor is low (M = 5). The plots in Figures 8 and 7 shows the results for max-of-linear models using weight as 10 with and without the cooccurrence results respectively. Our algorithm gives better results than the baselines for all M. Both the baselines converge faster than our method. In practice, intervals smaller than the optimum size give almost equally good results and converge faster. Figure 10 shows the plot for max-of-quadratic cases using weight as 5.

7.2 Image Inpainting and Denoising

Data. Given an image with noise and obscured/inpainted regions (regions with missing pixels), the task is to denoise it and fill the obscured regions in a way that is consistent with the surrounding regions. The images 'house' and 'penguin' from the Middlebury data set were used for the experiments. Since the images are grayscale, they have 256 labels in the interval [0, 255], each representing an intensity value. The unary potential for each pixel corresponding to a particular label equals the squared difference between the label and the intensity value at the pixel. We use high-order cliques as the super-pixels obtained using the mean-shift method [5]. The parameters ω_c, M and m are varied to give different truncated max-of-linear energy functions.

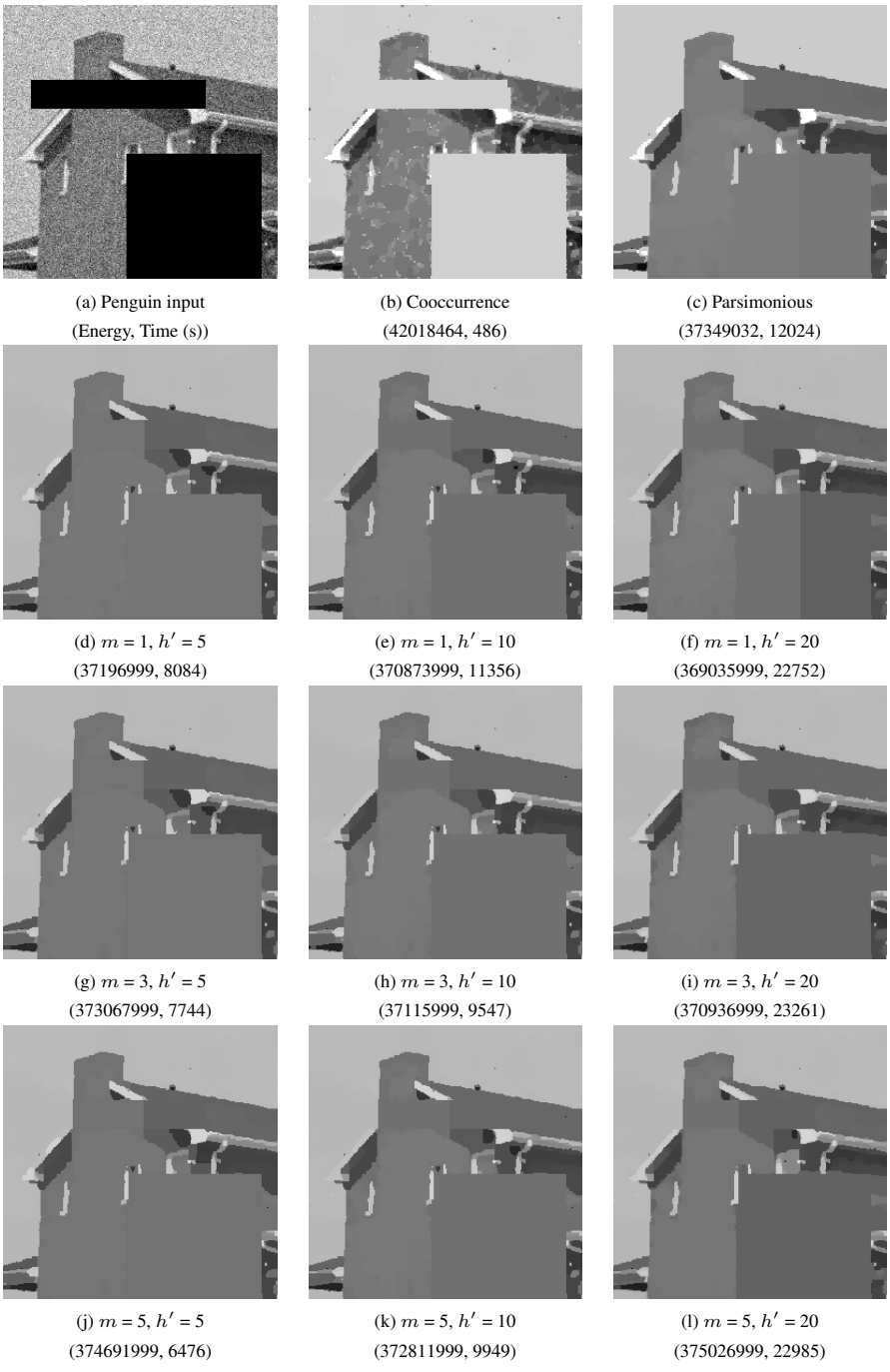
Method. For each parameter setting of ω_c, M and m, we vary the interval lengths for our algorithm and make a comparison with the baselines.

Results. Results for ω_c = 40, M = 40, and m = 1, 3 and 5 for 'penguin' and ω_c = 50, M = 50, and m = 1, 3 and 5 for 'house' are shown in Figure 12 for varying interval lengths $h' = \{5, 10, 20\}$. Our algorithm consistently gives lower energy labeling as compared to both [8] and [25] even for small h'. For 'penguin', our algorithm gives cleaner denoised image, preserving edges and details. On the other hand, both [8] and [25] exhibit significant blocking effect. Moreover, the output is more natural for m = 3 as compared to m = 1. Even for 'house', our output looks more visually appealing as compared to baselines.



(a) Penguin input (Energy, Time (s))
(b) Cooccurrence (14735411, 237)
(c) Parsimonious (12585846, 456)
(d) $m = 1$, $h' = 5$ (12541999, 1694)
(e) $m = 1$, $h' = 10$ (123598999, 2633)
(f) $m = 1$, $h' = 20$ (123018999, 3963)
(g) $m = 3$, $h' = 5$ (126481999, 1379)
(h) $m = 3$, $h' = 10$ (125784999, 2499)
(i) $m = 3$, $h' = 20$ (124044999, 5018)
(j) $m = 5$, $h' = 5$ (127329999, 1357)
(k) $m = 5$, $h' = 10$ (125284999, 2367)
(l) $m = 5$, $h' = 20$ (124501999, 5706)

Figure 11: *Image inpainting results for 'penguin'. Note that comparison with (b) and (c) makes sense only for $m = 1$. Also, we restricted our experiments to smaller (and suboptimal) h' due to computational issues.*



(a) Penguin input
(Energy, Time (s))

(b) Cooccurrence
(42018464, 486)

(c) Parsimonious
(37349032, 12024)

(d) $m = 1$, $h' = 5$
(37196999, 8084)

(e) $m = 1$, $h' = 10$
(370873999, 11356)

(f) $m = 1$, $h' = 20$
(369035999, 22752)

(g) $m = 3$, $h' = 5$
(373067999, 7744)

(h) $m = 3$, $h' = 10$
(37115999, 9547)

(i) $m = 3$, $h' = 20$
(370936999, 23261)

(j) $m = 5$, $h' = 5$
(374691999, 6476)

(k) $m = 5$, $h' = 10$
(372811999, 9949)

(l) $m = 5$, $h' = 20$
(375026999, 22985)

Figure 12: *Image inpainting results for 'house'. Note that comparison with (b) and (c) makes sense only for m = 1. Also, we restricted our experiments to smaller (and suboptimal) h' due to computational issues.*

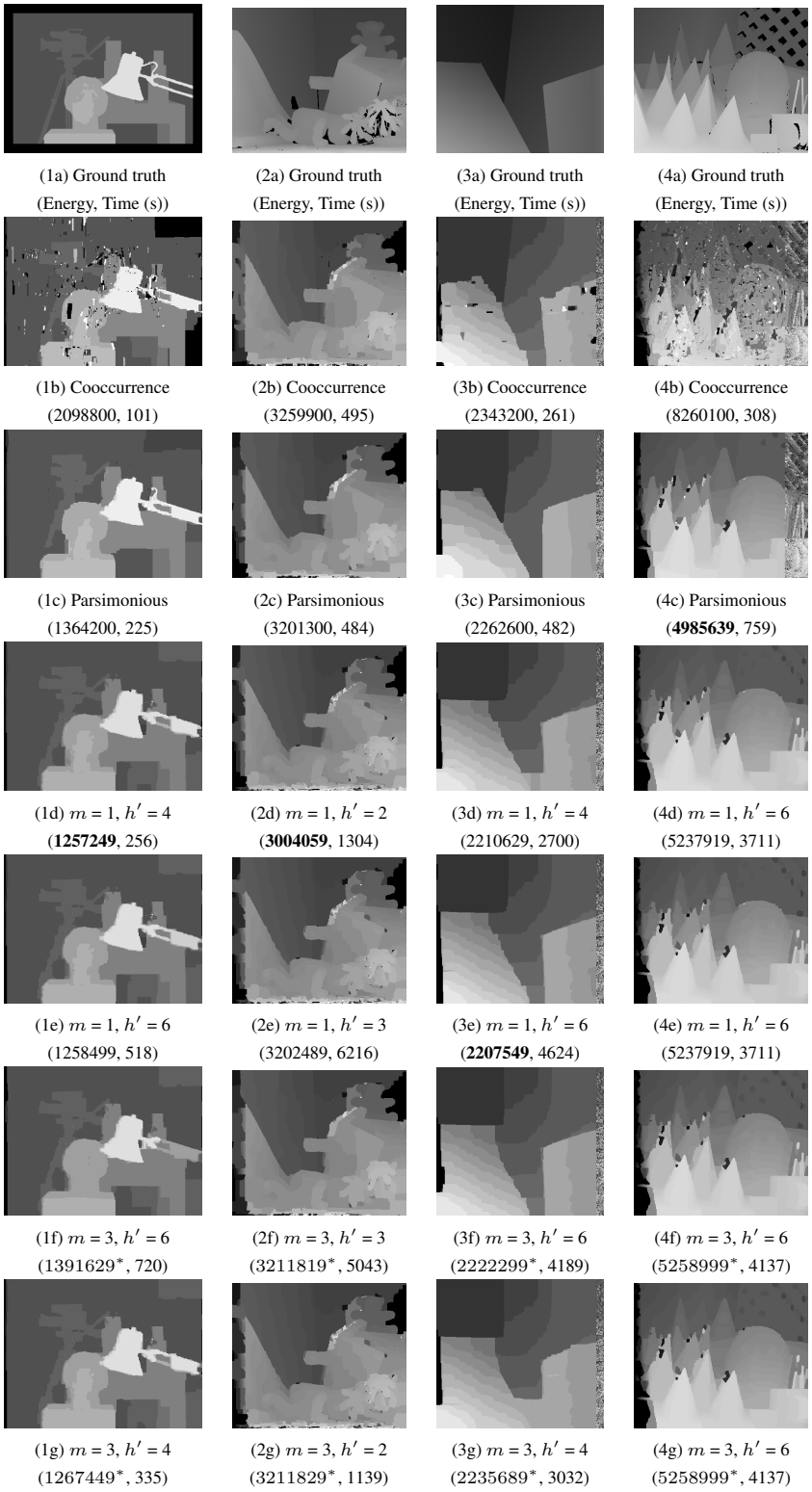
7.3 Stereo Matching

Data. In the stereo matching problem, we have two rectified images of the same scene from two cameras set slightly apart. We are required to estimate the horizontal disparity between a pixel in the right camera image from the corresponding pixel in the left camera. We use 'tsukuba' and 'teddy' data sets from the Middlebury stereo collection for our experiments. In each case, we have a pair of RGB images and ground truth disparities. We assume the unary potentials to be the $L1$-norm of the difference in RGB values of the left and right image pixels. There are 16 labels and 60 labels for 'tsukuba' and 'teddy' respectively. The high-order cliques are super-pixels obtained using mean-shift method [5]. The parameters ω_c, M and m are varied to give different truncated max-of-linear energy functions.

Method. For each parameter setting of ω_c, M and m, we vary the interval lengths for our algorithm and make a comparison with the baselines.

Results. Results for ω_c = 20, M = 5, and m = 1 and 3 for 'tsukuba', 'venus' and 'cones' and ω_c = 20, M = 1, and m = 1 and 3 for 'teddy' are shown in Figure 13. We used interval length h' as 4 for 'tsukuba' and 'venus', 6 for 'cones' and 2 for 'teddy'. Apart from 'cones', our algorithm consistently gives lower energy labeling as compared to both [8] and [25]. For 'tsukuba', our algorithm captures the details of the face better than [8] and [25]. For 'venus', we get smoother labeling for the front plane. Moreover, our results for m = 3 exhibit robustness to inaccurate clique definitions.

Results. Results for ω_c = 20, M = 5, and m = 1 and 3 for 'tsukuba', 'venus' and 'cones', and ω_c = 20, M = 1, and m = 1 and 3 for 'teddy' are shown in Figure 13. We show the results for interval length 3 for 'teddy', and 6 for others. Note that in our main paper, we used interval length h' as 4 for 'tsukuba' and 'venus', and 1 for 'teddy' and did not show the results for 'cones'. Apart from 'cones', our algorithm consistently gives lower energy labeling as compared to both [8] and [25].



(1a) Ground truth (Energy, Time (s)) | (2a) Ground truth (Energy, Time (s)) | (3a) Ground truth (Energy, Time (s)) | (4a) Ground truth (Energy, Time (s))

(1b) Cooccurrence (2098800, 101) | (2b) Cooccurrence (3259900, 495) | (3b) Cooccurrence (2343200, 261) | (4b) Cooccurrence (8260100, 308)

(1c) Parsimonious (1364200, 225) | (2c) Parsimonious (3201300, 484) | (3c) Parsimonious (2262600, 482) | (4c) Parsimonious (**4985639**, 759)

(1d) $m = 1$, $h' = 4$ (**1257249**, 256) | (2d) $m = 1$, $h' = 2$ (**3004059**, 1304) | (3d) $m = 1$, $h' = 4$ (2210629, 2700) | (4d) $m = 1$, $h' = 6$ (5237919, 3711)

(1e) $m = 1$, $h' = 6$ (1258499, 518) | (2e) $m = 1$, $h' = 3$ (3202489, 6216) | (3e) $m = 1$, $h' = 6$ (**2207549**, 4624) | (4e) $m = 1$, $h' = 6$ (5237919, 3711)

(1f) $m = 3$, $h' = 6$ (1391629^*, 720) | (2f) $m = 3$, $h' = 3$ (3211819^*, 5043) | (3f) $m = 3$, $h' = 6$ (2222299^*, 4189) | (4f) $m = 3$, $h' = 6$ (5258999^*, 4137)

(1g) $m = 3$, $h' = 4$ (1267449^*, 335) | (2g) $m = 3$, $h' = 2$ (3211829^*, 1139) | (3g) $m = 3$, $h' = 4$ (2235689^*, 3032) | (4g) $m = 3$, $h' = 6$ (5258999^*, 4137)

Figure 13: *Stereo matching results: Figures (1a), (1b), (1c) and (1d) are the ground truth disparity for 'tsukuba', 'teddy', 'venus' and 'cones' respectively. Our results for $m = 1$ (1d), (2d), and (3d) are significantly better than those of [25] (1b), (2b) and (3b), and of [8] (1c), (2c) and (3c) in terms of energy. For 'cones', [8] (4c) performs better than our algorithm. We also show results for $m = 3$. We use super-pixels obtained using mean-shift as cliques. *Note that $m = 3$ uses a different energy function from other cases.*

8 Discussion

We proposed a novel family of high-order random fields called truncated max-of-convex models (TMCM) which are generalization of truncated convex models (TCM). To perform inference in TMCM, we developed a novel range expansion algorithm for energy minimization that retains the efficiency of st-MINCUT and provides provably accurate solutions. The algorithm relies on an exact graph representation of max-of-convex models, a tight submodular overestimate of the energy function for any interval length and a graph construction that represents this overestimate, allowing the inference problem to be solved using st-MINCUT. From a theoretical point of view, our work can be thought of as a step towards the identification of graph representable submodular functions and automated construction of graphs for such functions.

References

[1] C. Arora and S. Maheshwari. Multi label generic cuts: Optimal infernce in multi label multi clique MRF-MAP problems. In *CVPR*, 2014.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[4] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *SODA*, 2001.

[5] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[6] A. Delong, L. Gorelick, O. Veksler, and Y. Boykov. Minimizing energies with hierarchical costs. *IJCV*, 2012.

[7] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *CVPR*, 2010.

[8] P. Dokania and M. P. Kumar. Parsimonious labeling. In *ICCV*, 2015.

[9] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 2006.

[10] A. Fix, C. Wang, and R. Zabih. A primal-dual algorithm for higher-order multilabel Markov random fields. In *CVPR*, 2014.

[11] S. Gould, F. Amat, and D. Koller. Alphabet soup: A framework for approximate energy minimization. In *CVPR*, 2009.

[12] A. Gupta and E. Tardos. A constant factor approximation algorithm for a class of classification problems. In *STOC*, 2000.

[13] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *PAMI*, 2003.

[14] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR*, 2011.

[15] P. Kohli, M. P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[16] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[17] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006.

[18] V. Kolmogorov. Minimizing a sum of submodular functions. *Discrete Applied Mathematics*, 2012.

[19] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006.

[20] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[21] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *CVPR*, 2007.

[22] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[23] M. P. Kumar and P. Torr. Improved moves for truncated convex models. In *NIPS*, 2008.

[24] M Pawan Kumar, Olga Veksler, and Philip HS Torr. Improved moves for truncated convex models. *JMLR*, 2011.

[25] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[26] Ofer Meshi, David Sontag, Tommi S. Jaakkola, and Amir Globerson. Learning efficiently with approximate inference via dual losses. In *ICML*, 2010.

[27] David Sontag, Talya Meltzer, Amir Globerson, Tommi S Jaakkola, and Yair Weiss. Tightening LP relaxations for map using message passing. In *UAI*, 2008.

[28] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *PAMI*, 2008.

[29] Daniel Tarlow, Inmar E Givoni, and Richard S Zemel. Hop-map: Efficient message passing with high order potentials. In *AISTATS*, 2010.

[30] O. Veksler. Graph cut based optimization for MRFs with truncated convex priors. In *CVPR*, 2007.

[31] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Trans. Information Theory*, 2005.

A Proof of Proposition 1

Proposition 1: $\delta_{a,b}(y_a, y_b)$ is submodular in the sense of label-set encoding used in [13]and is an overestimate of the truncated convex distance $\min\{d(x_a - x_b), M\}$.

Proof: Note that to prove that $\delta_{a,b}(y_a, y_b)$ is submodular, we need to show that the following inequality holds:

$$\delta_{a,b}(y_a, y_b) + \delta_{a,b}(y_a + 1, y_b + 1) \leq \delta_{a,b}(y_a + 1, y_b) + \delta_{a,b}(y_a, y_b + 1) \tag{14}$$

We consider the 4 cases corresponding to equation 6.

Case 1: $y_a = y_b = 0$

$$\begin{aligned} \delta_{a,b}(y_a, y_b) = \delta_{a,b}(0, 0) &= \min\{d(\hat{x}_a - \hat{x}_b), M\} \\ \min\{d(x_a - x_b), M\} &= \min\{d(\hat{x}_a - \hat{x}_b), M\} \end{aligned}$$

Hence, $\delta_{a,b}(y_a, y_b) = \min\{d(x_a - x_b), M\}$.

To prove submodularity, consider L.H.S of inequality 14:

$$\begin{aligned} \delta_{a,b}(y_a, y_b) + \delta_{a,b}(y_a + 1, y_b + 1) &= \delta_{a,b}(0, 0) + \delta_{a,b}(1, 1) \\ &= \min\{d(\hat{x}_a - \hat{x}_b), M\} + 0 \end{aligned}$$

The R.H.S of inequality 14 is given by:

$$\begin{aligned} \delta_{a,b}(y_a + 1, y_b) + \delta_{a,b}(y_a, y_b + 1) &= \delta_{a,b}(1, 0) + \delta_{a,b}(0, 1) \\ &= M + d(y_b - 1) + M + d(y_a - 1) \end{aligned}$$

Clearly L.H.S $<$ R.H.S for this case.

Case 2: $y_a = 0, y_b \neq 0$

$$\begin{aligned} \delta_{a,b}(y_a, y_b) = \delta_{a,b}(0, y_b) &= M + d(y_b - 1) \\ \min\{d(x_a - x_b), M\} &= \min\{d(\hat{x}_a - x_b), M\} \end{aligned}$$

Hence, $\delta_{a,b}(y_a, y_b) \geq \min\{d(x_a - x_b), M\}$.

For submodularity, L.H.S of inequality 14 is given by:

$$\begin{aligned} \delta_{a,b}(y_a, y_b) + \delta_{a,b}(y_a + 1, y_b + 1) &= \delta_{a,b}(0, y_b) + \delta_{a,b}(1, y_b + 1) \\ &= M + d(y_b - 1) + d(y_b) \end{aligned}$$

The R.H.S of inequality 14 can be written as:

$$\begin{aligned}\delta_{a,b}(y_a+1,y_b)+\delta_{a,b}(y_a,y_b+1) &= \delta_{a,b}(1,y_b)+\delta_{a,b}(0,y_b+1)\\ &= d(y_b-1)+M+d(y_b)\end{aligned}$$

Hence L.H.S = R.H.S for this case.

Case 3: $y_a \neq 0, y_b = 0$

$$\begin{aligned}\delta_{a,b}(y_a,y_b) &= \delta_{a,b}(y_a,0) = M+d(y_a-1)\\ \min\{d(x_a-x_b),M\} &= \min\{d(x_a-\hat{x}_b),M\}\end{aligned}$$

Hence, $\delta_{a,b}(y_a,y_b) \geq \min\{d(x_a-x_b),M\}$.

To prove submodularity, consider L.H.S of inequality 14:

$$\begin{aligned}\delta_{a,b}(y_a,y_b)+\delta_{a,b}(y_a+1,y_b+1) &= \delta_{a,b}(y_a,0)+\delta_{a,b}(y_a+1,1)\\ &= M+d(y_a-1)+d(y_a)\end{aligned}$$

The R.H.S of inequality 14 is given by:

$$\begin{aligned}\delta_{a,b}(y_a+1,y_b)+\delta_{a,b}(y_a,y_b+1) &= \delta_{a,b}(y_a+1,0)+\delta_{a,b}(y_a,1)\\ &= M+d(y_a)+d(y_a-1)\end{aligned}$$

Hence L.H.S = R.H.S for this case.

Case 4: $y_a \neq 0, y_b \neq 0$

$$\delta_{a,b}(y_a,y_b) = d(y_a-y_b)$$

Hence, $\delta_{a,b}(y_a,y_b) \geq \min\{d(x_a-x_b),M\}$.

To prove submodularity, consider L.H.S of inequality 14:

$$\delta_{a,b}(y_a,y_b)+\delta_{a,b}(y_a+1,y_b+1) = 2\cdot d(y_a-y_b)$$

The R.H.S of inequality 14 is given by:

$$\delta_{a,b}(y_a+1,y_b)+\delta_{a,b}(y_a,y_b+1) = d(y_a-y_b-1)+d(y_a-y_b+1)$$

Since $d()$ is convex, L.H.S $\leq$ R.H.S for this case.

This completes our proof of proposition 1. ■

B Proof of Lemma 1

Lemma 1: Graph (a) of figure 3 exactly models clique potentials that are proportional to the sum of m maximum convex distance functions over all disjoint pairs of random variables of the clique.

Proof: Let us arrange the labels assigned to $\mathbf{X_c}$ in increasing order as $\{l_{k_1}, \ldots, l_{k_c}\}$, where $c = |\mathbf{X_c}|$ indicates the size of the clique. The sum of m maximum convex distance functions over all disjoint pairs of random variables of the clique is given by:

$$\omega_{\mathbf{c}}(d(k_c - k_1) + d(k_{c-1} - k_2) + ... + d(k_{c-m+1} - k_m)) \tag{15}$$

When $i \in [k_p+1, k_{p+1}]$, exactly p variables in $\mathbf{x_c}$ are assigned a label between $[l_{k_1}, l_{k_p}]$ and the corresponding p vertices among $\{V_i^1, V_i^2....V_i^c\}$ will be in $\mathbf{V_t}$ (the remaining $c - t$ vertices will be in $\mathbf{V_s}$). Similarly, when $j \in [k_{c-p} + 1, k_{c-p+1}]$, exactly p variables in $\mathbf{x_c}$ are assigned a label in the range $[l_{k_{c-p+1}}, l_{k_c}]$ and the corresponding p vertices among $\{V_j^1, V_j^2....V_j^c\}$ will be in $\mathbf{V_s}$ (the remaining $c - p$ vertices will be in $\mathbf{V_t}$). st-MINCUT includes either the arcs (U_{ij}, V_i^a) for all $V_i^a \in \mathbf{V_t}$ or the arcs (V_j^a, W_{ij}) for all $V_j^a \in \mathbf{V_s}$ or the arc (W_{ij}, U_{ij}), whichever contribute the smallest cost to the cut. Let C_{ij} denote the sum of capacitites of arcs cut by st-MINCUT in graph (a) of figure 3 for a given pair i, j. Clearly, $0 \leq C_{ij} \leq m \cdot r_{ij}$.

Let us partition the interval $[k_1 + 1, k_c]$ to $2m - 1$ intervals $[k_1 + 1, k_2], [k_2 + 1, k_3], ...[k_m + 1, k_{c-m+1}]..., [k_{c-2} + 1, k_{c-1}], [k_{c-1} + 1, k_c]$. C_{ij} depends on the intervals in which i and j lie. Given a labeling $\mathbf{x_c}$, the total contribution of all the cut arcs is equal to

$$\begin{aligned}\sum_{i=k_1+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} = & \sum_{i=k_1+1}^{k_2} \sum_{j=i}^{k_c} C_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} C_{ij} \\ & + \cdots + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_c} C_{ij} + \cdots + \sum_{i=k_{c-1}+1}^{k_c} \sum_{j=i}^{k_c} C_{ij}\end{aligned} \tag{16}$$

Consider the second term of the series in (16). When $i \in [k_2 + 1, k_3]$, exactly 2 variables in $\mathbf{X_c}$ take up a label in the range $[k_1, k_2]$ and hence, two vertices among $\{V_i^1, V_i^2....V_i^c\}$ belong to $\mathbf{V_t}$. When $j \in [k_{c-1} + 1, k_c]$, only one vertex among $\{V_j^1, V_j^2....V_j^c\}$ belongs to $\mathbf{V_s}$, giving C_{ij} as r_{ij}. For j in all other intervals, at least two vertices among $\{V_j^1, V_j^2....V_j^c\}$ belong to $\mathbf{V_s}$, and C_{ij} equals $2 \cdot r_{ij}$. Thus, the following holds

$$\begin{aligned}\sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_2+1}^{k_3} \sum_{j=k_{c-1}+1}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} 2r_{ij} \\ &= \sum_{i=k_2+1}^{k_3} \sum_{j=k_{c-1}+1}^{k_c} r_{ij} + \left(\sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} \right) \\ &= \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij}\end{aligned}$$

where the last step is obtained by combining the first and the third terms in the summation.

