Study of Fast and Accurate Optimization Methods for Markov Random Fields

TOHOKU UNIVERSITY

Graduate School of Information Sciences

Study of Fast and Accurate Optimization Methods for Markov Random Fields
(高速・高精度なマルコフ確率場の最適化計算手法に関する研究)

A dissertation submitted for the degree of Doctor of Philosophy
(Information Science)

Department of System Information Sciences

by

Masaki SAITO

January 12, 2016

Study of Fast and Accurate Optimization Methods for Markov Random Fields

Masaki Saito

Abstract

We consider optimization problems using a probabilistic graphical model called Markov Random Fields (MRFs). The MRF models are one of the most fundamental probabilistic models in many fields including computer vision. It is used for inference of unknown parameters in a wide range of problems, such as image restoration, super-resolution, stereo matching, optical flow estimation, denoising, image segmentation, 3D reconstruction, and object recognition.

Although MRFs are probabilistic models that are used to accurately predict unknown values by using the prior that all the adjacent sites tend to be similar states, these computational costs are relatively large. To overcome this problem, a number of sophisticated algorithms have been proposed (e.g., graph cut and max-product algorithms). However, most of these algorithms are employed for the MAP inference problem, which estimates the state maximizing the joint probability; there exist a few algorithms for computing the marginal distributions of MRFs. Despite the fact that the marginal inference problem is nevertheless significant in many tasks such as parameter learning of Conditional Random Fields (CRFs) and Maximum Posterior Marginal (MPM) inference, the applicability of the existing methods for computing the marginal distribution is limited.

Based on the above discussions, we tackle this problem in three different directions. The first one is improving the accuracy of the existing methods for marginal distribution estimation by using the TAP (Thouless-Anderson-Palmer) equation, which was proposed in the field of physics. Although it has been confirmed that the estimated accuracy of the TAP is higher than that of Mean Field approximation widely used in computer vision, it has a disadvantage that it is only applicable to a binary MRF. Therefore, we first generalize

the TAP equation and enable it to deal with a wider range of problems. Moreover, we applied the generalized TAP equation to several vision problems, and illustrated its effectiveness.

The second is a study for discretizing the variable space of a continuous MRF model into a discrete one. Optimization algorithms developed for continuous variables are only applicable to a limited number of problems, whereas those for discrete variables are versatile. Thus, it is common to convert the continuous variables into discrete ones for the problems that ideally should be solved in the continuous domain. In this study, we propose novel algorithms for this continuous-discrete conversion: an extended Mean Field approximation and an extended Belief Propagation. These algorithms can correctly handle the variable space discretized in a non-uniform manner. By intentionally using such a non-uniform discretization, a higher balance between computational efficiency and accuracy of marginal distribution could be achieved.

The third study is efficiently solving the marginal inference problem of MRFs by transforming the original MRF into a smaller and simpler one. The goal of this study is to provide methods for solving the marginal inference problem more efficiently. Many existing methods that transform MRFs are only applicable to the MAP estimation problem, and empirically transform an energy function to a simpler one. In contrast, our method systematically derives transformed MRFs suited for the marginal inference problem. Using our formulation, we also propose three applications to transform MRFs: (1) discretization of variable space, (2) grouping of discrete labels, and (3) coarse graining of MRFs. The discretization of variable space transforms a MRF model, which has a continous variable and is impossible to derive marginal distributions, into a simpler MRF having a discrete variable. The grouping of discrete labels speeds up the computation of a marginal inference problem by grouping multiple discrete labels in a site into one label. The coarse graining of MRFs transforms graphs into smaller ones in such a way that a number of connected sites are grouped into a single site. In the study, we also show how some of these MRF transformations are combined in a coarse-to-fine manner, and how our MRF transformation approach is also applied to Markov chain Monte Carlo methods. Through several experiments, we confirmed the effectiveness of our formulation and the aforementioned three applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

We focus on optimization problems of a probabilistic model called Markov Random Field (MRF). MRF is one of the most fundamental probabilistic models studied in the field of computer vision, and is described as the probability distribution representing mutual influences of random variables. Since Geman and Geman [24] first applied the MRF model to the image reconstruction problem in Computer Vision (CV) as a seminal work, MRFs have been widely used to solve all levels of vision systems. Most of MRF models are employed for low-level vision, and include image restoration [24, 65], super resolution [64, 84], stereo matching [77, 83], image segmentation [14, 66], noise reduction [50, 4] and optical flow estimation [100, 95]. Some of popular applications of MRF models for high-level vision are general image recognition [43] and 3D reconstruction [13].

Using MRF theory, we can naturally describe the joint distribution of random variables defined by nodes of MRF models depending on problems we want solve, and efficiently estimate its solution. MRF theory has at least three advantages compared with the other approaches: i) convenience of modelling joint distributions, ii) accurate estimation, and iii) a number of sophisticated algorithms.

Convenience of modelling joint distributions: Appropriate formulation of *uncertainty* is beneficial in many cases. One of the advantages of MRF theory is that joint distributions represented by MRFs are able to naturally describe a statistical property of various vision problems. Suppose the noise reduction problem, in which we estimate a original image from a noisy reference. The MRF theory is used to first construct the joint distribution describing the statistical property behind the natural images (e.g., adjacent pixels in a natural image tend to have the same value). The significant property is that most of these correlations are associated with neighbouring pixels. MRFs are probabilistic models suited for naturally representing these correlations, and utilize them to improve the estimation accuracy.

Note that these correlations result from not only the noise reduction problem but also many other vision problems. For example, an image segmentation problem, which splits the input image into the foreground and the background regions, has a similar property that neighbouring pixels tend to be at the same state. An image recognition problem of estimating a semantic category (e.g., "car", "human", "road", etc.) per pixel, also assumes the validation of the same property, i.e., the adjacent pixels tend to be the same category.

Accurate estimation In computer vision there also exist other methods that directly compute the results without describing any statistical properties (e.g., bilateral filter [87]). One of the reasons why many researchers have used the MRF theory is that it could achieve a better estimation accuracy compared with the other existing methods. This is because that they take complex interactions among random variables into account.

We explain the effect of such interactions using the binary segmentation problem. When the probability that a certain pixel will have the "foreground" state is relatively high, the probability of its neighbouring pixel denotes the same tendency. Thus, the neighbouring pixels in MRF models interact each other, and its effect propagates over all the pixels. This means that all the pixels in a MRF model will affect each other, and therefore MRF theory can generate better estimation results than the other approaches.

A number of sophisticated algorithms: Although a MRF model succeeds to estimate good results by taking such complex interactions into account, its computational complexity is relatively larger than that of the conventional ones. Thus, a number of sophisticated algorithms have been proposed in many fields such as physics, computer science, and operations research.

Specifically, these are roughly classified into the following two categories: algorithms for estimating MAP solution of MRF, and algorithms for computing marginal distributions of MRF. We will provide the details of both of the inference algorithms in Sections 2.5.2 and 2.5.3.

Most of those are proposed for MAP solution; graph cut [9], sequential tree-reweighted message passing [40], dual decomposition [41], fastPD [42], spectral relaxation [12], and semidefinite programming relaxation [88] are these examples. Although the problems for computing marginal distributions is important, there exist only a few algorithms to estimate it; mean field approximation [32], belief propagation [98] and Gibbs sampling [21] are representative examples. It should be noted that the estimation of the marginal distributions is also called "marginal inference".

1.2 Problem definition

While there exist a number of sophisticated algorithms developed for MAP estimation problem, the number of algorithms employed for marginal inference problem is much smaller. One reason can be considered the computational complexity of the marginal inference, which is generally higher than that of the MAP inference. In particular, solving the MAP estimation problem is known to be NP-complete [74, 37], whereas computing the marginal inference problem of general MRFs is #P-complete. Therefore, early studies in the field of MRF mainly focused on the MAP inference.

In contrast to the preceding argument, the marginal inference is nevertheless significant. For example, many studies [49, 38, 43] use MPM (maximum posterior marginal) inference to estimates the results of the marginal distributions of MRF, as an alternative to the

MAP inference. They also use marginal distributions to learn parameters of conditional random fields (CRFs) [78], which are a kind of MRF models. In terms of generative models including Boltzmann Machine (BM) [68, 18] and Latent Dirichlet Allocation (LDA) [8], the marginal distributions are used for both learning of parameters and estimation.

1.2.1 Algorithms for computing marginal distributions

The algorithms proposed for the marginal inference problems are roughly classified into sampling-based algorithms [53, 79, 21, 55] and variational ones [98, 26, 36, 17]. While the sampling-based algorithms estimate true marginal distributions by generating a large number of samples from MRFs, the variational algorithms approximately compute the marginal distributions by iteratively updating marginal distributions by using equations derived from the variational principle [36].

The sampling-based algorithms have two advantages; the first one is that they are applicable a wide variety of MRF models such as a discrete MRF and a continuous one; the second one is that the estimated marginal distributions will approach to the *true* marginal distributions [55]. However, it is known that these computational cost are very expensive. In contrast, the variational algorithms do not estimate true marginal distributions in general, and they can apply only a few MRF models because they need to analytically implement iterative equations depending on MRF models, whereas it has a great advantage that their computations are much faster than those of sampling-based algorithms. The variational algorithms are especially significant when we are to apply MRFs into practical vision problems. Thus, in this thesis, we focus on the marginal inference problem using variational algorithms.

1.3 An overview of our studies

Based on the above discussions, we tackle the challenging problem that expands the applicability of the inference algorithms of the marginal distributions. We consider that there exist at least three directions to deal with this challenge: i) improving the estimation

accuracy of the existing algorithms, ii) enabling the existing algorithms to apply a wide variety of MRF models, and iii) solving existing problems more efficiently.

Chapter 2 Markov Random Fields and optimization methods: Before introducing the details of our studies, in Chapter 2 we describe the basis of the Markov Random Fields and fundamental optimization methods used for them. Specifically, we first briefly introduce a background of MRF theory, and then we describe the conditional independence property and Hammersley-Clifford theorem, which are crucial to define MRFs. In addition to that, we also explain several fundamental models and methods including Pairwise Markov Random Fields, Conditional Random Fields (CRFs), the MAP inference, and the marginal inference. Finally, we introduce two algorithms to approximately estimate the marginal distributions: mean field approximation and belief propagation. Based on these models and algorithms, we simplify the discussion regarding the following three studies.

Chapter 3 Generalization of TAP Equations and these applications: To improve the computational accuracy of the marginal distributions, in Chapter 3 we generalize a method called TAP equation, and use it for outperforming the mean field approximation, which is widely used in the field of computer science. Although the TAP equation has its roots in physics and it has been confirmed that the estimated accuracy of TAP is higher than that of MF, the applicability of the original TAP equation is limited, i.e., it is only applicable to a binary MRF. To overcome this problem, we generalize the TAP equation with Plefka expansion, and enable it to deal with wider range of problems including multi-label MRFs and Boltzmann machines having softmax units. Through several experiments, we confirmed its effectiveness.

Chapter 4 Discrete Inference of Markov Random Fields for non-uniformly Discretized Variable Space: To enable the existing algorithms to implement continuous MRF models, which are intractable to directly compute their marginal distributions, in Chapter 4 we propose general algorithms that can correctly discretize the continuous MRFs into the discrete ones. As these algorithms are extensions of existing inference al-

gorithms such as mean field approximation and belief propagation, and these algorithmic procedures are almost the same as the conventional algorithms, we can easily implement our proposed algorithms with the existing libraries. Moreover, our algorithms can correctly handle the case where the variable space is discretized in a non-uniform manner. Using such a non-uniform discretization, we can drastically reduce the computational cost while maintaining the accuracy. Experimental results show the effectiveness of our formulation.

Chapter 5 Transformation of Markov Random Fields for Marginal Distribution Estimation: To solve the existing problems more efficiently, in Chapter 5 we propose a general formulation for efficiently solving marginal distributions of the MRF by transforming original MRF model into a smaller, and simpler one. While many existing methods that transform MRFs only focus on the MAP estimation problem and empirically transform the energy function, our method systematically derives transformed MRFs suited for a marginal inference problem. In addition to proposing a general formulation, we also apply it into the following three applications: (1) discretization of variable space, (2) grouping of discrete labels, and (3) coarse graining of MRFs. It should be noted that this formulation is also considered as an extended version of the previous study since its applications include (1). Through several experiments, we confirmed the effectiveness of our proposed method.

Chapter 6 Conclusion: Chapter 6 concludes this thesis with a summary. We also discuss further developments of these studies, and the remaining questions as future work.

Chapter 2

Markov random fields and optimization methods

In this chapter, we discuss the MRF models, which are a type of graphical model, and introduce several significant optimization methods for MRFs.

Graphical models are probabilistic models for which a graph represents conditional dependencies between random variables [38, 55]. There exist two types of graphical models: the *Bayesian network* and the *Markov Random Field* (also known as the *undirected graphical model*). The Bayesian network represents conditional dependencies of a set of random variables as a directed acyclic graph (DAG), whereas an MRF represents the dependencies as an undirected graph.

In some domains, a DAG is inadequate to sufficiently represent the joint distribution of a problem (consider the case where we want to solve the low-vision problem in which pixels are described as random variables). Unlike DAG, the undirected graphical model is a much more natural form for many vision problems because it does not require that edge orientations be specified. In this chapter, we focus on MRF models, describe their fundamental properties, and derive some significant inference algorithms for MRFs.

2.1 Background of MRF theory

2.1.1 Ising model and mean field approximation

The Markov Random Field has its roots in the Ising model, which simplifies the classical Heisenberg model and aims to illustrate the effect of ferromagnetism in statistical mechanics. Even though the Ising model itself was first proposed by Wilhelm Lenz in 1920s [48], he was unable to analytically derive the expectation of the spins due to the multi-body interaction terms included in a Hamiltonian function.

Ising solved this problem in 1925 by proposing a method called the Mean Field approximation that effectively computes the multi-body effect in the Ising model [28]. This computational method aims to perform the calculation at a high speed by sacrificing its estimation accuracy. Afterwards, it was extended to apply to a wide range of probabilistic models including MRFs and directed graphical models. In computer science, the mean field approximation is also called "variational bayes".

2.1.2 Equivalence between MRF model and Gibbs distribution

Although the joint distribution of the Ising model is given by the Gibbs distribution, until the 1970s it was unclear that the original definition of MRFs, which was proposed in the field of computer science, and the Gibbs distribution are essentially equivalent. John Hammersley and Peter Clifford proved their equivalence in 1971 with a theorem called the Hammersley-Clifford theorem. In addition, in 1974, Julian Besag gave another rigorous proof of their equivalence[5]. As a result, many recent papers directly introduce the Gibbs distribution as an MRF model. We include an extensive discussion of the Gibbs distribution and the Hammersley-Clifford theorem in Sections 2.2 and 2.3, respectively.

2.1.3 Application of MRF theory to computer vision

MRF theories have long been used in computer vision, a trend that dates back to early work by Geman brothers [24] and Besag [6]. In initial studies regarding the application of an MRF model to computer vision, several approximate techniques (e.g., mean field approximation [58], simulated annealing, and iterated conditional modes (ICM)) were used to estimate the solution of an MRF. Several sophisticated algorithms that efficiently and accurately obtain solutions have been proposed in recent years (e.g., belief propagation and graph cuts). Thus, many researchers tend to use these algorithms instead.

2.2 Conditional independence property

In order to introduce the MRF model, we must first describe the conditional independence property used to define MRFs.

2.2.1 Preliminaries

Suppose we have a set of N random variables and set a X, where $X = \{X_1, X_2, \ldots, X_N\}$. Let $\mathcal{V} = \{1, 2, \ldots, N\}$ be the set of these indices. Each site $i \in \{1, \ldots, N\}$ has a variable x_i defined in space $\mathcal{X}_i$. The space for all variables $\mathbf{x} = (x_1, \ldots, x_N)^\top$ is expressed as $\mathcal{X} = \bigotimes_i \mathcal{X}_i$, where $\bigotimes$ is the Cartesian product.

The variables may be either continuous or discrete, i.e., the variable space $\mathcal{X}$ may be either a continuous or discrete set. In order to handle both cases, in this paper, we will use the symbol $\sum$ to represent not only a summation over discrete variables, but also an integral over continuous variables. However, when the variables are clearly continuous, we will use $\int$ instead of $\sum$.

2.2.2 Pairwise Markov property

Here, we consider a certain pair of sites u and v, and assume that there is no relation (i.e., they are independent) between them when we remove all the sites in $\mathcal{V}\backslash\{u,v\}$. In this case, the two random variables X_u and X_v are *conditionally independent* given $\text{rest}(u,v) \equiv \mathcal{V}\backslash\{u,v\}$, this is commonly written as

$$X_u \perp\!\!\!\perp X_v | X_{\text{rest}(u,v)}. \tag{2.1}$$

Eq. (2.1) is also called the *pairwise Markov property* for undirected graphical models.

We also assume that we know all the conditional properties of $\mathcal{V}$. The Markov Random Field is a probabilistic model representing these properties with an undirected graph.

In this MRF undirected graph, an edge exists iff the variables u and v are *not* conditionally independent given $\text{rest}(u,v)$. That is, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the general undirected graph satisfying all the conditional properties, then $\mathcal{G}$ satisfies $X_u \perp\!\!\!\perp X_v | X_{\text{rest}(u,v)}$ iff $\{u,v\} \nsubseteq \mathcal{E}$. This means that in an MRF model the graph $\mathcal{G}$ represents the conditional relation between i and j.

2.3 Hammersley-Clifford theorem

Although we are able to represent the conditional properties of all the random variables through the undirected graph, we have yet to define the specific form of the joint distribution satisfying all the given conditional properties.

Here we define the joint distribution of all the variables as $q(\mathbf{x})$. In this problem, John Hammersley and Peter Clifford gave necessary and sufficient conditions under which a positive probability distribution can be represented as an MRF [5, 22, 55]:

Theorem 2.3.1. (Hammersley-Clifford) *A positive distribution $q(\mathbf{x}) > 0$ satisfies the conditional independence properties of an undirected graph $\mathcal{G}$ if and only if $q(\mathbf{x})$ can be*

represented as a product of factors, one per maximal clique, i.e.,

$$q(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathcal{C}_\mathcal{G}} \psi_c(\mathbf{x}_c), \tag{2.2}$$

where $\mathcal{C}_\mathcal{G}$ *is the set of all the maximal cliques of* $\mathcal{G}$*,* $\mathbf{x}_c$ *is a subset of* $\mathbf{x}$ *that contains variables in the clique* c*,* $\psi_c(\mathbf{x}_c) > 0$ *is a positive* potential function *or* factor*, and* Z *is the normalization constant given by*

$$Z = \sum_{\mathbf{x}} \prod_{c \in \mathcal{C}_\mathcal{G}} \psi_c(\mathbf{x}_c). \tag{2.3}$$

Note that Z *is also called the partition function.*

Although the proof of this theorem was not officially published, it can be seen in [39].

For convenience, instead of directly defining all the factors in Eq. (2.2), most studies first introduce an indirect function called the *energy function*, and use it to represent factors in an MRF model. In this representation, the factor $\psi_c(\mathbf{x}_c)$ can be described as the function $f_c(\mathbf{x})$ given by

$$\psi_c(\mathbf{x}_c) = \exp\left(-\frac{1}{T} f_c(\mathbf{x}_c)\right), \tag{2.4}$$

where $T > 0$ is a positive constant called *temperature*. Note that the *inverse temperature* $\beta = 1/T$ may be used to simplify the notation instead of using the temperature directly.

Using Eq. (2.4), Eq. (2.2) can be rewritten as

$$q(\mathbf{x}) = \frac{1}{Z} \exp\left(-\beta E(\mathbf{x})\right), \tag{2.5}$$

where $E(\mathbf{x})$ is the energy function given by

$$E(\mathbf{x}) = \sum_{c \in \mathcal{C}_\mathcal{G}} f_c(\mathbf{x}_c). \tag{2.6}$$

The probabilistic model expressed in Eq. (2.5) and Eq. (2.6) is generally called the *Boltzmann distribution* or *Gibbs distribution.*

Instead of directly modeling $\psi_c(\mathbf{x}_c)$, many studies in computer vision first define the energy function depending on the problem, and then construct their probabilistic model.

The advantage of this approach is that it is easier for many researchers to represent the joint distribution with energy representations than factor representations. For example, in many vision problems the ψ_c function tends to be an exponential distribution. In such a case we can easily reduce the model for ψ_c to a simpler one such as the L1 and L2 norms. An additional advantage is that MAP estimation problems (we will describe this later) can be replaced by the minimization of the energy function, and therefore we can naturally apply several algorithms for energy minimization problems, such as graph cut [9], to this task.

2.4 Pairwise Markov random fields

As we discussed in Section 2.3, using the Hammersley-Clifford theorem, the joint distribution of an MRF is determined by a set of factors indicating the maximal cliques of the undirected graph. In many cases, we first assume the conditional properties of the random variables, draw the undirected graph representing these dependencies, model the potential functions depending on the problem, and construct the joint probability of the MRF.

Even if decomposing the joint distribution of an MRF into the product of the factors of the maximal cliques is successful, optimizing the joint distribution, including factors with more than two variables, is generally intractable. Thus, most studies simplify the problem by adding an additional assumption that all factors having more than two variables can be decomposed into the product of subfactors having only one or two variables. Using this assumption the energy function can be rewritten as

$$E(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i) + \sum_{(i,j)\in\mathcal{E}} f_{ij}(x_i, x_j), \tag{2.7}$$

where $f_i(x_i)$ and $f_{ij}(x_i, x_i)$ are called the *unary term* (or *unary potential*) and the *pairwise term*, respectively. $f_i(x_i)$ is generally used to represent the property of site i, and $f_{ij}(x_i, x_j)$ is used to determine the correlation between sites i and j. An MRF model whose energy function consists of only unary terms and pairwise terms is called a *pairwise Markov Random Field* or *first-order Markov Random Field*, whereas a MRF model whose energy

function includes higher-order terms (terms having more than two variables) is called a *higher-order Markov Random Field*.

2.5 CRFs and inference methods

In this section, we introduce Conditional Random Fields (CRFs), which have been frequently used in CV, and describe two fundamental inference methods (MAP inference and marginal inference). Most studies using probabilistic models consist of roughly three parts: i) define a probabilistic model depending on the problems, ii) train the parameters in the model with a large number of training samples, and iii) estimate the outputs from the input features and trained parameters. CRFs are probabilistic models used for solving such problems. The two inference methods (MAP inference and marginal inference), are frequently used to achieve steps i) and ii).

2.5.1 Conditional random fields

A conditional random field (CRF) is a type of MRF in which all potential functions are conditioned by the input features, i.e,

$$q(\mathbf{x}|\mathbf{y},\Theta) = \frac{1}{Z(\mathbf{y},\Theta)} \exp\left(-\beta E(\mathbf{x}|\mathbf{y},\Theta)\right) \tag{2.8}$$

$$E(\mathbf{x}|\mathbf{y},\Theta) = \sum_{c\in\mathcal{C}_\mathcal{G}} f_c(\mathbf{x}_c|\mathbf{y},\theta_c), \tag{2.9}$$

where $\mathbf{y}$ contains the input features and θ_c is the parameter of the clique c. Unlike MRF, a partition function in CRF has multiple arguments given by

$$Z(\mathbf{y},\Theta) = \sum_{\mathbf{x}} \exp\left(-\beta E(\mathbf{x}|\mathbf{y},\Theta)\right). \tag{2.10}$$

In order to simplify CRF training, the potential function $f_c(\mathbf{x}_c|\mathbf{y},\theta_c)$ is usually described by the following log-linear model:

$$f_c(\mathbf{x}_c|\mathbf{y},\theta_c) = \theta_c^\top \Phi(\mathbf{x}_c,\mathbf{y}), \tag{2.11}$$

where $\Phi(\mathbf{x}_c, \mathbf{y})$ is a feature vector derived from the inputs $\mathbf{y}$ and local outputs $\mathbf{x}_c$. Using these representations, CRFs estimate the parameters Θ from a pair of training samples $\{\mathbf{x}^n, \mathbf{y}^n\}$, and efficiently predict outputs $\mathbf{y}$ from unknown inputs $\mathbf{x}$.

Estimating CRF parameters has both advantages and disadvantages [55]. One advantage is that we can estimate these parameters based on data, meaning that the CRF potential functions would be data-dependent. For example, we consider the binary segmentation problem that predicts whether each pixel of an input image belongs to the "foreground" or "background". With a large number of training samples, the CRF can estimate the "appropriate" parameters that represent correlation between two neighboring sites.

A disadvantage is that training CRFs is very time-consuming because it requires a huge number of training samples. Thus, early studies in computer vision do not use this data-dependent approach, and empirically define the CRF parameters.

2.5.2 MAP inference

The MAP (Maximum A Posteriori) method estimates the most plausible output values by finding a maximum of the given $q(\mathbf{x}|\mathbf{y}, \Theta)$, i.e.,

$$\mathbf{x}^*_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} q(\mathbf{x}|\mathbf{y}, \Theta) = \arg\min_{\mathbf{x}} E(\mathbf{x}|\mathbf{y}, \Theta). \tag{2.12}$$

From Eq. (2.5), it is clear that the maximization of $q(\mathbf{x})$ is equivalent to the minimization of $E(\mathbf{x})$. MAP inference is mainly used to estimate the outputs from input features.

Solving the MAP inference problem is known to be NP-complete [74, 37]. Thus, computing the exact solution of an MAP problem is generally infeasible. However, in some simple cases, we can obtain globally optimal solutions of the CRF by applying several efficient minimization algorithms such as graph cuts [9]. Specifically, it is known that a max-product algorithm [49, 55, 37] can compute the true MAP solution if the graph in an MRF is a tree and has no loops. Note that the Viterbi algorithm [91] can be derived as a special version of the max-product algorithm.

2.5.3 Marginal inference

In contrast to MAP inference, the marginal inference method computes the marginal distribution of a factor c by integrating $q(\mathbf{x})$ with respect to all variables except for c. For example, when c indicates a site i, the marginal distribution of the site i can be computed by

$$\begin{aligned} q_i(x_i) &= \sum_{\mathbf{x}_{\setminus i}} q(\mathbf{x}|\mathbf{y}, \Theta) \\ &= \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_N} q(x_1, x_2, \ldots, x_N|\mathbf{y}, \Theta). \end{aligned} \tag{2.13}$$

$\mathbf{x}_{\setminus i}$ references the variable $\mathbf{x}$ except for the element i, i.e., $\mathbf{x}_{\setminus i} \in \mathcal{X}\setminus\mathcal{X}_i$.

As with Eq. (2.13), when c has multiple sites, the marginal distribution can be computed as

$$q_c(\mathbf{x}_c) = \sum_{\mathbf{x}_{\setminus c} \in \mathcal{X}\setminus\mathcal{X}_c} q(\mathbf{x}|\mathbf{y}, \Theta) = \sum_{\mathbf{x}_{\setminus c} \in \mathcal{X}\setminus\mathcal{X}_c} q(\mathbf{x}_c, \mathbf{x}_{\setminus c}|\mathbf{y}, \Theta), \tag{2.14}$$

where $\mathcal{X}_c$ is a subset of $\mathcal{X}$, and given by $\mathcal{X}_c = \bigotimes_{i \in c} \mathcal{X}_i$.

Although the marginal inference is mainly used to estimate parameters of CRFs (we will show this in detail in Section 2.5.5), in some cases it is also used to estimate the most plausible states, as in MAP inference. This estimation is called *MPM (Maximal Posterior Marginal) inference* or *marginal MAP inference* [37].

The i'th element of the MPM solution $\mathbf{x}^*_{\mathrm{MPM}}$ is given by

$$x_i^* = \arg\max_{\mathbf{x}_i} q_i(x_i). \tag{2.15}$$

Because solving an MPM inference problem for general MRFs is #P-complete, it is generally harder to solve than a MAP inference problem.

2.5.4 Relation between MAP solution and MPM solution

A significant property of the MPM solution is that $\mathbf{x}^*_{\mathrm{MPM}}$ of an MRF is equivalent to $\mathbf{x}^*_{\mathrm{MAP}}$ as the limit of $T \to 0$ if its energy function has a unique optimal solution. This

property can be derived by Eq. (2.27). As the limit of $T \to 0$ corresponds to $\beta \to \infty$, the second term of the free energy function can be discarded, i.e., $F[p] \to \langle E \rangle_p$. The minima of $F[p(\mathbf{x})]$ under this limit is clearly the probability that the function only takes the minimum state in $\mathcal{X}$. Hence, $p^*_{\mathrm{MPM}}(\mathbf{x})$ converges

$$p^*_{\mathrm{MPM}}(\mathbf{x}) \to \prod_{i=1}^{N} \delta(x_i - x_i^{\mathrm{MAP}}), \tag{2.16}$$

where $\delta(x)$ and x_i^{MAP} are the Dirac's delta function and i'th element of the MAP solution $\mathbf{x}_{\mathrm{MAP}}$, respectively. Thus, from Eq. (2.16) the MPM solution corresponds to the MAP solution as $T \to 0$.

2.5.5 Training conditional random fields

In this section, we describe how to estimate parameters of CRFs using maximum likelihood and MAP estimation.

As it is generally intractable to directly obtain the parameters of CRFs, gradient-based methods (e.g., Stochastic Gradient Descent (SGD) [57], AdaGrad [15], and L-BFGS [51]) have been used to estimate these parameters. We first consider the case where the parameters are estimated using the maximum likelihood method. In this case the optimal parameter Θ^* is given by

$$\begin{aligned} \Theta^* &= \arg\max_{\Theta} \sum_{n=1}^{N} \ln q(\mathbf{x}^n | \mathbf{y}^n, \Theta) \\ &= \arg\min_{\Theta} -\frac{1}{N} \sum_{n=1}^{N} \ln q(\mathbf{x}^n | \mathbf{y}^n, \Theta) \equiv \arg\min_{\Theta} \mathcal{L}(\Theta). \end{aligned} \tag{2.17}$$

Note that for numerical reasons we minimize the scaled negative log-likelihood function $\mathcal{L}(\Theta)$ instead of maximizing the likelihood.

Differentiating $\mathcal{L}(\Theta)$ with respect to θ_c, we have

$$\frac{\partial \mathcal{L}}{\partial \theta_c} = \beta \left\{ \frac{1}{N} \sum_{n=1}^{N} \frac{\partial f_c(\mathbf{x}_c^n | \mathbf{y}^n, \Theta)}{\partial \theta_c} - \frac{1}{N} \sum_{n=1}^{N} \sum_{\mathbf{x}} q_c(\mathbf{x}_c | \mathbf{y}^n, \Theta) \frac{\partial f_c(\mathbf{x}_c | \mathbf{y}^n, \Theta)}{\partial \theta_c} \right\}, \tag{2.18}$$

where q_c is a marginal distribution of the clique c. The first term on the right hand side in Eq. (2.18) is called a *data term*, and the second term is called a *model term*. In order to simplify the notation, we introduce the following distributions defined as

$$q_{\text{data}}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x} - \mathbf{x}^n)\delta(\mathbf{y} - \mathbf{y}^n) \tag{2.19a}$$

$$q_{\text{model}}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{y} - \mathbf{y}^n) q(\mathbf{x}|\mathbf{y}, \Theta). \tag{2.19b}$$

Note that $\delta(\mathbf{x})$ denotes the Dirac's delta function. Using Eqs. (2.19), Eq. (2.18) can be rewritten as

$$\frac{\partial \mathcal{L}}{\partial \Theta} = \beta \left\{ \left\langle \frac{\partial f_c}{\partial \Theta} \right\rangle_{\text{data}} - \left\langle \frac{\partial f_c}{\partial \Theta} \right\rangle_{\text{model}} \right\}, \tag{2.20}$$

where $\langle \cdot \rangle$ denotes the expectation of the internal function with respect to the specified distribution. Eq. (2.20) indicates that the gradient of $\mathcal{L}(\Theta)$ can be computed by taking the difference of the expectations between the two distributions.

While the computation of the data term is relatively tractable, that of the model term is time-consuming due to the marginal distribution with respect to c. Therefore, training CRFs to efficiently compute the marginal distributions is a significant task.

An interesting property with regard to CRF training is that the CRF negative log-likelihood function is a convex function with respect to Θ if it belongs to the log-linear model (Eq. (2.11)). This means that a CRF with a log-linear model has a unique global optimum parameter, and is guaranteed to converge [55].

2.5.6 Training CRFs with a regularization term

In order to avoid over-fitting, some studies add an extra regularization term to the objective function, i.e.,

$$\mathcal{L}'(\Theta) = \mathcal{L}(\Theta) + \lambda \|\Theta\|_2^2. \tag{2.21}$$

From a probabilistic perspective, Eq. (2.21) corresponds to the MAP estimation of parameters for CRFs. Suppose that all the parameters in CRFs are random variables, and

the joint distribution of $\mathbf{x}$ and Θ is described as follows:

$$q(\mathbf{x}, \Theta|\mathbf{y}) = q(\mathbf{x}|\mathbf{y}, \Theta)q(\Theta). \tag{2.22}$$

Here we assume that $q(\Theta)$ is a Gaussian distribution whose covariance matrix is a diagonal matrix and all elements are identical. When estimating the most plausible parameters using MAP estimation, the objective function is identical to Eq. (2.21). Because both terms in Eq. (2.21) are convex, if the CRF is the log-linear model, Eq. (2.21) is also convex function and has a unique optimal solution.

2.6 Variational principle

In this section we describe several algorithms for efficiently computing the marginal distributions of an MRF model.

2.6.1 Approximate estimation of marginal distributions

Analytically computing the marginal distribution $q_i(x_i)$ from $q(\mathbf{x})$ is generally intractable. Even if the case where all the random variables in an MRF are binary, we have to evaluate 2^N states with regard to $q(\mathbf{x})$ in order to analytically compute the marginal distribution of site i.

As a result, there exist two types of approaches for approximately computing marginal distributions: the first is a sampling-based approach, and the second is a variational inference approach. Sampling-based approaches, which are also known as Markov chain Monte Carlo (MCMC) based approaches, estimate marginal distributions by generating a number of samples from $q(\mathbf{x})$. There are a large number of MCMC-based methods, such as Metropolis-Hastings [53], Swendsen-Wang [79], and Gibbs Sampling [21]. Although it is known that marginal distributions estimated by MCMC-based approaches asymptotically tend towards the exact marginals by generating infinite samples, they also require a large amount of computational time.

In contrast, the variational inference method approximately computes marginal distributions through iterative updates from an equation derived by a variational principle we will describe later. Although these marginal distributions are estimated by variational inference methods and are generally incorrect, the computational time costs are much less then those of MCMC. In the variational inference methods, *Mean Field approximation* (MF) and *Belief Propagation* (BP) are the most popular algorithms. Both methods utilize iterative equations that generate approximate marginal distributions by minimizing the free energy functional. In this paper we focus on the variational inference methods, and derive the iterative functions.