In general, for $p \in [1, m-1]$ we can write

$$\sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_c} C_{ij} = \sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_{c-1}} r_{ij} + \cdots + \sum_{i=k_p+1}^{k_{p+1}} \sum_{j=i}^{k_{c-p+1}} r_{ij} \tag{17}$$

Similar argument can be extended for $p \in [c-m+1, c-1]$

Using (17), each term of (16) can be written as

$$\begin{aligned}
\sum_{i=k_1+1}^{k_2} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_1+1}^{k_2} \sum_{j=i}^{k_c} r_{ij} \\
\sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_3} \sum_{j=i}^{k_{c-1}} r_{ij} \\
\sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_{c-1}} r_{ij} + \sum_{i=k_3+1}^{k_4} \sum_{j=i}^{k_{c-2}} r_{ij} \\
&\vdots \\
\sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_{c-1}} r_{ij} + \cdots\cdots + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_{c-m+1}} r_{ij} \\
&\vdots \\
\sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_{c-1}} r_{ij} + \sum_{i=k_{c-3}+1}^{k_{c-2}} \sum_{j=i}^{k_{c-2}} r_{ij} \\
\sum_{i=k_{c-2}+1}^{k_{c-1}} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_{c-2}+1}^{k_{c-1}} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_{c-2}+1}^{k_{c-1}} \sum_{j=i}^{k_{c-1}} r_{ij} \\
\sum_{i=k_{c-1}+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_{c-1}+1}^{k_c} \sum_{j=i}^{k_c} r_{ij}
\end{aligned}$$

Adding all the above equations by adding all terms vertically, we obtain

$$\begin{aligned}
\sum_{i=k_1+1}^{k_c} \sum_{j=i}^{k_c} C_{ij} &= \sum_{i=k_1+1}^{k_c} \sum_{j=i}^{k_c} r_{ij} + \sum_{i=k_2+1}^{k_{c-1}} \sum_{j=i}^{k_{c-1}} r_{ij} + \cdots + \sum_{i=k_m+1}^{k_{c-m+1}} \sum_{j=i}^{k_{c-m+1}} r_{ij} \\
&= \omega_{\mathbf{c}} d(k_c - k_1) + \omega_{\mathbf{c}} d(k_{c-1} - k_2) + \cdots + \omega_{\mathbf{c}} d(k_{c-m+1} - k_m)
\end{aligned}$$

Hence, graph (a) of figure 3 represents exactly clique potentials that are proportional to the sum of m maximum convex distance functions over all disjoint pairs of random variables of the clique. ■

C Proof of Lemma 2

Lemma 2: At an iteration of our algorithm, given the current labeling f_n and an interval $I_n = [i_n + 1, j_n]$, the new labeling f_{n+1} obtained by solving the st-mincut problem reduces the energy by at least the following:

$$\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x_c}) - \left(\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right)$$

Proof: It is evident from the properties of the graph discussed in subsection 6.2.1 that the energy of the new labeling f_{n+1} is bounded from above by the cost of the st-mincut. The cost of the st-mincut itself is bounded from above by the cost of any other st-cut in the graph. Consider one such st-cut that gives the following labeling:

$$f(a) = \begin{cases} f^*(a) & if \;\; X_a \in \mathbf{X}(f^*, I_n) \\ f_n(a) & otherwise. \end{cases}$$

We can derive the cost of this st-cut using the properties of subsection 6.2.1. The energy of f equals the sum of the terms in the properties minus $\sum_{c \in \mathcal{C}} \kappa_c$. The energy of f_{n+1} is less than that of f. Hence, the difference between the energy of the current labeling f_n and the new labeling f_{n+1}, i.e $E(f_n) - E(f_{n+1})$, is at least

$$\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x_c}) - \left(\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_c^n + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s_c^n \right)$$

This proves the lemma. ■

D Proof of Lemma 3

Lemma 3: When $d(.)$ is linear, that is, $d(x) = |x|$, the following inequality holds true:

$$\begin{aligned} &\frac{1}{L}\sum_{r}\sum_{I_n\in\Gamma_r}\left(\sum_{\mathbf{X_c}\in\mathcal{A}(f,I_n)} t_c^n + \sum_{\mathbf{X_c}\in\mathcal{B}(f,I_n)} s_c^n\right) \\ &\leq max\left\{\frac{c}{2}\left(2+\frac{L}{M}\right), \left(2+\frac{2M}{L}\right)\right\}\sum_{\mathbf{c}\in\mathcal{C}}\theta_{\mathbf{c}}(\mathbf{x_c}) \end{aligned} \tag{18}$$

where c is the largest clique in the random field.

Proof: Let us denote the set of optimum labels in a clique $\mathbf{X_c}$ arranged in an increasing order as $l_1, l_2 ....., l_{c-1}, l_c$. For notational simplicity, we introduce a new notation $L = h'$. Since we are dealing with the truncated linear metric, the terms t_c^n and s_c^n can be simplified as:

$$t_c^n = \omega_c(l_c - l_1), s_c^n = \omega_c(l_c - i_n - 1 + M) \tag{19}$$

The LHS of inequality in lemma 3 can be written as:

$$\frac{1}{L}\sum_{\mathbf{c}\in\mathcal{C}}\left(\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} t_c^n + \sum_{\mathcal{B}(f,I_n)\ni\mathbf{c}} s_c^n\right) \tag{20}$$

In order to prove the lemma, we consider the following three cases for each clique $\mathbf{X_c}$:

$Case - I : |l_c - l_1| \geq L$ and hence $\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$

In this case, $\mathbf{X_c} \notin \mathcal{A}(f, I_n)$ for all intervals I_n since the length of each interval is L. Moreover, the conditions for $\mathbf{X_c} \in \mathcal{B}(f, I_n)$ are given by

$$\mathbf{X_c} \in \mathcal{B}(f, I_n) \iff i_n \in [l_1 - L, l_c - 1]$$

We observe that

$$\begin{aligned} &\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} t_c^n + \sum_{\mathcal{B}(f,I_n)\ni\mathbf{c}} s_c^n \\ &\leq \omega_c \sum_{p=1}^{c}\sum_{l_p-L}^{l_p-1}(M + l_p - i_n - 1) \\ &= \omega_c \sum_{p=1}^{c}\left(L\cdot M + \frac{L\cdot(L-1)}{2}\right) \\ &= L\cdot\frac{c}{2}\left(2+\frac{L-1}{M}\right)\theta_{\mathbf{c}}(\mathbf{x_c}) \\ &\leq L\cdot\frac{c}{2}\left(2+\frac{L}{M}\right)\theta_{\mathbf{c}}(\mathbf{x_c}) \end{aligned} \tag{21}$$

$Case - II : M \leq |l_c - l_1| < L$ and hence $\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$
In this case, the conditions for $\mathbf{X_c} \in \mathcal{A}(f, I_n)$ and $X_c \in \mathcal{B}(f, I_n)$ are given by

$$\begin{aligned}
\mathbf{X_c} \in \mathcal{A}(f, I_n) &\iff i_n \in [l_c - L, l_1 - 1] \\
\mathbf{X_c} \in \mathcal{B}(f, I_n) &\iff i_n \in [l_1 - L, l_c - L - 1] \cup [l_1, l_c - 1]
\end{aligned}$$

We observe that

$$\begin{aligned}
&\sum_{\mathcal{A}(f,I_n) \ni \mathbf{c}} t_c^n + \sum_{\mathcal{B}(f,I_n) \ni \mathbf{c}} s_c^n \\
&= \omega_c \left(\sum_{i_n = l_c - L}^{l_1 - 1} (l_c - l_1) + \left\{ \sum_{p=1}^{c-1} \sum_{i_n = l_p - L}^{l_{p+1} - L - 1} (M + l_p - (i_n + 1)) \right.\right. \\
&\quad \left.\left. + \sum_{i_n = l_1}^{l_c - 1} (M + l_c - (i_n + 1)) \right\} \right) \\
&= \omega_c \left(A + B + C \right)
\end{aligned}$$

where

$$\begin{aligned}
A &= \sum_{i_n = l_c - L}^{l_1 - 1} (l_c - l_1) \\
B &= \sum_{p=1}^{c-1} \sum_{i_n = l_p - L}^{l_{(p+1)} - L - 1} (M + l_p - (i_n + 1)) \\
C &= \sum_{i_n = l_1}^{l_c - 1} (M + l_c - (i_n + 1))
\end{aligned}$$

Let $(l_c - l_1) = r$. Hence, we have:

$$\begin{aligned}
A &= (l_c - l_1) \cdot (L - (l_c - l_1)) \\
&= r \cdot (L - r) = L \cdot r - r^2
\end{aligned} \tag{22}$$

$$
\begin{aligned}
B &= \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{(p+1)}-L-1} (M + l_p - (i_n + 1)) \\
&= \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{(p+1)}-L-1} M + \\
&\sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{(p+1)}-L-1} (l_p - (i_n + 1)) \\
&= \sum_{p=1}^{c-1} M \cdot (l_{(p+1)} - l_p) + \\
&\sum_{p=1}^{c-1} \{(L-1) + ... + (L - (l_{p+1} - l_p))\}
\end{aligned}
$$

Let $y_p = l_{(p+1)} - l_p$. Clearly

$$
\sum_{p=1}^{c-1} y_p = l_c - l_1
$$

Hence,

$$
\begin{aligned}
B &= M \cdot r + L \cdot r - \sum_{p=1}^{c-1} \frac{y_p \cdot (y_p + 1)}{2} \\
&= M \cdot r + L \cdot r - \frac{r}{2} - \sum_{p=1}^{c-1} \frac{(y_p)^2}{2}
\end{aligned} \tag{23}
$$

$$
\begin{aligned}
C &= \sum_{i_n=l_1}^{l_c-1} (M + l_c - (i_n + 1)) \\
&= \sum_{i_n=l_1}^{l_c-1} M + \sum_{i_n=l_1}^{l_c-1} (l_c - (i_n + 1)) \\
&= M \cdot (l_c - l_1) + \{(l_c - (l_1 + 1)) + (l_c - (l_1 + 2)) ... + 1\} \\
&= M \cdot (l_c - l_1) + \frac{(l_c - l_1 - 1)(l_c - l_1)}{2} \\
&= M \cdot r + \frac{(r-1) \cdot r}{2} \\
&= M \cdot r + \frac{r^2}{2} - \frac{r}{2}
\end{aligned} \tag{24}
$$

Using (22), (23), (24), we have

$$A + B + C = (2L + 2M - r)r + \frac{r^2}{2} - r - \sum_{p=1}^{c-1} \frac{y_p^2}{2} \tag{25}$$

We have

$$\sum_{p=1}^{c-1} \frac{{y_p}^2}{2} \geq \frac{r^2}{2(c-1)}$$

(Taking each $y_p = r/(c-1)$)

Hence,

$$A + B + C \leq \left\{ 2L + 2M - 1 - \frac{r}{2} \left(\frac{c}{c-1} \right) \right\} \cdot r$$

Let

$$f(r) = \left\{ 2L + 2M - 1 - \frac{r}{2} \left(\frac{c}{c-1} \right) \right\} \cdot r$$

Taking derivative of $f(r)$, we obtain

$$f'(r) = 2L + 2M - 1 - \frac{r \cdot c}{(c-1)}$$

$f(r)$ is a quadratic function which opens downwards (coefficient of r^2 is negative). Also

$$f'(L) = \left(\frac{c-2}{c-1} \right) \cdot L + 2M - 1 > 0$$

Hence, using $r = L$,

$$\begin{aligned} A + B + C &< L \cdot \left\{ 2L + 2M - 1 - \frac{L}{2} \cdot \left(\frac{c}{c-1} \right) \right\} \\ &< L \cdot \left\{ \frac{2L}{M} + 2 - \frac{L}{2M} \cdot \left(\frac{c}{c-1} \right) \right\} \theta_{\mathbf{c}}(\mathbf{x_c}) \end{aligned} \tag{26}$$

where the last expression is obtained using the fact that $\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$.

$Case - III : |l_c - l_1| < M$ and hence $\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}}(\mathbf{l_c} - \mathbf{l_1})$

In this case, the conditions for $\mathbf{X_c} \in \mathcal{A}(f, I_n)$ and $\mathbf{X_c} \in \mathcal{B}(f, I_n)$ are given by

$$\begin{aligned} \mathbf{X_c} \in \mathcal{A}(f, I_n) &\iff i_n \in [l_c - L, l_1 - 1] \\ \mathbf{X_c} \in \mathcal{B}(f, I_n) &\iff i_n \in [l_1 - L, l_c - L - 1] \cup [l_1, l_c - 1] \end{aligned}$$

We observe that

$$\begin{aligned}
&\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} t_c^n + \sum_{\mathcal{B}(f,I_n)\ni\mathbf{c}} s_c^n \\
&= \omega_c \left(\sum_{i_n=l_c-L}^{l_1-1} (l_c - l_1) + \left\{ \sum_{p=1}^{c-1} \sum_{i_n=l_p-L}^{l_{p+1}-L-1} (M + l_p - (i_n+1)) \right.\right. \\
&\quad \left.\left. + \sum_{i_n=l_1}^{l_c-1} (M + l_c - (i_n+1)) \right\} \right) \\
&= \omega_c \left(A + B + C\right)
\end{aligned}$$

Similar to case II, we can write

$$A + B + C = (2L + 2M - r/2)r - r - \sum_{p=1}^{c-1} \frac{y_p^2}{2}$$

Since $r < M$

$$\begin{aligned}
A + B + C &\leq (2L + 2M)(l_c - l_1) \\
&= L\left(2 + \frac{2M}{L}\right)\theta_{\mathbf{c}}(\mathbf{x_c})
\end{aligned} \tag{27}$$

where the last expression is obtained using the fact that $\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \cdot \mathbf{M}$.

Substituting inequalities (21), (26) and (27) in expression (20) and dividing both sides by L for all $\mathbf{X_c}$, we obtain inequality of lemma 3. This proves the lemma. ■

E Proof of Proposition 3

Proposition 3: The range expansion algorithm with $h' = M$ has a multiplicative bound of $O(C)$ for truncated max-of-linear model when $m = 1$. The term C equals the size of the largest clique. Hence, if $\mathbf{x}^*$ is a labeling with minimum energy and $\hat{\mathbf{x}}$ is the labeling estimated by range expansion algorithm then

$$\sum_{a \in \mathcal{V}} \theta_a(\hat{x}_a) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \sum_{a \in \mathcal{V}} \theta_a(x^*_a) + O(C) \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}^*_{\mathbf{c}}).$$

The above inequality holds for arbitrary set of unary potentials and non-negative clique weights.

Proof: The following equation can be deduced from the above definitions:

$$\sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) = \sum_{I_n \in \mathcal{I}_r} \sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) \tag{28}$$

since $f^*(a)$ belongs to exactly one interval in I_r for all X_a.

For the final labeling f of the range expansion algorithm, the term in lemma 2 should be non-positive for all intervals I_n because f is a local optimum. Hence,

$$\begin{aligned} &\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n) \cup \mathcal{B}(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x_c}) \\ &\leq \left(\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t^n_c + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s^n_c \right), \forall I_n \end{aligned}$$

We sum the above inequality over all $I_n \in \Gamma_r$. The summation of the LHS is at least $E(f)$. Also, using (28), the summation of the above inequality can be written as:

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \sum_{I_n \in \Gamma_r} \left(\sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t^n_c + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s^n_c \right)$$

We now take the expectation of the above inequality over the uniformly distributed random integer $r \in [0, L-1]$. The LHS of the inequality and the first term on the RHS (that is, $\sum \theta_a(f^*(a))$) are constants with respect to r. Hence, we get

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + \frac{1}{L} \sum_r \sum_{I_n \in \Gamma_r} \left(\sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t^n_c + \sum_{\mathbf{X_c} \in \mathcal{B}(f, I_n)} s^n_c \right) \tag{29}$$

Lemma 3 allows us to write the above inequality as

$$E(f) \leq \sum_{X_a \in \mathbf{X}} \theta_a(f^*(a)) + max \left\{ \frac{c}{2} \left(2 + \frac{L}{M} \right), \left(2 + \frac{2M}{L} \right) \right\} \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x_c}) \tag{30}$$

The R.H.S of lemma 3 is minimum under the following condition:

$$\frac{c}{2}\left(2+\frac{L}{M}\right)=\left(2+\frac{2M}{L}\right) \tag{31}$$

The positive solution of the above quadratic equation gives the optimum value L_{opt} of interval length:

$$L_{opt}=\left\{\frac{2-c+\sqrt{c^2+4}}{c}\right\}\cdot M \tag{32}$$

However, we bound the minimum value of L as M, giving:

$$L_{opt}=max\left\{\left\{\frac{2-c+\sqrt{c^2+4}}{c}\right\}\cdot M, M\right\} \tag{33}$$

Note L_{opt} equals $\sqrt{2}M$ for $c=2$ and M for $c>2$. Substituting the optimum value of L from equation 33 in inequality of lemma 3 and simplifying inequality 12 gives the multiplicative bound for truncated max-of-linear model as $\frac{(c+2)+\sqrt{c^2+4}}{2}$. Hence, the multiplicative bound is $O(c)$ where c is the size of the maximal clique. For $c=2$, this gives a bound of 2 + $\sqrt{2}$ using $L=\sqrt{2}M$. ■

F Proof of Proposition 4

Proposition 4: The range expansion algorithm with $h' = M$ has a multiplicative bound of $O(m \cdot C)$ for truncated max-of-linear model for any general value of m. The term C equals the size of the largest clique. Hence, if $\mathbf{x}^*$ is a labeling with minimum energy and $\hat{\mathbf{x}}$ is the labeling estimated by range expansion algorithm then

$$\sum_{a \in \mathcal{V}} \theta_a(\hat{x}_a) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\hat{\mathbf{x}}_{\mathbf{c}}) \leq \sum_{a \in \mathcal{V}} \theta_a(x_a^*) + O(m \cdot C) \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}^*).$$

The above inequality holds for arbitrary set of unary potentials and non-negative clique weights.

Proof: We introduce some notations for our proof. Let $\mathcal{A}(f, I_n)$ be the set of all cliques for which all variables take optimum label in the interval I_n and $\mathcal{B}_k(f, I_n)$ where $k \in [1, \ldots, m-1]$ be the set of all cliques for which exactly k variables retain their old label. Let $\mathcal{B}_m(f, I_n)$ be the set of all cliques for which m or more variables retain their old label.

We also introduce the following shorthand notation:

- We denote $\omega_{\mathbf{c}}\{\sum_{i=1}^{m} d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c)\}$ as $t_{m,c}^n$
- We denote $\omega_{\mathbf{c}}\{\sum_{i=2}^{m-1} d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c) + d(p_c(\mathbf{x}_c) - i_m - 1)\} + \omega_{\mathbf{c}} \cdot M$ as $s_{1,c}^n$
- In general, we denote $\omega_{\mathbf{c}}\{\sum_{i=k+1}^{m-k} d(p_i(\mathbf{x}_c) - p_{c-i+1}(\mathbf{x}_c) + \sum_{i=1}^{k} d(p_{c-i+1}(\mathbf{x}_c) - i_m - 1)\} + \mathrm{k} \cdot \omega_{\mathbf{c}} \cdot M$ as $s_{k,c}^n$

We state the following lemma which is a generalization of lemma 2.

Lemma 4. *At an iteration of our algorithm, given the current labeling f_n and an interval $I_n = [i_n + 1, j_n]$, the new labeling f_{n+1} obtained by solving the st-mincut problem reduces the energy by at least the following:*

$$\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f_n(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n) \cup \mathcal{B}_1(f, I_n) \cup \cdots \cup \mathcal{B}_m(f, I_n)} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}})$$
$$- \left(\sum_{X_a \in \mathbf{X}(f^*, I_n)} \theta_a(f^*(a)) + \sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_{m,c}^n + \sum_{\mathbf{X_c} \in \mathcal{B}_1(f, I_n)} s_{1,c}^n + \cdots + \sum_{\mathbf{X_c} \in \mathcal{B}_m(f, I_n)} s_{m,c}^n \right)$$

We also make use of the following lemma which generalizes lemma 3:

Lemma 5. *When $d(.)$ is linear, that is, $d(x) = |x|$, the following inequality holds true:*

$$\begin{aligned} &\frac{1}{L} \sum_{r} \sum_{I_n \in \Gamma_r} \left(\sum_{\mathbf{X_c} \in \mathcal{A}(f, I_n)} t_{m,c}^n + \sum_{\mathbf{X_c} \in \mathcal{B}_1(f, I_n)} s_{1,c}^n + \cdots + \sum_{\mathbf{X_c} \in \mathcal{B}_m(f, I_n)} s_{m,c}^n \right) \\ &\leq max \left\{ m \cdot \frac{c}{2} \left(2 + \frac{L}{M} \right), \left(2 + \frac{2M}{L} \right) \right\} \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x}_{\mathbf{c}}) \end{aligned} \tag{34}$$

where c is the largest clique in the random field.

Let us denote the set of optimum labels in a clique $\mathbf{X_c}$ arranged in an increasing order as $l_1, l_2....., l_{c-1}, l_c$. Since we are dealing with the truncated linear metric, the terms $t^n_{m,c}$ and $s^n_{k,c}$ can be simplified as:

$$\begin{aligned}
t^n_{m,c} &= \omega_c\{(l_c - l_1) + (l_{c-1} - l_2) + \cdots + (l_{c-m+1} - l_m)\} \\
s^n_{1,c} &= \omega_c\{(l_c - i_n - 1 + M) + (l_{c-1} - l_2) + \cdots + (l_{c-n+1} - l_n)\} \\
&\vdots \\
s^n_{m,c} &= \omega_c\{(l_c - i_n - 1 + M) + (l_{c-1} - i_n - 1 + M) + \cdots + (l_{c-n+1} - i_n - 1 + M)\}
\end{aligned}$$

Since the labels are sorted in ascending order $\{l_1, \ldots, l_c\}$, it follows that

$$\begin{aligned}
t^n_{m,c} &\leq \omega_c \cdot m \cdot \{l_c - l_1\} \\
s^n_{1,c} &\leq \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\} \\
&\vdots \\
s^n_{m,c} &\leq \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\}
\end{aligned}$$

The LHS of inequality in lemma 5 can be written as:

$$\begin{aligned}
&\frac{1}{L}\sum_{\mathbf{c}\in\mathcal{C}}\left(\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} t^n_{m,c} + \sum_{\mathcal{B}_1(f,I_n)\ni\mathbf{c}} s^n_{1,c} + \cdots + \sum_{\mathcal{B}_m(f,I_n)\ni\mathbf{c}} s^n_{m,c}\right) \\
&\leq \frac{1}{L}\sum_{\mathbf{c}\in\mathcal{C}}\left(\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} \omega_c \cdot m \cdot \{l_c - l_1\} + \sum_{\mathcal{B}_1(f,I_n)\ni\mathbf{c}} \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\} + \ldots\right. \\
&\left. + \sum_{\mathcal{B}_m(f,I_n)\ni\mathbf{c}} \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\}\right) \\
&\leq \frac{1}{L}\sum_{\mathbf{c}\in\mathcal{C}}\left(\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} \omega_c \cdot m \cdot \{l_c - l_1\} + \sum_{\mathcal{B}_1(f,I_n)\cup\cdots\cup\mathcal{B}_m(f,I_n)\ni\mathbf{c}} \omega_c \cdot m \cdot \{l_c - i_n - 1 + M\}\right) \\
&= m \cdot \frac{1}{L}\sum_{\mathbf{c}\in\mathcal{C}}\left(\sum_{\mathcal{A}(f,I_n)\ni\mathbf{c}} \omega_c \cdot t^n_c + \sum_{\mathcal{B}(f,I_n)\ni\mathbf{c}} \omega_c \cdot s^n_c\right)
\end{aligned} \tag{35}$$

The last expression in equation 35 is m times expression 20. The analysis below follows the same steps as in lemma 3.

In order to prove the lemma, we consider the following three cases for each clique $\mathbf{X_c}$:

$Case - I : |l_c - l_1| \geq L$ and hence $\omega_c \cdot M \leq \theta_{\mathbf{c}}(\mathbf{x_c}) \leq \omega_c \cdot m \cdot M$

This results in

$$m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left(\sum_{\mathcal{A}(f,I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f,I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right) \leq m \cdot \frac{c}{2} \left(2 + \frac{L}{M} \right) \omega_c \cdot M$$
$$\leq m \cdot \frac{c}{2} \left(2 + \frac{L}{M} \right) \theta_{\mathbf{c}}(\mathbf{x_c}) \tag{36}$$

$Case - II : M \leq |l_c - l_1| < L$ and hence $\omega_c \cdot M \leq \theta_{\mathbf{c}}(\mathbf{x_c}) \leq \omega_c \cdot m \cdot M$

This gives

$$m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left(\sum_{\mathcal{A}(f,I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f,I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right) \leq m \cdot \left\{ \frac{2L}{M} + 2 - \frac{L}{2M} \cdot \left(\frac{c}{c-1} \right) \right\} \omega_c \cdot M$$
$$\leq m \cdot \left\{ \frac{2L}{M} + 2 - \frac{L}{2M} \cdot \left(\frac{c}{c-1} \right) \right\} \theta_{\mathbf{c}}(\mathbf{x_c}) \tag{37}$$

$Case - III : |l_c - l_1| < M$ and hence $\omega_c(l_c - l_1) \leq \theta_{\mathbf{c}}(\mathbf{x_c}) \leq m \cdot \omega_c(l_c - l_1)$

This leads to

oo

$$m \cdot \frac{1}{L} \sum_{\mathbf{c} \in \mathcal{C}} \left(\sum_{\mathcal{A}(f,I_n) \ni \mathbf{c}} \omega_c \cdot t_c^n + \sum_{\mathcal{B}(f,I_n) \ni \mathbf{c}} \omega_c \cdot s_c^n \right) \leq m \cdot (2L + 2M)(l_c - l_1)$$
$$\leq m \cdot L \left(2 + \frac{2M}{L} \right) \theta_{\mathbf{c}}(\mathbf{x_c}) \tag{38}$$

Substituting inequalities (36), (37) and (38) in expression (35), we obtain inequality of lemma 5. This proves the lemma. ■

Making use of lemmas 4 and 5, the proof of proposition 4 follows exactly the same steps as that of proposition 3.

CHAPTER 5

Worst-case Optimal Submodular Extensions for Marginal Estimation

This work was presented as a poster at the 21st International Conference on Artificial Intelligence and Statistics (AISTATS), 2018.

The appendix for this chapter is the same as that for chapter 6, and can be found at the end of the latter.