2.6.2 Derivation of free energy

As it is generally infeasible to compute the marginal distributions with regard to $q(\mathbf{x})$, variational methods such as the Mean Field approximation and Belief Propagation first introduce a new approximate distribution $p(\mathbf{x})$ that can efficiently compute marginal distributions. Then, the variational methods introduce the following functional called *Kullback-Leibler* (KL) divergence, and compute the similarity between $p(\mathbf{x})$ and $q(\mathbf{x})$:

$$\mathcal{D}[p\|q] = \sum_{\mathbf{x}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})}. \tag{2.23}$$

An interesting property of the KL divergence is that it is a non-negative functional and is equivalent to zero if and only if $p(\mathbf{x}) = q(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$. Thus, the KL divergence can be regarded as a type of distance functional and can be called the *KL distance*. Although the KL divergence is also called *KL distance*, the KL distance is not strictly a distance functional because it does not satisfy the symmetric property, i.e., $\mathcal{D}[p\|q] \neq \mathcal{D}[q\|p]$.

Finally, the variational methods minimize the KL distance between p and q with respect to p, this is regarded as the approximate distribution of q, i.e.,

$$\hat{p}(\mathbf{x}) = \arg\min_{p} \mathcal{D}[p\|q]. \tag{2.24}$$

The substitution of Eq. (2.5) into Eq. (2.23) yields

$$\begin{aligned}\mathcal{D}[p\|q] &= \frac{1}{T}\sum_{\mathbf{x}} p(\mathbf{x})E(\mathbf{x}) - \left(-\sum_{\mathbf{x}} p(\mathbf{x})\ln\right) + \ln Z \\ &= \beta \left\langle E\right\rangle_p - \mathcal{H}[p] + \ln Z, \end{aligned} \tag{2.25}$$

where $\left\langle E\right\rangle_p = \sum_{\mathbf{x}} E(\mathbf{x})p(\mathbf{x})$ is the expectation of the energy $E(\mathbf{x})$ with respect to $p(\mathbf{x})$, and $\mathcal{H}[p]$ is the entropy of $p(\mathbf{x})$. $\left\langle E\right\rangle_p$ is also called the *mean energy* with respect to $p(\mathbf{x})$. Because the third term of Eq. (2.25) is independent of $p(\mathbf{x})$, the minimization of Eq. (2.25) is strictly equivalent to the minimization of the following functional:

$$p^*(\mathbf{x}) = \arg\min_p F[p], \tag{2.26}$$

where $\mathcal{F}[p]$ is given by

$$\mathcal{F}[p] = \beta \left\langle E\right\rangle_p - \mathcal{H}[p]. \tag{2.27}$$

We call the functional $\mathcal{F}[p]$ the *free energy* [1]. Both MF and BP algorithms estimate the marginal distributions of $q(\mathbf{x})$ by minimizing the free energy, defined by Eq. (2.27).

Note that when the approximate distribution has no constraints, the minima of the free energy is:

$$\mathcal{F}[p^*] = -\ln Z, \tag{2.28}$$

and p^* is given by $p^*(\mathbf{x}) = q(\mathbf{x})$. This equation can be used to derive the TAP equation, which is described later.

2.7 Mean field approximation

In this section we describe the Mean Field (MF) approximation, which is the most basic and popular algorithm in variational inference methods. The MF approximation stems from the field of solid state physics. In solid state physics, a simple model called the

[1] For notational simplicity, we call Eq. (2.27) the free energy instead of $\left\langle E\right\rangle_p - T\mathcal{H}[p]$, which is widely used in the field of physics.

Ising model was used as a simple representation of the multi-body system and has been used to illustrate the effect of ferromagnetism in statistical mechanics. Although it is generally unfeasible to use this model to analytically compute the behavior of a magnetic body due to the existence of an interaction term (identical to the *pairwise term* in Eq. (2.32)), the original MF was used to approximately compute the interactions by replacing the interaction terms with a mean of the expectations. In information science, the MF approximation can be regarded as one type of variational method that uses the variational principle. The advantage of such an approach is that it can be used to derive the MF fixed-point equations, not through using physical concepts (e.g., the thermodynamic limit), but using only probability theory. Thus, in this section, we derive the MF algorithm with the latter approach.

2.7.1 Mean field free energy

The MF first minimizes the free energy under the distribution that all the sites in $p(\mathbf{x})$ are independent. This means that $p(\mathbf{x})$ is defined as

$$p_{\mathrm{MF}}(\mathbf{x}) = \prod_{i=1}^{N} p_i(x_i). \tag{2.29}$$

Although this assumption is generally incorrect, it allows us to efficiently compute its marginal distributions. From Eq. (2.29), the i'th marginal distribution of $p_{\mathrm{MF}}(\mathbf{x})$ is equivalent to $p_i(x_i)$.

We focus on pairwise MRF models in order to simplify the MF derivation. Substituting Eq. (2.29) into $\mathcal{F}[p]$, we have

$$\mathcal{F}[p_{\mathrm{MF}}] = \beta \sum_{i=1}^{N} \sum_{x_i} p_i(x_i) f_i(x_i) + \beta \sum_{(i,j)\in\mathcal{E}} \sum_{x_i} \sum_{x_j} p_i(x_i) p_j(x_j) f_{ij}(x_i, x_j) + \sum_{i=1}^{N} \sum_{x_i} p_i(x_i) \ln p_i(x_i). \tag{2.30}$$

Note that the first and the second terms of Eq. (2.30) correspond to the mean energy of $F[p]$, and the third term corresponds to the entropy of $\mathcal{H}[p]$. Eq. (2.30) describes the *Mean Field free energy*.

Next, we find a stationary point in $\mathcal{F}[p_{\mathrm{MF}}]$ from Eq. (2.30). To minimize Eq. (2.30) under the constraint $\sum_{x_i} p_i(x_i) = 1$, we introduce the following Lagrange function $\mathcal{J}_{\mathrm{MF}}$ by adding the Lagrange multiplier γ_i to Eq. (2.30):

$$\mathcal{J}_{\mathrm{MF}} = \mathcal{F}[p_{\mathrm{MF}}] + \sum_{i=1}^{N} \gamma_i \left(1 - \sum_{x_i} p_i(x_i) \right). \tag{2.31}$$

Finally, we find the stationary point $\mathcal{J}_{\mathrm{MF}}$ using the Euler-Lagrange equation. As the above functional does not include any derived functions (e.g., $\partial p_{\mathrm{MF}}/\partial \mathbf{x}$), the stationary point can be derived by simply differentiating $\mathcal{J}_{\mathrm{MF}}$ with respect to $p_i(x_i)$. Thus, differentiating $\mathcal{J}_{\mathrm{MF}}$ and discarding γ_i from the relation $\sum_{x_i} p_i(x_i) = 1$, we have

$$p_i(x_i) \propto \exp\left[-\beta \left(f_i(x_i) + \sum_{j \in \mathcal{N}_i} \sum_{x_j} p_j(x_j) f_{ij}(x_i, x_j) \right) \right], \tag{2.32}$$

where $\mathcal{N}_i$ is the set of the neighboring sites of site i. Therefore, we have found that the stationary point $\mathcal{F}[p_{\mathrm{MF}}]$ can be derived by estimating a $p_i(x_i)$ that satisfies Eq. (2.32).

2.7.2 Estimation of local minima

From Eq. (2.32), the stationary point of $\mathcal{F}[p]$ must satisfy non-linear fixed point equations such as $p_{\mathrm{MF}}(\mathbf{x}) = G[p_{\mathrm{MF}}(\mathbf{x})]$. The MF generally utilizes an *iterative method* to obtain a set of solutions with regard to such fixed-point equations.

We describe this method in detail as follows. First, the MF initializes the current distribution $p_i(x_i)$ to some appropriate distribution. Then, p_i is iteratively updated by the fixed-point equation (Eq. (2.32)). It is well known that the MF iterations are guaranteed to converge and the distributions p_{MF} estimated by the MF are stationary points of $\mathcal{F}[p_{\mathrm{MF}}]$ [36]. We show the algorithm in Alg.1. Note that we define Z_i as the normalization constant.

2.7.3 Mean field algorithm for binary segmentation

In this subsection, we derive a fixed-point equation for a binary MRF model in which all the nodes take only binary states.

Algorithm 1 The Mean-Field approximation algorithm for a pairwise MRF

for all i **do**
  $p_i^0(x_i) \leftarrow \exp\left[-\beta f_i(x_i)\right] / Z_i$
end for
for $t = 1$ to T **do**
  for all i **do**
    $p_i^t(x_i) \leftarrow \exp\left[-\beta \left(f_i(x_i) + \sum_{j \in \mathcal{N}_i} \sum_{x_j} p_j^{t-1}(x_j) f_{ij}(x_i, x_j)\right)\right] / Z_i$
  end for
end for

Suppose $x_i \in \{-1, +1\}$ for all i and let $E(\mathbf{x})$ be

$$E(\mathbf{x}) = -\sum_i h_i x_i - \sum_{i,j} J_{ij} x_i x_j, \tag{2.33}$$

where $J_{ij} \in \mathbb{R}$ and $h_i \in \mathbb{R}$ are constant parameters, and we are to determine the bias of site i and the interaction between sites i and j, respectively. For example, if J_{ij} is a positive constant, the pair of sites i and j tend to take the same state, and vice versa.

Using Eq. (2.33), we derive the fixed-point MF equation. We also define m_i to be the expectation of the site i:

$$m_i = \langle x_i \rangle_{p_i} = p_i(+1) - p_i(-1). \tag{2.34}$$

In a binary MRF the marginal distribution $p_i(x_i)$ can be rewritten with m_i, that is

$$p_i(x_i = +1) = \frac{1 + m_i}{2} \tag{2.35a}$$

$$p_i(x_i = -1) = \frac{1 - m_i}{2}. \tag{2.35b}$$

Substituting Eq. (2.35a) and Eq. (2.35b) into $\mathcal{F}[p]$, we have

$$\mathcal{F}[\mathbf{m}] = \sum_i \left[\frac{1 + m_i}{2} \ln\left(\frac{1 + m_i}{2}\right) + \frac{1 - m_i}{2} \ln\left(\frac{1 - m_i}{2}\right)\right] - \sum_i h_i m_i - \sum_{i,j} J_{ij} m_i m_j. \tag{2.36}$$

Differentiating Eq. (2.36) with respect to m_i, we can write the fixed-point equation as

$$m_i = \tanh\left[\beta \left(h_i + \sum_{j \in \mathcal{N}_i} J_{ij} m_j\right)\right]. \tag{2.37}$$

2.8 Belief propagation

In this section, we describe Belief Propagation (BP), which is another algorithm for computing marginal distributions. The BP method has its roots in both information science and solid state physics. In information science, Pearl [62] first formulated a BP algorithm that analytically computes the marginal distributions in a graphical model when the factor graph is a tree. In solid state physics, Bethe proposed a similar algorithm, called Bethe approximation [7], which aims to illustrate the effect of a magnetic body in the Ising model. Currently, it is known that both algorithms are essentially the same, and can be interpreted as finding a stationary point of Bethe free energy, which we will describe later. Therefore, in this section, we derive the specific BP algorithm using the variational principle [98, 36].

There exist two types of BP algorithms: the Sum-Product algorithm and the Max-Product algorithm (a.k.a. Min-Sum algorithm) [55]. The Sum-Product is an algorithm for computing marginal distributions, and can be derived through the variational principle, whereas the Max-Product is an algorithm that estimates the MAP solution and can be derived using the Sum-Product algorithm under the limit as $T \to 0$. Thus, in this chapter we will only derive the Sum-Product algorithm.

2.8.1 Bethe free energy

As with the MF algorithm, we minimize the variational free energy in Eq. (2.27) under the constraint that an approximate distribution belongs to a certain class. In BP, the approximate distribution $p(\mathbf{x})$ is represented by

$$p_{\mathrm{BP}}(\mathbf{x}) = \frac{\prod_{ij} p_{ij}(x_i, x_j)}{\prod_i p_i(x_i)^{z_i - 1}}, \tag{2.38}$$

where z_i is the number of neighboring sites to site i, i.e., $z_i = |\mathcal{N}_i|$. In addition to Eq. (2.38), in order to satisfy the condition $\sum_{\mathbf{x}} p_{\mathrm{BP}}(\mathbf{x}) = 1$, we add further constraints given

by

$$\sum_{x_i} p_i(x_i) = 1 \tag{2.39a}$$

$$\sum_{x_i}\sum_{x_j} p_{ij}(x_i, x_j) = 1 \tag{2.39b}$$

$$\sum_{x_j} p_{ij}(x_i, x_j) = p_i(x_i). \tag{2.39c}$$

We minimize $\mathcal{F}[p]$ under the $p_{\mathrm{BP}}(\mathbf{x})$ distribution. Substituting Eq. (2.38) into Eq. (2.27), we have

$$\begin{aligned}\mathcal{F}[P_{\mathrm{BP}}] = \beta \sum_i \sum_{x_i} p_i(x_i) f_i(x_i) + \beta \sum_{(i,j)\in\mathcal{E}} \sum_{x_i}\sum_{x_j} p_{ij}(x_i, x_j) f_{ij}(x_i, x_j) \\ - \sum_i (z_i - 1) \sum_{x_i} p_i(x_i) \ln p_i(x_i) + \sum_{(i,j)\in\mathcal{E}} \sum_{x_i}\sum_{x_j} p_{ij}(x_i, x_j) \ln p_{ij}(x_i, x_j).\end{aligned} \tag{2.40}$$

Note that the first and the second terms in Eq. (2.40) correspond to the mean energy terms in $\mathcal{F}[p]$, and the third and the forth terms correspond to the entropy $\mathcal{H}[p]$. The expression in Eq. (2.40) is generally called *Bethe free energy* [7].

We now introduce $\hat{f}_{ij}(x_i, x_j)$ and $\hat{f}_i(x_i)$ called *local energy* [98] to simplify the notation in Eq. (2.40). Let $\hat{f}_{ij}$ and $\hat{f}_i$ be

$$\hat{f}_{ij}(x_i, x_j) = f_{ij}(x_i, x_j) + f_i(x_i) + f_j(x_j) \tag{2.41a}$$

$$\hat{f}_i(x_i) = f_i(x_i). \tag{2.41b}$$

Utilizing Eq. (2.41a) and Eq. (2.41b), Eq. (2.40) can be rewritten as:

$$\begin{aligned}\mathcal{F}[P_{\mathrm{BP}}] = -\sum_i (z_i - 1) \sum_{x_i} p_i(x_i) \left(\beta \hat{f}_i(x_i) + \ln p_i(x_i)\right) \\ + \sum_{i,j}\sum_{x_i}\sum_{x_j} p_{ij}(x_i, x_j) \left(\beta \hat{f}_{ij}(x_i, x_j) + \ln p_{ij}(x_i, x_j)\right).\end{aligned} \tag{2.42}$$

Next, we minimize $\mathcal{F}[P_{\mathrm{BP}}]$ using Eq. (2.42). Unlike the MF, which has only one type of constraint, the BP has three different types of constraints. Hence, we introduce the

Lagrange function defined as

$$\begin{aligned}\mathcal{J}_{\mathrm{BP}} = F[P_{\mathrm{BP}}] &+ \sum_i \gamma_i \left(1 - \sum_{x_i} p_i(x_i)\right) + \sum_{(i,j)\in\mathcal{E}} \gamma_{ij} \left(1 - \sum_{x_i}\sum_{x_j} p_{ij}(x_i, x_j)\right) \\ &+ \sum_{(i,j)\in\mathcal{E}} \sum_{x_j} \lambda_{ij}(x_j) \left(p_j(x_j) - \sum_{x_i} p_{ij}(x_i, x_j)\right) \\ &+ \sum_{(i,j)\in\mathcal{E}} \sum_{x_i} \lambda_{ji}(x_i) \left(p_i(x_i) - \sum_{x_j} p_{ij}(x_i, x_j)\right), \quad (2.43)\end{aligned}$$

where γ_i, γ_{ij}, and λ_{ij} are Lagrange multipliers.

Next, we apply the Euler-Lagrange equation and obtain the stationary points $p_{ij}(x_i, x_j)$ and $p_i(x_i)$. Differentiating $\mathcal{J}_{\mathrm{BP}}$ with respect to $p_{ij}(x_i, x_j)$ and $p_i(x_i)$, we have

$$\ln p_{ij}(x_i, x_j) = -\beta \hat{f}_{ij}(x_i, x_j) + \lambda_{ij}(x_j) + \lambda_{ji}(x_i) + \gamma_{ij} - 1 \quad (2.44a)$$

$$(z_i - 1)(\ln p_i(x_i) + 1) = -\beta(z_i - 1)\hat{f}_i(x_i) + \sum_{i\in\mathcal{N}_i} \lambda_{ji}(x_i) + \gamma_i. \quad (2.44b)$$

We redefine the Lagrange multiplier λ_{ji} using the following function m_{kj}:

$$\lambda_{ij}(x_j) = \ln \prod_{k\in\mathcal{N}_j\setminus i} m_{kj}(x_j). \quad (2.45)$$

m_{kj} is called the *message*. The substitution of Eq. (2.45) into Eq. (2.44a) and Eq. (2.44b) yields

$$p_{ij}(x_i, x_j) \propto \psi_{ij}(x_i, x_j)\phi_i(x_i)\phi_j(x_j) \prod_{k\in\mathcal{N}_i\setminus j} m_{ki}(x_i) \prod_{l\in\mathcal{N}_j\setminus i} m_{lj}(x_j) \quad (2.46a)$$

$$p_i(x_i) \propto \phi_i(x_i) \prod_{k\in\mathcal{N}_i} m_{ki}(x_i), \quad (2.46b)$$

Finally, from Eq. (2.46a) and Eq. (2.46b) we have

$$m_{ij}(x_j) \propto \sum_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k\in\mathcal{N}_i\setminus j} m_{ki}(x_i), \quad (2.47)$$

where expressions $\phi_i(x_i) = -T \ln f_i(x_i)$ and $\psi_{ij}(x_i, x_j) = -T \ln f_{ij}(x_i, x_j)$ denote the factors of site i and edge ij, respectively.

Algorithm 2 The Belief Propagation algorithm for a pairwise MRFs

for all m_{ij} **do**
  $m_{ij}^0(x_j) \leftarrow 1$
end for
for $t = 0$ to $T - 1$ **do**
  for all m_{ij}^t **do**
    $m_{ij}^{t+1}(x_j) \leftarrow \sum_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}_i \setminus j} m_{ki}^t(x_i)$
  end for
end for
for all i **do**
  $p_i(x_i) \leftarrow \left(\phi_i(x_i) \prod_{k \in \mathcal{N}_i} m_{ki}^T(x_i)\right) / Z_i$
end for

The BP algorithm first initializes all messages m_{ij} to appropriate values. In many cases all messages are initialized to 1 (uniform distribution). The algorithm then updates the messages and evaluates Eq. (2.47), this is repeated until convergence. Finally, the marginal distributions are estimated using Eq. (2.46b). We show the specifics of this algorithm in Alg.2. In the algorithm, we set Z_i to the normalization constant of site i.

One significant property of BP is that, in the case where the graph of an MRF is a tree, the BP algorithm can compute the exact marginal distributions. This also means that the approximate distribution p_{BP} will converge to the original distribution q.

2.8.2 Loopy belief propagation

In general, approximate marginal distributions estimated by BP are not exact wherever the graph contains loops. Moreover, in some cases the BP algorithm does fail to converge or the approximate marginals oscillate between several states. However, it is empirically known that the marginals estimated by Alg.2 are close to the exact marginals of q. These BP methods for estimating marginals of an MRF with cycles are called *Loopy Belief Propagation* (LBP) [56]. The LBP algorithm is identical to Alg.2.

It is interesting to compare the naive MF and LBP algorithms. There are advantages to

the LBP method. First, the results from LBP will converge to the exact marginals when the graphical model is a tree. Second, it has been experimentally confirmed that the LBP algorithm is more accurate than the MF. However, the disadvantage of the LBP is that it can handle fewer distributions than the MF algorithm. In fact, LBP can handle only Gaussian and discrete distributions, whereas the MF algorithm is not limited to these distributions. Moreover, the MF can give a lower bound to the partition function, which is especially useful when it is being used to learn graphical models. This method is called the *variational Bayes method*.

Chapter 3

Generalization of TAP equations and their applications

This chapter discusses the Mean Field approximation and a TAP equation that theoretically extend the MF equation. In the field of solid-state physics, it has been well recognized that the TAP equation developed by Thouless, Anderson, and Palmer yields more accurate estimates than the MF approximation. In the field of machine learning and the related fields, the TAP equation has not been so popular, although there are a few studies showing its effectiveness. This unpopularity can be explained to some extent by the limitation of the conventional TAP equation that it is applicable only to binary MRFs.

In this chapter, we first evaluate whether the conventional TAP equation is applicable to several problems in the field of Computer Vision. Next, we generalize the conventional TAP equation to be able to deal with more general MRF models, leading to a general-purpose expression of the second-order TAP equation. Using this result, we also derive specific forms of the second-order TAP equation for multi-label MRFs and Boltzmann machines having softmax units. Several experimental results show the effectiveness of our approach.

3.1 Introduction

This chapter focuses on the estimation of marginal distributions by the TAP equation. Attention has been paid to the same problem of estimating marginal distributions for MRFs in a different research field, the field of solid-state physics. One of the important achievements in the field is the TAP (Thouless-Anderson-Palmer) equation [86] [1]. The TAP equation can be regarded as an improvement of the MF approximation. (MF is the simplest and the least accurate; for example, MF is faster but usually less accurate than BP [55].) The TAP equation was originally developed for the purpose of exactly solving the Sherrington-Kirkpatrick (SK) model [72]. In the study, a solution is obtained by replacing variables in the energy function with their mean values and then taking the limit $N(\text{the number of particles}) \to \infty$. Later, Plefka showed [63] that the TAP equation can be derived by a different approach based on a free energy, which is now referred to as the Plefka expansion.

The objective function derived by the Plefka expansion, which is called the TAP free energy, has the form of a Taylor series expansion with respect to the parameter called inverse temperature. While the derivation of the TAP free energy differs from that of the free energy associated with MF, the TAP free energy up to the first order term coincides with the MF free energy. Thus, the TAP equation can be regarded as an improvement of MF, where the inclusion of higher order terms is expected to contribute to improve accuracy of the estimates of the marginal distributions.

Following the above developments in solid-state physics, the TAP equations and their applications were actively studied in the field of machine learning from 90's to the beginning of 00's [30, 99, 25, 47, 82, 32, 29, 94]. Despite the fact that it was reported in the literatures that the TAP equations are indeed effective for Boltzmann machines (a class of MRFs), there have been few studies of the TAP equations and their applications since then; an only exception to the authors' knowledge is [97].

We consider that this is mainly because of the limitation of the original TAP equations

[1] It should be noted that Morita previously derived the same version of the TAP equation [54], however, it is generally considered that Thouless, Anderson, and Palmer first derived the TAP equation.

that they can deal with only binary-label MRFs, i.e. those having binary site variables. They are not directly applicable to MRFs having more general types of variables such as discrete multi-label variables, which are more popular in the application fields such as computer vision and natural language processing. Furthermore, as the derivation of the TAP equation for binary-label MRFs is by itself complicated, its generalization to be able to deal with arbitrary MRFs such as multi-label MRFs appears to be even more difficult. Although there are a few studies [32, 82] implying that the original TAP equation is extensible to arbitrary graphical models from the viewpoint of information geometry, these studies focus only on binary graphical models such as sigmoid belief networks and do not actually show an expression of the TAP equation that is applicable to them.

Meanwhile, from the beginning of 00's to date, an increasing number of papers dealing with applications of MRFs have been published, where more general types of MRFs including binary-level MRFs were considered. For example, the introduction of multi-label MRFs enables multi-class classification rather than binary-class classification [66], such as visual recognition of object categories [43]. For the Boltzmann machines, Gaussian units and softmax units that can handle continuous values and discrete multi-labels, respectively, contributing to widen the application area of the Boltzmann machines [44, 69, 16, 75]. However, despite such developments and applications of more general MRFs, the classical MF has been used to estimate their marginal distributions, as was done in 90's.

With these past trends in mind, this study sheds light again on the TAP equation, where the goal is to confirm whether the conventional TAP equation is also applicable to several Computer Vision problems such as binary segmentation problem, and generalize it to be able to deal with a wider range of problems. In particular, we derive a general-purpose expression of the second-order TAP equation that could deal with any arbitrary MRFs. As examples, we also derive two specific second-order TAP equations that can be applied to the multi-label MRFs and the Boltzmann Machines having softmax units.

This chapter is organized as follows. In Section 3.2 we derive the generalized TAP equation. Section 3.3 shows the three TAP equations for binary-label MRFs, multi-label MRFs, and Boltzmann machines having softmax units. Section 3.4 discusses the advan-

tages of the TAP equation. Section 3.5 presents several experimental results. Section 3.6 concludes this chapter.

3.2 Generalization of the TAP equation

We derive the TAP equation that can be applied to a wider class of MRFs. There exists several approaches for deriving the conventional TAP equation: the Cavity method [71, 31], the Plefka method [63, 94], Parisi and Potter's method [61], and methodology using information geometry [82, 32, 2]. In this section, we use the Plefka method and Georges' derivation [25] for generalizing the conventional TAP equation.

3.2.1 Revisiting the conventional TAP equation for binary-label MRFs

We now summarize Plefka's original derivation of the TAP equation for binary-label MRFs [63, 60].

As described in Section 2.7, the approach of the MF approximation first chooses the approximate distribution p of Eq. (2.29) and then calculates the marginal distributions by making it close to the true distribution q. Plefka's method also searches for an approximate distribution p that is maximally close to the true distribution q by minimizing the free energy. It differs from MF in that Plefka's method does not assume a specific class of approximate distributions p's like Eq. (2.29). Moreover, instead of directly solving the minimization with respect to p, Plefka's method considers a constrained minimization problem of the free energy by introducing a constraint with respect to the marginal distributions to estimate.

The original TAP equation can deal with only binary-label MRFs. Then let x_i be the binary variable of the i-th site here: $x_i \in \{-1, +1\}$. In this case, the marginal distribution $p_i(x_i)$ of the site is simply represented by $m_i = \langle x_i \rangle_{p_i(x_i)}$, where $\langle \cdot \rangle_p$ denotes the expectation with respect to p. Thus, the problem is to estimate the expectation m_i

for each site.

Plefka's method starts with introducing a constraint with respect to the approximate distribution $p(\mathbf{x})$ as

$$\langle x_i \rangle_p = m_i \ \text{ for all } i \in \{1, \ldots, N\}. \tag{3.1}$$

This means that the expectation of x_i with respect to $p(\mathbf{x})$ coincides with m_i. (Note that m_i is unknown and to be determined later.) We then consider the minimization of Eq. (2.27) over $p(\mathbf{x})$ under this constraint. Using Lagrange multipliers λ_i's, this minimization can be rewritten as

$$G(\mathbf{m}) = \max_{\lambda} \min_{p} \left(\beta \langle E \rangle_p - \mathcal{H}[p] + \sum_i \lambda_i (m_i - \langle x_i \rangle_p) \right) \tag{3.2}$$

$$= \max_{\lambda} \min_{p} \left(\left\langle \beta E(\mathbf{x}) - \sum_i \lambda_i x_i \right\rangle_p - S[p] + \sum_i \lambda_i m_i \right). \tag{3.3}$$

Note that $G(\mathbf{m})$, a function of $\mathbf{m} = [m_1, \ldots, m_N]$, is the minimizer to $F[p]$.

We denote the two terms in the parenthesis of Eq. (3.3) by $F'[p] = \langle \beta E(\mathbf{x}) - \sum_i \lambda_i x_i \rangle_p - \mathcal{H}[p]$. The distribution p^* minimizing $F'[p]$ is given by $p^*(\mathbf{x}) \propto \exp(-\beta E(\mathbf{x}) + \sum_i \lambda_i x_i)$. Thus, using the fact that the minimum of $F[p]$ is represented by $-\ln Z$, Eq. (3.2) reduces to

$$G(\mathbf{m}) = \max_{\lambda} \sum_i \lambda_i m_i - \ln \left[\sum_{\mathbf{x}} \exp \left(-\beta E(\mathbf{x}) + \sum_i \lambda_i x_i \right) \right]. \tag{3.4}$$

This is referred to as the TAP free energy.

As no arbitrary p is chosen unlike the MF approximation, this approach could yield more accurate marginal distributions. In fact, if $G(\mathbf{m})$ can be rigorously minimized, the solution $\mathbf{m}^* = \arg\min_{\mathbf{m}} G(\mathbf{m})$ should be equal to the true expectations (i.e., the true marginal distributions).

However, it is mathematically intractable to rigorously minimize $G(\mathbf{m})$ of Eq. (3.4) because of the multiple integrals in the second term. To cope with this difficulty, Plefka's method approximates $G(\mathbf{m})$ by its Taylor series expansion at $\beta = 0$ and minimizes the

approximated function. This makes sense since $G(\mathbf{m})$ and its derivatives can analytically be calculated at $\beta = 0$. The Taylor series expansion, or the Plefka expansion, is as follows:

$$G(\mathbf{m}) = G^0(\mathbf{m}) + \beta G^1(\mathbf{m}) + \frac{\beta^2}{2} G^2(\mathbf{m}) + \cdots \quad \text{where} \quad G^n(\mathbf{m}) = \left. \frac{\partial^n G}{\partial \beta^n} \right|_{\beta=0}. \tag{3.5}$$

The analytic expressions of a few lowest-order terms are given in [63]. An interesting fact is that the approximation up to the first order term (called the first-order TAP free energy) coincides with the MF free energy. We will show this property later.

3.2.2 Transformation of the free energy

Now we consider deriving the TAP equation for more general classes of MRFs. In the case of binary labels, the marginal distribution of x_i is efficiently represented by the expectation m_i, as mentioned above. As this does not apply to the case where the sites have multi-labels or continuous values, we need to revise the formulation.

Our approach here is to generalize Eq. (3.1) as follows:

$$p_i(x_i) = \hat{p}_i(x_i) \ \text{ for all } i \in \{1, \ldots, N\}. \tag{3.6}$$

This constrains the marginal distribution $p_i(x_i)$ of each site so that it should be equal to a certain distribution $\hat{p}_i(x_i)$. By using this constraint instead of Eq. (3.1), the minimization of $G(\mathbf{m})$ should (at least formally) be able to be converted to the minimization of a functional $G[\hat{p}_1, \ldots, \hat{p}_N]$ of the distributions $\hat{p}_i(x_i)$'s.

Using Lagrange multipliers $\lambda_i(x_i)$'s, the minimization of free energy can be rewritten as

$$G[\hat{p}_1, \ldots, \hat{p}_N] = \max_{\lambda} \min_{P} \left(\beta \langle E \rangle_P - \mathcal{H}[P] + \sum_i \lambda_i(x_i)(\hat{p}_i(x_i) - p_i(x_i)) \right) \tag{3.7}$$

$$= \max_{\lambda} \min_{P} \left(\left\langle \beta E(\mathbf{x}) - \sum_i \lambda_i(x_i) \right\rangle_P - \mathcal{H}[P] + \sum_i \sum_{x_i} \lambda_i(x_i) \hat{p}_i(x_i) \right). \tag{3.8}$$

We denote the two terms in the parenthesis of Eq. (3.8) by $F'[P] = \langle \beta E(\mathbf{x}) - \sum_i \lambda_i(x_i) \rangle_P - \mathcal{H}[P]$. As with the original derivation of the TAP equation, the distribution p^* minimizing $F'[p]$ is given by

$$q_\beta(\mathbf{x}) \propto \exp\left(-\beta E(\mathbf{x}) + \sum_i \lambda_i(x_i) \right). \tag{3.9}$$

Thus, using the fact that the minimum of $F[p]$ is $-\ln Z$ (see Eq. (2.28)), Eq. (3.8) reduces to

$$G[\hat{p}] = \max_{\lambda} \sum_i \sum_{x_i} \lambda_i(x_i)\hat{p}_i(x_i) - \ln \left[\sum_{\mathbf{x}} \exp\left(\beta E(x) + \sum_i \lambda_i(x_i) \right) \right], \tag{3.10}$$

where $\hat{p}(\mathbf{x})$ is defined as

$$\hat{p}(\mathbf{x}) = \prod_i \hat{p}_i(x_i). \tag{3.11}$$

Note that $\hat{p}(\mathbf{x})$ is merely the product of $\hat{p}_i(x_i)$'s and is introduced for making the notation simpler. Thus, it does not represent the joint distribution of the MRF. Using Eq. (3.11), we will denote $G[\hat{p}_1, \ldots, \hat{p}_N]$ by $G[\hat{p}]$.

As it is mathematically intractable to rigorously minimize Eq. (3.10), we approximate $G[\hat{p}]$ by its Taylor series expansion at $\beta = 0$ and minimize the approximated function. The As with the original version of the TAP equation, The Taylor series expansion is given by Eq. (3.5). Each term in Eq. (3.5) is calculated by (1) obtaining the stationary point of the Lagrange multiplier $\lambda_i(x_i)$ under the condition of $\beta \to 0$ and then (2) calculating G^n by using the obtained $\lambda_i(x_i)$.

3.2.3 Formula for deriving the TAP free energy

The derivation of $G^n[\hat{p}]$'s requires several formula, which will be derived in what follows.