Worst-case Optimal Submodular Extensions for Marginal Estimation

Pankaj Pansari[1,2]
1: University of Oxford
pankaj@robots.ox.ac.uk

Chris Russell [2,3]
2: The Alan Turing Institute
crussell@turing.ac.uk

M. Pawan Kumar[1,2]
3: University of Surrey
pawan@robots.ox.ac.uk

Abstract

Submodular extensions of an energy function can be used to efficiently compute approximate marginals via variational inference. The accuracy of the marginals depends crucially on the quality of the submodular extension. To identify the best possible extension, we show an equivalence between the submodular extensions of the energy and the objective functions of linear programming (LP) relaxations for the corresponding MAP estimation problem. This allows us to (i) establish the worst-case optimality of the submodular extension for Potts model used in the literature; (ii) identify the worst-case optimal submodular extension for the more general class of metric labeling; and (iii) efficiently compute the marginals for the widely used dense CRF model with the help of a recently proposed Gaussian filtering method. Using synthetic and real data, we show that our approach provides comparable upper bounds on the log-partition function to those obtained using tree-reweighted message passing (TRW) in cases where the latter is computationally feasible. Importantly, unlike TRW, our approach provides the first practical algorithm to compute an upper bound on the dense CRF model.

1 Introduction

The desirable optimization properties of submodular set functions have been widely exploited in the design of approximate MAP estimation algorithms for discrete conditional random fields (CRFs) [Boykov et al., 2001; Kumar et al., 2011]. Submodularity has also been recently used to design an elegant variational inference algorithm to compute the marginals of a discrete CRF by minimising an upper-bound on the log-partition function. In the initial work of [Djolonga and Krause, 2014], the energy of the CRF was restricted to be submodular. In a later work [Zhang et al., 2015], the algorithm was extended to handle more general Potts energy functions. The key idea here is to define a large ground set such that its subsets represent valid labelings, sublabelings or even incorrect labelings (these may assign two separate labels to a random variable and hence be invalid). Given the large ground set, it is possible to define a submodular set function whose value is equal to the energy of the CRF for subsets that specify a valid labeling of the model. We refer to such a set function as a *submodular extension* of the energy.

For a given energy function, there exists a large number of possible submodular extensions. The accuracy of the variational inference algorithm depends crucially on the choice of the submodular extension. Yet, previous work has largely ignored the question of identifying the best extension. Indeed, the difficulty of identifying submodular extensions of general energy functions could be a major reason why the experiments of [Zhang et al., 2015] were restricted to the special case of models specified by the Potts energy functions.

In this work, we establish a hitherto unknown connection between the submodular extension of the Potts model proposed by Zhang et al. [2015], and the objective function of an accurate linear programming (LP) relaxation of the corresponding MAP estimation problem [Kleinberg and Tardos, 2002]. This connection has three important practical consequences. First, it establishes the accuracy of the submodular extension of the Potts model, via the UGC-hardness worst-case optimality of the LP relaxation. Second, it provides an accurate submodular extension of the hierarchical Potts model, via the LP relaxation of the corresponding MAP estimation problem proposed by Kleinberg and Tardos [2002]. Since any metric can be accurately approximated as a mixture of hierarchical Potts models [Bartal, 1996, 1998], this result also provides a computationally feasible algorithm for estimating the

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

marginals for metric labeling. Third, it establishes the equivalence between the subgradient of the LP relaxation and the conditional gradient of the problem of minimising the upper bound of the log-partition. This allows us to employ the widely used dense CRF, since the subgradient of its LP relaxation can be efficiently computed using a recently proposed modified Gaussian filtering algorithm [Ajanthan et al., 2017]. As a consequence, we provide the first efficient algorithm to compute an upper bound of the log-partition function of dense CRFs. This provides complementary information to the popular mean-field inference algorithm for dense CRFs, which computes a lower bound on the log-partition [Koltun and Krahenbuhl, 2011]. We show that the quality of our solution is comparable to tree reweighted message passing (TRW) [Wainwright et al., 2005] for the case of sparse CRFs. Unlike our approach, TRW is computationally infeasible for dense CRFs, thereby limiting its use in practice. Using dense CRF models, we perform stereo matching on standard data sets and obtain better results than [Koltun and Krahenbuhl, 2011]. The complete code is available at `https://github.com/pankajpansari/denseCRF`.

2 Preliminaries

We now introduce the notation and definitions that we will make use of in the remainder of the paper.

Submodular Functions Given a ground set $U = \{1, \dots, N\}$, denote by 2^U its power set. A set function $F : 2^U \to \mathbb{R}$ is *submodular* if, for all subsets $A, B \subseteq U$, we have

$$F(A \cup B) + F(A \cap B) \leq F(A) + F(B). \quad (1)$$

The set function F is *modular* if there exists $\mathbf{s} \in \mathbb{R}^N$ such that $F(A) = \sum_{k \in A} s_k \ \forall \ A \subseteq 2^U$. Henceforth, we will use the shorthand $s(A)$ to denote $\sum_{k \in A} s_k$.

Extended Polymatroid Associated with any submodular function F is a special polytope known as the *extended polymatroid* defined as

$$EP(F) = \{\mathbf{s} \in \mathbb{R}^N | \ \forall A \subseteq U : s(A) \leq F(A)\}, \quad (2)$$

where $\mathbf{s}$ denotes the modular function $s(.)$ considered as a vector.

Lovasz Extension For a given set function F with $F(\emptyset) = 0$, the value of its Lovasz extension $f(\mathbf{w}) : \mathbb{R}^N \to \mathbb{R}$ is defined as follows: order the components of $\mathbf{w}$ in decreasing order such that $w_{j_1} \geq w_{j_2} \geq \cdots \geq w_{j_N}$, where $(j_1, j_2, \dots, j_N)$ is the corresponding permutation of the indices. Then,

$$f(\mathbf{w}) = \sum_{k=1}^{N} w_{j_k} \left(F(j_1, \dots, j_k) - F(j_1, \dots, j_{k-1}) \right). \quad (3)$$

The function f is an extension because it equals F on the vertices of the unit cube. That is, for any $A \subseteq V$, $f(\mathbf{1}_A) = F(A)$ where $\mathbf{1}_A$ is the 0-1 indicator vector corresponding to the elements of A.

Property 1. *By Edmond's greedy algorithm [Edmonds, 1970], if* $\mathbf{w} \geq 0$ *(non-negative elements),*

$$f(\mathbf{w}) = \max_{\mathbf{s} \in EP(F)} \langle \mathbf{w}, \mathbf{s} \rangle. \quad (4)$$

Property 1 implies that an LP over $EP(F)$ can be solved by computing the value of the Lovasz extension using equation (3).

Property 2. *The Lovasz extension f of a submodular function F is a convex piecewise linear function.*

Property 2 holds since $f(\mathbf{w})$ is the pointwise maximum of linear functions according to equation (4).

CRF and Energy Functions A CRF is defined as a graph on a set of random variables $\mathcal{X} = \{X_1, \dots, X_N\}$ related by a set of edges $\mathcal{N}$. We wish to assign every variable X_a one of the labels from the set $\mathcal{L} = \{1, 2, \dots, L\}$. The quality of a labeling $\mathbf{x}$ is given by an energy function defined as

$$E(\mathbf{x}) = \sum_{a \in \mathcal{X}} \phi_a(x_a) + \sum_{(a,b) \in \mathcal{N}} \phi_{ab}(x_a, x_b), \quad (5)$$

where ϕ_a and ϕ_{ab} are the unary and pairwise potentials respectively. In computer vision, we often think of $\mathcal{X}$ as arranged on a grid. A *sparse CRF* has $\mathcal{N}$ defined by 4-connected or 8-connected neighbourhood relationships. In a *dense CRF*, on the other hand, every variable is connected to every other variable.

The energy function also defines a probability distribution $P(\mathbf{x})$ as follows:

$$P(\mathbf{x}) = \begin{cases} \frac{1}{Z} \exp(-E(\mathbf{x})) & \text{if } \mathbf{x} \in \mathcal{L}^N, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The normalization factor $Z = \sum_{\mathbf{x} \in \mathcal{L}^N} \exp(-E(\mathbf{x}))$ is known as the *partition function.*

Inference There are two inference problems in CRFs:

(i) Marginal inference: We want to compute the marginal probabilities $P(X_a = i)$ for every $a = 1, 2, \dots, N$ and $i = 1, 2, \dots, L$.

(ii) MAP inference: We want to find a labeling with the minimum energy, that is, $\min_{\mathbf{x} \in \mathcal{L}^N} E(\mathbf{x})$. Equivalently, MAP inference finds the mode of $P(\mathbf{x})$.

3 Review: Variational Inference Using Submodular Extensions

We now summarise the marginal inference method of Zhang et al. [2015] which uses submodular extensions.

Submodular Extensions A submodular extension is defined using a ground set such that some of its

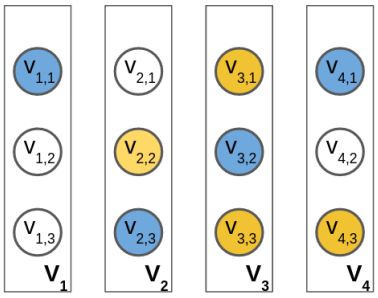


Figure 1: *Illustration of* 1*-of-L encoding used in [Zhang et al., 2015] with 4 variables and 3 labels. The blue labeling, corresponding to* $X_1 = 1, X_2 = 3, X_3 = 2, X_4 = 1$*, is valid. The yellow labeling, corresponding to* $X_2 = 2, X_3 = 1, 3, X_4 = 3$*, is invalid since* X_3 *has been assigned multiple labels and* X_1 *has been assigned none.*

subsets correspond to valid CRF labelings. To such an extension, we need an encoding scheme which gives the sets corresponding to valid CRF labelings.

One example of an encoding scheme is the 1-of-L encoding, illustrated in figure 1. Let each variable X_a take one of L possible labels. In this scheme, we represent the set of possible assignments for X_a by the set $V_a = \{v_{a1}, v_{a2}, \ldots, v_{aL}\}$. If X_a is assigned label i, then we select the element v_{ai}. Extending to all variables, our ground set becomes $V = \cup_{a=1}^{N} V_a$. A valid assignment $A \subseteq V$ assigns each variable exactly one label, that is, $|A \cap V_a| = 1$ for all V_a. We denote the set of valid assignments by $\mathcal{M}$ where $\mathcal{M} = \cap_{a=1}^{N} \mathcal{M}_a$ and $\mathcal{M}_a = \{A : |A \cap V_a| = 1\}$.

Using our ground set V, we can define a submodular function F which equals $E(\mathbf{x})$ for all sets corresponding to valid labelings, that is, $F(A_{\mathbf{x}}) = E(\mathbf{x})$, $\mathbf{x} \in \mathcal{L}^N$ where $A_{\mathbf{x}}$ is the set encoding of $\mathbf{x}$. We call such a function F a *submodular extension* of $E(\mathbf{x})$.

Upper-Bound on Log-Partition Using a submodular extension F and given any $\mathbf{s} \in EP(F)$, we can obtain an upper-bound on the partition function as

$$\mathcal{Z} = \sum_{A \in \mathcal{M}} \exp(-F(A)) \leq \sum_{A \in \mathcal{M}} \exp(-s(A)), \qquad (7)$$

where $\mathcal{M}$ is the set of valid labelings. The upper-bound is the partition function of the distribution $Q(A) \propto \exp(-s(A))$, which factorises fully because $s(.)$ is modular. Since $\mathbf{s} \in EP(F)$ is a free parameter, we can obtain good approximate marginals of the distribution $P(\cdot)$ by minimising the upper-bound. After taking logs, we can equivalently write our optimisation problem as

$$\min_{\mathbf{s} \in EP(F)} g(\mathbf{s}), \text{where } g(\mathbf{s}) = \log \sum_{A \in \mathcal{M}} \exp(-s(A)). \quad (8)$$

Conditional Gradient Algorithm The conditional gradient algorithm (algorithm 1) [Frank and Wolfe, 1956] is a good candidate for solving problem (8) due to two reasons. First, problem (8) is convex. Second, as solving an LP over EP(F) is computationally tractable (property 1), the conditional gradient can be found efficiently. The algorithm starts with an initial solution $\mathbf{s_0}$ (line 1). At each iteration, we compute the conditional gradient $\mathbf{s}^*$ (line 3), which minimises the linear approximation $g(\mathbf{s}_k) + \nabla g(\mathbf{s}_k)^T(\mathbf{s} - \mathbf{s}_k)$ of the objective function. Finally, $\mathbf{s}$ is updated by either (i) fixed step size schedule, as in line 7 of algorithm 1, or (ii) by doing line search $\mathbf{s}_{k+1} = \min_{0 \leq \gamma \leq 1} g(\gamma \mathbf{s}^* + (1-\gamma)\mathbf{s}_k)$.

Algorithm 1 Upper Bound Minimisation using Conditional Gradient Descent

1: Initialize $\mathbf{s} = \mathbf{s}_0 \in EP(F)$
2: **for** k = 1 to `MAX_ITER` **do**
3: $\quad \mathbf{s}^* = \text{argmin}_{\mathbf{s} \in EP(F)} \langle \nabla g(\mathbf{s}_k), \mathbf{s} \rangle$
4: $\quad$ **if** $\langle \mathbf{s}^* - \mathbf{s}_k, \nabla g(\mathbf{s}_k) \rangle \leq \epsilon$ **then**
5: $\quad\quad$ **break**
6: $\quad$ **end if**
7: $\quad \mathbf{s}_{k+1} = \mathbf{s}_k + \gamma(\mathbf{s}^* - \mathbf{s}_k)$ with $\gamma = 2/(k+2)$
8: **end for**
9: **return** $\mathbf{s}$

4 Worst-case Optimal Submodular Extensions via LP Relaxations

Worst-case Optimal Submodular Extensions Different choices of extensions F change the domain in problem (8), leading to different upper bounds on the log-partition function. How does one come up with an extension which yields the tightest bound?

In this paper, we focus on submodular extension families $\mathcal{F}(.)$ which for each instance of the energy function $E(.)$ belonging to a given class $\mathcal{E}$ gives a corresponding submodular extension $\mathcal{F}(E)$. We find the extension family $\mathcal{F}_{opt}$ that is *worst-case optimal*. This implies that there does not exist another submodular extension family $\mathcal{F}$ that gives a tighter upper bound for problem (8) than $\mathcal{F}_{opt}$ for all instances of the energy function in $\mathcal{E}$. Formally,

$$\nexists \mathcal{F} : \min_{\mathbf{s} \in EP(\mathcal{F}(E))} g(\mathbf{s}) \leq \min_{\mathbf{s} \in EP(\mathcal{F}_{opt}(E))} g(\mathbf{s}) \ \forall\ E(.) \in \mathcal{E}. \qquad (9)$$

Note that our problem is different from taking a given energy model and obtaining a submodular extension which is optimal for that model. Also, we seek a closed-form analytical expression for $\mathcal{F}$. For the sake of clarity, in the analysis that follows we use F to represent $\mathcal{F}(E)$ where the meaning is clear from context. The two classes of energy functions we consider in this paper are Potts and hierarchical Potts families.

Using LP Relaxations If we introduce a temperature parameter in $P(\mathbf{x})$ (equation (6)) by using $E(\mathbf{x})/T$

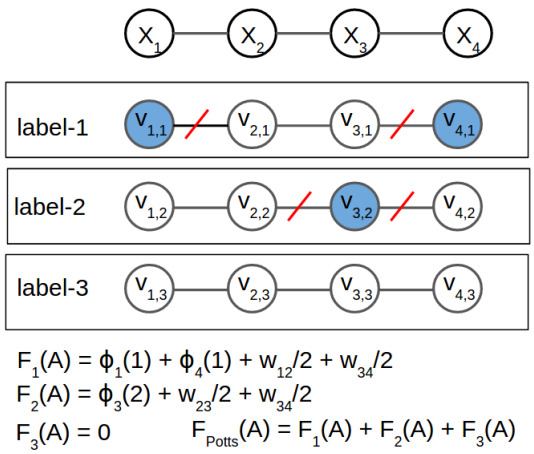


Figure 2: *An illustration of the worst-case optimal submodular extension for Potts model for a chain graph of 4 variables, each of which can take 3 labels. The figure shows the way to compute the extension values of the set* $A = \{v_{1,1}, v_{4,1}, v_{3,2}\}$.

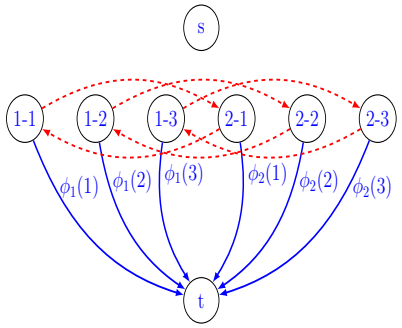


Figure 3: *An st-graph specifying the worst-case optimal submodular extension for Potts model for 2 variables with 3 labels each and connected to each other. There is a node for each variable and each label, that is, for all elements of the ground set. The nodes have been labeled as 'variable-label', hence node 1-1 represents the element* v_{11} *and so on. The solid blue arcs model the unary potentials, and the dotted red arcs represent the pairwise potentials. Each dotted red arc has weight* $w_{12}/2$.

and decrease T, the resulting distribution starts to peak more sharply around its mode. As $T \to 0$, marginal estimation becomes the same as MAP inference since the resulting distribution $P^0(\mathbf{x})$ has mass 1 at its mode $\mathbf{x}^*$ and is 0 everywhere else. Given the MAP solution $\mathbf{x}^*$, one can compute the marginals as $P(X_i = j) = [x_i^* = j]$, where [.] is the Iverson bracket.

Motivated by this connection, we ask if one can introduce a temperature parameter to our problem (8) and transform it to an LP relaxation in the limit $T \to 0$? We can then hope to use the tightest LP relaxations of MAP problems known in literature to find worst-case optimal submodular extensions. We answer this question in affirmative. Specifically, in the following two sections we show how one can select the set encoding and submodular extension to convert problem (8) to the tightest known LP relaxations for Potts and hierarchical Potts models. Importantly, we prove the worst-case optimality of the extensions thus obtained.

5 Potts Model

The Potts model, also known as the uniform metric, specifies the pairwise potentials $\phi_{ab}(x_a, x_b)$ in equation (5) as follows:

$$\phi_{ab}(x_a, x_b) = w_{ab} \cdot [x_a \neq x_b], \tag{10}$$

where w_{ab} is the weight associated with edge (a, b).

Tightest LP Relaxation Before describing our set encoding and submodular extension, we briefly outline the LP relaxation of the corresponding MAP estimation problem. To this end, we define indicator variables y_{ai} which equal 1 if $X_a = i$, and 0 otherwise. The following LP relaxation is the tightest known for Potts model in the worst-case, assuming the Unique Games Conjecture to be true [Manokaran et al., 2008]

$$\text{(P-LP)} \quad \min_{\mathbf{y}} \; E(\mathbf{y}) = \sum_a \sum_i \phi_a(i) y_{ai} + \sum_{(a,b)\in\mathcal{N}} \sum_i \frac{w_{ab}}{2} \cdot |y_{ai} - y_{bi}|$$
$$\text{s.t} \quad \mathbf{y} \in \Delta. \tag{11}$$

The set Δ is the union of N probability simplices:

$$\Delta = \{\mathbf{y}_a \in \mathbb{R}^L | \mathbf{y}_a \geq 0 \text{ and } \langle \mathbf{1}, \mathbf{y}_a \rangle = 1\}, \tag{12}$$

where $\mathbf{y}$ is the vector of all variables and $\mathbf{y}_a$ is the component of $\mathbf{y}$ corresponding to X_a.

Set Encoding We choose to use the 1-of-L encoding for Potts model as described in section 3. With the encoding scheme for Potts model above, $g(\mathbf{s})$ can be factorised and problem (8) can be rewritten as:

$$\min_{\mathbf{s}\in EP(F)} \quad \sum_{a=1}^{N} \log \sum_{i=1}^{L} \exp(-s_{ai}). \tag{13}$$

(See Remark 1 in appendix)

Marginal Estimation with Temperature We now introduce a temperature parameter $T > 0$ to problem (13) which divides $E(\mathbf{x})$, or equivalently divides $\mathbf{s}$ belonging to $EP(F)$. Also, since $T > 0$, we can multiply the objective by T leaving the problem unchanged. Without changing the solution, we can transform problem (13) as follows

$$\min_{\mathbf{s}\in EP(F)} \quad g_T(\mathbf{s}) = \sum_{a=1}^{N} T \cdot \log \sum_{i=1}^{L} \exp(-\frac{s_{ai}}{T}). \tag{14}$$

Worst-case Optimal Submodular Extension We now connect our marginal estimation problem (8) with LP relaxations using the following proposition.

Proposition 1. *Using the* 1*-of-*L *encoding scheme, in*

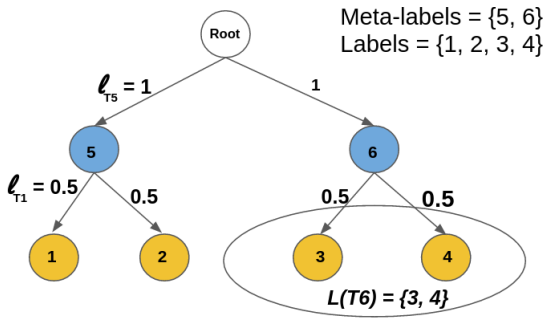


Figure 4: *A hierarchical Potts model instance illustrating notations with 2 meta-labels (blue) and 4 labels (yellow). All labels are at the same level. $r = 2$, that is, edge-length decreases by 2 at each level. Also, distance between labels 1 and 3 is $d(1,3) = 0.5 + 1 + 1 + 0.5 = 3$.*

the limit $T \to 0$, problem (14) for Potts model becomes:

$$-\min_{\mathbf{y} \in \Delta} f(\mathbf{y}) \tag{15}$$

where $f(.)$ is the Lovasz extension of $F(.)$.
(Proof in appendix)

The above problem is equivalent to an LP relaxation of the corresponding MAP estimtation problem (see Remark 2 in appendix). We note that $g_T(\mathbf{s})$ in problem (14) becomes the objective function of an LP relaxation in the limit $T \to 0$. We seek to obtain the worst-case optimal submodular extension by making $g_T(\mathbf{s})$ same as the objective of (P-LP) as $T \to 0$. Since at $T = 1$, problems (14) and (13) are equivalent, this gives us the worst-case optimal extension for our problem (13).

The question now becomes how to recover the worst-case optimal submodular extension using $E(\mathbf{y})$. The following propositions answers this question.

Proposition 2. *The worst-case optimal submodular extension for Potts model is given by $F_{Potts}(A) = \sum_{i=1}^{L} F_i(A)$, where*

$$F_i(A) = \sum_a \phi_a(i)[|A \cap \{v_{ai}\}| = 1] + \sum_{(a,b) \in \mathcal{N}} \frac{w_{ab}}{2} \cdot [|A \cap \{v_{ai}, v_{bi}\}| = 1] \tag{16}$$

Also, $E(\mathbf{y})$ in (P-LP) is the Lovasz extension of F_{Potts}.
(Proof in appendix)

Proposition 2 paves the way for us to identify the worst-case optimal extension for hierarchical Potts model, which we discuss in the following section.

6 Hierarchical Potts Model

Potts model imposes the same penalty for unequal assignment of labels to neighbouring variables, regardless of the label dissimilarity. A more natural approach is to vary the penalty based on how different the labels are. A hierarchical Potts model permits this by specifying the distance between labels using a tree with the following properties:

1. The vertices are of two types: (i) the leaf nodes representing labels, and (ii) the non-leaf nodes, except the root, representing meta-labels.
2. The lengths of all the edges from a parent to its children are the same.
3. The lengths of the edges along any path from the root to a leaf decreases by a factor of at least $r \geq 2$ at each step.
4. The metric distance between nodes of the tree is the sum of the edge lengths on the unique path between them.

A subtree T of an hierarchical Potts model is a tree comprising all the descendants of some node v (not root). Given a subtree T, l_T denotes the length of the tree-edge leading upward from the root of T and $L(T)$ denotes the set of leaves of T. We call the leaves of the tree as labels and all other nodes of the tree expect the root as *meta-labels*. Figure 4 illustrates the notations in the context of a hierarchical Potts model.

Tightest LP Relaxation We use the same indicator variables y_{ai} that were employed in the LP relaxation of Potts model. Let $y_a(T) = \sum_{i \in L(T)} y_{ai}$. The following LP relaxation is the tightest known for hierarchical Potts model in the worst-case, assuming the Unique Games Conjecture to be true [Manokaran et al., 2008]

$$\text{(T-LP)} \quad \min_{\mathbf{y}} \ \widetilde{E}(\mathbf{y}) = \sum_a \sum_i \phi_a(i) y_{ai} + \sum_{(a,b) \in \mathcal{N}} w_{ab} \sum_T l_T \cdot |y_a(T) - y_b(T)|$$
$$\text{such that} \quad \mathbf{y} \in \Delta. \tag{17}$$

The set Δ is the same domain as defined in equation (12). We rewrite this LP relaxation using indicator variables z_{ai} for all labels and meta-labels as

$$\text{(T-LP-FULL)} \quad \min \widetilde{E}(\mathbf{z})$$
$$\text{such that} \quad \mathbf{z} \in \Delta' \tag{18}$$

where Δ' is the convex hull of the vectors satisfying

$$\sum_{i \in \mathcal{L}} z_{ai} = 1, \ \ z_{ai} \in \{0,1\} \ \ \forall a \in \mathcal{X}, i \in \mathcal{L} \tag{19}$$

$$\text{and} \ \ z_{ai} = \sum_{j \in L(T_i)} z_{aj}. \ \ \forall a \in \mathcal{X}, i \in \mathcal{R} - \mathcal{L} \tag{20}$$

The details of the new relaxation (T-LP-FULL) can be found in the appendix.

Set Encoding For any variable X_a, let the set of possible assignment of labels and meta-labels be the set $V_a = \{v_{a1}, \ldots, v_{aM}\}$, where M is the total number of nodes in the tree except the root. Our ground set is $V = \cup_{a=1}^{N} V_a$ of size $N \cdot M$.

A consistent labeling of a variable assigns it one label, and all meta-labels on the path from root to the label. Let us represent the set of consistent assignments for X_a by the set $P_a = \{p_{a1}, \ldots, p_{aL}\}$, where p_{ai} is the collection of elements from V_a for label i and all meta-labels on the path from root to label i. The set of valid labelings $A \subseteq V$ assigns each variable exactly one consistent label. This constraint can be formally written as $\mathcal{M} = \cap_{a=1}^{N} \mathcal{M}_a$ where $\mathcal{M}_a$ has exactly one element from P_a. Let s'_{ai} be the sum of the components of $\mathbf{s}$ corresponding to the elements of p_{ai}, that is,

$$s'_{ai} = \sum_{t \in p_{ai}} s_t. \tag{21}$$

Using our encoding scheme, we rewrite problem (8) as:

$$\min_{\mathbf{s} \in EP(F)} \quad \sum_{a=1}^{N} \log \sum_{i=1}^{L} \exp(-s'_{ai}). \tag{22}$$

Marginal Estimation with Temperature Similar to Potts model, we now introduce a temperature parameter $T > 0$ to problem (22). The transformed problem becomes

$$\min_{\mathbf{s} \in EP(F)} \quad g_T(\mathbf{s}) = \sum_{a=1}^{N} T \cdot \log \sum_{i=1}^{L} \exp(-\frac{s'_{ai}}{T}). \tag{23}$$

Worst-case Optimal Submodular Extension The following proposition connects the marginal estimation problem (8) with LP relaxations:

Proposition 3. *In the limit $T \to 0$, problem (23) for hierarchical Potts energies becomes:*

$$-\min_{\mathbf{z} \in \Delta'} f(\mathbf{z}) \tag{24}$$

(Proof in appendix).