Notation of expectations

To simplify the notation, we employ the following braket notation for representing expectations. First, we denote an expectation of $\mathcal{O}(\mathbf{x})$ with respect to $\hat{p}_i(x_i)$ by

$$\langle \mathcal{O} \rangle_i \equiv \sum_{x_i} \hat{p}_i(x_i) \mathcal{O}(\mathbf{x}). \tag{3.12}$$

We also denote an expectation of $\mathcal{O}(\mathbf{x})$ with respect to $\hat{p}_i(x_i)$ and $\hat{p}_j(x_j)$ by

$$\langle \mathcal{O} \rangle_{ij} \equiv \sum_{x_i} \sum_{x_j} \hat{p}_i(x_i) \hat{p}_j(x_j) \mathcal{O}(\mathbf{x}). \tag{3.13}$$

Next, we define $\langle \cdot \rangle$ to be an expectation with respect to $\hat{P}(\mathbf{x}) = \prod_i \hat{p}(x_i)$, i.e.,

$$\langle \mathcal{O} \rangle \equiv \sum_{\mathbf{x}} \left(\prod_i \hat{p}_i(x_i) \right) \mathcal{O}(\mathbf{x}). \tag{3.14}$$

For example, $\langle f_{ij} \rangle$ represents the expectation of $f_{ij}(x_i, x_j)$ with respect to $\hat{p}(\mathbf{x})$.

$$\langle f_{ij} \rangle = \langle f_{ij} \rangle_{ij} = \sum_{x_i, x_j} \hat{p}_i(x_i) \hat{p}_j(x_j) f_{ij}(x_i, x_j) \tag{3.15}$$

$G[\hat{P}]$ in the limit of $\beta \to 0$ (0th-order TAP free energy)

We first derive $G[\hat{P}]$ in the limit of $\beta \to 0$, which is the 0th-order TAP free energy. Using the fact that $E(\mathbf{x})$ vanishes as $\beta \to 0$, Eq. (3.8) reduces to

$$\begin{aligned} \lim_{\beta \to 0} G[\hat{P}] &= \sum_i \sum_{x_i} \lambda_i(x_i) \hat{p}_i(x_i) - \ln \left[\sum_{x_i} \exp \left(\sum_i \lambda_i(x_i) \right) \right] \\ &= \sum_i \sum_{x_i} \lambda_i(x_i) \hat{p}_i(x_i) - \sum_i \ln \left[\sum_{x_i} \exp \left(\lambda_i(x_i) \right) \right]. \end{aligned} \tag{3.16}$$

From the Euler-Lagrange equation, we have the following equation for $\lambda_i(x_i)$:

$$\hat{p}_i(x_i) = \frac{\exp \left(\lambda_i(x_i) \right)}{\sum_{x_i} \exp \left(\lambda_i(x_i) \right)}. \tag{3.17}$$

Although the stationary point of $\lambda_i(x_i)$ is given in the form of $\lambda_i(x_i) = \ln \hat{p}_i + C$, where C is an arbitrary constant, C does not affect the minimization of $G[\hat{p}]$. Hence, we select the fixed point of $\lambda_i(x_i)$ to be $\ln \hat{p}_i(x_i)$, resulting in that Eq. (3.16) is given by

$$\lim_{\beta \to 0} G[\hat{p}] = \sum_i \sum_{x_i} \hat{p}_i(x_i) \ln \hat{p}_i(x_i) = -\mathcal{H}[\hat{p}]. \tag{3.18}$$

Eq. (3.18) means that the 0th TAP free energy is equivalent to the entropy of the distribution $\hat{p}(\mathbf{x})$.

Defining $\langle\cdot\rangle_\beta$ and $U_\beta(\mathbf{x})$

We denote the expectation of a function $\mathcal{O}(\mathbf{x})$ with respect to the distribution $Q_\beta(\mathbf{x})$ introduced in Eq. (3.9) by $\langle\mathcal{O}\rangle_\beta$:

$$\langle\mathcal{O}\rangle_\beta \equiv \sum_{\mathbf{x}} q_\beta(\mathbf{x})\mathcal{O}(\mathbf{x}). \tag{3.19}$$

As $\beta \to 0$, the fixed point of $\lambda_i(x_i)$ tends to $\ln \hat{p}_i(x_i)$. Hence, as $\beta \to 0$, $\langle\mathcal{O}\rangle_\beta$ coincides with the expectation of the distribution $\hat{p}$, namely,

$$\lim_{\beta\to 0}\langle\mathcal{O}\rangle_\beta = \langle\mathcal{O}\rangle_0 = \langle\mathcal{O}\rangle = \sum_{\mathbf{x}}\left(\prod_i \hat{p}_i(x_i)\right)\mathcal{O}(\mathbf{x}). \tag{3.20}$$

Next, we also derive a new equation with respect to $\partial\langle\mathcal{O}\rangle_\beta/\partial\beta$ we will use later. Differentiating $\langle\mathcal{O}\rangle_\beta$ with respect to β, we have

$$\frac{\partial\langle\mathcal{O}\rangle_\beta}{\partial\beta} = \left\langle\frac{\partial\mathcal{O}}{\partial\beta}\right\rangle_\beta + \left\langle\mathcal{O}(\mathbf{x})\left(-E(\mathbf{x}) + \sum_i \frac{\partial\lambda_i}{\partial\beta}\right)\right\rangle_\beta - \langle\mathcal{O}\rangle_\beta\left\langle -E(\mathbf{x}) + \sum_i\frac{\partial\lambda_i}{\partial\beta}\right\rangle_\beta. \tag{3.21}$$

For the brevity, we introduce a new function $U_\beta(\mathbf{x})$:

$$U_\beta(\mathbf{x}) \equiv E - \langle E\rangle_\beta - \sum_i\left(\frac{\partial\lambda_i}{\partial\beta} - \left\langle\frac{\partial\lambda_i}{\partial\beta}\right\rangle_\beta\right). \tag{3.22}$$

Using Eq. (3.22), Eq. (3.21) can be rewritten as the following simpler form:

$$\frac{\partial\langle\mathcal{O}\rangle_\beta}{\partial\beta} = \left\langle\frac{\partial\mathcal{O}}{\partial\beta}\right\rangle_\beta - \langle\mathcal{O}U_\beta\rangle_\beta + \langle\mathcal{O}\rangle_\beta\langle U_\beta\rangle_\beta. \tag{3.23}$$

As $\langle U_\beta\rangle_\beta$ is clearly equivalent to zero by definition, Eq. (3.23) turns to

$$\frac{\partial\langle\mathcal{O}\rangle_\beta}{\partial\beta} = \left\langle\frac{\partial\mathcal{O}}{\partial\beta}\right\rangle_\beta - \langle\mathcal{O}U_\beta\rangle_\beta. \tag{3.24}$$

$U_\beta(\mathbf{x})$ in the condition of $\beta \to 0$

Next, we examine the behaviour of $U_\beta(\mathbf{x})$ of Eq. (3.22) in the limit of $\beta \to 0$. Using Eq. (3.10), $\partial\lambda_i/\partial\beta$ can be represented as

$$\frac{\partial \lambda_i(x_i)}{\partial \beta} = \frac{\partial^2 G[\hat{p}]}{\partial \hat{p}_i(x_i)\partial \beta}. \tag{3.25}$$

Here, assuming that $G[\hat{p}]$ is a smooth function with respect to β as well as $\hat{p}_i(x_i)$, $\partial\lambda_i/\partial\beta$ as $\beta \to 0$ is written as

$$\begin{aligned}\left.\frac{\partial \lambda_i(x_i)}{\partial \beta}\right|_{\beta=0} &= \frac{\partial \langle E \rangle}{\partial \hat{p}_i(x_i)} = f_i(x_i) + \sum_{j\in\mathcal{N}_i}\sum_{x_i} \hat{p}_j(x_j) f_{ij}(x_i, x_j) \\ &= f_i + \sum_{j\in\mathcal{N}_i} \langle f_{ij} \rangle_j \,. \end{aligned} \tag{3.26}$$

Substituting Eq. (3.26) into Eq. (3.22), we have

$$\begin{aligned}\lim_{\beta\to 0} U_\beta &= \sum_i f_i + \sum_{i,j} f_{ij} - \sum_i \langle f_i \rangle - \sum_{i,j} \langle f_{ij} \rangle \\ &\quad - \sum_i \left(f_i + \sum_{j\in\mathcal{N}_i} \langle f_{ij} \rangle_j - \langle f_i \rangle - \sum_{j\in\mathcal{N}_i} \langle \langle f_{ij} \rangle_j \rangle_i \right) \\ &= \sum_{i,j} f_{ij} - \sum_{i,j} \langle f_{ij} \rangle - \sum_{i,j} \langle f_{ij} \rangle_j - \sum_{i,j} \langle f_{ij} \rangle_i + 2\sum_{i,j} \langle \langle f_{ij} \rangle_j \rangle_i \\ &= \sum_{i,j} f_{ij} - \sum_{i,j} \langle f_{ij} \rangle_j - \sum_{i,j} \langle f_{ij} \rangle_i + \sum_{i,j} \langle f_{ij} \rangle \,. \end{aligned} \tag{3.27}$$

As for the derivation of Eq. (3.27), we used the symmetric nature of the pairwise term (i.e., $f_{ij}(x_i, x_j) = f_{ji}(x_j, x_i)$) as

$$\sum_i \sum_{j\in\mathcal{N}_i} f_{ij}(x_i, x_j) = 2\sum_{i,j} f_{ij}(x_i, x_j).$$

Thus, $U_0(\mathbf{x})$ can be represented using $U_{ij}(x_i, x_j)$ by

$$U_0 = \sum_{i,j} U_{ij}(x_i, x_j) = \sum_{i,j} \Big(f_{ij} - \langle f_{ij} \rangle_j - \langle f_{ij} \rangle_i + \langle f_{ij} \rangle \Big) \,. \tag{3.28}$$

3.2.4 The derivation of the 1st-order TAP free energy

Using the above equations, we derive the 1st-order TAP free energy. Differentiating $G[\hat{P}]$ in Eq. (3.10) with respect to β, we have

$$\frac{\partial G[\hat{P}]}{\partial \beta} = \sum_i \left\langle \frac{\partial \lambda_i}{\partial \beta} \right\rangle_0 + \beta \left\langle E \right\rangle_\beta - \sum_i \left\langle \frac{\partial \lambda_i}{\partial \beta} \right\rangle_\beta . \tag{3.29}$$

As $\beta \to 0$, the first term and the third term of Eq. (3.29) are cancelled out. Making use of the fact that the 0th-order TAP free energy is equal to $-S[\hat{P}]$, the 1st-order TAP free energy can be represented by

$$G[\hat{p}] \approx -\mathcal{H}[\hat{p}] + \beta \left\langle E \right\rangle_{\hat{p}} . \tag{3.30}$$

Eq. (3.30) is equivalent to the MF free energy.

3.2.5 The derivation of the 2nd-order TAP free energy

In a similar manner to the 1st-order TAP free energy, we can derive the 2nd-order TAP free energy. Evaluating the second derivative of $G[\hat{p}]$ in Eq. (3.10) with respect to β, we have

$$\frac{\partial^2 G[\hat{p}]}{\partial \beta^2} = \sum_i \left\{ \left\langle \frac{\partial^2 \lambda_i}{\partial \beta^2} \right\rangle_0 - \left\langle \frac{\partial^2 \lambda_i}{\partial \beta^2} \right\rangle_\beta + \left\langle \frac{\partial \lambda_i}{\partial \beta} U_\beta \right\rangle_\beta \right\} - \left\langle E U_\beta \right\rangle_\beta . \tag{3.31}$$

Taking the limit of the third term in Eq. (3.31) with $\beta \to 0$, we have

$$\left\langle \frac{\partial \lambda_i}{\partial \beta} U_\beta \right\rangle_\beta \to \left\langle (f_i + \sum_{j \in \mathcal{N}_i} \left\langle f_{ij} \right\rangle_j)(f_{ij} - \left\langle f_{ij} \right\rangle_i - \left\langle f_{ij} \right\rangle_j + \left\langle f_{ij} \right\rangle_{ij}) \right\rangle . \tag{3.32}$$

The calculation of Eq. (3.32) needs some efforts. After tedious calculations, we reach the following expressions:

$$\begin{aligned}
&\langle f_i (f_{ij} - \langle f_{ij} \rangle_i - \langle f_{ij} \rangle_j + \langle f_{ij} \rangle_{ij}) \rangle \\
&\qquad = \langle f_i \langle f_{ij} \rangle_j \rangle_i - \langle f_i \rangle_i \langle f_{ij} \rangle_{ij} - \langle f_i \langle f_{ij} \rangle_j \rangle_i + \langle f_i \rangle_i \langle f_{ij} \rangle_{ij} = 0 \\
&\langle \langle f_{ij} \rangle_j f_{ij} \rangle_{ij} - \langle \langle f_{ij} \rangle_j \langle f_{ij} \rangle_i \rangle_{ij} - \langle \langle f_{ij} \rangle_j^2 \rangle_{ij} + \langle \langle f_{ij} \rangle_j \langle f_{ij} \rangle_{ij} \rangle_{ij} \\
&\qquad = \langle \langle f_{ij} \rangle_j^2 \rangle_i - \langle f_{ij} \rangle_{ij}^2 - \langle \langle f_{ij} \rangle_j^2 \rangle_i + \langle f_{ij} \rangle_{ij}^2 = 0.
\end{aligned}$$

Thus, as $\beta \to 0$, the third term of Eq. (3.31) tends to

$$\left\langle \frac{\partial \lambda_i}{\partial \beta} U_\beta \right\rangle_\beta \to 0.$$

We find that the first and the second term of Eq. (3.31) are cancelled out, and the third term approaches to 0 as $\beta \to 0$. Hence, $\partial^2 G[\hat{p}]/\partial \beta^2$ is equivalent to

$$\left.\frac{\partial^2 G}{\partial \beta^2}\right|_{\beta=0} = -\langle E U_0 \rangle = -\langle U_0^2 \rangle \tag{3.33}$$

in the limit of $\beta \to 0$. Finally, using Eq. (3.28), $\langle U_0^2 \rangle$ reduces to

$$\langle U_0^2 \rangle = \sum_{(i,j)\in\mathcal{E}} \sum_{(k,l)\in\mathcal{E}} \langle U_{ij} U_{kl} \rangle . \tag{3.34}$$

In order to calculate Eq. (3.34), we divide $\sum_{i,j}\sum_{k,l} \langle U_{ij}U_{kl}\rangle$ of Eq. (3.34) into the following three cases:

The case where $i \neq j \neq k \neq l$

Using the fact that U_{ij} and U_{kl} are independent of each other and $\langle U_{ij} \rangle = 0$, it turns to

$$\langle U_{ij} U_{kl} \rangle = \langle U_{ij} \rangle \langle U_{kl} \rangle = 0. \tag{3.35}$$

The case where $(i,j) = (k,l)$

After some calculations, $\left\langle U_{ij}^2 \right\rangle$ satisfies

$$\begin{aligned}
\langle U_{ij} U_{ij} \rangle &= \left\langle \Big(f_{ij} - \langle f_{ij} \rangle_j - \langle f_{ij} \rangle_i + \langle f_{ij} \rangle \Big)^2 \right\rangle \\
&= \langle f_{ij}^2 \rangle + \langle \langle f_{ij} \rangle_i^2 \rangle_j + \langle \langle f_{ij} \rangle_j^2 \rangle_i + \langle f_{ij} \rangle^2 \\
&\qquad - 2 \langle f_{ij} \langle f_{ij} \rangle_j \rangle_{ij} - 2 \langle f_{ij} \langle f_{ij} \rangle_i \rangle_{ij} \\
&= \langle f_{ij}^2 \rangle - \langle \langle f_{ij} \rangle_i^2 \rangle_j - \langle \langle f_{ij} \rangle_j^2 \rangle_i + \langle f_{ij} \rangle^2 .
\end{aligned} \tag{3.36}$$

The case where $i = k$ and $j \neq l$

The calculation of $\langle U_{ij}U_{kl}\rangle$ reveals that all the terms in $\langle U_{ij}U_{il}\rangle$ are cancelled out, and thus we have

$$\begin{aligned}
\langle U_{ij} U_{il} \rangle &= \Big\langle \big(f_{ij} - \langle f_{ij} \rangle_j - \langle f_{ij} \rangle_i + \langle f_{ij} \rangle \big) \big(f_{il} - \langle f_{il} \rangle_l - \langle f_{il} \rangle_i + \langle f_{il} \rangle \big) \Big\rangle \\
&= \langle f_{ij} f_{il} \rangle - \langle f_{ij} \langle f_{il} \rangle_l \rangle - \langle \langle f_{ij} \rangle_j f_{il} \rangle + \langle \langle f_{ij} \rangle_j \langle f_{il} \rangle_l \rangle \\
&= \langle f_{ij} f_{il} \rangle - \langle f_{ij} f_{il} \rangle - \langle f_{ij} f_{il} \rangle + \langle f_{ij} f_{il} \rangle = 0.
\end{aligned} \tag{3.37}$$

Substituting Eqs. (3.35)-(3.37) into Eq. (3.35), it is rewritten as

$$\left\langle U_0^2 \right\rangle = \sum_{(i,j)\in\mathcal{E}} \left\langle U_{ij}^2 \right\rangle = \sum_{(i,j)\in\mathcal{E}} \left(\langle f_{ij}^2 \rangle - \langle \langle f_{ij} \rangle_i^2 \rangle_j - \langle \langle f_{ij} \rangle_j^2 \rangle_i + \langle f_{ij} \rangle^2 \right). \tag{3.38}$$

Therefore, using Eq. (3.33) and Eq. (3.38), we finally have the 2nd-order TAP free energy as follows:

$$\begin{aligned} G[\hat{p}] &\approx -\mathcal{H}[\hat{p}] + \beta \left\langle E \right\rangle_{\hat{p}} - \frac{\beta^2}{2} \left\langle U_0^2 \right\rangle_{\hat{p}} \\ &= -\mathcal{H}[\hat{p}] + \beta \left\langle E \right\rangle_{\hat{p}} - \frac{\beta^2}{2} \sum_{(i,j)\in\mathcal{E}} \left\langle U_{ij}^2 \right\rangle_{\hat{p}}. \end{aligned} \tag{3.39}$$

We refer to Eq. (3.39) as the *2nd-order TAP free energy*.

3.2.6 The derivation of the 2nd-order TAP equation

Using Eq. (3.39), we derive the 2nd-order TAP equation for $\hat{p}_i(x_i)$. The derivation of the TAP equation is similar to that of MF. In order to minimize $G[\hat{p}]$ subject to the constraint $\sum_{x_i} \hat{p}_i(x_i) = 1$, we add the Lagrange multiplier η_i and transform the minimization of $G[\hat{p}]$ into

$$G[\hat{p}] + \sum_i \eta_i (1 - \sum_{x_i} \hat{p}_i(x_i)). \tag{3.40}$$

Differentiating Eq. (3.40) with respect to $\hat{p}_i(x_i)$ and equating the derivative to zero, we have the following fixed point equation:

$$\begin{aligned} \hat{p}_i(x_i) \propto \exp \Bigg[& - f_i(x_i) - \sum_{j\in\mathcal{N}_i} \sum_{x_j} \hat{p}_j(x_j) f_{ij}(x_i, x_j) \\ & + \frac{1}{2} \sum_{j\in\mathcal{N}_i} \left(\sum_{x_j} \hat{p}_j(x_j) f_{ij}^2(x_i, x_j) - \left(\sum_{x_j} \hat{p}_j(x_j) f_{ij}(x_i, x_j) \right)^2 \right. \\ & \left. - 2 \left(\sum_{x_j} \hat{p}_j(x_j) \left(\sum_{x_i'} \hat{p}_i(x_i') f_{ij}(x_i', x_j) \right) \left(f_{ij}(x_i, x_j) - \sum_{x_j'} \hat{p}_j(x_j') f_{ij}(x_i, x_j') \right) \right) \right) \Bigg]. \end{aligned} \tag{3.41}$$

where we set $\beta = 1$ for the conciseness. We refer to Eq. (3.41) as the *2nd-order TAP equation*. This is a general-purpose expression of the second-order TAP equation that can be applied to a general class of MRFs.

Using this new equation, estimates of the marginal distributions are computed in an iterative manner similar to MF. Starting with some initial estimates of the marginal distributions, it iteratively updates $\hat{p}_i(x_i)$'s according to Eq. (3.41) until convergence. The $\hat{p}_i(x_i)$'s after convergence give estimates of the marginal distributions, which can be used for inferences or learning with the MRF.

3.2.7 The derivation of the 3rd-order TAP free energy

As with the 2nd-order TAP free energy, the 3rd-order TAP free energy can also be derived in the same manner. Firstly, differentiating Eq. (3.10) with respect to β three times, we have the following equation:

$$\frac{\partial^3 G[\hat{p}]}{\partial \beta^3} = \sum_i \left\{ \left\langle \frac{\partial^3 \lambda_i}{\partial \beta^3} \right\rangle_0 - \left\langle \frac{\partial^3 \lambda_i}{\partial \beta^3} \right\rangle_\beta + 2 \left\langle \frac{\partial^2 \lambda_i}{\partial \beta^2} U_\beta \right\rangle_\beta \right. \\ \left. - \left\langle \frac{\partial \lambda_i}{\partial \beta} U_\beta^2 \right\rangle_\beta + \left\langle \frac{\partial \lambda_i}{\partial \beta} \frac{\partial U_\beta}{\partial \beta} \right\rangle_\beta \right\} - \left\langle E \frac{\partial U_\beta}{\partial \beta} \right\rangle_\beta + \left\langle E U_\beta^2 \right\rangle_\beta . \quad (3.42)$$

Performing the same computation in Section 3.2.6, we found that the first and the second terms in Eq. (3.42) are canceled out, and from the third term to the sixth term are converged to zero in the limit of $\beta \to 0$. Therefore, Eq. (3.42) converges to

$$\left. \frac{\partial^2 G}{\partial \beta^2} \right|_{\beta=0} = \left\langle E U_0^2 \right\rangle_0 = \left\langle U_0^3 \right\rangle = \sum_{i,j} \left\langle U_{ij}^3 \right\rangle + 6 \sum_{i,j,k} \left\langle U_{ij} U_{jk} U_{ki} \right\rangle , \quad (3.43)$$

where (i, j, k) represents all possible triples of neighbour sites in a graph such that $\{(i, j), (j, k), (k, i)\} \in \mathcal{E}^3$ and $i \neq j \neq k$. Hence, the 3rd-order TAP free energy can be represented by

$$\begin{aligned} G[\hat{p}] &\approx -\mathcal{H}[\hat{p}] + \langle E \rangle_{\hat{p}} - \frac{\beta^2}{2} \left\langle U_0^2 \right\rangle_{\hat{p}} + \frac{\beta^3}{6} \left\langle U_0^3 \right\rangle_{\hat{p}} \\ &= -\mathcal{H}[\hat{p}] + \langle E \rangle_{\hat{p}} - \frac{\beta^2}{2} \sum_{i,j} \left\langle U_{ij}^2 \right\rangle_{\hat{p}} \\ &\quad + \frac{\beta^3}{6} \sum_{i,j} \left\langle U_{ij}^3 \right\rangle_{\hat{p}} + \beta^3 \sum_{i,j,k} \left\langle U_{ij} U_{jk} U_{ki} \right\rangle_{\hat{p}} . \end{aligned} \quad (3.44)$$

3.3 The TAP equations for several specific MRFs

In this section, we show three example applications of the derived TAP equation (3.41). One is the application to binary MRF having binary-label variables; second is that to the discrete MRFs having multi-label variables; and the other is that to the Boltzmann machines having softmax units.

3.3.1 Binary MRFs

The class of MRFs in which each site has a binary variable has been considered in many application areas. When we want to estimate the marginal distributions, the first choice has been the MF approximation or BP. As we have extended the TAP equation to be able to deal with multi-labels, the extended one (Eq. (3.41)) will be yet another choice.

A remarkable property of the TAP equation is that it tends to be as fast as MF and can be more accurate than MF. Thus, its advantage to BP is that it could be faster and as accurate as or hopefully even more accurate than BP. The computation of the TAP equation, which is to update the marginal distribution of a site using those of its neighboring sites, is about equivalent in terms of computational cost to the computation of a single message in (loopy) BP. Therefore, the method of the TAP equation will be faster than BP by approximately a multiple equal to the number of the neighboring sites.

The computation of Eq. (3.41) in the binary-label MRFs is relatively simple because it only includes an additional term made by the 2nd-order TAP equation to the original MF equation (Eq. (2.37)). Specifically, suppose that the binary MRF model with the energy function of Eq. (2.33) and $x_i \in \{-1, +1\}$. Substituting Eq. (2.33) into the 3rd-order TAP

free energy, we have

$$G[\mathbf{m}] = \sum_i \left[\frac{1+m_i}{2} \ln\left(\frac{1+m_i}{2}\right) + \frac{1-m_i}{2} \ln\left(\frac{1-m_i}{2}\right) \right] - \beta \sum_i h_i m_i - \beta \sum_{i,j} J_{ij} m_i m_j - \frac{\beta^2}{2} \sum_{i,j} J_{ij}^2 (1-m_i^2)(1-m_j^2) - \frac{2\beta^3}{3} \sum_{i,j} J_{ij}^3 m_i m_j (1-m_i^2)(1-m_j^2) - \beta^3 \sum_{i,j,k} J_{ij} J_{jk} J_{ki} (1-m_i^2)(1-m_j^2)(1-m_k^2). \quad (3.45)$$

Note that this TAP free energy computed by the generalized TAP free energy is exactly the same as the original TAP free energy, which is targeted only to the binary MRFs. As Eq. (3.45) is also considered as the extended version of the MF free energy (Eq. (2.36)), we differentiate $G[\mathbf{m}]$ with respect to m_i, and get the similar fixed-point equation defined as

$$m_i = \tanh\left[\beta h_i + \beta \sum_{j \in \mathcal{N}_i} J_{ij} m_j - \beta^2 \sum_{j \in \mathcal{N}_i} J_{ij}^2 m_i (1-m_j^2) + \frac{2\beta^3}{3} \sum_{j \in \mathcal{N}_i} J_{ij}^3 (1-3m_i^2) m_j (1-m_j^2) - 2\beta^3 \sum_{(j,k) \in \mathcal{N}_i^2} J_{ij} J_{jk} J_{ki} m_i (1-m_j^2)(1-m_k^2) \right], \quad (3.46)$$

where $(j,k) \in \mathcal{N}_i^2$ represents all possible pairs of neighbouring sites of site i. We call Eq. (3.46) the TAP equation for the binary-label MRFs. Since it has the form of self-consistency equations, updating $\mathbf{m}$ according to this equation constitutes an iterative algorithm similar to naive MF.

The order of the terms on the right hand side of Eq. (3.46) directly corresponds to the Taylor series expansion of G, which theoretically implies that more accurate solution will be obtained by using higher-order terms. It is an interesting coincidence that the naive MF equation is equivalent to the first-order TAP equation.

Since the above method is based on the Taylor series expansion with small β, it will be theoretically effective only when β is small (i.e., T is large). However, β appears only in the form of βE in G, and thus it does not make sense to discuss the choice of β

independently of the design of E. Therefore, the effects of the approximation with small β can only be investigated through experiments.

The overall algorithm is the same as Alg.1 except that the variable m_i is updated using Eq. (3.46) instead of Eq. (2.37).

3.3.2 Discrete MRFs

As with the binary MRFs, the class of MRFs in which each site has a discrete variable (or multi-label) plays an important role in many application areas. Using Eq. (3.41), we also derive the TAP equations for multi-label MRFs.

The computation of Eq. (3.41) in the case of multi-label MRFs tends to be complicated. When all the pairwise terms in the energy function (Eq. (2.7)) are given by the Potts model, its computation is simple and thus its cost tends to be small. We consider here a class of MRFs in which the pairwise term is given by

$$f_{ij}(x_i, x_j) = J_{ij}(x_i)\delta(x_i - x_j), \tag{3.47}$$

where $\delta(x)$ is Kronecker's delta function. When representing this pairwise term as a matrix, it becomes a square matrix whose diagonal components are $J_{ij}(x)$'s and non-diagonal components are all zero. The substitution of Eq. (3.47) into Eq. (3.41) yields

$$\hat{p}_i(x_i) \propto \exp\Bigg[- f_i(x_i) - \sum_{j\in\mathcal{N}_i} \hat{p}_j(x_i) J_{ij}(x_i) + \frac{1}{2}\sum_{j\in\mathcal{N}_i} \hat{p}_j(x_i) J_{ij}(x_i)\Big\{(1 - p_j(x_i))J_{ij}(x_i) + 2(\langle J_{ij}\rangle - p_i(x_i)J_{ij}(x_i))\Big\}\Bigg], \tag{3.48}$$

where $\langle J_{ij}\rangle = \sum_x \hat{p}_i(x)\hat{p}_j(x)J_{ij}(x)$.

3.3.3 Boltzmann machines having softmax units

The introduction of softmax units to the Boltzmann machines (BMs) has widened the range of applications of the BMs, as they can deal with not only binary labels but multiple

labels [69, 44]. We show here how the derived TAP equation can be applied to the inference of marginal distributions of the BMs.

Suppose a BM that has D visible units, P hidden units, and L softmax units that represent L labels. We denote each of them by $\mathbf{v} \in \{0,1\}^D$, $\mathbf{h} \in \{0,1\}^P$, and $\mathbf{y} \in \{1,\dots,L\}^C$, respectively. We assume here for the BM that there only exist connections between visible–hidden units, hidden–hidden units, and hidden–softmax units. Then, the energy function of the BM is represented as

$$E(\mathbf{x},\mathbf{h},\mathbf{y};\theta) = -\sum_{i,j} x_i W_{ij} h_j - \frac{1}{2}\sum_{j\neq j'} h_j J_{jj'} h_{j'} - \sum_{j,k}\sum_{l} h_j V_{jk}^l y_k^l, \tag{3.49}$$

where $\theta = \{\mathbf{W},\mathbf{J},\mathbf{V}\}$ represents the parameters of the BM, i.e., the weights of the connections. We neglect the bias terms for the sake of brevity.

To perform learning of or inference by BMs, it is necessary to compute the marginal distribution of each unit with respect to their joint distribution or conditional distribution [68, 55]. For example, unsupervised learning from only a set of input data ($\mathbf{x}$'s) requires the computation of $p(\mathbf{h},\mathbf{y}|\mathbf{x};\theta)$), the marginal distributions of the conditional probability with respect to an input. Similarly, supervised learning using a set of pairs of the input $\mathbf{x}$ and the output $\mathbf{y}$ requires the computation of the conditional distribution $p(\mathbf{h}|\mathbf{x},\mathbf{y};\theta)$ for the learning and $p(\mathbf{h},\mathbf{y}|\mathbf{x};\theta)$ for the inference using the learned model. To compute these marginal distributions, the MF approximation has been the most widely used so far [68]. It is possible to use the derived TAP equation instead of MF, by which we can expect some improvement in estimation accuracy of these marginal distributions without sacrificing computational efficiency.

This is done as follows. We first consider the problem of estimating the marginal distributions with respect to the conditional distribution $p(\mathbf{v},\mathbf{h}|\mathbf{y};\theta)$. Letting $p(h_j = 1|\mathbf{x})$

be μ_j and $p_k(y_k = l|\mathbf{x})$ be μ_k^l, Eq. (3.41) can be rewritten using Eq. (3.49) as

$$\mu_j \leftarrow \sigma\Bigg(\sum_i W_{ij}x_i + \sum_{j'\backslash j} J_{j'j}\mu_{j'} + \frac{1}{2}(1-2\mu_j)\sum_{j'\backslash j} J_{j'j}^2\mu_{j'}(1-\mu_{j'}) + \sum_k \langle V_{jk}\rangle_l + \frac{1}{2}(1-2\mu_j)\sum_k(\langle V_{jk}^2\rangle_l - \langle V_{jk}\rangle_l^2)\Bigg) \tag{3.50}$$

$$\mu_k^l \propto \exp\left(\sum_j V_{jk}^l\mu_j + \frac{1}{2}\sum_j V_{jk}^l(1-2\langle V_{jk}\rangle_l)\mu_j(1-\mu_j)\right), \tag{3.51}$$

where $\langle V_{jk}\rangle_l$ and $\langle V_{jk}^2\rangle_l$ are represented as

$$\langle V_{jk}\rangle_l = \sum_l V_{jk}^l\mu_k^l, \quad \langle V_{jk}^2\rangle_l = \sum_l (V_{jk}^l)^2\mu_k^l. \tag{3.52}$$

The third and fifth terms of Eq. (3.50) correspond to the newly added terms of the second TAP equation (Eq. (3.41)). The third term originates from the connections between the hidden units, and the fifth term originates from those between hidden units and softmax ones. The former is equivalent to the term that was already derived in [94], as the hidden units of our BM are the same binary units as [94]. The latter (the fifth term) has not shown before and novel, as the TAP equation has not been applied to softmax units.

It is possible to deal with $p(\mathbf{h}|\mathbf{x},\mathbf{y};\theta)$ in the same way as $p(\mathbf{x},\mathbf{h}|\mathbf{y};\theta)$. Some calculation yields an equation that is the same as Eq. (3.50) but the third term.

3.4 Advantages of TAP equation

This section discusses several real and potential advantages of the TAP equation to LBP. LBP iterates message passing between neighboring sites until convergence; at each iteration, the following two steps are performed alternately and independently at each site:

(a) the addition of messages from the neighboring sites and the data term of the site

(b) the computation of messages to be sent to the neighboring sites.

3.4.1 More flexible choice of MRF models

The TAP equations are more flexible than LBP in the choice of MRF models, especially of the representation of the marginal distribution $p_i(x_i)$. LBP can handle only the Gaussian distribution in the continuous domain, whereas the distribution functions that the TAP equation can deal with are not limited to the Gaussian distribution. The limitation of LBP stems from the fact that in step (b), LBP marginalizes over the variables of the neighboring sites; it is required that the marginalized distribution should be represented by the same function. Although there are methods based on particle filtering to overcome this limitation [76, 27], there will emerge other issues such as large computational complexity and difficulty with maintaining accuracy. There is no such requirement in the TAP equation, and they could deal with all sorts of continuous parametric distribution functions besides discrete representations. However, it is necessary to derive a different algorithm for each assumed parametric function; moreover, it is another issue whether or not the derived algorithm will be convergent.

3.4.2 Faster computation

Even in the discrete domain, as far as the computational complexity per iteration is concerned, the TAP equation are faster than at least a naive implementation of LBP. In each iteration, the TAP equations merely updates the state of the site by referring to the states of its neighboring sites. Its computational complexity is roughly comparable to the computation of a single message in step (b) of LBP. Thus, LBP will be several times (i.e., the number of edges per site) slower than MF and TAP. Moreover, LBP needs to access all the neighboring sites to compute a single message, and thus the number of total memory accesses is by the same factor larger than MF and TAP. When implementing on parallel systems such as GPU, the gap could become larger, since overall speed tends to be constrained by the number of memory accesses in these systems.

Of course, smaller computational complexity per iteration does not mean smaller overall computational cost. The other equally important factor is the number of iterations needed until convergence. This basically depends on each problem and datum, and can

be investigated only by experiments. According to our experiments, the TAP equations are basically comparable to LBP in this repsect.

It should be noted that there are variants of LBP algorithms that perform step (b) efficiently based on distance transform [19, 1]. To be specific, the naive implementation of step (b) has computational complexity of $O(n^2)$ where n is the number of labels, while the efficient algorithms perform this in $O(n)$, although they can be used for a particular class of energy. A similar efficient method is not known for MF and TAP. Thus, the efficient LBP algorithms will be faster than our current implementations of the TAP equations, especially when n is large. Note, however, that this will not be a problem if n is small.

3.4.3 Accuracy

As is described above, MF computes the marginal distribution at each site in an approximate sense, so does LBP. Thus, our concern is with the accuracy of the approximations. As mentioned above, MF is based on the assumption of the independence of each site, i.e., Eq. (2.29). LBP is based on a supposedly more accurate assumption such that the marginal distribution of a site is represented by Eq. (2.38). Thus, LBP is considered to be more accurate than the naive MF equations to the extent of the difference between Eqs.(2.29) and (2.38).