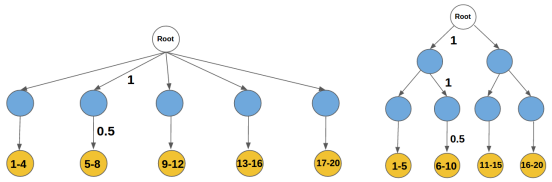
The above problem is equivalent to an LP relaxation of the corresponding MAP estimtation problem (see Remark 3 in appendix). Hence, $g_T(\mathbf{s})$ becomes the objective function of an LP relaxation in the limit $T \to 0$. We seek to make this objective same as $\widetilde{E}(\mathbf{z})$ of (T-LP-FULL) in the limit $T \to 0$. The question now becomes how to recover the worst-case optimal submodular extension from $\widetilde{E}(\mathbf{z})$.

Proposition 4. *The worst-case optimal submodular extension for hierarchical Potts model is given by $F_{hier}(A) = \sum_{i=1}^{M} F_i(A)$, where*

$$F_i(A) = \sum_{a} \phi'_a(i)[|A \cap \{v_{ai}\}| = 1] + \sum_{(a,b) \in \mathcal{N}} w_{ab} \cdot l_{T_i} \cdot [|A \cap \{v_{ai}, v_{bi}\}| = 1] \tag{25}$$

Also, $\widetilde{E}(\mathbf{z})$ in (T-LP-FULL) is the Lovasz extension of F_{hier}. (Proof in appendix)

Since any finite metric space can be probabilistically approximated by mixture of tree metric [Bartal, 1996], the worst-case optimal submodular extension for metric energies can be obtained using F_{hier}. Note that F_{hier} reduces to F_{Potts} for Potts model. One can see this by considering the Potts model as a star-shaped tree with edge weights as 0.5.



(a) Tree A (b) Tree B

Figure 5: *The hierarchical Potts models defining pairwise distance among 20 labels used for upper-bound comparison with TRW. Blue nodes are the meta-labels and yellow nodes are labels. All the edges at a particular level have the same edge weights. The sequence of weights from root level to leaf level is 1, 0.5 for tree A and 1, 1, 0.5 for tree B. The yellow node is shown to clump together 4 and 5 leaf nodes for tree A and B respectively.*

7 Fast Conditional Gradient Computation for Dense CRFs

Dense CRF Energy Function A dense CRF is specified by the following energy function

$$E(\mathbf{x}) = \sum_{a \in \mathcal{X}} \phi_a(x_a) + \sum_{a \in \mathcal{X}} \sum_{b \in \mathcal{X}, b \neq a} \phi_{ab}(x_a, x_b). \tag{26}$$

Note that every random variable is a neighbour of every other random variable in a dense CRF. Similar to previous work [Koltun and Krahenbuhl, 2011], we consider the pairwise potentials to be to be Gaussian, that is,

$$\phi(i,j) = \mu(i,j) \sum_{m} w^{(m)} k(\mathbf{f}_a^{(m)}, \mathbf{f}_b^{(m)}), \tag{27}$$

$$k(\mathbf{f}_a^{(m)}, \mathbf{f}_b^{(m)}) = \exp\left(\frac{-||\mathbf{f}_a - \mathbf{f}_b||^2}{2}\right). \tag{28}$$

The term $\mu(i,j)$ is known as *label compatibility* function between labels i and j. Potts model and hierarchical Potts models are examples of $\mu(i,j)$. The other term is a mixture of Gaussian kernels $k(.,.)$ and is called the *pixel compatibility* function. The terms $\mathbf{f}_a^{(m)}$ are features that describe the random variable X_a. In practice, similar to [Koltun and Krahenbuhl, 2011], we use x, y coordinates and RGB values associated to a pixel as its features.

Algorithm 1 assumes that the conditional gradient $\mathbf{s}^*$ in step 3 can be computed efficiently. This is certainly not the case for dense CRFs, since computing s^* involves NL function evaluations of the submodular extension F, where N is the number of variables, and L is the number of labels. Each F evaluation has complexity $\mathcal{O}(N)$ using the efficient Gaussian filtering algorithm

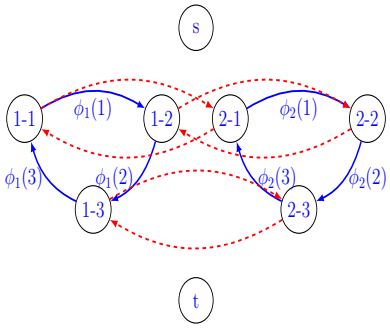


Figure 6: *An st-graph specifying the alternate submodular extension for Potts model for 2 variables with 3 labels each and connected to each other. The convention used is same as in figure 3. Each dotted red arc has weight $w_{12}/2$. This alternate extension was also used to derive the extension for hierarchical Potts model.*

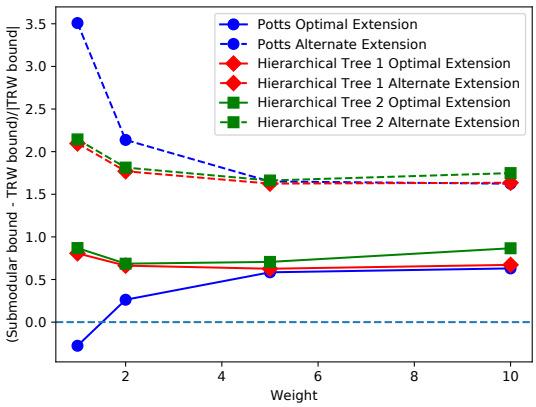


Figure 7: *Upper-bound comparison using synthetic data. The plot shows the ratio (Submod bound - TRW bound)//TRW bound/ averaged over 100 unary instances as a function of pairwise weights using the worst-case optimal and alternate extension for Potts and hierarchical Potts models. We observe that the worst-case optimal extension (solid) results in tighter bounds as compared to the respective alternate extensions (dotted). Also, the worst-case optimal extension bounds are in similar range as the TRW bounds.*

of [Koltun and Krahenbuhl, 2011]. However, computation of s^* would still be $\mathcal{O}(N^2)$ this way, which is clearly impractical for computer-vision applications where $N \sim 10^5 - 10^6$.

However, using the equivalence of relaxed LP objectives and the Lovasz extension of submodular extensions in proposition 1, we are able to compute s^* in $\mathcal{O}(NL)$ time. Specifically, we use the algorithm of Ajanthan et al. [2017], which provides an efficient filtering procedure to compute the subgradient of the LP relaxation objective $E(\mathbf{y})$ of (P-LP).

Proposition 5. *Computing the subgradient of $E(\mathbf{y})$ in (P-LP) is equivalent to computing the conditional gradient for the submodular function F_{Potts}.*
(Proof in appendix).

A similar observation can be made in case of hierarchical Potts model. Hence we have the first practical algorithm to compute upper bound of log-partition function of a dense CRF for Potts and metric energies.

8 Experiments

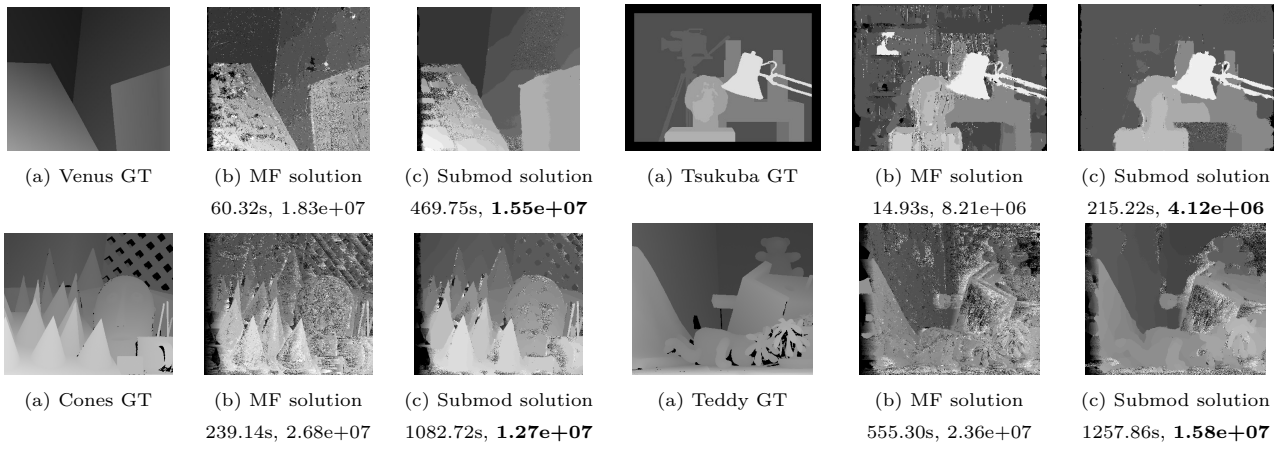
Using synthetic data, we show that our upper-bound compares favorably with TRW for both Potts and hierarchical Potts models. For comparison, we restrict ourselves to sparse CRFs, as the code available for TRW does not scale well to dense CRFs. We also perform stereo matching using dense CRF models and compare our results with the mean-field-based approach of [Koltun and Krahenbuhl, 2011]. All experiments were run on a x86-64, 3.8GHz machine with 16GB RAM. In this section, we refer to our algorithm as *Submod* and mean field as *MF*.

8.1 Upper-bound Comparison using Synthetic Data

Data We generate lattices of size 100 × 100, where each lattice point represents a variable taking one of 20 labels. The pairwise relations of the sparse CRFs are defined by 4-connected neighbourhoods. The unary potentials are uniformly sampled in the range [0, 10]. We consider (a) Potts model and (b) hierarchical Potts models with pairwise distance between labels given by the trees of figure 5. The pairwise weights are varied in the range $\{1, 2, 5, 10\}$. We compare the results of our worst-case optimal submodular extension with an alternate submodular extension as given in figure 6.

Method For our algorithm, we use the standard schedule $\gamma = 2/(k+2)$ to obtain step size γ at iteration k. We run our algorithm till convergence - 100 iterations suffice for this. The experiments are repeated for 100 randomly generated unaries for each model and each weight. For TRW, we used the MATLAB toolbox of [Domke, 2013]. The baseline code does not optimise over tree distributions. We varied the edge-appearance probability in trees over the range [0.1 - 0.5] and found 0.5 to give tightest upper bound.

Results We plot the ratio of the normalised difference of the upper bound values of our method with TRW as a function of pairwise weights. The ratios are averaged over 100 instances of unaries. Figure 7 shows the plots for Potts and hierarchical Potts models for the worst-case optimal and alternate extension. We find that the optimal extension (solid) results in tighter upper-bounds than the alternate extension (dotted) for both models. This is because the representation of the submodular function using figure 6 necessitates that $\phi_a(i)$ be non-negative. This implies that $F(A)$ values



(a) Venus GT (b) MF solution 60.32s, 1.83e+07 (c) Submod solution 469.75s, **1.55e+07** (a) Tsukuba GT (b) MF solution 14.93s, 8.21e+06 (c) Submod solution 215.22s, **4.12e+06**

(a) Cones GT (b) MF solution 239.14s, 2.68e+07 (c) Submod solution 1082.72s, **1.27e+07** (a) Teddy GT (b) MF solution 555.30s, 2.36e+07 (c) Submod solution 1257.86s, **1.58e+07**

Figure 8: *Stereo matching using dense CRFs with Potts compatibility and Gaussian pairwise potentials. We compare our solution with the mean-field algorithm of Koltun and Krahenbuhl [2011]. We observe that our method gives better-looking solutions with lower energy value at the cost of higher computational time.*

are larger for the worst-case optimal extension of figure 3 as compared to the alternate extension. Hence the minimisation problem 8 has larger domain $EP(F)$ for the optimal extension, thereby resulting in better minima. Figure 7 also indicates that our algorithm with optimal extension provides similar range of upper bound as TRW, thereby providing empirical justification of our method. TRW makes use of the standard LP relaxation [Chekuri et al., 2004] which is tighter than Kleinberg-Tardos relaxation, resulting in better approximation. However, TRW does not scale well with neighborhood size, thereby prohibiting its use in dense CRFs.

8.2 Stereo Matching using Dense CRFs

Data We demonstrate the benefit our algorithm for stereo matching on images extracted from the Middlebury stereo matching dataset [Scharstein et al., 2001]. We use dense CRF models with Potts compatibility term and Gaussian pairwise potentials. The unary terms are obtained using the absolute difference matching function of [Scharstein et al., 2001].

Method We use the implementation of mean-field algorithm for dense CRFs of [Koltun and Krahenbuhl, 2011] as our baseline. For our algorithm, we make use of the modified Gaussian filtering implementation for dense CRFs by [Ajanthan et al., 2017] to compute the conditional gradient at each step. The step size γ at each iteration is selected by doing line search. We run our algorithm till 100 iterations, since the visual quality of the solution does not show much improvement beyond this point. We run mean-field up to convergence, with a threshold of 0.001 for change in KL-divergence.

Results Figure 8 shows some example solutions obtained by picking the label with maximum marginal probability for each variable for mean-field and for our algorithm. We also report the time and energy values of the solution for both methods. Though we are not performing MAP estimation, energy values give us a quantitative indication of the quality of solutions. For the full set of 21 image pairs (2006 dataset), the average ratio of the energies of the solutions from our method compared to mean-field is 0.943. The avearge time ratio is 10.66. We observe that our algorithm results in more natural looking stereo matching results with lower energy values for all images. However, mean-field runs faster than our method for each instance.

9 Discussion

We have established the relation between submodular extension for the Potts model and the LP relaxation for MAP estimation using Lovasz extension. This allowed us to identify the worst-case optimal submodular extension for Potts as well as the general metric labeling problems. It is worth noting that it might still be possible to obtain an improved submodular extension for a given problem instance. The design of a computationally feasible algorithm for this task is an interesting direction of future research. While our current work has focused on pairwise graphical models, it can be readily applied to high-order potentials by considering the corresponding LP relaxation objective as the Lovasz extension of a submodular extension. The identification of such extensions for popular high-order potentials such as the P^n Potts model or its robust version could further improve the accuracy of important computer vision applications such as semantic segmentation.

Acknowledgements

This work was supported by the EPSRC grants EP/P020658/1, TU/B/000048, and EP/P022529/1 and Google Deepmind PhD studentship.

References

Ajanthan, T., Desmaison, A., Bunel, R., Salzmann, M., Torr, P., and Kumar, M. (2017). Efficient linear programming for dense crfs. In *CVPR*.

Bartal, Y. (1996). Probabilistic approximation of metric spaces and its algorithmic applications. In *Foundations of Computer Science*.

Bartal, Y. (1998). On approximating arbitrary metrices by tree metrics. In *ACM Symposium on Theory of Computing*.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *PAMI*.

Chekuri, C., Khanna, S., Naor, J., and Zosin, L. (2004). A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics*.

Djolonga, J. and Krause, A. (2014). From map to marginals: Variational inference in bayesian submodular models. In *NIPS*.

Domke, J. (2013). Learning graphical model parameters with approximate marginal inference. *PAMI*.

Edmonds, J. (1970). Submodular functions, matroids, and certain polyhedra. *Combinatorial Optimization — Eureka, You Shrink!*

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*.

Kleinberg, J. and Tardos, E. (2002). Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *IEEE Symposium on the Foundations of Computer Science*.

Koltun, V. and Krahenbuhl, P. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*.

Kumar, M., Veksler, O., and Torr, P. (2011). Improved moves for truncated convex models. *JMLR*.

Manokaran, R., Naor, J., Raghavendra, P., and Schwartz, R. (2008). Sdp gaps and ugc hardness for multiway cut, 0-extension, and metric labeling. In *ACM Symposium on Theory of Computing*.

Scharstein, D., Szeliski, R., and Zabih, R. (2001). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision*.

Wainwright, M., Jaakkola, T., and Willsky, A. (2005). A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*.

Zhang, J., Djolonga, J., and Krause, A. (2015). Higher-order inference for multi-class log-supermodular models. In *ICCV*.

CHAPTER 6

LP-based Submodular Extensions for Marginal Estimation

This work was published in the journal Computer Vision and Image Understanding, 2019.

Computer Vision and Image Understanding

journal homepage: www.elsevier.com

Linear Programming-based Submodular Extensions for Marginal Estimation

Pankaj Pansari[a,], Chris Russell[b], M. Pawan Kumar[c]

[a] *University of Oxford*
[b] *University of Surrey, The Alan Turing Institute*
[c] *University of Oxford, The Alan Turing Institute*

ABSTRACT

Submodular extensions of an energy function can be used to efficiently compute approximate marginals via variational inference. The accuracy of the marginals depends crucially on the quality of the submodular extension. To identify accurate extensions for different classes of energy functions, we establish a relationship between the submodular extensions of the energy and linear programming (LP) relaxations for the corresponding MAP estimation problem. This allows us to (i) establish the worst-case optimality of the submodular extension for Potts model used in the literature; (ii) identify the worst-case optimal submodular extension for the more general class of metric labeling; (iii) efficiently compute the marginals for the widely used dense CRF model with the help of a recently proposed Gaussian filtering method; and (iv) propose an accurate submodular extension based on an LP relaxation for a higher-order diversity model. Using synthetic and real data, we show that our approach provides comparable upper bounds on the log-partition function to those obtained using tree-reweighted message passing (TRW) in cases where the latter is computationally feasible. Importantly, unlike TRW, our approach provides the first computationally tractable algorithm to compute an upper bound on dense CRF model with higher-order Potts potentials.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The desirable optimization properties of submodular set functions have been widely exploited in the design of approximate MAP estimation algorithms for discrete conditional random fields (CRFs) (Boykov et al., 2001; Kumar et al., 2011). Submodularity has also been recently used to design an elegant variational inference algorithm to compute the marginals of a discrete CRF by minimising an upper-bound on the log-partition function. In the initial work of Djolonga and Krause (2014), the energy of the CRF was restricted to be submodular. In a later work (Zhang et al., 2015), the algorithm was extended to handle more general Potts energy functions. The key idea there was to define a large ground set such that its subsets represent valid labelings, sublabelings or even incorrect labelings (these may assign two separate labels to a random variable and hence be invalid). Given the large ground set, it is possible to define a submodular set function whose value is equal to the energy of the CRF for subsets that specify a valid labeling of the model. We refer to such a set function as a *submodular extension* of the energy.

For a given energy function, there exists a large number of possible submodular extensions. The accuracy of the variational inference algorithm depends crucially on the

e-mail: pankaj@robots.ox.ac.uk (Pankaj Pansari)

choice of the submodular extension. Yet, previous work has largely ignored the question of identifying accurate extensions for different energy classes. Indeed, the difficulty of identifying submodular extensions of general energy functions could be a major reason why the experiments of Zhang et al. (2015) were restricted to the special case of models specified by the Potts energy functions.

In this work, we establish a hitherto unknown connection between the submodular extension of the Potts model proposed by Zhang et al. (2015), and the objective function of an accurate linear programming (LP) relaxation of the corresponding MAP estimation problem (Kleinberg and Tardos, 2002). Specifically, the Lovasz extension of a submodular extension can be shown to be an objective function for the LP relaxation. This connection has four important practical consequences. First, it establishes the optimality of the submodular extension of the Potts model, via the tightest LP relaxation (among first-level relaxations of the Sherali-Adams hierarchy (Sherali and Adams, 1990)) under UGC-hardness assumptions (Manokaran et al., 2008). Second, it provides an accurate submodular extension of the hierarchical Potts model, via the LP relaxation of the corresponding MAP estimation problem proposed by Kleinberg and Tardos (2002). Since any metric can be accurately approximated as a mixture of hierarchical Potts models (Bartal, 1996, 1998), this result also provides a computationally feasible algorithm for estimating the marginals for metric labeling. Third, it establishes the equivalence between the subgradient of the LP relaxation and the conditional gradient of the problem of minimising the upper bound of the log-partition. This allows us to employ the widely used dense CRF, since a subgradient of its LP relaxation can be efficiently computed using a recently proposed modified Gaussian filtering algorithm (Ajanthan et al., 2017). As a consequence, we provide the first efficient algorithm to compute an upper bound of the log-partition function of dense CRFs. This provides complementary information to the popular mean-field inference algorithm for dense CRFs, which computes a lower bound on the log-partition (Koltun and Krahenbuhl, 2011). Fourth, we obtain an accurate submodular extension for a higher-order diversity model based on an LP relaxation. Higher-order models can capture interesting properties of the image that cannot be expressed using pairwise models (Vineet et al., 2014; Kohli et al., 2007). We show that our upper-bounds on synthetic problems are comparable to those from tree reweighted message passing (TRW) (Wainwright et al., 2005) for the case of sparse CRFs. Unlike our approach, TRW is computationally infeasible for dense CRFs, thereby limiting its use in practice. Using pairwise dense CRF models, we perform stereo matching on standard data sets and obtain better results than Koltun and Krahenbuhl (2011). We also perform semantic segmentation on the MSRC-21 dataset using a combination of dense pairwise and higher-order diversity model. The complete code is available at `https://github.com/pankajpansari/denseCRF`.

2. Preliminaries

We now introduce the notation and definitions that we make use of in the remainder of the paper.

Submodular Functions: Given a ground set $U = \{1, \dots, N\}$, let us denote by 2^U its power set. A set function $F : 2^U \to \mathbb{R}$ is *submodular* if, for all subsets $A, B \subseteq U$

$$F(A \cup B) + F(A \cap B) \leq F(A) + F(B). \tag{1}$$

The set function F is *modular* if there exists $\mathbf{s} \in \mathbb{R}^N$ such that $F(A) = \sum_{k \in A} s_k \ \forall \ A \subseteq 2^U$. Henceforth, we will use the shorthand $s(A)$ to denote $\sum_{k \in A} s_k$.

Extended Polymatroid: Associated with any submodular function F is a special polytope known as the *extended polymatroid* defined as

$$EP(F) = \{\mathbf{s} \in \mathbb{R}^N | \ \forall A \subseteq U : s(A) \leq F(A)\}, \tag{2}$$

where $\mathbf{s}$ denotes the modular function $s(.)$ represented as a vector (Bach, 2013).

Lovasz Extension For a given set function F with $F(\emptyset) = 0$, the value of its Lovasz extension $f(\mathbf{w}) : \mathbb{R}^N \to \mathbb{R}$ is defined as follows (Bach, 2013) : order the components of $\mathbf{w}$ in decreasing order such that $w_{j_1} \geq w_{j_2} \geq \cdots \geq w_{j_N}$, where $(j_1, j_2, \ldots, j_N)$ is the corresponding permutation of the indices. Then,

$$f(\mathbf{w}) = \sum_{k=1}^{N} w_{j_k} \left(F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\}) \right). \tag{3}$$

The function f is an extension because it equals F on the vertices of the unit cube. That is, for any $A \subseteq V$, $f(\mathbf{1}_A) = F(A)$, where $\mathbf{1}_A$ is the 0-1 indicator vector corresponding to the elements of A.

Property 1. *By Edmond's greedy algorithm (Edmonds, 1970), if* $\mathbf{w} \geq 0$ *(non-negative elements),*

$$f(\mathbf{w}) = \max_{\mathbf{s} \in EP(F)} \langle \mathbf{w}, \mathbf{s} \rangle. \tag{4}$$

Property 1 implies that an LP over $EP(F)$ can be solved by computing the value of the Lovasz extension using equation (3).

Property 2. *The Lovasz extension f of a submodular function F is a convex piecewise linear function.*

Property 2 holds since $f(\mathbf{w})$ is the pointwise maximum of linear functions according to equation (4).

CRF and Energy Functions A CRF is defined as a graph on a set of random variables $\mathcal{X} = \{X_1, \ldots, X_N\}$ related by a set of edges $\mathcal{N}$. We wish to assign every variable X_a one of the labels from the set $\mathcal{L} = \{1, 2, \ldots, L\}$. The quality of a labeling $\mathbf{x}$ is given by an energy function defined as

$$E(\mathbf{x}) = \sum_{a \in \mathcal{X}} \phi_a(x_a) + \sum_{(a,b) \in \mathcal{N}} \phi_{ab}(x_a, x_b), \tag{5}$$

where ϕ_a and ϕ_{ab} are the unary and pairwise potentials respectively. In computer vision, we often think of $\mathcal{X}$ as arranged on a grid. A *sparse CRF* has $\mathcal{N}$ defined by 4-connected or 8-connected neighbourhood relationships. In a *dense CRF*, on the other hand, every variable is connected to every other variable.

We can augment the above pairwise models with higher-order potentials. Let $\mathcal{C}$ be the set of cliques on subgroups of variables. These cliques can, for instance, be the set of superpixels from a clustering method, such as k-means or mean-shift Comaniciu and Meer (2002). Also, let $\mathbf{x}_c$ be the component of $\mathbf{x}$ formed by elements in clique c. The energy function now also contains higher-order potentials

$$E(\mathbf{x}) = \sum_{a \in \mathcal{X}} \phi_a(x_a) + \sum_{(a,b) \in \mathcal{N}} \phi_{ab}(x_a, x_b) + \sum_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c), \tag{6}$$

The energy function can be interpreted as defining a probability distribution $P(\mathbf{x})$ as:

$$P(\mathbf{x}) = \begin{cases} \frac{1}{Z} \exp(-E(\mathbf{x})) & \text{if } \mathbf{x} \in \mathcal{L}^N, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The normalization factor $Z = \sum_{\mathbf{x} \in \mathcal{L}^N} \exp(-E(\mathbf{x}))$ is known as the *partition function.*

Inference There are two inference problems in CRFs:

(i) Marginal inference: We want to compute the marginal probabilities $P(X_a = i)$ for every $a = 1, 2, \ldots, N$ and $i = 1, 2, \ldots, L$.

(ii) MAP inference: We want to find a labeling with the minimum energy, that is, $\min_{\mathbf{x} \in \mathcal{L}^N} E(\mathbf{x})$. Equivalently, MAP inference finds the mode of $P(\mathbf{x})$.

3. Review: Variational Inference Using Submodular Extensions

We now summarise the inference method of Zhang et al. (2015) which made use of a submodular extension.

Submodular Extensions A submodular extension is defined using a ground set such that some of its subsets correspond to valid CRF labelings. Note that not every subset needs to represent a valid labeling - some of them could correspond to incomplete or invalid labelings. To obtain such an extension, we need an encoding scheme which gives the sets corresponding to valid CRF labelings.