Although its derivation is considerably different, the method of the TAP equation can be regarded as improving the accuracy of the naive MF equation. Thus, which prevails between the above difference of naive MF from LBP and the improvement by the TAP equation. This generally has to be investigated by experiments. According to our experiments shown in Sec. 3.5, the 3rd-order TAP method generally yields more accurate results than LBP.

3.5 Experimental results

In order to examine the effectiveness of the derived TAP equation, we conducted three experiments: the first is binary segmentation problem with a binary MRF; the second is stereo matching problem with a discrete MRF; and the third is parameter learning problem with a Boltzmann Machine having softmax units. For LBP, we used the naive implementation of the sum-product algorithm.

In the three experiments, Intel Core i7 2.67GHz CPU and nVidia GeForce GTX480 GPU were used. MSVC is used for implementations on the CPU; /fp:fast option is specified to maximize the performance of floating point arithmetic and further the code is parallelized using OpenMP. CUDA is used for implementations on the GPU.

3.5.1 Binary segmentation problem (interactive segmentation)

Following the same procedure as GrabCut [67], we use brushes to roughly specify the foreground and the background pixels of an image, as shown in Fig. 3.1, from which their color models are learned. We used the functions `calcNWeights()` and `constructGCGraph()` from OpenCV2.3 to generate the energy function, where the parameters were set as $\gamma = 50$ and $\lambda = 450$. Defining the variable x_i to indicate whether the pixel i is foreground or background, each method estimates the marginal distribution of $q(\mathbf{x})$ at each pixel. The parameter T was empirically chosen as $T = 80$. For target images, we used an image from [59] and four images from the dataset of [52].

In the original GrabCut, optimization is iteratively performed a few times, while the Gaussian mixture models (GMMs) of the foreground and background pixels are updated at each iteration. When the same procedure is carried out in our case, *MF, TAP, and LBP yielded almost identical results; they are too close to find a significant difference in terms of accuracy.* (The results are also almost the same as those of GC, when each pixel is classified as foreground and background by thresholding with $p = 0.5$.) Therefore, we carry out the optimization only once while fixing the energy initially determined by manual brushes. For the purpose of evaluating the accuracy of the estimation of the marginal

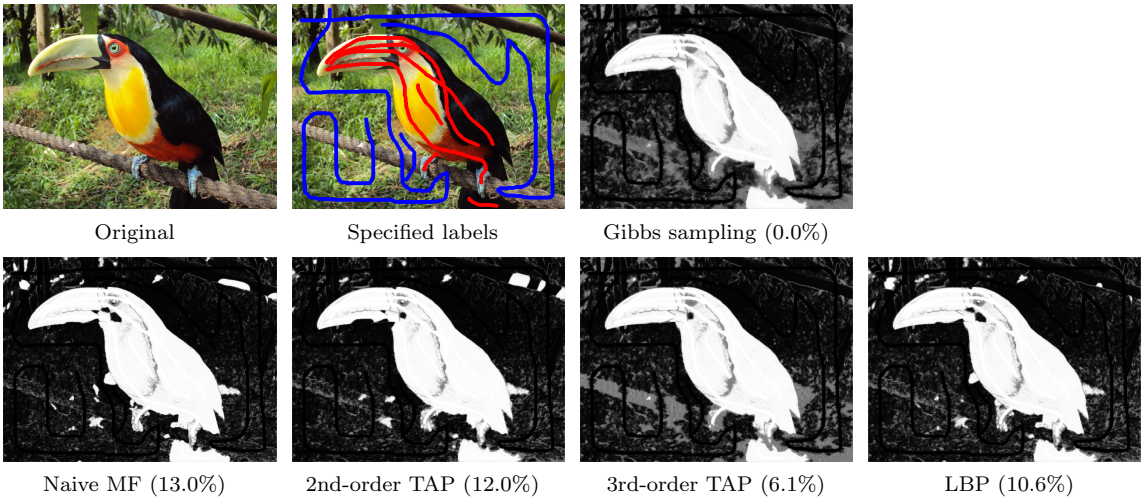

Figure 3.1: Results of interactive segmentation for an image *Bird*. The numbers in the parenthesis are the residual errors after convergence of each method. The methods are ordered from inaccurate to accurate: MF, 2nd-order TAP, LBP, and 3rd-order TAP.

distributions, we also estimate them by Gibbs sampling [38] and use the estimates as the ground truths. In this computation, a sufficient number (= 20000) of samples are generated and used per pixel.

Fig. 3.1 shows the results for the image *Bird* from [59]. The size of the images is 640×480 pixels. The brightness of each pixel represents the probability that the pixel belongs to the foreground; white is 1.0 and black is 0.0. Comparing the results of the MF and two TAP methods with the ground truth obtained by Gibbs sampling, it is observed that the errors tend to decrease in the order of MF, 2nd-order TAP, and 3rd-order TAP. This is more clearly seen in Fig. 3.2 which shows how the errors decrease with the number of iterations. This improvement in accuracy is considered to be due to the effect of the higher-order terms of the TAP equations. Moreover, it is observed from Fig. 3.2 that LBP has smaller errors than 2nd-order TAP, but has larger errors than 3rd-order TAP. This can be visually confirmed in Fig. 3.1.

Table 3.1 shows the computational time of 100 iterations for each method. It is seen from this table that as compared with LBP, the three MF methods are 3-5 times faster on CPU and 6-10 times faster on GPU. This increase in speed is due to the fact that at each iteration, LBP needs to compute messages in eight directions, one of which is

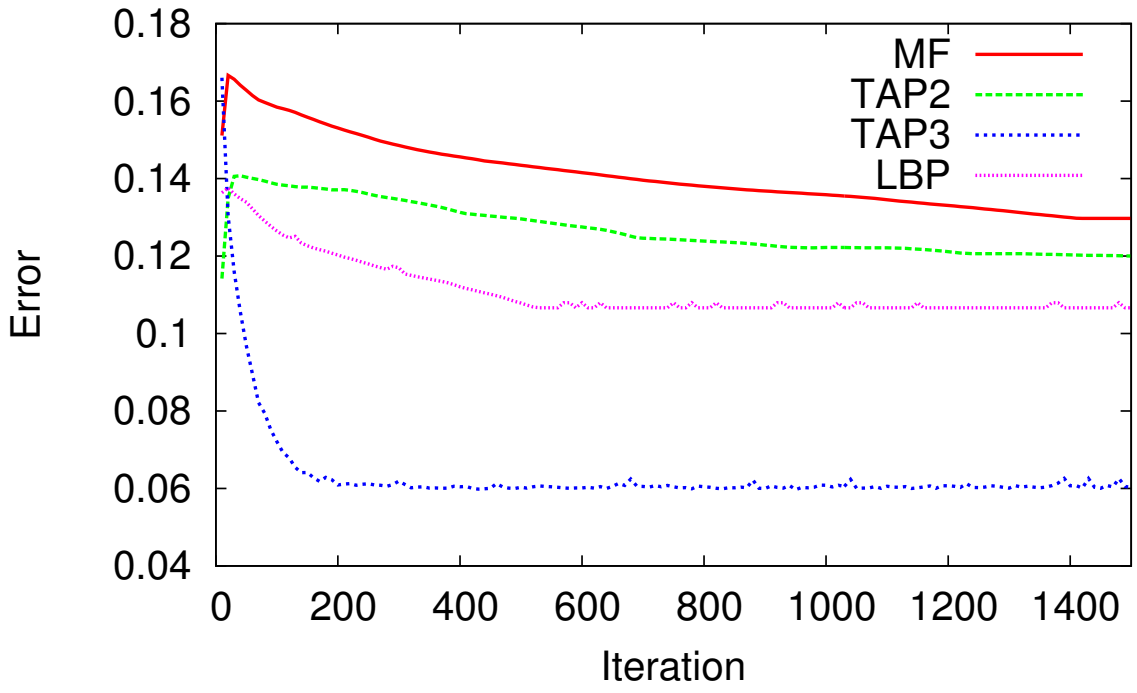

Figure 3.2: Errors per pixel for interactive segmentation vs. the number of iterations.

Table 3.1: Computational time for 100 iterations of each method.

	CPU[ms]	GPU[ms]	CPU/GPU
Mean Field Approximation	730.0	16.8	43.5
TAP equation (2nd)	782.12	17.6	44.4
TAP equation (3rd)	1344.45	26.3	51.1
Loopy Belief Propagation	4253.01	163.61	26.0

computationally comparable to a single iteration of the MF and the TAP methods. It should also be noted that comparing CPU and GPU implementations, the speed ratios are 40-50 for MF and TAP, whereas that for LBP is only 26.0. This is attributable to the fact that LBP requires more memory accesses than MF and TAP.

Figures 3.3 and 3.4 show the results for *Flower*, *Horse*, *Starfish*, and *Tiger* from the dataset of [52]. Figure 3.5 presents the residual errors after convergence for each method. It can be seen that the same observation as above holds true for these images; the result is more accurate in the order of MF, 2nd-order TAP, LBP, and 3rd-order TAP.

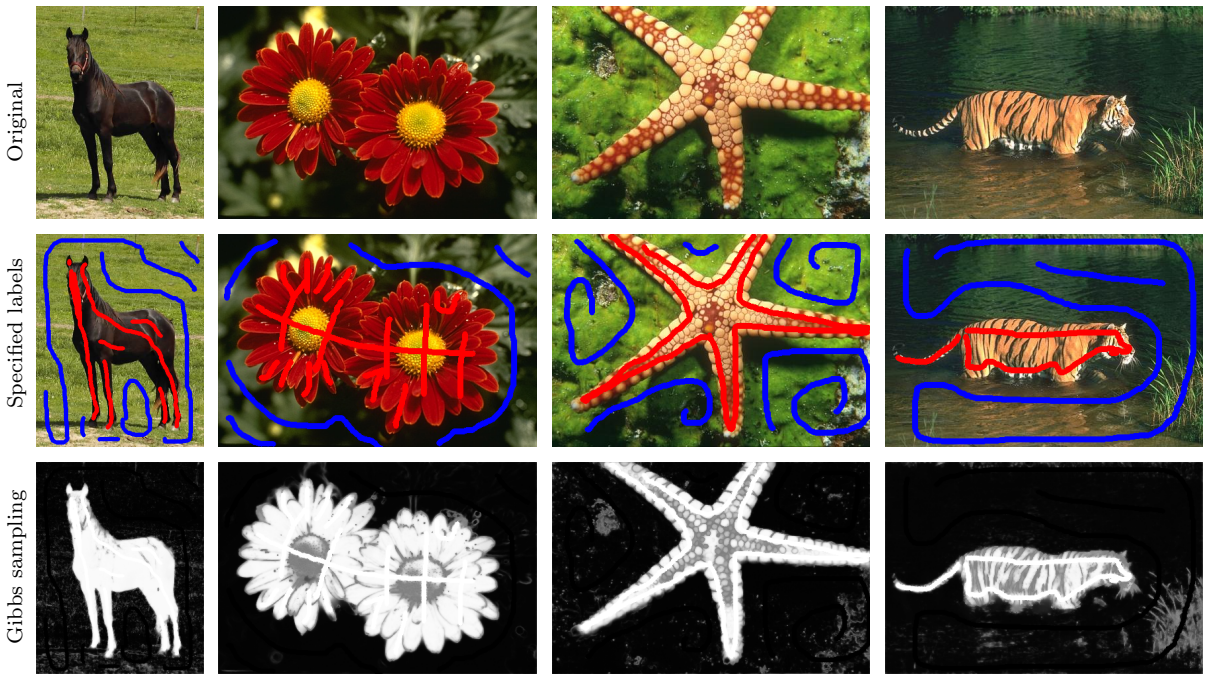

Figure 3.3: Input images and specified labels for interactive segmentation (*Horse, Flower, Starfish* and *Tiger*).

3.5.2 Stereo matching

As an example of discrete MRFs, we choose the problem of matching a pair of stereo images, which is widely studied in the field of computer vision. We compare the proposed TAP equation with the MF approximation and the loopy BP (LBP) (or equivalently, the Sum-Product algorithm). In the experiment we measured the computational time of these compared methods. All of them are implemented on a PC with Intel Core i7 2.67GHz; the code is parallelized by using OpenMP.

In the experiment, we considered a 4-neighbour grid MRF, for which we generated the energy by using the Middlebury MRF library [81]. We set $|L| = 16$, $\lambda = 90$, and truncated $= 1$, which means that the pairwise term belongs to a class of the Potts model. We empirically set the temperature $T = 50$. To evaluate the accuracies, we also estimate the marginal distributions by using the Gibbs sampling [38] with a sufficient number ($=$ 20000) of samples; assuming them to be the true ones, we compare them with those estimated by the above methods.

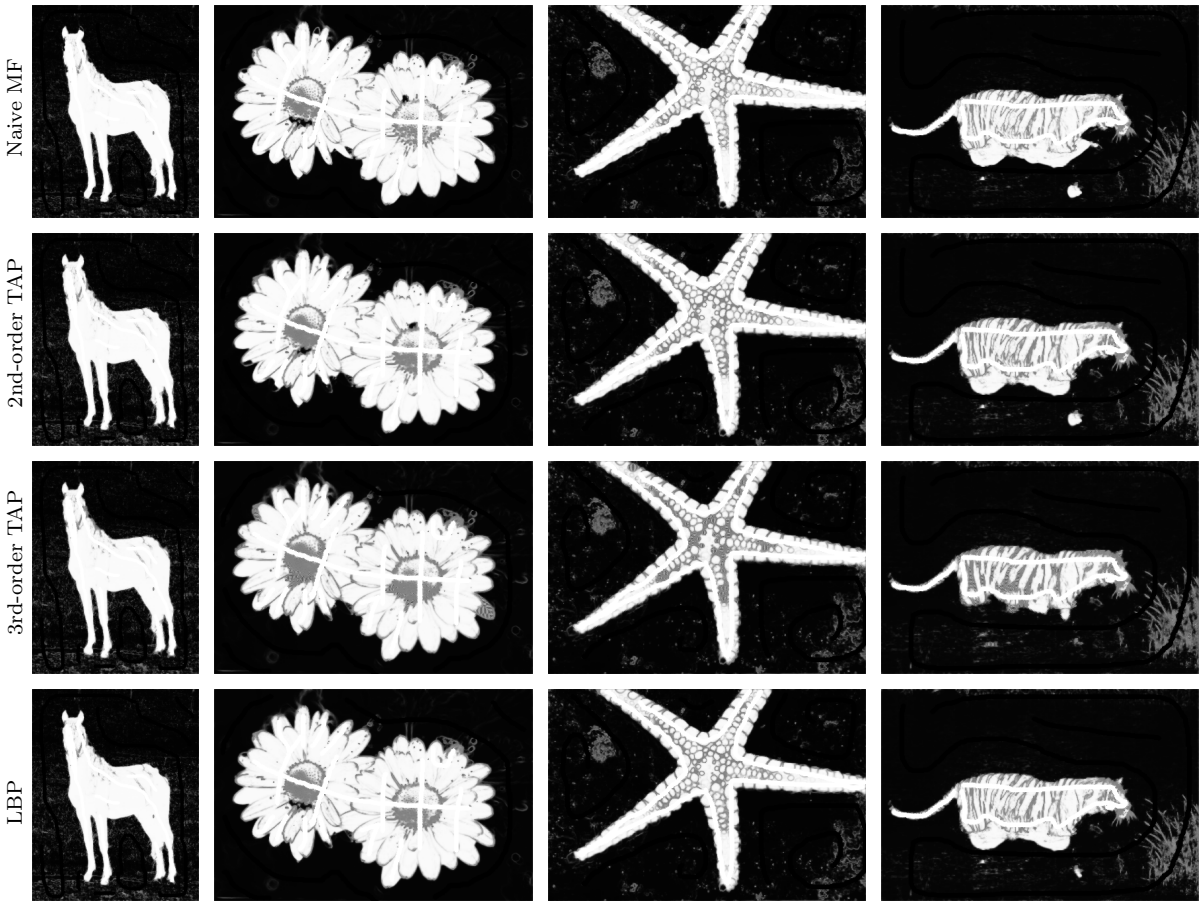

Figure 3.4: Results for the four images (*Horse*, *Flower*, *Starfish*, and *Tiger*).