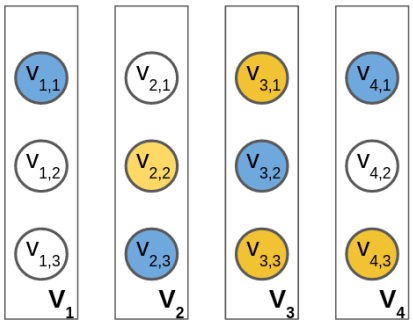


Fig. 1: *Illustration of* 1*-of-L encoding used in Zhang et al. (2015) with 4 variables and 3 labels. The blue labeling, corresponding to* $X_1 = 1, X_2 = 3, X_3 = 2, X_4 = 1$*, is valid. The yellow labeling, corresponding to* $X_2 = 2, X_3 = 1, 3, X_4 = 3$*, is invalid since* X_3 *has been assigned multiple labels and* X_1 *has been assigned none.*

One example of an encoding scheme is the 1-of-L encoding, illustrated in Figure 1. Let each variable X_a take one of L possible labels. In this scheme, we represent the set of possible assignments for X_a by the set $V_a = \{v_{a1}, v_{a2}, \ldots, v_{aL}\}$. If X_a is assigned label i, then we select the element v_{ai}. Extending to all variables, our ground set becomes $V = \cup_{a=1}^{N} V_a$. A valid assignment $A \subseteq V$ assigns each variable exactly one label, that is, $|A \cap V_a| = 1$ for all V_a. We denote the set of valid assignments by $\mathcal{M}$ where $\mathcal{M} = \cap_{a=1}^{N} \mathcal{M}_a$ and $\mathcal{M}_a = \{A : |A \cap V_a| = 1\}$.

Using our ground set V, we can define a submodular function F which equals $E(\mathbf{x})$ for all sets corresponding to valid labelings, that is, $F(A_{\mathbf{x}}) = E(\mathbf{x}),\ \mathbf{x} \in \mathcal{L}^N$ where $A_{\mathbf{x}}$ is the set encoding of $\mathbf{x}$. We call such a function F a *submodular extension* of $E(\mathbf{x})$.

Upper-Bound on Log-Partition Using a submodular extension F and given any $\mathbf{s} \in EP(F)$, we can obtain an upper-bound on the partition function as

$$\mathcal{Z} = \sum_{A \in \mathcal{M}} \exp(-F(A)) \leq \sum_{A \in \mathcal{M}} \exp(-s(A)), \qquad (8)$$

where $\mathcal{M}$ is the set of valid labelings. The upper-bound is the partition function of the distribution $Q(A) \propto \exp(-s(A))$, which factorises fully because $s(.)$ is modular. We can minimise this upper-bound, and the corresponding optimum $\mathbf{s}$ can help us obtain good approximate marginals of the distribution $P(\cdot)$. After taking logs for easy factorisaton, we can equivalently write our optimisation problem as

$$\min_{\mathbf{s} \in EP(F)} g(\mathbf{s}), \text{where } g(\mathbf{s}) = \log \sum_{A \in \mathcal{M}} \exp(-s(A)). \qquad (9)$$

Conditional Gradient Algorithm The conditional gradient algorithm (Algorithm 1) (Frank and Wolfe, 1956) is a good candidate for solving problem (9) due to two reasons. First, problem (9) is convex. Second, as solving an LP over EP(F) is computationally tractable (property 1), the conditional gradient can be found efficiently. The algorithm starts with an initial solution $\mathbf{s_0}$ (line 1). At each iteration, we compute the conditional gradient $\mathbf{s}^*$ (line 3), which minimises the linear approximation $g(\mathbf{s}_k) + \nabla g(\mathbf{s}_k)^T(\mathbf{s} - \mathbf{s}_k)$ of the objective function. Finally, $\mathbf{s}$ is updated by either (i) fixed step size schedule, as in line 7 of algorithm 1, or (ii) by doing a line search and setting $\mathbf{s}_{k+1} = \min_{0 \leq \gamma \leq 1} g(\gamma \mathbf{s}^* + (1-\gamma)\mathbf{s}_k)$.

Algorithm 1 Upper Bound Minimisation using Conditional Gradient Descent

1: Initialize $\mathbf{s} = \mathbf{s}_0 \in EP(F)$
2: **for** k = 1 to `MAX_ITER` **do**
3: $\quad \mathbf{s}^* = \text{argmin}_{\mathbf{s} \in EP(F)} \langle \nabla g(\mathbf{s}_k), \mathbf{s} \rangle$
4: $\quad$ **if** $\langle \mathbf{s}^* - \mathbf{s}_k, \nabla g(\mathbf{s}_k) \rangle \leq \epsilon$ **then**
5: $\quad\quad$ **break**
6: $\quad$ **end if**
7: $\quad \mathbf{s}_{k+1} = \mathbf{s}_k + \gamma(\mathbf{s}^* - \mathbf{s}_k)$ with $\gamma = 2/(k+2)$
8: **end for**
9: **return** $\mathbf{s}$

4. Overview: Accurate Submodular Extensions from LP Relaxations

Different extensions F change the domain in problem (9), leading to different upper bounds on the log-partition function. How does one come up with accurate extensions for different classes of CRF energy? Is it possible to identify optimal extensions which yield the tightest bound?

If we introduce a temperature parameter in $P(\mathbf{x})$ (equation (7)) by using $E(\mathbf{x})/T$ and decrease T, the resulting distribution starts to peak more sharply around its mode. We assume that $P(\mathbf{x})$ is unimodal, and there aren't multiple solutions with the same minima. As $T \to 0$, marginal estimation becomes the same as MAP inference since the resulting distribution $P^0(\mathbf{x})$ has mass 1 at its mode $\mathbf{x}^*$ and is 0 everywhere else. Given the MAP solution $\mathbf{x}^*$, one can compute the marginals as $P(X_i = j) = [x_i^* = j]$, where [.] is the Iverson bracket. We point out that the notion of a temperature parameter T in a probability distribution and indeed variational methods for approximating complicated distributions have their origins in statistical physics literature - the interested reader is referred to MacKay (2003, chapter 33). Motivated by this connection, we ask if one can introduce a temperature parameter to our problem (9) and transform it to an LP relaxation in the limit $T \to 0$? We can then use accurate LP relaxations of MAP problems known in literature to find good submodular extensions for different classes of energy functions.

We answer this question in the affirmative. Specifically, in the following two sections we show how one can select the set encoding and submodular extension to convert problem (9) to accurate LP relaxations for Potts, hierarchical Potts and higher-order diversity models. When the LP relaxation has tightness guarantees, we obtain worst-case optimal submodular extensions - a notion we elucidate shortly.

In this work, we focus on obtaining submodular extensions with closed-form analytical expressions for different classes of energy functions. Formally, for a specific energy class (such as the Potts model) $\mathcal{E}$, we derive a family of submodular functions $\mathcal{F}(.)$. Given an instance of the energy function $E(.)$ from the class $\mathcal{E}$, the corresponding submodular extension is $\mathcal{F}(E)$.

5. Worst-case Optimal Submodular Extensions

Worst-case Optimality

Our extensions are derived from LP relaxations belonging to the first-level of the Sherali-Adams hierarchy. This level yields LP relaxations having $\mathcal{O}(N)$ relaxed variables for N variables in the CRF. In principle, these LP relaxations can be made tighter by introducing more variables and constraints, thereby moving to higher levels in the hierarchy ($\mathcal{O}(N^k)$ relaxed variables for k-th level). However, in practice, working with these higher-order relaxations is computationally infeasible for large-scale vision problems. In the discussion that follows, we assume that the LP relaxations belong to the first-level of Sherali-Adams hierarchy. When the LP relaxation is the tightest possible, the extension family $\mathcal{F}_{opt}$ we obtain is *worst-case optimal*. That is, there does not exist another submodular extension family $\mathcal{F}$ that gives a tighter upper bound for problem (9) than $\mathcal{F}_{opt}$ for all instances of the energy function in $\mathcal{E}$. Alternatively, for any other submodular extension family $\mathcal{F}$, we can find at least one instance of energy function $E(.)$ such that $\mathcal{F}_{opt}$ results in a tighter upper-bound. Formally, $\mathcal{F}_{opt}$ is worst-case optimal if

$$\nexists \mathcal{F} : \min_{\mathbf{s} \in EP(\mathcal{F}(E))} g(\mathbf{s}) \leq \min_{\mathbf{s} \in EP(\mathcal{F}_{opt}(E))} g(\mathbf{s}) \ \ \forall \ E(.) \in \mathcal{E}. \tag{10}$$

Note that the notion of worst-case optimality does not guarantee that the extension is best for every instance of the energy function. It may be possible to solve an optimisation problem to obtain the best extension for every given instance of energy $E(.)$. However, in this paper, we do not take that approach, and instead provide a general solution for a given class of energy functions.

In subsection 5.1, we prove the worst-case optimality of the submodular extension used in literature for the Potts model. In subsection 5.2, we obtain worst-case optimal submodular extension for the more general hierarchical Potts model. Finally, in subsection 5.3, we provide a way to make our inference algorithm efficient for the dense CRF model. Using our approach, the inference algorithm

has linear time-complexity per iteration in the number of variables and labels.

5.1. Pairwise Potts Model

The pairwise Potts model, also known as the uniform metric, specifies the pairwise potentials $\phi_{ab}(x_a, x_b)$ in equation (5) as follows:

$$\phi_{ab}(x_a, x_b) = w_{ab} \cdot [x_a \neq x_b], \tag{11}$$

where w_{ab} is the weight associated with edge (a, b). There are no higher-order potential terms in this model as in equation (5). We note that the Potts model is non-submodular according to the popular encoding used in Ishikawa (2003). However, we use the 1-of-L encoding to construct the submodular extension.

Tightest LP Relaxation Before describing our set encoding and submodular extension, we briefly outline the LP relaxation of the corresponding MAP estimation problem. To this end, we define indicator variables y_{ai} which equal 1 if $X_a = i$, and 0 otherwise. The following is the tightest possible LP relaxation (among first-level LP relaxations) for Potts model in the worst-case, assuming the Unique Games Conjecture is true (see (EM-LP) for metric labeling in Manokaran et al. (2008) and its connection to (P-LP) in Chekuri et al. (2004)):

$$\begin{aligned} \text{(P-LP)} \quad \min_{\mathbf{y}} \; & E(\mathbf{y}) = \sum_a \sum_i \phi_a(i) y_{ai} + \\ & \sum_{(a,b)\in\mathcal{N}} \sum_i \frac{w_{ab}}{2} \cdot |y_{ai} - y_{bi}| \\ \text{s.t} \quad & \mathbf{y} \in \Delta. \end{aligned} \tag{12}$$

The set Δ is the union of N probability simplices:

$$\Delta = \{\mathbf{y}_a \in \mathbb{R}^L | \mathbf{y}_a \geq 0 \text{ and } \langle \mathbf{1}, \mathbf{y}_a \rangle = 1\}, \tag{13}$$

where $\mathbf{y}$ is the vector of all variables and $\mathbf{y}_a$ is the component of $\mathbf{y}$ corresponding to X_a.

Set Encoding We choose to use the 1-of-L encoding for Potts model as described in Section 3. With the encoding scheme for Potts model above, $g(\mathbf{s})$ can be factorised and problem (9) can be rewritten as:

$$\min_{\mathbf{s}\in EP(F)} \quad \sum_{a=1}^{N} \log \sum_{i=1}^{L} \exp(-s_{ai}). \tag{14}$$

(See Remark 1 in appendix A for proof).

We note that the gradient $\nabla g(\mathbf{s})$ can be computed efficiently as $[\nabla g(\mathbf{s})]_{ai} = -\exp^{-s_{ai}} / \sum_{i=1}^{L} \exp^{-s_{ai}}$, these being the marginals of the fully-factorised distribution $Q(A)$.

Marginal Estimation with Temperature We now introduce a temperature parameter $T > 0$ to problem (14) which divides $E(\mathbf{x})$, or equivalently divides $\mathbf{s}$ belonging to $EP(F)$. Also, since $T > 0$, we can multiply the objective by T leaving the problem unchanged. Without changing the solution, we can transform problem (14) as follows

$$\min_{\mathbf{s}\in EP(F)} \quad g_T(\mathbf{s}) = \sum_{a=1}^{N} T \cdot \log \sum_{i=1}^{L} \exp(-\frac{s_{ai}}{T}). \tag{15}$$

Worst-case Optimal Submodular Extension We now connect our marginal estimation problem (9) with LP relaxations using the following proposition.

Proposition 1. *In the limit $T \to 0$, problem (15) becomes:*

$$-\min_{\mathbf{y}\in\Delta} f(\mathbf{y}) \tag{16}$$

where $f(.)$ is the Lovasz extension of $F(.)$.

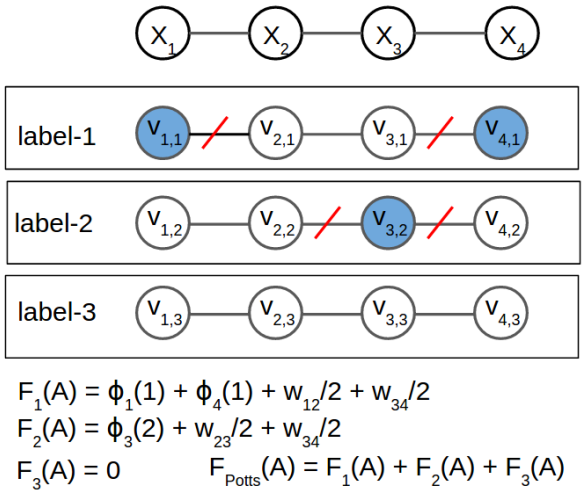


Fig. 2: *An illustration of the worst-case optimal submodular extension for Potts model for a chain graph of 4 variables, each of which can take 3 labels. The figure shows how to compute extension for $A = \{v_{1,1}, v_{4,1}, v_{3,2}\}$.*

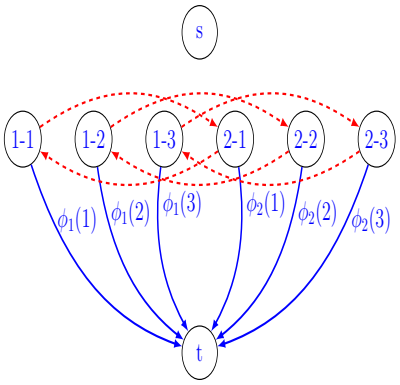


Fig. 3: *An st-graph specifying the worst-case optimal submodular extension for Potts model for 2 variables with 3 labels each and connected to each other. There is a node for each variable and each label, that is, for all elements of the ground set. The nodes have been labeled as 'variable-label', hence node 1-1 represents the element v_{11} and so on. The solid blue arcs model the unary potentials, and the dotted red arcs represent the pairwise potentials. Each dotted red arc has weight $w_{12}/2$.*

Proof. In the limit $T \to 0$, we can rewrite problem (15) as

$$\min_{\mathbf{s} \in EP(F)} \quad \sum_{a=1}^{N} \max_{i}(-s_{ai}). \tag{17}$$

In vector form, the problem becomes

$$\min_{\mathbf{s} \in EP(F)} \max_{\mathbf{y} \in \Delta} -\langle \mathbf{y}, \mathbf{s} \rangle \tag{18}$$

$$= - \max_{\mathbf{s} \in EP(F)} \min_{\mathbf{y} \in \Delta} \langle \mathbf{y}, \mathbf{s} \rangle \tag{19}$$

where Δ is the domain as defined as in equation (13). By the minimax theorem Boyd and Vandenberghe (2004) for LP, we can reorder the terms:

$$- \min_{\mathbf{y} \in \Delta} \max_{\mathbf{s} \in EP(F)} \langle \mathbf{y}, \mathbf{s} \rangle \tag{20}$$

Recall that $\max_{\mathbf{s} \in EP(F)} \langle \mathbf{y}, \mathbf{s} \rangle$ is the value of the Lovasz extension of F at $\mathbf{y}$, that is, $f(\mathbf{y})$. Hence, as $T \to 0$, the marginal inference problem converts to minimising the Lovasz extension under the simplices constraint:

$$- \min_{\mathbf{y} \in \Delta} f(\mathbf{y}) \tag{21}$$

□

The problem (21) is equivalent to an LP relaxation of the corresponding MAP estimation problem (see Lemma 1 in appendix). Indeed, $g_T(\mathbf{s})$ in problem (15) becomes the objective function of an LP relaxation in the limit $T \to 0$. We seek to obtain the worst-case optimal submodular extension by making $g_T(\mathbf{s})$ same as the objective of (P-LP) as $T \to 0$. We note that problems (15) and (14) become equivalent at $T = 1$, and our worst-case optimality guarantee was derived in the limit $T \to 0$. However, it is possible to derive such a guarantee only in the limit case, and we claim the worst-case optimality in this sense.

The question now becomes how to recover the worst-case optimal submodular extension using $E(\mathbf{y})$. The following proposition answers this question.

Proposition 2. *The worst-case optimal submodular extension for Potts model is given by $F_{Potts}(A) = \sum_{i=1}^{L} F_i(A)$, where*

$$F_i(A) = \sum_{a} \phi_a(i)[|A \cap \{v_{ai}\}| = 1] + \sum_{(a,b) \in \mathcal{N}} \frac{w_{ab}}{2} \cdot [|A \cap \{v_{ai}, v_{bi}\}| = 1] \tag{22}$$

Also, $E(\mathbf{y})$ in (P-LP) is the Lovasz extension of F_{Potts}.

Proof. Since F_{Potts} is sum of Ising models F_i, we first focus on a particular label i and then generalize. Consider a graph with only two variables X_a and X_b with an edge between them. The ground set in this case is $\{v_{ai}, v_{bi}\}$. Let the corresponding relaxed indicator variables be $\mathbf{y} = \{y_{aj}, y_{bj}\}$, such that $y_{ai}, y_{bi} \in [0, 1]$ and assume $y_{ai} > y_{bi}$. The Lovasz extension in this case is:

$$\begin{aligned} f(\mathbf{y}) &= y_{ai} \cdot [F_i(\{v_{ai}\}) - F_i(\{\})] \\ &\quad + y_{bi} \cdot [F_i(\{v_{ai}, v_{bi}\}) - F_i(\{v_{ai}\})] \\ &= y_{ai} \cdot [\Big(\phi_a(j) + \frac{w_{ab}}{2}\Big) - 0] + y_{bi} \cdot [(\phi_a(j) + \phi_b(j)) \\ &\quad - \Big(\phi_a(j) + \frac{w_{ab}}{2}\Big)] \\ &= \phi_a(j) \cdot y_{ai} + \phi_b(j) \cdot y_{bi} + \frac{w_{ab}}{2} \cdot (y_{ai} - y_{bi}) \end{aligned} \tag{23}$$

In general for both orderings of y_{ab} and y_{bi}, we can write

$$f(\mathbf{y}) = \phi_a(j) \cdot y_{ai} + \phi_b(j) \cdot y_{bi} + \frac{w_{ab}}{2} \cdot |y_{ai} - y_{bi}| \tag{24}$$

Extending the Lovasz extension of equation (24) to all variables and all labels gives us $E(\mathbf{y})$ in (P-LP).

□

Proposition 2 paves the way for us to identify the worst-case optimal extension for hierarchical Potts model. Figure 2 shows an example where we compute $F_{Potts}(A)$ for a small graph and a given set A representing an invalid labeling. As an alternate representation, Figure 3 shows the *st*-graph corresponding to F_{Potts} for a small instance. Each *st*-cut corresponds to a labeling, and the cost of each cut matches F_{Potts} (subsection 4.1 of Wang et al. (2013)).

5.2. Hierarchical Potts Model

Potts model imposes the same penalty for unequal assignment of labels to neighbouring variables, regardless of the label dissimilarity. In some scenarios, a more natural approach is to vary the penalty based on how different the labels are. A hierarchical Potts model (Kleinberg and Tardos, 2002) permits this by specifying the distance between labels using a tree with the following properties:

1. The vertices are of two types: (i) the leaf nodes representing labels, and (ii) the non-leaf nodes, except the root, representing meta-labels.

2. The lengths of all the edges from a parent to its children are the same.

3. The lengths of the edges along any path from the root to a leaf decreases by a factor of at least $r \geq 2$ at each step.

4. The metric distance between nodes of the tree is the sum of the edge lengths on the unique path between them.

A subtree T of an hierarchical Potts model is a tree comprising all the descendants of some node v (not root). Given a subtree T, l_T denotes the length of the tree-edge leading upward from the root of T and $L(T)$ denotes the set of leaves of T. We call the leaves of the tree as labels and all other nodes of the tree expect the root as *meta-labels*. Figure 4 illustrates the notations in the context of a hierarchical Potts model.

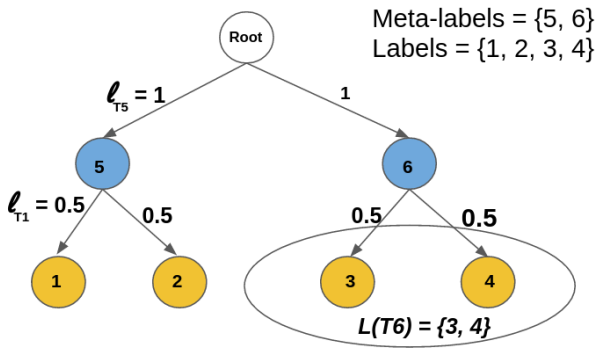


Fig. 4: *A hierarchical Potts model instance illustrating notations with 2 meta-labels (blue) and 4 labels (yellow). All labels are at the same level. $r = 2$, that is, edge-length decreases by 2 at each level. Also, distance between labels 1 and 3 is $d(1,3) = 0.5 + 1 + 1 + 0.5 = 3$.*

We note that the hierarchical Potts model is non-submodular according to the popular encoding used in Ishikawa (2003). However, we use the 1-of-L encoding to construct its submodular extension.

Tightest LP Relaxation We use the same indicator variables y_{ai} that were employed in the LP relaxation of Potts model. Let $y_a(T) = \sum_{i \in L(T)} y_{ai}$. The following LP relaxation is the tightest known (among the LP relaxations of first-level Sherali-Adams hierarchy) for hierarchical Potts model in the worst-case, assuming the Unique Games Conjecture is true (Manokaran et al., 2008)

$$\text{(T-LP)} \quad \min_{\mathbf{y}} \ \widetilde{E}(\mathbf{y}) = \sum_a \sum_i \phi_a(i) y_{ai} + \sum_{(a,b) \in \mathcal{N}} w_{ab} \sum_T l_T \cdot |y_a(T) - y_b(T)|$$
$$\text{such that} \quad \mathbf{y} \in \Delta. \tag{25}$$

where we sum T over all subtrees. The set Δ is the same domain as defined in equation (13).

Transformed Tightest LP Relaxation We take (T-LP) and rewrite it using indicator variables for all labels and meta-labels. Let $\mathcal{R}$ denote the set of all labels and meta-labels, that is, all nodes in the tree apart from the

root. Also, let $\mathcal{L}$ denote the set of labels, that is, the leaves of the tree. Let T_i denote the subtree which is rooted at the i-th node. We introduce an indicator variable $z_{ai} \in \{0, 1\}$

$$z_{ai} = \begin{cases} y_{ai} & \text{if } i \in \mathcal{L} \\ y_a(T_i) & \text{if } i \in \mathcal{R} - \mathcal{L} \end{cases} \tag{26}$$

We need to extend the definition of unary potentials to the expanded label space as follows:

$$\text{where } \phi'_a(i) = \begin{cases} \phi_a(i) & \text{if } i \in \mathcal{L} \\ 0 & \text{if } i \in \mathcal{R} - L \end{cases} \tag{27}$$

We can now rewrite problem (25) in terms of new indicator variables z_{ai}:

$$\begin{aligned} \text{(T-LP-FULL)} \quad \min \widetilde{E}(\mathbf{z}) = & \sum_{i \in \mathcal{R}} \sum_{a \in \mathcal{X}} \phi'_a(i) \cdot z_{ai} + \\ & \sum_{i \in \mathcal{R}} \sum_{(a,b) \in \mathcal{N}} w_{ab} \cdot l_{T_i} \cdot |z_{ai} - z_{bi}| \\ \text{such that} \quad & \mathbf{z} \in \Delta' \end{aligned} \tag{28}$$

where Δ' is the convex hull of the vectors satisfying

$$\sum_{i \in \mathcal{L}} z_{ai} = 1, \;\; z_{ai} \in \{0, 1\} \;\; \forall a \in \mathcal{X}, i \in \mathcal{L} \tag{29}$$

$$\text{and } \; z_{ai} = \sum_{j \in L(T_i)} z_{aj}. \;\; \forall a \in \mathcal{X}, i \in \mathcal{R} - \mathcal{L} \tag{30}$$

Constraint (30) ensures consistency among labels and meta-labels, that is, if a label is assigned then all the meta-labels which lie on the path from the root to the label should be assigned as well.

Set Encoding For any variable X_a, let the set of possible assignment of labels and meta-labels be the set $V_a = \{v_{a1}, \ldots, v_{aM}\}$, where M is the total number of nodes in the tree except the root. Our ground set is $V = \cup_{a=1}^{N} V_a$ of size $N \cdot M$.

A consistent labeling of a variable assigns it one label, and all meta-labels on the path from root to the label. Let us represent the set of consistent assignments for X_a by the set $P_a = \{p_{a1}, \ldots, p_{aL}\}$, where p_{ai} is the collection of elements from V_a for label i and all meta-labels on the path from root to label i. The set of valid labelings $A \subseteq V$ assigns each variable exactly one consistent label. This constraint can be formally written as $\mathcal{M} = \cap_{a=1}^{N} \mathcal{M}_a$ where $\mathcal{M}_a$ has exactly one element from P_a. Let s'_{ai} be the sum of the components of $\mathbf{s}$ corresponding to the elements of p_{ai}, that is,

$$s'_{ai} = \sum_{t \in p_{ai}} s_t. \tag{31}$$

Using our encoding scheme, we rewrite problem (9) as:

$$\min_{\mathbf{s} \in EP(F)} \quad \sum_{a=1}^{N} \log \sum_{i=1}^{L} \exp(-s'_{ai}). \tag{32}$$

Marginal Estimation with Temperature Similar to Potts model, we now introduce a temperature parameter $T > 0$ to problem (32). The transformed problem becomes

$$\min_{\mathbf{s} \in EP(F)} \quad g_T(\mathbf{s}) = \sum_{a=1}^{N} T \cdot \log \sum_{i=1}^{L} \exp(-\frac{s'_{ai}}{T}). \tag{33}$$

Worst-case Optimal Submodular Extension The following proposition connects the marginal estimation problem (9) with LP relaxations:

Proposition 3. *In the limit $T \to 0$, problem (33) becomes:*

$$-\min_{\mathbf{z} \in \Delta'} f(\mathbf{z}) \tag{34}$$

(Proof in appendix).

The above problem is equivalent to an LP relaxation of the corresponding MAP estimation problem (see Lemma 2 in appendix). Hence, $g_T(\mathbf{s})$ becomes the objective function of an LP relaxation in the limit $T \to 0$. We seek to make this objective same as $\widetilde{E}(\mathbf{z})$ of (T-LP-FULL) in the limit $T \to 0$. The question now becomes how to recover the worst-case optimal submodular extension from $\widetilde{E}(\mathbf{z})$.

Proposition 4. *The worst-case optimal submodular extension for hierarchical Potts model is given by $F_{hier}(A) = \sum_{i=1}^{M} F_i(A)$, where*

$$\begin{aligned} F_i(A) = & \sum_{a} \phi'_a(i)[|A \cap \{v_{ai}\}| = 1] + \\ & \sum_{(a,b) \in \mathcal{N}} w_{ab} \cdot l_{T_i} \cdot [|A \cap \{v_{ai}, v_{bi}\}| = 1] \end{aligned} \tag{35}$$

Also, $\widetilde{E}(\mathbf{z})$ in (T-LP-FULL) is the Lovasz extension of F_{hier}. (Proof in appendix)

Since any finite metric space can be probabilistically approximated by mixture of tree metric (Bartal, 1996), the worst-case optimal submodular extension for metric energies can be obtained using F_{hier}. Note that F_{hier} reduces to F_{Potts} for Potts model. One can see this by considering the Potts model as a star-shaped tree with edge weights as 0.5.

5.3. Fast Conditional Gradient Computation for Dense Conditional Random Fields

Dense CRF Energy Function A dense CRF is specified by the following energy function

$$E(\mathbf{x}) = \sum_{a\in\mathcal{X}} \phi_a(x_a) + \sum_{a\in\mathcal{X}} \sum_{b\in\mathcal{X}, b\neq a} \phi_{ab}(x_a, x_b). \tag{36}$$

Note that every random variable is a neighbour of every other random variable in a dense CRF. Similar to previous work (Koltun and Krahenbuhl, 2011), we consider the pairwise potentials to be Gaussian, that is,

$$\phi_{ab}(i,j) = \mu(i,j) \sum_m w^{(m)} k(\mathbf{f}_a^{(m)}, \mathbf{f}_b^{(m)}), \tag{37}$$

$$k(\mathbf{f}_a^{(m)}, \mathbf{f}_b^{(m)}) = \exp\left(\frac{-||\mathbf{f}_a - \mathbf{f}_b||^2}{2}\right). \tag{38}$$

The term $\mu(i,j)$ is known as *label compatibility* function between labels i and j. Potts model and hierarchical Potts models are examples of $\mu(i,j)$. The other term is a mixture of Gaussian kernels $k(.,.)$ and is called the *pixel compatibility* function. The terms $\mathbf{f}_a^{(m)}$ are features that describe the random variable X_a. In practice, similar to Koltun and Krahenbuhl (2011), we use x, y coordinates and RGB values associated to a pixel as its features.

Algorithm 1 assumes that the conditional gradient $\mathbf{s}^*$ in step 3 can be computed efficiently. This is certainly not the case for dense CRFs, since computing s^* involves NL function evaluations of the submodular extension F, where N is the number of variables, and L is the number of labels. Each F evaluation has complexity $\mathcal{O}(N)$ using the efficient Gaussian filtering algorithm of Koltun and Krahenbuhl (2011). However, computation of s^* would still be $\mathcal{O}(N^2)$, which is clearly impractical for computer-vision applications where $N \sim 10^5 - 10^6$.

However, using the equivalence of relaxed LP objectives and the Lovasz extension of submodular extensions in Proposition 1, we are able to compute s^* in $\mathcal{O}(NL)$ time. Specifically, we use the algorithm of Ajanthan et al. (2017), which provides an efficient filtering procedure to compute a subgradient of the LP relaxation objective $E(\mathbf{y})$ of (P-LP).

Proposition 5. *Computing a subgradient of $E(\mathbf{y})$ in (P-LP) is equivalent to computing the conditional gradient for the submodular function F_{Potts}.*

Proof. For the Potts model, we derived the worst-case optimal extension F_{Potts} by making its Lovasz extension $f(\mathbf{y})$ same as the objective function $E(\mathbf{y})$ of the worst-case optimal LP relaxation. Hence, we have

$$\begin{aligned} E(\mathbf{y}) &= f(\mathbf{y}) \\ &= \max_{\mathbf{s}\in EP(F)} \langle \mathbf{y}, \mathbf{s} \rangle. \end{aligned}$$

The subgradient of $E(\mathbf{y})$ at $\mathbf{y_0}$ is an 'active' linear function at $\mathbf{y_0}$. Hence,

$$\partial E(\mathbf{y})|_{\mathbf{y}=\mathbf{y_0}} \in \underset{\mathbf{s}\in EP(F)}{\operatorname{argmax}} \langle \mathbf{y_0}, \mathbf{s} \rangle \tag{39}$$

Equation (39) is equivalent to $\mathbf{s}^*$ computation in line 3 of algorithm 1, with $\mathbf{y_0} = -\nabla g(\mathbf{s}_k)$. □

A similar observation can be made in case of hierarchical Potts model. Hence we have the first practical algorithm to compute upper bound of log-partition function of a dense CRF for Potts and metric energies.

6. Accurate Submodular Extension for Higher-order Diversity Model

The pairwise Potts model often fails to capture useful image statistics because it restricts the order of the potentials to be at most two. Higher order clique potentials can model complex interactions of random variables, and thereby overcome this difficulty.

A higher-order model useful in real-world applications is the *diversity* model, which favours labeling where variables in a clique have fewer number of unique labels. For instance, in semantic segmentation we often first obtain superpixels from a clustering method (Comaniciu and Meer, 2002), which we consider as cliques in our model. We expect the labeling in a superpixel to be homogeneous. As a result, it is preferable to have pixels belonging to a superpixel to be incorrectly labeled with two class labels rather than three or more class labels. Let $\Gamma(\mathbf{x}_c)$ be the set of unique labels assigned to variables in the clique c, and ω_c be the weight associated with the clique. In our case, we take the clique potentials as proportional to the number of unique labels in the clique c:

$$\phi_c(\mathbf{x}_c) = \omega_c |\Gamma(\mathbf{x}_c)| \tag{40}$$

where the notation $|A|$ denotes the cardinality of a set A.

LP Relaxation First, let us consider the IP formulation of the MAP problem for our diversity model. Using the same set of indicator variables $y_{ai} = \{0, 1\}$ as for the Potts model, which are binary now, the clique potential $\phi(\mathbf{x}_c)$ can be represented as

$$\phi_c(\mathbf{x}_c) = \omega_c \cdot \sum_{i=1}^{L} \max_{(a,b)\in c} |y_{ai} - y_{bi}| \tag{41}$$

For example, let the set of unique labels in a clique be $\Gamma = \{l_1, l_2, l_3\}$. This implies that $y_{ai} = 1$ for some, but not all, variables in the clique for $i = \{1, 2, 3\}$. Hence, $|y_{ai} - y_{bi}| = 1$ for some pairs $(a, b) \in c$ for $i = \{1, 2, 3\}$. Also, for any other label i, $|y_{ai} - y_{bi}| = 0$ for all $(a, b) \in c$ since $y_{ai} = 0$ for all variables a for these labels i. For binary variables, the maximum possible value of $|y_{ai} - y_{bi}|$ is 1. Hence, $\phi_c(\mathbf{x}_c) = \omega_c \cdot (1 + 1 + 1) = 3 \cdot \omega_c = \omega_c |\{l_1, l_2, l_3\}|$.

We now relax the indicator variables y_{ai} to lie in $[0, 1]$. Assuming the pairwise potentials in equation (6) to be Potts, an accurate LP relaxation for the higher-order diversity model is given by:

$$\begin{aligned}
\text{(HOD-LP)} \quad \min_{\mathbf{y}} \quad & E(\mathbf{y}) = \sum_{a=1}^{N}\sum_{i=1}^{L} \phi_a(i) y_{ai} \\
& + \sum_{(a,b)\in\mathcal{N}} \sum_{i} \frac{w_{ab}}{2} \cdot |y_{ai} - y_{bi}| \\
& + \sum_{c\in\mathcal{C}} w_c \cdot \sum_{i=1}^{L} \max_{(a,b)\in c} |y_{ai} - y_{bi}| \\
\text{s.t} \quad & \mathbf{y} \in \Delta.
\end{aligned} \tag{42}$$

where the label (HOD-LP) stands for LP relaxation for higher-order diversity model. The objective function is similar to that for the Potts model along with the additional terms corresponding to higher-order potentials. The constraints for (HOD-LP) are the same as for (P-LP). The above LP has not been formally analysed in literature, and we do not make any optimality claims. However, its similarity in form to (P-LP) is an indication of its accuracy. We establish its accuracy empirically in section 7.

LP-based Submodular Extension We use the same 1-of-L encoding scheme as for Potts model. Problem 9 then factorises as:

$$\min_{\mathbf{s}\in EP(F)} \quad \sum_{a=1}^{N} \log \sum_{i=1}^{L} \exp(-s_{ai}). \tag{43}$$

We introduce a temperature parameter T to problem (43) to obtain the following new problem:

$$\min_{\mathbf{s}\in EP(F)} \quad g_T(\mathbf{s}) = \sum_{a=1}^{N} T \cdot \log \sum_{i=1}^{L} \exp(-\frac{s_{ai}}{T}). \tag{44}$$

In the limit $T \to 0$, problem (44) becomes

$$-\min_{\mathbf{y}\in\Delta} f(\mathbf{y}) \tag{45}$$

where $f(.)$ is the Lovasz extension of $F(.)$. We are interested in making this LP the same as (HOD-LP) of equation (42), thereby enabling us to obtain the submodular extension for the higher-order diversity model. We will make use of the following lemma in our proof.

Lemma 1. *The Lovasz extension of the set function* $F_{HOD}(A) =$

$$F_{Potts}(A) + \sum_{c\in\mathcal{C}} w_c \cdot \sum_{i=1}^{L} \max_{(a,b)\in c} [|A \cap \{v_{ai}, v_{bi}\}| = 1]$$

is the objective function $E(\mathbf{y})$ of the LP relaxation referred to as (HOD-LP).

Proof. Making use of Proposition 2, it suffices for us to show that the Lovasz extension of the i-th Ising model $F_i(A) = \max_{(a,b)\in c}[|A \cap \{v_{ai}, v_{bi}\}| = 1]$ is $\max_{(a,b)\in c} |y_{ai} - y_{bi}|$.

Let the clique c have M variables. Given $\mathbf{y} \in [0,1]^M$, let us order its components in decreasing order $y_{j_1} > \cdots > y_{j_M}$, where $(j_1, \ldots, j_M)$ is a permutation. Then the Lovasz extension of $F_i()$ is:

$$f_i(\mathbf{y}) = \sum_{k=1}^{M} y_{j_k}[F_i(\{j_1, \ldots, j_k\}) - F_i(\{j_1, \ldots, j_{k-1}\})]$$

We note that when $A = \{\}$ or $\{j_1, \ldots, j_M\}$, that is, when A represents homogeneous labelings, $F_i(A) = 0$. For any other $A \subset \{j_1, \ldots, j_M\}$, $F_i(A) = 1$. Hence, the Lovasz extension simplifies to

$$\begin{aligned} f_i(\mathbf{y}) &= y_{j_1} - y_{j_M} \\ &= \max_{(a,b)\in c} |y_{ai} - y_{bi}| \end{aligned}$$

This proves our lemma. □

We are now in a position to state our main result:

Proposition 6. *The set function F_{HOD} of lemma 1 is the submodular extension of our higher-order diversity model whose Lovasz extension is $E(\mathbf{y})$ in problem (HOD-LP).*

Proof. Note that $\forall \mathbf{y} \in \mathcal{L}^N$, that is, for all valid labelings, $F_{HOD}(A_\mathbf{y})$ equals $E(\mathbf{y})$. Hence, F_{HOD} is an extension.

Lemma 1 showed that the Lovasz extension of F_{HOD} is $E(\mathbf{y})$ in problem (HOD-LP). $E(\mathbf{y})$ being a sum of terms corresponding to maximum of linear functions, is convex. The convexity of Lovasz extension implies submodularity of the set function Bach (2013). Hence, F_{HOD} is submodular. □

We now show the effectiveness of the accurate submodular extensions for different classes of models by means of experiments on synthetic and real-world datasets.

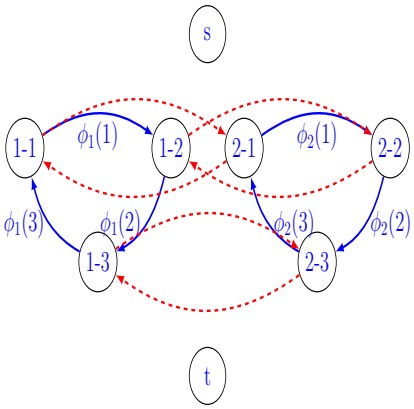


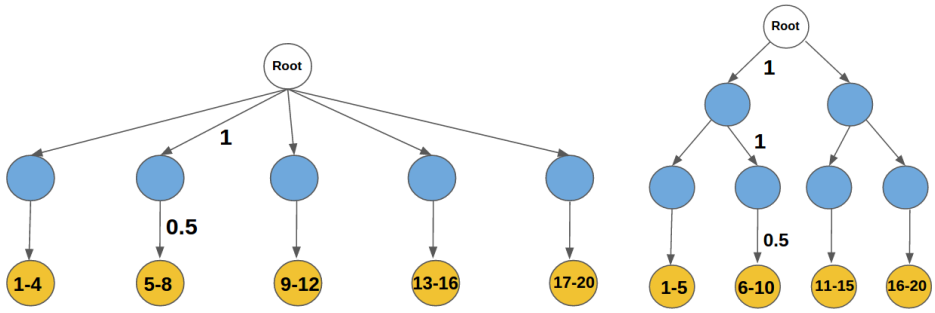
Fig. 5: **Alternate extension for synthetic experiments**: *An st-graph specifying the alternate submodular extension for Potts model for 2 variables with 3 labels each and connected to each other. The convention used is same as in figure 3. Each dotted red arc has weight $w_{12}/2$. This alternate extension was also used to derive the extension for hierarchical Potts model.*

7. Experiments

Using synthetic data, we show that our upper-bound compares favorably with TRW for both Potts and hierarchical Potts models. For comparison, we restrict ourselves to sparse CRFs, as the code available for TRW does not scale well to dense CRFs. We also perform stereo matching using dense CRF models and compare our results with the mean-field-based approach of Koltun and Krahenbuhl (2011). All experiments were run on a x86-64, 3.8GHz machine with 16GB RAM. In this section, we refer to our algorithm as *Submod* and mean field as *MF*.

7.1. Upper-bound Comparison using Synthetic Data

Data We generate lattices of size 100 × 100, where each lattice point represents a variable taking one of 20 labels. The pairwise relations of the sparse CRFs are defined by 4-connected neighbourhoods. The unary potentials are uniformly sampled in the range [0, 10]. We consider (a) Potts model and (b) hierarchical Potts models with pairwise distance between labels given by the trees of Figure 6. The pairwise weights are varied in the range $\{1, 2, 5, 10\}$. We compare the results of our worst-case optimal submodular



(a) Tree A (b) Tree B

Fig. 6: **Trees for synthetic experiments:** *The hierarchical Potts models defining pairwise distance among 20 labels used for upper-bound comparison with TRW. Blue nodes are the meta-labels and yellow nodes are labels. All the edges at a particular level have the same edge weights. The sequence of weights from root level to leaf level is 1, 0.5 for tree A and 1, 1, 0.5 for tree B. The yellow node is shown to clump together 4 and 5 leaf nodes for tree A and B respectively.*

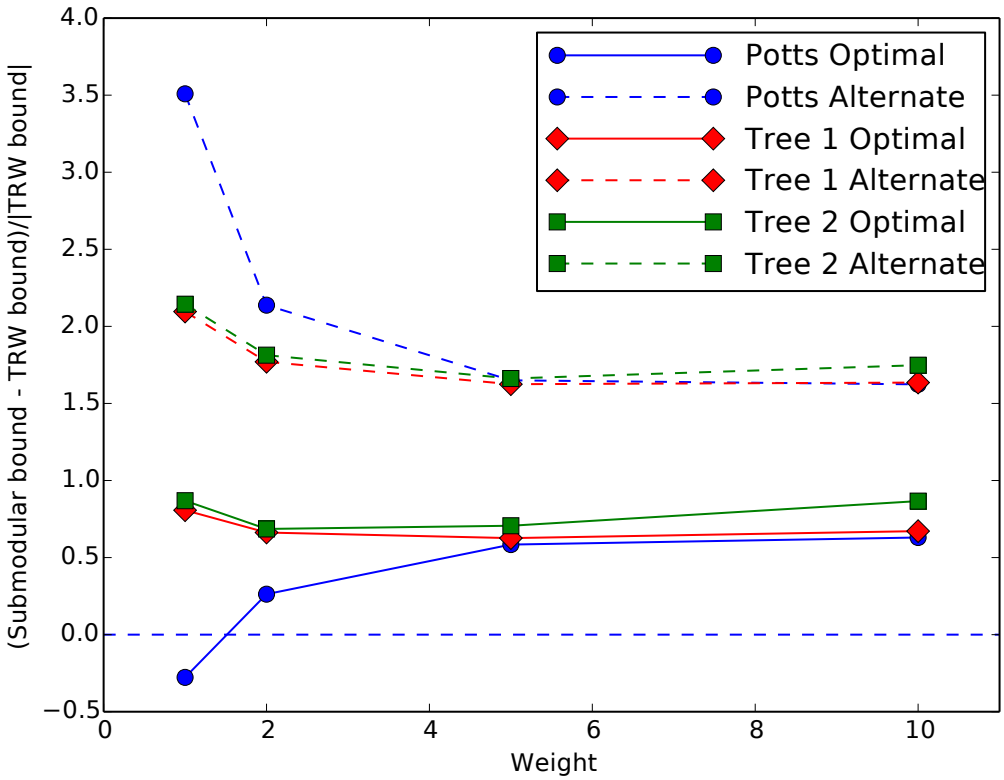


Fig. 7: **Upper-bound comparison using synthetic data:** *The plot shows the ratio (Submod bound - TRW bound)/|TRW bound| averaged over 100 unary instances as a function of pairwise weights using the worst-case optimal and alternate extension for Potts and hierarchical Potts models. We observe that the worst-case optimal extension (solid) results in tighter bounds as compared to the respective alternate extensions (dotted). Also, the worst-case optimal extension bounds are in similar range as the TRW bounds.*

extension with an alternate submodular extension as given in Figure 5.

Method For our algorithm, we use the standard schedule $\gamma = 2/(k+2)$ to obtain step size γ at iteration k. We run our algorithm till convergence - 100 iterations suffices for this. The experiments are repeated for 100 randomly generated unaries for each model and each weight. For TRW, we used the MATLAB toolbox of Domke (2013). The baseline code does not optimise over tree distributions. We varied the edge-appearance probability in trees over the range [0.1 - 0.5] and found 0.5 to give tightest upper bound.

Results We plot the ratio of the normalised difference of the upper bound values of our method with TRW as a function of pairwise weights. The ratios are averaged over 100 instances of unaries. Figure 7 shows the plots for Potts and hierarchical Potts models for the worst-case optimal and alternate extension. We find that the optimal extension (solid) results in tighter upper-bounds than the alternate extension (dotted) for both models. This is because the representation of the submodular function using Figure 5 requires that $\phi_a(i)$ be non-negative. This implies that $F(A)$ values are larger for the worst-case optimal extension of Figure 3 as compared to the alternate extension. Hence the minimisation problem 9 has larger domain $EP(F)$ for the optimal extension, thereby resulting in better minima. Figure 7 indicates that our algorithm does not provide as tight upper-bounds as TRW, however they are of similar magnitude. TRW makes use of the standard LP relaxation (Chekuri et al., 2004), from the second-level of Sherali-Adams hierarchy (having $\mathcal{O}(N^2)$ number of relaxed variables), and is tighter than Kleinberg-Tardos relaxation, resulting in better approximation. However, TRW does not scale well with neighborhood size, thereby prohibiting its use in dense CRFs.

7.2. Stereo Matching using Dense CRFs

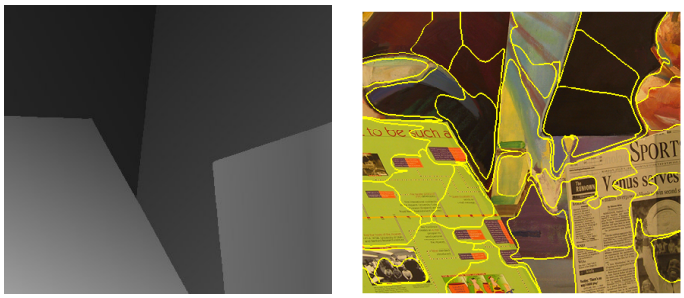
Data We demonstrate the benefit our algorithm for stereo matching on images extracted from the Middlebury stereo matching dataset (Scharstein et al., 2001). We use dense CRF models with Potts compatibility term and Gaussian pairwise potentials. The unary terms are obtained using the absolute difference matching function of Scharstein et al. (2001).

Method We use the implementation of mean-field algorithm for dense CRFs of Koltun and Krahenbuhl (2011) as our baseline. For our algorithm, we make use of the modified Gaussian filtering implementation for dense CRFs by Ajanthan et al. (2017) to compute the conditional gradient at each step. The step size γ at each iteration is selected by doing line search in $[0, 1]$ (we try step sizes at 0.1 interval and pick the one that decreases the objective most). We run our algorithm till 100 iterations, since the visual quality of the solution does not show much improvement beyond this point. We run mean-field up to convergence, with a threshold of 0.001 for change in KL-divergence.

Results Figure 9 shows some example solutions obtained by picking the label with maximum marginal probability for each variable for mean-field and for our algorithm. We also report the time and energy values of the solution for both methods. Though we are not performing MAP estimation, energy values give us a quantitative indication of the quality of solutions. For the full set of 21 image pairs (2006 dataset), the average ratio of the energies of the solutions from our method compared to mean-field is 0.943. The average time ratio is 10.66. We observe that our algorithm results in more natural looking stereo matching results with lower energy values for all images. However, mean-field runs faster than our method for each instance.The set of hyperparameters that we used can be found in the appendix.

7.3. Stereo Matching using Higher-order Diversity Model

Data Next, we use the higher-order diversity model for stereo matching on the Middlebury dataset. A higher-order model is suitable to be used for only some images in the dataset, since others have gradually sloping surfaces.



(a) Ground truth disparity (b) Superpixels

Fig. 8: *An example for stereo matching that is unsuitable to be used with higher-order model. Figure (a) shows the ground truth disparity for this example. The image has gradually sloping surfaces, and hence gradually varying disparity values. Figure (b) shows the superpixels for this example. Using these superpixels as higher-order cliques will force pixels on gradually sloping surfaces to take similar disparity values.*

Higher-order cliques will force all pixels on gradually sloping surfaces to take similar disparity values and we want to avoid this (Figure 8). We take a dense CRF model with Potts compatibility term and Gaussian pairwise potentials, and augment it with higher-order diversity term. We obtain higher-order cliques by using super-pixels obtained from the mean-shift algorithm (Comaniciu and Meer, 2002).

Method We again use the modified Gaussian filtering implementation for dense CRFs by Ajanthan et al. (2017) to compute the conditional gradient at each step. To compute the contribution of the higher-order model, we make use of an efficient strategy to reuse computation. Recall that computing the conditional gradient coordinates $s^*_{\sigma(i)}$ requires us to obtain differences of the nested sets $F(S_i) - F(S_{i-1})$. We store the clique state at each i, and this enables us to compute the difference for each coordinate in a constant amount of time. In this case, computation of conditional gradient takes $\mathcal{O}(cNL)$ where c is the number of higher-order cliques. The step size γ at each iteration is selected by doing line search in $[0, 1]$ (we try step sizes at 0.1 interval and pick the one that decreases the objective most).We ran our algorithm for 200 iterations.

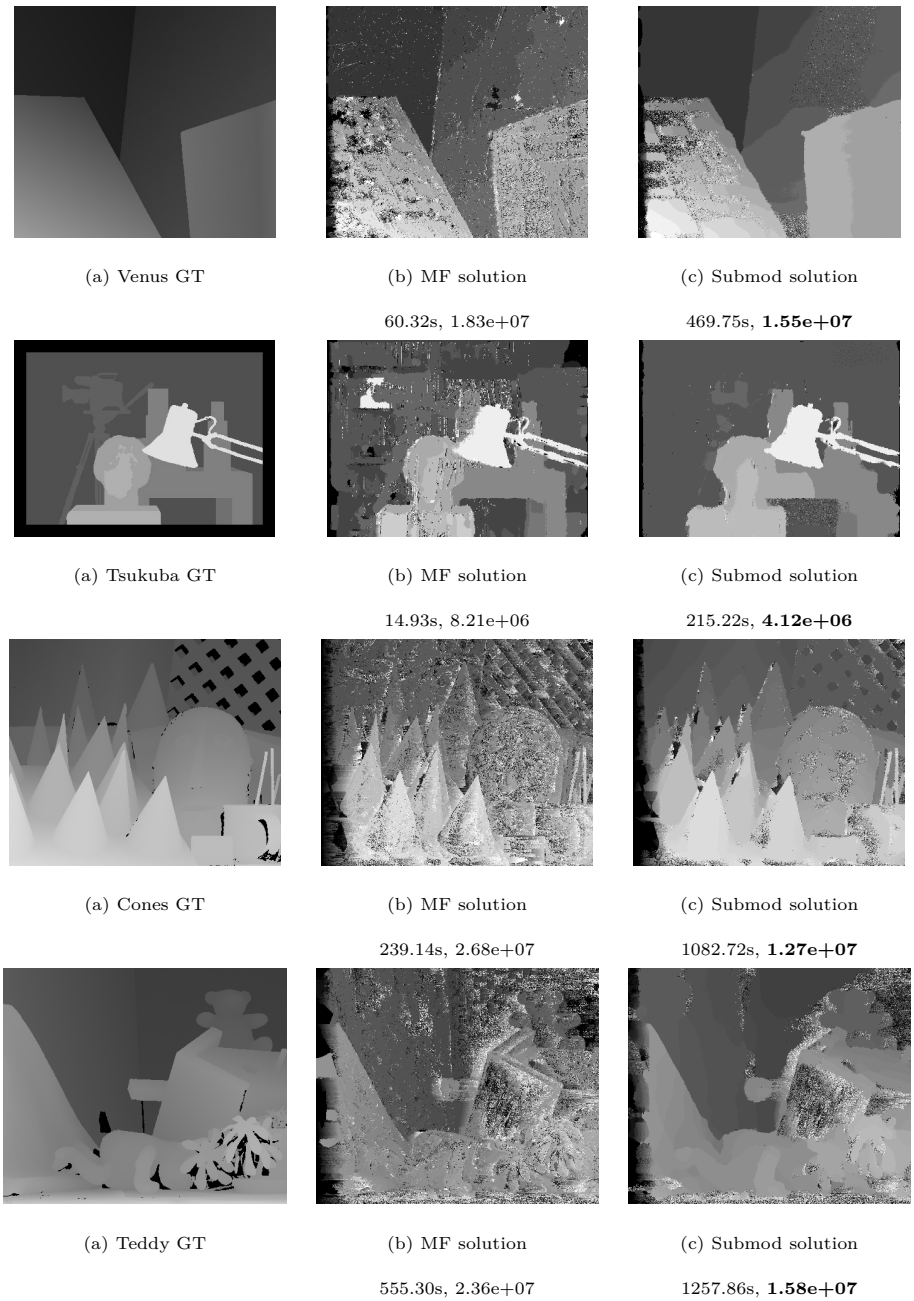
Results Figure 10 shows some example solutions obtained by picking the label with maximum marginal probability for each variable. Our method gives results that are reasonably close to the ground-truth. The set of hyperparameters that we used can be found in the appendix.

7.4. Semantic Segmentation using Higher-order Diversity Model

Data We evaluate our approach on the task of semantic segmentation on the MSRC-21 dataset (Shotton et al., 2009). It consists of 591 color images of size 320×213 with corresponding ground truth labelings of 21 object classes. We made use of the unary features from the *TextonBoost* classifier (Shotton et al., 2009). As for stereo matching (subsection 7.3), we augment the dense CRF with Gaussian pairwise potentials with our higher-order diversity model. Superpixels for higher-order modeling were obtained from the mean-shift algorithm (Comaniciu and Meer, 2002).

Method As for stereo matching, we used the modified Gaussian filtering implementation (Ajanthan et al., 2017) for the contribution of the dense pairwise terms to the conditional gradient. We used the same strategy as in subsection 7.3 to compute the higher-order component of the conditional gradient in $\mathcal{O}(cNL)$ time-complexity. We ran our algorithm for 100 iterations.

Results The ground-truth labelings provided with the MSRC-21 dataset are coarse. We evaluate our algorithm on the set of 94 images for which fine-grain annotations are available (Koltun and Krahenbuhl, 2011). On this subset, our method correctly labeled 81.18% of pixels. It took 275.02s on average per instance to run our method for 100 iterations. Some representative results are shown



(a) Venus GT (b) MF solution 60.32s, 1.83e+07 (c) Submod solution 469.75s, **1.55e+07**

(a) Tsukuba GT (b) MF solution 14.93s, 8.21e+06 (c) Submod solution 215.22s, **4.12e+06**

(a) Cones GT (b) MF solution 239.14s, 2.68e+07 (c) Submod solution 1082.72s, **1.27e+07**

(a) Teddy GT (b) MF solution 555.30s, 2.36e+07 (c) Submod solution 1257.86s, **1.58e+07**

Fig. 9: *Stereo matching using dense CRFs with Potts compatibility and Gaussian pairwise potentials. We compare our solution with the mean-field algorithm of Koltun and Krahenbuhl (2011). We observe that our method gives better-looking solutions with lower energy value at the cost of higher computational time.*

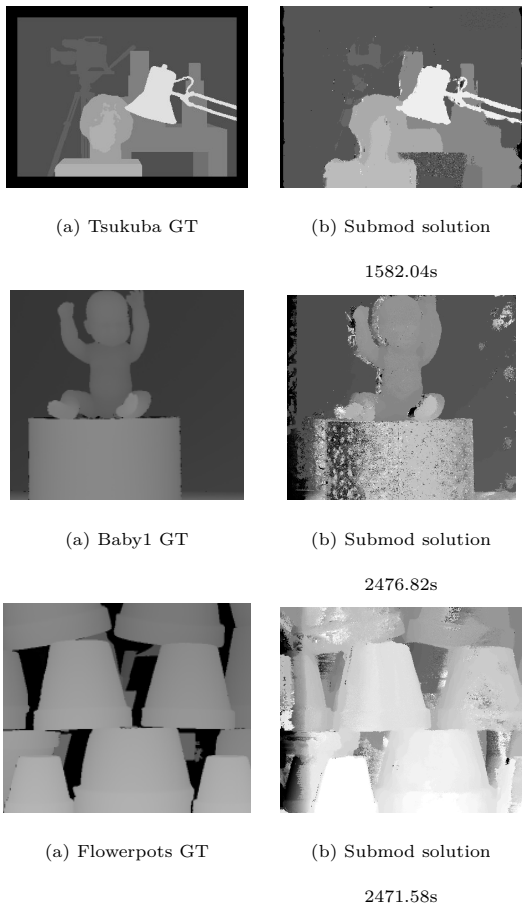
in Figure 11 where we label each pixel with the label having the maximum marginal probability. Our set of hyperparameters can be found in the appendix.

8. Discussion

We have established the relation between submodular extension and the LP relaxation for MAP estimation using Lovasz extension for various CRF models. This allowed us to identify the worst-case optimal submodular extension for Potts as well as the general metric labeling problems. In addition, we obtained an accurate submodular extension for a higher-order model based on label-diversity in cliques. It is worth noting that it might still be possible to obtain an improved submodular extension for a given problem instance. The design of a computationally feasible algorithm for this task is an interesting direction of future research. While our work focused on one class of higher-order model, there is potential for our approach to be used to identify accurate submodular extensions for other energy families, such as truncated max-of-convex models (Pansari and Kumar, 2017).

Acknowledgement

This work was supported by the EPSRC grantsEP/P020658/1, TU/B/000048, and EP/P022529/1 and Google Deepmind PhD studentship.

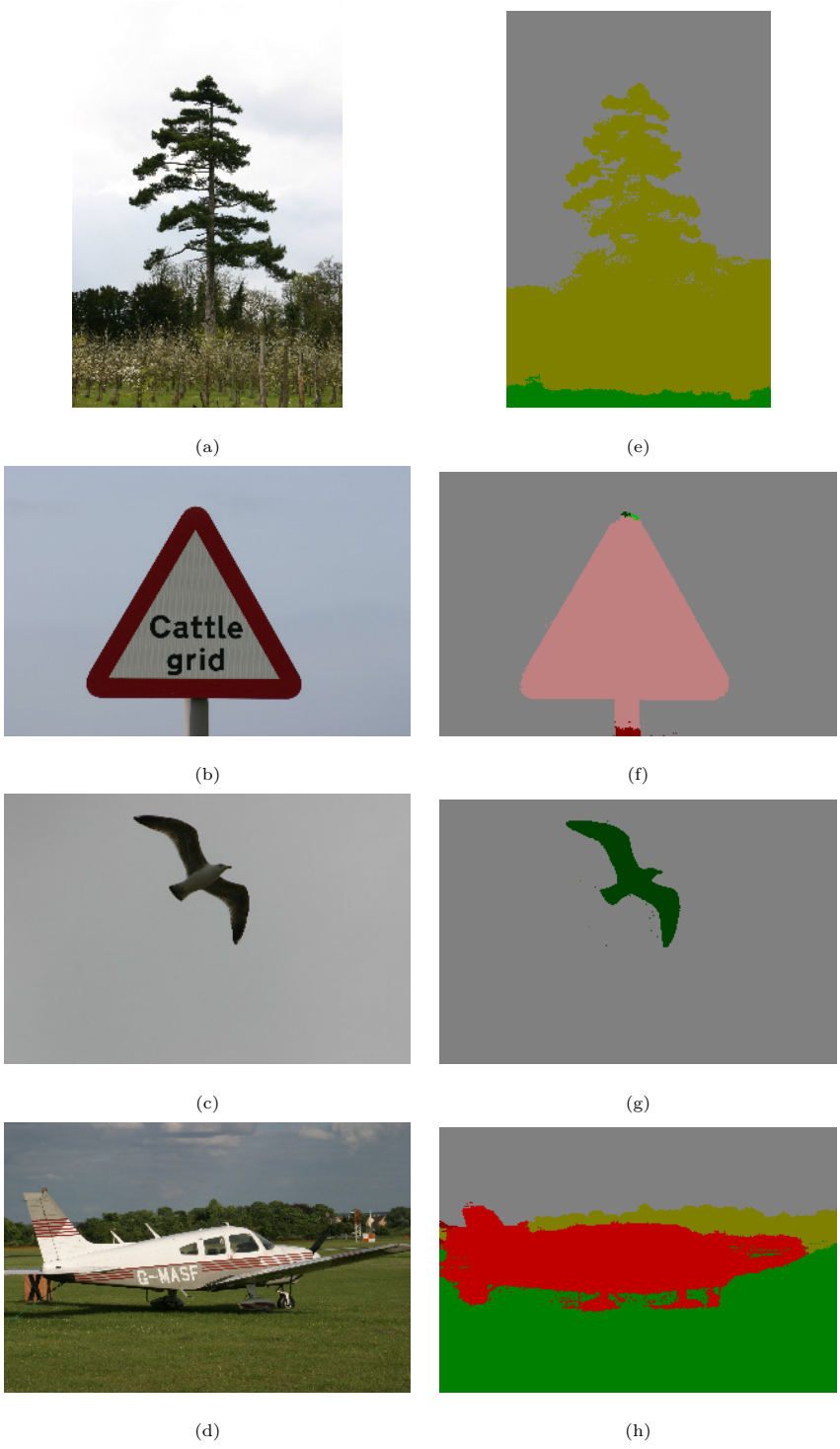


(a) Tsukuba GT (b) Submod solution 1582.04s

(a) Baby1 GT (b) Submod solution 2476.82s

(a) Flowerpots GT (b) Submod solution 2471.58s

Fig. 10: *Stereo matching using dense CRFs with higher-order diversity term in addition to Potts compatibility and Gaussian pairwise potentials.*



(a) (e) (b) (f) (c) (g) (d) (h)

Fig. 11: *Sample results from semantic segmentation on the MSRC-21 dataset using dense CRFs with higher-order diversity term in addition to Potts compatibility and Gaussian pairwise potentials. (a)-(d) show the input image, and (e)-(f) show the corresponding results using our method. We observe that our method is able to segment out objects with complex boundaries fairly accurately.*

References

Ajanthan, T., Desmaison, A., Bunel, R., Salzmann, M., Torr, P., Kumar, M., 2017. Efficient linear programming for dense crfs, in: CVPR.

Bach, F., 2013. Learning with submodular functions: A convex optimization perspective. Foundations and Trends in Machine Learning .

Bartal, Y., 1996. Probabilistic approximation of metric spaces and its algorithmic applications, in: Foundations of Computer Science.

Bartal, Y., 1998. On approximating arbitrary metrices by tree metrics, in: ACM Symposium on Theory of Computing.

Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. PAMI .

Chekuri, C., Khanna, S., Naor, J., Zosin, L., 2004. A linear programming formulation and approximation algorithms for the metric labeling problem. SIAM Journal on Discrete Mathematics .

Comaniciu, D., Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. PAMI .

Djolonga, J., Krause, A., 2014. From map to marginals: Variational inference in bayesian submodular models, in: NIPS.

Domke, J., 2013. Learning graphical model parameters with approximate marginal inference. PAMI .

Edmonds, J., 1970. Submodular functions, matroids, and certain polyhedra. Combinatorial Optimization — Eureka, You Shrink! .

Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. Naval research logistics quarterly .

Ishikawa, H., 2003. Exact optimization for markov random fields with convex priors. IEEE Trans. Pattern Anal. Mach. Intell. 25, 1333–1336.

Kleinberg, J., Tardos, E., 2002. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. IEEE Symposium on the Foundations of Computer Science .

Kohli, P., Kumar, M.P., Torr, P.H., 2007. P3 & beyond: Solving energies with higher order cliques, in: CVPR.

Koltun, V., Krahenbuhl, P., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. NIPS .

Kumar, M., Veksler, O., Torr, P., 2011. Improved moves for truncated convex models. JMLR .

MacKay, D.J.C., 2003. Information theory, inference, and learning algorithms. Cambridge University Press.

Manokaran, R., Naor, J., Raghavendra, P., Schwartz, R., 2008. Sdp gaps and ugc hardness for multiway cut, 0-extension, and metric labeling, in: ACM Symposium on Theory of Computing.

Pansari, P., Kumar, M.P., 2017. Truncated max-of-convex models., in: CVPR.

Scharstein, D., Szeliski, R., Zabih, R., 2001. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, in: Stereo and Multi-Baseline Vision.

Sherali, H.D., Adams, W.P., 1990. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM J. Discrete Math. 3, 411–430.

Shotton, J., Winn, J., Rother, C., Criminisi, A., 2009. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. International Journal of Computer Vision .

Vineet, V., Warrell, J., Torr, P., 2014. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. IJCV .

Wainwright, M., Jaakkola, T., Willsky, A., 2005. A new class of upper bounds on the log partition function. IEEE Transactions on Information Theory .

Wang, C., Komodakis, N., Paragios, N., 2013. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. Computer Vision and Image Understanding 117, 1610–1627.

Zhang, J., Djolonga, J., Krause, A., 2015. Higher-order inference for multi-class log-supermodular models, in: ICCV.

Appendix: Worst-case Optimal Submodular Extensions for Marginal Estimation

1 Proofs for Potts Model Extension

Remark 1 We show using induction over the number of variables that with 1-of-L encoding for Potts,

$$\sum_{A\in\mathcal{M}} \exp(-s(A)) = \prod_{a=1}^{N}\sum_{i=1}^{L}\exp(-s_{ai}) \tag{1}$$

Proof. Let t be the number of variables, V^t be the corresponding ground set and $\mathcal{M}^t$ be the sets corresponding to valid labelings. Equation (1) clearly holds for $t = 1$.

Let us assume that the relation holds for $t = N$, that is,

$$\sum_{A^N\in\mathcal{M}^N} \exp(-s(A^N)) = \prod_{a=1}^{N}\sum_{i=1}^{L}\exp(-s_{ai}) \tag{2}$$

For $t = N + 1$,

$$\begin{aligned}
\sum_{A^{N+1}\in\mathcal{M}^{N+1}} \exp(-s(A^{N+1})) &= \sum_{i=1}^{L}\sum_{A^N\in\mathcal{M}^N} \exp(-s(A^N) - s_{N+1,i}) \\
&= \sum_{i=1}^{L}\exp(-s_{N+1,i})\sum_{A^N\in\mathcal{M}^N} \exp(-s(A^N)) \\
&= \sum_{i=1}^{L}\exp(-s_{N+1,i})\prod_{a=1}^{N}\sum_{j=1}^{L}\exp(-s_{aj}) \\
&= \prod_{a=1}^{N+1}\sum_{i=1}^{L}\exp(-s_{ai})
\end{aligned} \tag{3}$$

□

Lemma 1. *Given any submodular extension $F(.)$ of a Potts energy function $E(.)$, its Lovasz extension $f(.)$ defines an LP relaxation of the MAP problem for $E(.)$ as*

$$\min_{\mathbf{y}\in\Delta} f(\mathbf{y}). \tag{4}$$

Proof. By definition of a submodular extension and the Lovasz extension, $E(\mathbf{x}) = F(A_{\mathbf{x}}) = f(1_{A_{\mathbf{x}}})$ for all valid labelings $\mathbf{x}$. Also, from property 1, $f(\mathbf{y})$ is maximum of linear functions. Hence, $f(\mathbf{y})$ is a piecewise linear relaxation of $E(\mathbf{x})$.

The domain Δ is a polytope formed by union of N probability simplices

$$\Delta = \{\mathbf{y}_a \in \mathbb{R}^L | \mathbf{y}_a \succeq 0 \text{ and } \langle \mathbf{1}, \mathbf{y}_a \rangle = 1\} \tag{5}$$

With objective as maximum of linear functions and domain as a polytope, we have an LP relaxation of the corresponding MAP problem. □

2 Proofs for Hierarchical Potts Model Extension

Lemma 2. *Given any submodular extension $F(.)$ of an hierarchical Potts energy function $E(.)$, its Lovasz extension $f(.)$ defines an LP relaxation of the corresponding MAP estimation problem as*

$$\min_{\mathbf{z} \in \Delta'} f(\mathbf{z}) \tag{6}$$

Proof. By definition of a submodular extension and the Lovasz extension, $E(\mathbf{x}) = F(A_{\mathbf{x}}) = f(1_{A_{\mathbf{x}}})$ for all valid labelings $\mathbf{x}$. Also, from property 1, $f(\mathbf{y})$ is maximum of linear functions. Hence, $f(\mathbf{y})$ is a piecewise linear relaxation of $E(\mathbf{x})$.

We can write the domain Δ' as

$$\Delta' = \{\mathbf{y}_a \in \mathbb{R}^M | \mathbf{y}_a \succeq 0,\ \langle \mathbf{1}, \mathbf{y}_a^{label} \rangle = 1,\ \mathbf{y}_a(p_{ai}) = \mathbf{1} \text{ or } \mathbf{y}_a(p_{ai}) = \mathbf{0} \forall i \in [1, L]\} \tag{7}$$

where $\mathbf{y}_a$ is the component of $\mathbf{y}$ corresponding to the a-th variable, $\mathbf{y}_a^{label}$ is the component of $\mathbf{y}_a$ corresponding to the L labels, and $\mathbf{y}_a(p_{ai})$ is the component of $\mathbf{y}_a$ corresponding to the elements of p_{ai}.

Since Δ' is defined by linear equalities and inequalities, it is a polytope. With objective as maximum of linear functions and domain as a polytope, we have an LP relaxation of the corresponding MAP problem. □

Proposition 1. *In the limit $T \to 0$, the following problem for hierarchical Potts energies*

$$\min_{\mathbf{s} \in EP(F)} \quad g_T(\mathbf{s}) = \sum_{a=1}^{N} T \cdot \log \sum_{i=1}^{L} \exp(-\frac{s'_{ai}}{T}). \tag{8}$$

becomes:

$$-\min_{\mathbf{z} \in \Delta'} f(\mathbf{z}) \tag{9}$$

Proof. In the limit of $T \to 0$, we can rewrite the above problem as

$$\min_{\mathbf{s} \in EP(F)} \quad \sum_{a=1}^{N} \max_i (-s'_{ai}). \tag{10}$$

In vector form, the problem becomes

$$\min_{\mathbf{s}\in EP(F)} \max_{\mathbf{z}\in\Delta} -\langle \mathbf{z}, \mathbf{s}' \rangle \tag{11}$$

$$= - \max_{\mathbf{s}\in EP(F)} \min_{\mathbf{z}\in\Delta} \langle \mathbf{z}, \mathbf{s}' \rangle \tag{12}$$

$$\text{where } \Delta = \{\mathbf{z}_a \in \mathbb{R}^L | \mathbf{z}_a \succeq 0 \text{ and } \langle \mathbf{1}, \mathbf{z}_a \rangle = 1\} \tag{13}$$

where $\mathbf{z}_a$ is the component of $\mathbf{z}$ corresponding to the a-th variable. We can unpack $\mathbf{s}'$ using

$$s'_{ai} = \sum_{t\in p_{ai}} s_t. \tag{14}$$

and rewrite problem (12) as

$$- \max_{\mathbf{s}\in EP(F)} \min_{\mathbf{y}\in\Delta'} \langle \mathbf{y}, \mathbf{s} \rangle \tag{15}$$

The new constraint set Δ' ensures that the binary entries of labels and meta-labels is consistent:

$$\begin{aligned} \text{where } \Delta' = \{\mathbf{y}_a \in \mathbb{R}^M | \mathbf{y}_a \succeq 0, \;\; \langle \mathbf{1}, \mathbf{y}_a^{label} \rangle = 1, \\ \mathbf{y}_a(p_{ai}) = \mathbf{1} \text{ or } \mathbf{y}_a(p_{ai}) = \mathbf{0} \forall i \in [1, L]\} \end{aligned} \tag{16}$$

where $\mathbf{y}_a$ is the component of $\mathbf{y}$ corresponding to the a-th variable, $\mathbf{y}_a^{label}$ is the component of $\mathbf{y}_a$ corresponding to the L labels, and $\mathbf{y}_a(p_{ai})$ is the component of $\mathbf{y}_a$ corresponding to the elements of p_{ai}.

By the minimax theorem for LP, we can reorder the terms:

$$- \min_{\mathbf{y}\in\Delta'} \max_{\mathbf{s}\in EP(F)} \langle \mathbf{y}, \mathbf{s} \rangle \tag{17}$$

Recall that $\max_{\mathbf{s}\in EP(F)} \langle \mathbf{y}, \mathbf{s} \rangle$ is the value of the Lovasz extension of F at $\mathbf{y}$, that is, $f(\mathbf{y})$. Hence, as $T \to 0$, the marginal inference problem converts to minimising the Lovasz extension under the constraints Δ':

$$- \min_{\mathbf{y}\in\Delta'} f(\mathbf{y}) \tag{18}$$

□

Proposition 2. *The objective function $\widetilde{E}(\mathbf{z})$ of (T-LP-FULL) is the Lovasz extension of $F_{r-HST}(A) = \sum_{i=1}^{M} F_i(A)$, where*

$$\begin{aligned} F_i(A) = \sum_a \phi'_a(i)[|A \cap \{v_{ai}\}| = 1] + \\ \sum_{(a,b)\in\mathcal{N}} w_{ab} \cdot l_{T_i} \cdot [|A \cap \{v_{ai}, v_{bi}\}| = 1] \end{aligned} \tag{19}$$

Proof. We observe that $F_{r-\text{HST}}$ is of exactly the same form as F_{Potts}, except that the Ising models F_i are defined over not just labels, but meta-labels as well. Using the same logic as in the proof of Potts extension, each F_i is the Lovasz extension of

$$\widetilde{E}_i(\mathbf{z}) = \left(\sum_{a\in\mathcal{X}} \phi'_a(i) \cdot z_{ai} + \sum_{(a,b)\in\mathcal{N}} w_{ab} \cdot l_{T_i} \cdot |z_{ai} - z_{bi}| \right) \tag{20}$$

and the results follows. □

	$w^{(1)}$	θ_α	θ_β	$w^{(1)}$	θ_γ	w_c
Stereo matching (dense pairwise, Submod)	1.64	12.99	50.84	1.51	1.10	0
Stereo matching (dense pairwise, MF)	12.43	22.38	17.26	4.59	8.75	0
Stereo matching (higher-order, Submod)	12.43	22.38	17.26	4.59	8.75	10.00
Semantic Segmentation (higher-order, Submod)	1.00	1.00	50.84	1.51	1.10	10.00

Table 1: List of hyperparameters used in experiments

3 List of Hyperparameters

For pairwise dense CRFs, following Koltun and Krahenbuhl [2011], we use constrast-sensitive two-kernel potentials for both stereo matching and semantic segmentation. These are defined in terms of the color vectors I_a and I_b and positions p_a and p_b as:

$$k(\mathbf{f}_a, \mathbf{f}_b) = w^{(1)} \exp\left(-\frac{|p_a - p_b|^2}{2\theta_\alpha^2} - \frac{|I_a - I_b|^2}{2\theta_\beta^2}\right) + w^{(2)} \exp\left(-\frac{|p_a - p_b|^2}{2\theta_\gamma^2}\right) \tag{21}$$

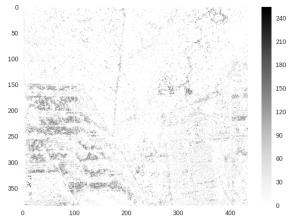
For the higher-order diversity model, we have a clique weight w_c that is the same across all cliques. We give the set of hyperparameters for our experiments in table 1.

4 Entropy Plots for Stereo Matching using Dense CRFs with Potts Model

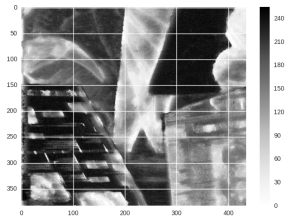
We plotted the entropy of the marginals obtained from mean-field Koltun and Krahenbuhl [2011] and our inference algorithm for the Potts models for all stereo matching instances. The entropy values are scaled to lie between 0-255 for visualisation. We observe that mean-field marginal inference results in almost 0/1 marginals, and there is not enough uncertainty information. Our method on the other hand yields more meaningful marginals to indicate uncertainity in prediction. For instance, in dark regions with not enough detail, the entropy for our method is high. However, we also observe that in some cases our method yields marginals with high uncertainty even in homogeneous regions, such as object interior.



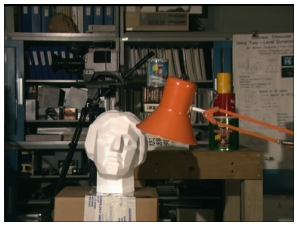
(a) Venus left image



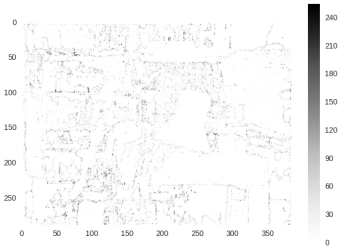
(b) MF entropy



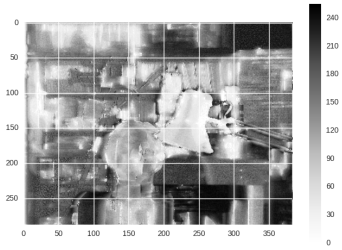
(c) Submod entropy



(a) Tsukuba left image



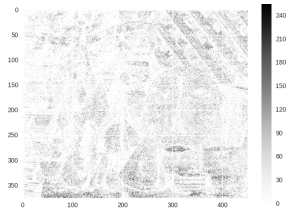
(b) MF entropy



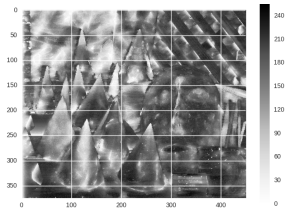
(c) Submod entropy



(a) Cones left image



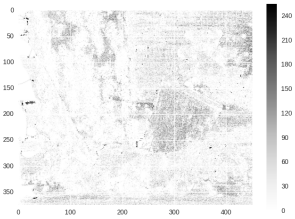
(b) MF entropy



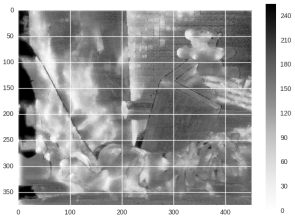
(c) Submod entropy



(a) Teddy left image



(b) MF entropy



(c) Submod entropy

Figure 1: *Entropy (scaled to lie in 0-255) plots for stereo matching using dense CRFs with Potts compatibility and Gaussian pairwise potentials for the mean-field algorithm of Koltun and Krahenbuhl [2011] and our method.*

References

Koltun, V. and Krahenbuhl, P. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*. 4, 5

CHAPTER 7

Conclusion

We have seen the central role played by submodularity in our design of MAP and variational inference algorithms for higher-order and dense CRFs. First, they enabled us to develop an efficient *st*-mincut-based MAP inference method for our novel higher-order model called TMCM, which is a generalisation of TCMs. Although, TMCMs are not graph-representable, and hence not submodular, our work relied crucially on the fact that max-of-convex models can be represented as graphs. This fact enabled us to design our inference algorithm based on the majorization-minimization paradigm. It is possible to use our approach to develop efficient MAP inference algorithms for other classes of higher-order CRFs. This line of work will be greatly facilitated by a better understanding of the class of higher-order energies that are graph-representable.

Second, submodularity had been elegantly applied in literature for variational inference in simple, pairwise CRFs. The approach was based on exploiting the ease of solving LPs over extended polytope of submodular functions. We brought rigor in this approach by providing a principled way to design submodular extensions of higher-order and dense CRF energy functions. This was possible thanks to known connections between submodularity and convexity via the

Lovasz extension of submodular functions. A knowledge of tightness of LP relaxations for MAP estimation enabled us to claim the worst-case optimality of submodular extensions in some cases.

There are two avenues for further work on the use of submodularity for the variational inference approach we have focussed on. First, one can study submodular extensions for more general higher-order models, such as P^n Potts and its robust version, as well as TMCM. We attempted to design extensions for these higher-order models, but our research was not very fruitful. In particular, we could not prove the submodularity of the extensions we came up with. This research will benefit from studies of tight LP relaxations for higher-order CRFs.

The second avenue for further research is formulating the optimisation problem for obtaining the best possible submodular extension for a given energy function, even from the classes of CRFs we have studied. The worst-case optimality guarantee that we have provided still leaves the door open for existence of better extensions. The first step in tackling this problem is likely to be deciding a good representation/parameterisation for submodular extension that is amenable to efficient optimisation.

We conclude by expressing our optimism that submodular functions still have a vital role to play in efficient inference in more general classes of CRFs, in the design and optimisation of binary/discrete neural networks, and in other hard discrete optimisation problems.

CHAPTER 8

Errata

We thank the examiners for pointing out errors in the first version of this thesis, as well as for providing valuable suggestions to improve the clarity of the material.

Chapter 4: Truncated Max-of-Convex Models:

1. In section 3 (convexity in discrete domain): $d : \mathbb{Z} \to \mathbb{R}^n$ is not a convex function in the strict sense. The domain of a convex function is a convex set, whereas $d(.)$ has as its domain the set of integers which is not convex. Hence, one needs to extend the notion of convexity to a discrete domain. Following [37], we define convexity on a discrete domain A as:

   Definition 15. *A real-valued function $d(.)$ on a set A of integers is convex provided that*

   $$d(ax + (1-a)y) \leq ad(x) + (1-a)d(y) \tag{8.1}$$

   holds for any $x, y \in A$ and $0 \leq a \leq 1$ such that $ax + (1-a)y \in A$.

   When A is a set of consecutive integers, it is necessary and sufficient for

convexity that all second-order differences are nonnegative:

$$d(x+1) - 2d(x) + d(x-1) \geq 0 \tag{8.2}$$

2. In section 4:

   (a) Graph definition: Using the same notation as in section 3, the graph is defined by the set of vertices $\mathbf{X} = \{X_a \in \mathcal{V}\}$ where $\mathcal{V} = \{1, 2, \ldots, n\}$ and the set of undirected edges $\mathcal{E}$ where $(a, b) \in \mathcal{E}$ if X_a and X_b are connected. In TCM, $\mathcal{E}$ is such that the maximum size of a clique - that is, a set of vertices such that every vertex in it has an edge to every other vertex in the set - is only 2. However, TMCM permits cliques of cardinality 3 or more, and we call these *higher-order* cliques.

   (b) Energy of TMCM: Let $\mathcal{C}$ be the set of all cliques. The set of vertices in a given clique $\mathbf{c} \in \mathcal{C}$ is denoted by $\mathbf{X_c} = \{X_a, a \in \mathbf{c} \subseteq \mathcal{V}, |\mathbf{c}| \geq 3\}$, and $\mathbf{x_c}$ is an assignment to $\mathbf{X_c}$. The energy function of a TMCM is then:

$$E(\mathbf{x}) = \sum_{a \in \mathcal{V}} \theta_a(x_a) + \sum_{(a,b) \in \mathcal{E}} \theta_{ab}(x_a, x_b) + \sum_{\mathbf{c} \in \mathcal{C}} \theta_{\mathbf{c}}(\mathbf{x_c}) \tag{8.3}$$

   where

$$\theta_a(.) : \text{arbitrary unary potentials} \tag{8.4a}$$

$$\theta_{ab}(x_a, x_b) = \sum_{(a,b) \in \mathcal{E}} \theta_{ab}(x_a, x_b) \tag{8.4b}$$

$$\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \sum_{i=1}^{m} \min\{d(p_i(\mathbf{x_c}) - p_{c-i+1}(\mathbf{x_c})), M\} \tag{8.4c}$$

   (c) In describing the higher-order MRF as above, we followed the conven-

tion used in MRF literature in computer vison. Alternatively we can use the concept of hypergraphs for our description. A hypergraph H is a pair $H = (\mathbf{X}, \mathcal{E}_H)$ where $\mathbf{X}$ is the set of vertices, and $\mathcal{E}_H$ is a set of non-empty subsets of $\mathbf{X}$ called hyperedges. We assume that $\mathcal{E}_H$ has vertex subsets of cardinality 2 or more. For TCM, $\mathcal{E}_H = \mathcal{E}$, where $\mathcal{E}$ is as defined above in part 2(a). For TMCM, $\mathcal{E}_H = \mathcal{E} \cup \mathcal{C}$, for $\mathcal{C}$ as in part 2(b). Using the unary potentials $\theta_a(x_a)$ and $\theta_{\mathbf{c}}(\mathbf{x_c})$ as in equation (8.4c), the energy for both TCM and TMCM can now be written as:

$$E(\mathbf{x}) = \sum_{a \in \mathcal{V}} \theta_a(x_a) + \sum_{\mathbf{c} \in \mathcal{E}_H} \theta_{\mathbf{c}}(\mathbf{x_c}) \tag{8.5}$$

3. Section 4: It is better to represent the labelings $\mathbf{x_c}$ as tuples instead of sets because elements can be repeated, and there is an ordering. Hence, it is preferable to use $\mathbf{x_c} = (3, 2, 1, 4, 1, 3)$ instead of $\mathbf{x_c} = \{3, 2, 1, 4, 1, 3\}$.

4. In the proof of proposition 1 in the appendix, we make use of the following inequality as the condition for pairwise potential $f(.) = \delta_{a,b}(.)$ to be submodular:

$$f(l, l') + f(l+1, l'+1) \leq f(l+1, l') + f(l, l'+1). \tag{8.6}$$

The above condition is equivalent to

$$f(l_1, l'_1) + f(l_2, l'_2) \leq f(l_1, l'_2) + f(l_2, l'_1). \tag{8.7}$$

for each four-tuple such that $l_2 \geq l_1, l'_2 \geq l'_1$ (the equivalence is shown in

subsubsection 2.5.2 of chapter 2). Inequality (8.7) can be rewritten as:

$$f(\min(l_1, l_2), \min(l'_1, l'_2)) + f(\max(l_1, l_2), \max(l'_1, l'_2)) \leq f(l_1, l'_2) + f(l_2, l'_1). \tag{8.8}$$

We now look upon $f(.)$ as a k-ary function with each parameter (representing a pixel of the k-clique) taking a value from the label set $\mathbf{L}$. Let $\mathbf{x}, \mathbf{y}$ be k-tuple of assignments. Let $\vee$ and $\wedge$ be max and min operators respectively, and let $(\mathbf{x} \vee \mathbf{y})$, $(\mathbf{x} \wedge \mathbf{y})$ denote k-tuples resulting from element-wise application of the max and min operations over the k-tuples $\mathbf{x}$ and $\mathbf{y}$. We can now rewrite submodularity condition (8.8) in vector form as

$$f(\mathbf{x} \wedge \mathbf{y}) + f(\mathbf{x} \vee \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}). \tag{8.9}$$

If we take the pairwise case (set $k = 2$), and use $\mathbf{x} = [l_1, l'_2]$ and $\mathbf{y} = [l_2, l'_1]$ in inequality (8.9), we recover inequality (8.8).

5. We proved that the function

$$\theta'_{\mathbf{c}}(\mathbf{y_c}) = \omega_{\mathbf{c}} \sum_{i=1}^{m} \delta_{q_i, q_{c-i+1}}(p_i, p_{c-i+1}) \tag{8.10}$$

is submodular by representing it as a graph-cut function in section 6.2. We are not aware of an easier argument to prove the submodularity of this function. We have proved in Proposition 1 that the functions $\delta_{a,b}(y_a, y_b)$ are submodular for all pairs (a, b). The sum of submodular functions (with positive coefficients) is submodular. The difficulty with our function (8.10) is that the choice of the component submodular functions that form the sum depends on $\mathbf{y_c}$.

6. Subsection 5.2: Lemma 1 can be alternatively stated as:

Lemma 1. *Graph (a) of figure 3 exactly models TMCM with no truncation, that is, when $M \to \infty$:*

$$E(\mathbf{x}) = \sum_{a \in \mathcal{V}} \theta_a(x_a) + \sum_{\mathbf{c} \in \mathcal{C}} \sum_{i=1}^{m} d(p_i(\mathbf{x_c}) - p_{c-i+1}(\mathbf{x_c})) \tag{8.11}$$

7. In the original version of the thesis, the lemmas and propositions of chapter 4 were mis-referenced in the associated appendix. To rectify this as well as to provide more detailed proofs and results, we have now replaced the manuscript of the CVPR 2017 paper with an extended version of the same paper which was uploaded on arXiv.org.

8. Section 6: We are able to compare the final energy values of our truncated max-of-linear model with $m = 1$, parsimonious labeling model of [23], and label co-occurence-based model of [65] because the three models are the same due to our choice of diversity and co-occurence functions. We proceed to show this relation more clearly.

   - Let $\Gamma(\mathbf{x_c})$ be the set of unique labels in the labeling of the clique $\mathbf{c}$ and $\delta(.)$ be a valid *diversity function* satisfying the conditions in [23]. The clique potentials of the parsimony-based model is given by

   $$\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \delta(\Gamma(\mathbf{x_c}). \tag{8.12}$$

   Given a metric function $d(.)$, a special type of diversity called *diameter diversity* is defined as

   $$\delta^{dia}(\Gamma) = \max_{l_i, l_j \in \Gamma} d(l_i, l_j). \tag{8.13}$$

Taking $d(.)$ as truncated linear metric

$$d(l_i, l_j) = \min(|l_i - l_j|, M) \tag{8.14}$$

in equation (8.13) gives us the truncated max-of-linear model with $m = 1$.

- Given a co-occurrence cost $C(.)$, the energy function used in [65] is

$$E(\mathbf{x}) = \sum_{a \in \mathcal{V}} \theta_a(x_a) + \sum_{\mathbf{c} \in \mathcal{E}_H} \theta_{\mathbf{c}}(\mathbf{x_c}) + C(\Gamma(\mathbf{x})). \tag{8.15}$$

The only requirement on $C(.)$ is that it must be monotonically increasing with the size of $\Gamma(\mathbf{x})$. In the model (8.15) above, the co-occurence potential is a global one. For comparison with TMCM, we instead use clique-wise co-occurence potentials. By taking

$$\theta_{\mathbf{c}}(\mathbf{x_c}) = C_{\mathbf{c}}(\Gamma(\mathbf{x_c})) = \theta_{\mathbf{c}} \max_{l_i, l_j \in \mathbf{x_c}} \min(|l_i - l_j|, M) \tag{8.16}$$

we obtain truncated max-of-linear model with $m = 1$.

Chapter 6: LP-based Submodular Extensions for Marginal Estimation

1. Equation (10) of the main paper should read as

$$\nexists \mathcal{F} \neq \mathcal{F}_{opt} : \min_{\mathbf{s} \in EP(\mathcal{F}(E))} g(\mathbf{s}) \leq \min_{\mathbf{s} \in EP(\mathcal{F}_{opt}(E))} g(\mathbf{s}) \ \ \forall \ E(.) \in \mathcal{E}. \tag{8.17}$$

2. Section 5.1: In equation (12) of the main paper, the set Δ is the cartesian product of N probability simplices Δ_a:

$$\Delta_a = \{\mathbf{y}_a \in \mathbb{R}^L | \mathbf{y}_a \geq 0 \text{ and } \langle \mathbf{1}, \mathbf{y}_a \rangle = 1\}, \tag{8.18}$$

where $\mathbf{y}$ is the vector of all variables and $\mathbf{y}_a$ is the component of $\mathbf{y}$ corresponding to X_a.

3. The problem

$$-\min_{\mathbf{y}\in\Delta} f(\mathbf{y})$$

can be rewritten as

$$-\min_{\mathbf{y}\in\Delta}\max_{\mathbf{s}\in EP(F)}\langle\mathbf{y},\mathbf{s}\rangle.$$

The problem

$$\min_{\mathbf{s}\in EP(F)} \quad \sum_{a=1}^{N}\max_{i}(-s_{ai}).$$

can be rewritten as

$$-\max_{\mathbf{s}\in EP(F)}\min_{\mathbf{y}\in\Delta}\langle\mathbf{y},\mathbf{s}\rangle.$$

The above two problems have compact, convex domains and linear objective functions. Hence, they satisfy the conditions for the minimax theorem of LPs and are dual of each other.

4. The proofs of propositions 1 and 2 are given in the main paper. The proofs of propositions 3 and 4 are provided in the appendix. However, propositions 3 and 4 have been mistakenly called propositions 1 and 2 respectively in the appendix. We apologise for the inconvenience caused to the reader due to this.

5. We claim that in the limit $T \to 0$,

$$\nexists \mathcal{F} \neq \mathcal{F}_{Potts} : \min_{\mathbf{s}\in EP(\mathcal{F}(E))} g_T(\mathbf{s}) \leq \min_{\mathbf{s}\in EP(\mathcal{F}_{Potts}(E))} g_T(\mathbf{s}) \ \ \forall\, E(.) \in \mathcal{E}. \tag{8.19}$$

Let us denote by $f(.)$ and $f_{Potts}(.)$ the Lovasz extensions of the submodular functions $F(.)$ and F_{Potts}. Using proposition 1, in the limit $T \to 0$, we can

rewrite the above claim as

$$\nexists \mathcal{F} \neq \mathcal{F}_{Potts} : -\min_{\mathbf{y}\in\Delta} f(\mathbf{y}) \leq -\min_{\mathbf{y}\in\Delta} f_{Potts}(\mathbf{y}) \ \forall\ E(.) \in \mathcal{E}.$$

$$\min_{\mathbf{y}\in\Delta} f(\mathbf{y}) \geq \min_{\mathbf{y}\in\Delta} f_{Potts}(\mathbf{y}) \ \forall\ E(.) \in \mathcal{E}.$$

By lemma 1,

$$\min_{\mathbf{y}\in\Delta} f(\mathbf{y}) \tag{8.20}$$

is an LP relaxation of the corresponding MAP problem. Also,

$$\min_{\mathbf{y}\in\Delta} f_{Potts}(\mathbf{y}), \tag{8.21}$$

designated by (P-LP) in our paper (refer equation (12) of the main paper), is the tightest possible LP relaxation (among the first-level LP relaxations) for the Potts model in the worst-case, the Unique Games Conjecture is true [68]. This implies that there does not exist a first-level LP relaxation that gives higher optimum values for the energy minimisation problem for all instances of the energy function. In other words,

$$\nexists \mathcal{F} \neq \mathcal{F}_{Potts} : \min_{\mathbf{y}\in\Delta} f(\mathbf{y}) \geq \min_{\mathbf{y}\in\Delta} f_{Potts}(\mathbf{y}) \ \forall\ E(.) \in \mathcal{E}.$$

which is our claim. Our worst-case optimality guarantee is derived in the limit case $T \to 0$, and it is in this limited sense that we make our claim.

6. Proposition 5: "the conditional gradient for the submodular function F_{Potts}" should read "the conditional gradient of problem (9) (main paper) with $F = F_{Potts}$".

7. Both mean-field and our proposed approach yield marginals. For qualitative comparison, we visualise the solutions by picking for each pixel the label

with maximum marginal probability. We also visualise the entropy values resulting from these marginals in the appendix. We compared the energy values even though neither our mean-field nor our method performs MAP estimation. This was done based on the premise that the energy values give a quantitative indication of the quality of the solutions. A comparison of the final segmentation with respect to the ground truth would have been a better criterion.

Bibliography

[1] Gridcut. `http://gridcut.com/`.

[2] T. Ajanthan, A. Desmaison, R. Bunel, M. Salzmann, P.H.S Torr, and M.P Kumar. Efficient linear programming for dense crfs. In *CVPR*, 2017.

[3] Asem M Ali, Aly A Farag, and Georgy L Gimelfarb. Optimizing binary mrfs with higher order cliques. In *ECCV*, 2008.

[4] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 2013.

[5] Francis R. Bach. Structured sparsity-inducing norms through submodular functions. In *NIPS*, 2010.

[6] Pierre Baqué, Timur Bagautdinov, François Fleuret, and Pascal Fua. Principled parallel mean-field inference for discrete random fields. In *CVPR*, 2016.

[7] Michael Bleyer, Carsten Rother, Pushmeet Kohli, Daniel Scharstein, and Sudipta Sinha. Object stereo–joint stereo matching and object segmentation. In *CVPR*, 2011.

[8] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[9] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[10] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[11] Anton Chechetka and Carlos Guestrin. Efficient principled learning of thin junction trees. In *NIPS*, 2007.

[12] Chandra Chekuri, Sanjeev Khanna, Joseph Seffi Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *ACM-SIAM Symposium on Discrete algorithms*, 2001.

[13] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[14] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[15] William H. Cunningham. On submodular function minimization. *Combinatorica*, 5(3):185–192, 1985.

[16] Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *ICML*, 2011.

[17] A. Delong, L. Gorelick, O. Veksler, and Y. Boykov. Minimizing energies with hierarchical costs. *IJCV*, 2012.

[18] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *CVPR*, 2010.

[19] A. Desmaison, R. Bunel, P. Kohli, P.H.S Torr, and M. P. Kumar. Efficient continuous relaxations for dense crf. In *ECCV*, 2016.

[20] J. Djolonga and A. Krause. From map to marginals: Variational inference in bayesian submodular models. In *NIPS*, 2014.

[21] Josip Djolonga, Stefanie Jegelka, Sebastian Tschiatschek, and Andreas Krause. Cooperative graphical models. In *NIPS*, 2016.

[22] Josip Djolonga and Andreas Krause. Scalable variational inference in log-supermodular models. In *ICML*, 2015.

[23] P. Dokania and M. P. Kumar. Parsimonious labeling. In *ICCV*, 2015.

[24] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[25] Pedro F Felzenszwalb and Julian J McAuley. Fast inference with min-sum matrix product. *PAMI*, 33(12):2549–2554, 2011.

[26] Alexander Fix, Aritanan Gruber, Endre Boros, and Ramin Zabih. A graph cut algorithm for higher-order markov random fields. In *ICCV*, 2011.

[27] B. Flach and D. Schlesinger. Transforming an arbitrary minsum problem into a binary one. Technical report, TU Dresden, 2006.

[28] Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR*, 2005.

[29] William T Freeman, Egon C Pasztor, and Owen T Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000.

[30] Satoru Fujishige and Sachin B. Patkar. Realization of set functions as cut functions of graphs and hypergraphs. *Discrete Mathematics*, 226(1-3):199–210, 2001.

[31] Andrew Gallagher, Dhruv Batra, and Devi Parikh. Inference for order reduction in markov random fields. In *CVPR*, 2011.

[32] Alan E Gelfand, Susan E Hills, Amy Racine-Poon, and Adrian FM Smith. Illustration of bayesian inference in normal data models using gibbs sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

[33] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, (6):721–741, 1984.

[34] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):21, 2012.

[35] S. Gould, F. Amat, and D. Koller. Alphabet soup: A framework for approximate energy minimization. In *CVPR*, 2009.

[36] Andrew Guillory and Jeff A Bilmes. Active semi-supervised learning using submodular functions. *arXiv preprint arXiv:1202.3726*, 2012.

[37] Hiroshi Ishikawa. Exact optimization for markov random fields with convex priors. *PAMI*, 25(10):1333–1336, 2003.

[38] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[39] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR*, 2011.

[40] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on computing*, 22(5):1087–1116, 1993.

[41] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

[42] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *IEEE Symposium on the Foundations of Computer Science*, 2002.

[43] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

[44] P. Kohli, M. P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[45] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[46] Pushmeet Kohli and Carsten Rother. Higher-order models in computer vision. *Image Processing and Analysis with Graphs*, pages 65–100, 2012.

[47] Pushmeet Kohli and Philip HS Torr. Efficiently solving dynamic markov random fields using graph cuts. In *ICCV*, 2005.

[48] Pushmeet Kohli and Philip HS Torr. Dynamic graph cuts for efficient inference in markov random fields. *PAMI*, 29(12):2079–2088, 2007.

[49] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[50] Daphne Koller, Nir Friedman, Lise Getoor, and Ben Taskar. Graphical models in a nutshell. *Introduction to statistical relational learning*, 43, 2007.

[51] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006.

[52] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimizedvia graph cuts? *PAMI*, (2):147–159, 2004.

[53] V. Koltun and P. Krahenbuhl. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[54] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[55] Nikos Komodakis and Nikos Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *CVPR*, 2009.

[56] Nikos Komodakis, Georgios Tziritas, and Nikos Paragios. Fast, approximately optimal solutions for single and dynamic mrfs. In *CVPR*, 2007.

[57] Nikos Komodakis, Georgios Tziritas, and Nikos Paragios. Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29, 2008.

[58] Arie MCA Koster, Stan PM Van Hoesel, and Antoon WJ Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 23(3-5):89–97, 1998.

[59] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013.

[60] Volodymyr Kuleshov and Stefano Ermon. Lecture notes: Cs 228 graphical models. Available at `https://ermongroup.github.io/cs228-notes/`.

[61] M. P. Kumar and P. Torr. Improved moves for truncated convex models. In *NIPS*, 2008.

[62] M Pawan Kumar, Vladimir Kolmogorov, and Philip HS Torr. An analysis of convex relaxations for map estimation of discrete mrfs. *JMLR*, 10(Jan):71–106, 2009.

[63] M Pawan Kumar and Philip HS Torr. Fast memory-efficient generalized belief propagation. In *ECCV*, 2006.

[64] M Pawan Kumar, Olga Veksler, and Philip HS Torr. Improved moves for truncated convex models. *JMLR*, 12(Jan):31–67, 2011.

[65] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[66] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. VanBriesen, and Natalie S. Glance. Cost-effective outbreak detection in networks. In *SIGKDD*, 2007.

[67] Hui Lin and Jeff A. Bilmes. Optimal selection of limited vocabulary speech corpora. In *INTERSPEECH*, 2011.

[68] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. Sdp gaps and ugc hardness for multiway cut, 0-extension, and metric labeling. In *ACM Symposium on Theory of Computing*, 2008.

[69] Vikash K Mansinghka, Tejas D Kulkarni, Yura N Perov, and Josh Tenenbaum. Approximate bayesian image interpretation using generative probabilistic graphics programs. In *NIPS*, 2013.

[70] Julian J McAuley and Tibério S Caetano. Faster algorithms for max-product message-passing. *JMLR*, 12(Apr):1349–1388, 2011.

[71] S. Thomas McCormick. Submodular function minimization. *Handbook on Discrete Optimization*, 11:321–391, 2006.

[72] Masakatsu Muramatsu and Tsunehiro Suzuki. A new second-order cone programming relaxation for max-cut problems. *Journal of the Operations Research Society of Japan*, 46(2):164–177, 2003.

[73] Mukund Narasimhan and Jeff A. Bilmes. Pac-learning bounded tree-width graphical models. In *UAI*, 2004.

[74] Mukund Narasimhan, Nebojsa Jojic, and Jeff A. Bilmes. Q-clustering. In *NIPS*, 2005.

[75] Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

[76] Sebastian Nowozin and Christoph H Lampert. *Structured learning and prediction in computer vision*, volume 6. Now publishers Inc, 2011.

[77] Pankaj Pansari. Notes: Submodularity in energy minimisation using graph cuts. `http://www.robots.ox.ac.uk/~pankaj/notes/submodularity_energy_min.pdf`.

[78] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.

[79] Kersten Petersen, Janis Fehr, and Hans Burkhardt. Fast generalized belief propagation for map estimation on 2d and 3d grid-like markov random fields. In *Joint Pattern Recognition Symposium*, pages 41–50. Springer, 2008.

[80] Martyn Plummer et al. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *International Workshop on Distributed Statistical Computing*, 2003.

[81] Brian Potetz and Tai Sing Lee. Efficient belief propagation for higher-order cliques using linear constraint nodes. *Computer Vision and Image Understanding*, 112(1):39–54, 2008.

[82] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *ICML*, 2006.

[83] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[84] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, 2004.

[85] Dmitrij Schlesinger and Boris Flach. *Transforming an arbitrary minsum problem into a binary one*. TU, Fak. Informatik, 2006.

[86] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[87] Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, 1994.

[88] MI Shlezinger. Syntactic analysis of two-dimensional visual signals in the presence of noise. *Cybernetics*, 12(4):612–628, 1976.

[89] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[90] Ying Sun, Prabhu Babu, and Daniel P Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2017.

[91] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *PAMI*, 30(6):1068–1080, 2008.

[92] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV*, 2003.

[93] Daniel Tarlow, Inmar E Givoni, and Richard S Zemel. Hop-map: Efficient message passing with high order potentials. In *AISTATS*, 2010.

[94] O. Veksler. Graph cut based optimization for MRFs with truncated convex priors. In *CVPR*, 2007.

[95] Olga Veksler. Dynamic programming for approximate expansion algorithm. In *ECCV*, 2012.

[96] Olga Veksler. Multi-label moves for mrfs with truncated convex priors. *IJCV*, 98(1):1–14, 2012.

[97] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[98] V. Vineet, J. Warrell, P. Sturgess, and P.H.S Torr. Improved initialization and gaussian mixture pairwise terms for dense random fields with mean-field inference. In *BMVC*, 2012.

[99] Vibhav Vineet, Jonathan Warrell, and Philip HS Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *IJCV*, 110(3):290–307, 2014.

[100] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[101] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.

[102] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 2008.

[103] Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

[104] Yair Weiss and William T Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001.

[105] Tomás Werner. A linear programming approach to max-sum problem: A review. *PAMI*, 29(7), 2007.

[106] Tomás Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (map-mrf). In *CVPR*, 2008.

[107] Tomás Werner. Revisiting the linear programming relaxation approach to gibbs energy minimization and weighted constraint satisfaction. *PAMI*, 32(8):1474–1488, 2010.

[108] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

[109] Alan L Yuille and Anand Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.

[110] J. Zhang, J. Djolonga, and A. Krause. Higher-order inference for multi-class log-supermodular models. In *ICCV*, 2015.

[111] Liang Zhao, Hiroshi Nagamochi, and Toshihide Ibaraki. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming*, pages 17–183, 2005.

[112] Stanislav Zivny, David A. Cohen, and Peter G. Jeavons. The expressive power of binary submodular functions. *Discrete Applied Mathematics*, 157(15):3347–3358, 2009.

[113] Stanislav Zivny and Peter G. Jeavons. Classes of submodular constraints expressible by graph cuts. *Constraints*, 15(3):430–452, 2010.