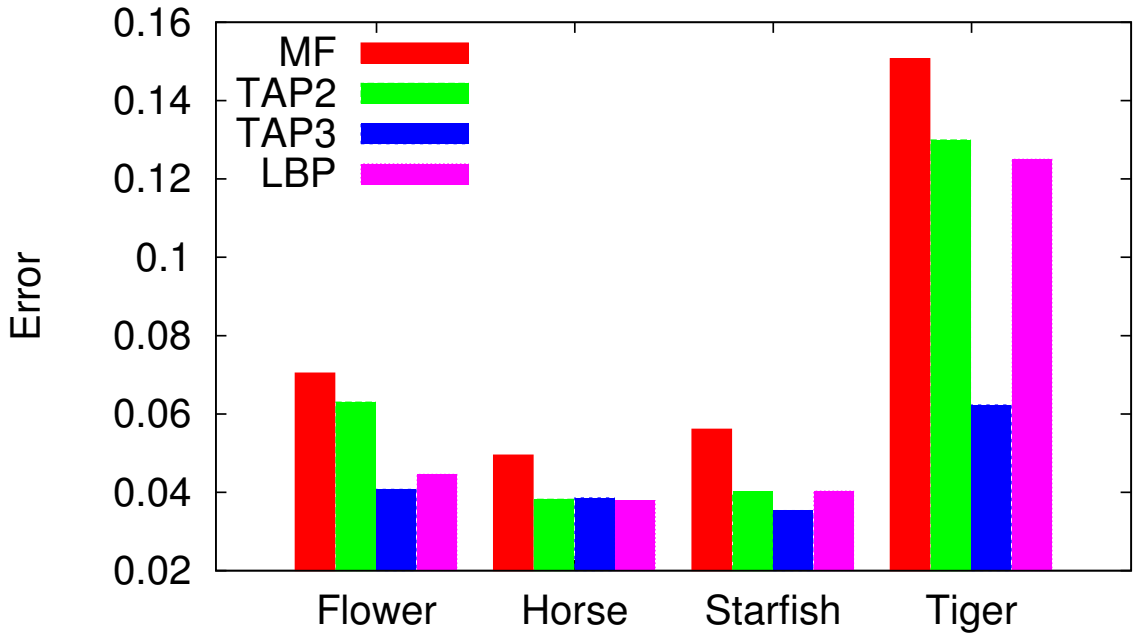

Figure 3.5: Residual errors per pixel for the four images

Fig. 3.6 shows one of the experimental results, which is for *Tsukuba* in the Middlebury dataset [81]. These images show the value of the label x_i at each pixel which maximizes the estimated marginal distribution of the pixel. The number in the parenthesis below each image is the errors of the estimated marginal distributions (or, the difference from the Gibbs sampling). Fig. 3.7 also shows how the errors of *tsukuba* decrease with the number of iterations. It is observed that the errors decrease in the order of MF, 2nd-order TAP, and LBP. Thus, 2nd-order TAP is more accurate than MF as intended but less accurate than BP in this case. The computational times of MF, 2nd-order TAP, and LBP are 1.94, 2.75, and 6.95 seconds, respectively.

3.5.3 Boltzmann machines

To examine how the derived TAP equation improves accuracy of estimating the marginal distributions for BMs, we conducted two experiments with Deep Boltzmann Machines (DBMs). One is supervised learning of a DBM using the MNIST dataset, and the other is unsupervised learning of a DBM using the NORB dataset.

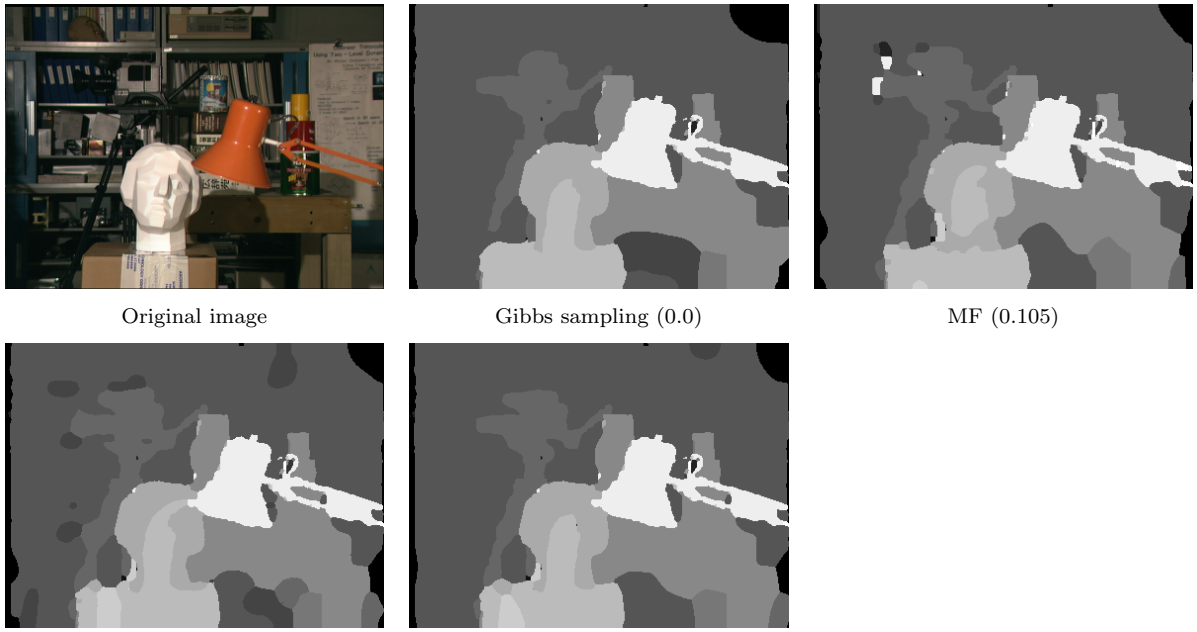

Figure 3.6: The result of stereo matching. The numbers in the parenthesis are the estimation errors of the marginal distributions, evaluated by the difference from the Gibbs sampling.

MNIST: supervised learning

In the experiment, we consider a 3-layer DBM consisting of two hidden layers and one softmax layer. The first and the second layers consist of 500 and 1000 binary units, respectively. The third layer is a softmax unit that can take one of 10 discrete values, which corresponds to the ten digit categories. We normalized the pixel brightness of the MNIST images in the range of $[0, 1]$ and used mini-batches of 100 images for training the DBM.

We performed learning of the DBM in the following way. We first determined the weights between the visible and the first hidden layers by treating them as a RBM. We then calculate the conditional distribution of the units in the first hidden layer when an input image is set to the visible layers. We then determined the weights between the first and second hidden layers and those between the second hidden layer and third (i.e. output) layers, where the above conditional distributions are fed to the first hidden layer and the labels of their inputs are fed to the third layer. Using the weights thus determined,

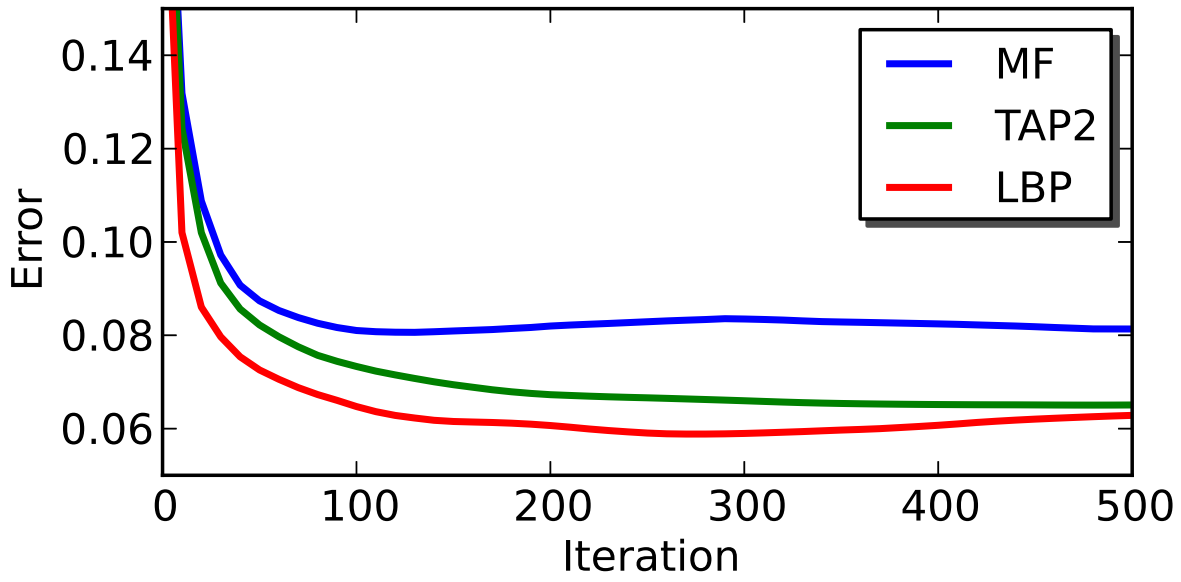

Figure 3.7: Errors per pixel for interactive segmentation of *tsukuba* vs. the number of iterations.

we construct the entire DBM. The resulting DBM achieves 94.92% accuracy for the entire test data.

For the DBM thus obtained, we estimate the marginal distributions $p(\mathbf{h}, \mathbf{y}|\mathbf{v}; \theta)$ of the hidden units $\mathbf{h}$ (of the first and second hidden layers) using the MF approximation and the derived 2nd-order TAP equation. To measure their accuracies, we compare the estimates with those obtained by a Gibbs sampler with a sufficient number (10,000) of particles, as in the above experiment of stereo matching. We performed the estimation by the Gibbs sampler for 100 times and calculated their mean and variance. Table 3.2 shows the accuracies of the estimated marginal distributions for each layer; the "error" columns show the average differences from the mean of the estimates by the Gibbs sampler over a set of 100 randomly chosen images; the "error $\pm\, \hat{\sigma}$" columns indicate their range considering the variance of the Gibbs sampling. It is observed that the TAP equation significantly outperforms MF, which confirms the effectiveness of the derived TAP equation.

Table 3.2: Accuracy of the marginal distributions with respect to the MF and the 2nd-order TAP. The numbers in the table represent the error of the estimated marginal distributions of $p(\mathbf{h}|\mathbf{v};\theta)$. $\hat{\sigma}$ is the measurement error from the Gibbs sampler. To simplify the notation, all the numbers are multiplied by 100.

	Layer1 (500 units)		Layer2 (1000 units)		Layer3 (1 unit)	
MNIST	error	error $\pm\hat{\sigma}$	error	error $\pm\hat{\sigma}$	error	error $\pm\hat{\sigma}$
MF	2.368	2.361, 2.375	1.420	1.414, 1.427	1.489	1.468, 1.510
TAP2	**1.756**	1.749, 1.763	**1.091**	1.084, 1.097	**1.049**	1.020, 1.078
	Layer1 (4096 units)		Layer2 (4096 units)		Layer3 (4096 units)	
NORB	error	error $\pm\hat{\sigma}$	error	error $\pm\hat{\sigma}$	error	error $\pm\hat{\sigma}$
MF	1.029	1.027, 1.030	1.389	1.386, 1.391	1.500	1.498, 1.502
TAP2	**0.992**	0.991, 0.994	**1.272**	1.269, 1.274	**1.448**	1.446, 1.450

NORB: unsupervised learning

We also conducted an experiment of unsupervised learning of a DBM using the NORB dataset [45].

In the experiment, we downsized the problem by downsampling the images from 96×96 to 48×48 pixels. As in [90], we multiplied a scalar to each image so that the averaged pixel values are the same for all the images. Following [16], we then normalized the images so that each pixel has zero mean and unit variance over the entire dataset.

In the experiment, we consider a DBM consisting of one visible and three hidden layers. Each of the three hidden layers has 4096 binary units. For the visible layer, we chose a set of Gaussian units having the variance parameter $\sigma_i = 1$. For the learning of the DBM, we employed the greedy pre-training procedure proposed by [70]. To be specific, following the approach of [70] we train the three RBMs and construct a DBM by concatenating them. We divided the dataset into mini-batches of 100 images and used them for the learning.

As with the experiment of MNIST, we measured accuracy of the marginal distributions of $p(\mathbf{h}|\mathbf{v};\theta)$ by using the MF approximation and the 2nd-order TAP equation. The true marginal distributions were computed by a Gibbs sampler with 10,000 particles. To evaluate their accuracies, we performed the estimation for 100 times and computed their means and variances. The results are shown in Table 3.2. As with MNIST, it is observed that the 2nd-order TAP equation significantly outperforms the MF approximation, which confirms the effectiveness of the TAP equation applied to unsupervised learning of DBMs.

3.6 Summary

In many applications with MRFs, it is necessary to estimate the marginal distributions of their sites. To do this, the classical MF approximation has been used for years in the field of machine learning and the related fields. The TAP equation, which was developed in the field of solid state physics and has been known to do improve the accuracy of the MF approximation, has not been so popular in other fields. This may be because of the limitation of the original TAP equation that it is applicable only to binary MRFs and not to more general MRFs.

To eliminate this limitation, we first generalize the conventional TAP equation and derive a general-purpose expression of the second-order TAP equation that can be applied to more general MRFs. As examples of its application, we then derive the specific TAP equations for binary-label MRFs, multi-label MRFs, and for BMs having softmax units. We show the results of several experiments with discrete multi-label MRFs for stereo matching and with DBMs for supervised learning and unsupervised learning using the MNIST and NORB datasets. They demonstrate the effectiveness of our approach.

Chapter 4

Discrete inference of Markov random fields for non-uniformly discretized variable space

The optimization algorithms for continuous variables are only applicable to a limited number of problems, whereas those for discrete variables are versatile. Thus, it is quite common to convert the continuous variables into discrete ones for the problems that ideally should be solved in the continuous domain, such as stereo matching and optical flow estimation.

In this chapter, we show a novel formulation for this continuous-discrete conversion. The key idea is to estimate the marginal distributions in the continuous domain by approximating them with mixtures of rectangular distributions. Based on this formulation, we derive a Mean Field (MF) algorithm and a Belief Propagation (BP) algorithm. These algorithms can correctly handle the case where the variable space is discretized in a non-uniform manner. By intentionally using such a non-uniform discretization, a higher balance between computational efficiency and accuracy of marginal distribution estimates could be achieved. We present a method for actually doing this, which dynamically discretizes the variable space in a coarse-to-fine manner in the course of the computation. Experimental results show the effectiveness of our approach.

4.1 Introduction

As we have noted in Chapter 2, There are basically two methods for inference using MRF models, MAP (Maximum A Posteriori) inference and MPM (Maximum Posterior Marginal) inference. Both are built upon the Boltzmann distribution $q(\mathbf{x}) \propto \exp(-E(\mathbf{x}))$. MAP directly obtains the maximizer to $q(\mathbf{x})$ and uses it as an estimate of $\mathbf{x}$. MPM first computes the marginal distribution of each variable x_i; it then obtains its maximizer and uses it as the estimate of x_i [85, 35, 49].

As we have mentioned before, we consider the estimation of marginal distributions. Although MAP is in general computationally easier to perform and thus MPM is unlikely to be the first choice when both can be used, there is no other choice when the marginal distributions themselves are necessary, e.g., learning the parameters in CRF (Conditional Random Field) models [43, 68].

The computation of the marginal distributions is differently formulated depending on whether the variable x_i is continuous or discrete. There are two practical algorithms, the Belief Propagation (BP) and the Mean Field (MF) algorithms. Both iteratively estimate the marginal distributions by repeatedly exchanging information, or messages, among the neighboring sites. In the case of continuous variables, the marginal distributions are represented by some parametric distribution function and its parameters are iteratively updated at each site. In the case of discrete variables, the marginal distributions are represented as discrete distributions, and they are iteratively updated at each site.

The former formulation for continuous variables can be used only for a small class of problems, as there are only a few choices for the parametric function representing the marginal distributions. In fact, for the BP algorithm, the Gaussian function is practically an only choice. (This limitation comes from the constraint that in the message updating step, the distributions before and after the update should be represented by the same parametric function.) For the MF algorithm, this limitation is somewhat relaxed but it is in general difficult to derive an iterative algorithm having good convergence property.

On the other hand, the formulation for discrete variables is free from such a limita-

tion, and it can be used for a wide range of problem. Thus, it is used not only for the problems originally defined in discrete domain (e.g., multi-label image segmentation) but also for those originally defined in continuous domain. In the latter case, the continuous variables are discretized into discrete ones. For example, in stereo matching and optical flow estimation, the site variable is disparity and a flow vector, respectively, which are both continuous; they are discretized and the energy function is then defined based on the resulting discrete variables.

In this chapter, we present a novel formulation for this continuous-discrete conversion (i.e., that the problems that should ideally be dealt with in continuous domain are solved by discretization of the variables). In the conventional formulation, the variables are first discretized and the energy of Eq. (2.7) is then defined based on those discrete variables. Then, the marginal distributions defined in the discrete domain are estimated using the discrete MF or BP algorithm. On the other hand, our formulation starts with the energy defined in the continuous domain. To make its minimization feasible, we "discretize" the marginal distribution of each site, or more rigorously, approximate the marginal distribution with a discrete distribution. We then search for the marginal distributions that minimize the energy in the space of the approximating discrete distributions.

For the approximating distribution, we choose a mixture of rectangular distributions in this study. The center of each rectangular distribution corresponds to a discrete value in the conventional formulation, and its height is the parameter to be determined in the minimization. Based on this formulation, we derive the MF and BP algorithms.

In our formulation, the rectangular functions in the mixture are allowed to have arbitrary locations and sizes (as long as any two of them are not overlapped in the variable space), which provides a core practical value of our formulation. In fact, when they are placed on a regular grid and have the same size, the new MF and BP algorithms coincide with the conventional ones, whereas otherwise the two are different. To be specific, the updating terms in the new MF and BP algorithms have additional terms as compared with conventional ones; these additional terms are regarded as compensating the non-uniform distribution of rectangular functions. Note that the conventional MF and BP algorithms are independent of how the continuous variables are discretized; as the energy is defined

after the discretization, differences in the discretization simply change the meaning of the energy.

This flexibility with our formulation enables the followings:

- One can discretize the variable space in a non-uniform manner (e.g., sampled densely in some region and sparsely in others) to improve the estimation accuracy of the marginal distributions without increasing the computational cost.
- One can deal with the case where the variable space is non-Euclidean and is difficult to uniformly discretize, e.g., spherical surface.

The former could be particularly effective for the variable space of two or higher dimensions. For effective non-uniform discretization, some prior knowledge could be used if it is available.

In this chapter, taking one step further, we present a method that performs this non-uniform discretization dynamically in the course of the optimization. Our method employs a coarse-to-fine strategy; starting with coarsely divided blocks of the variable space, it recursively divides the block of the largest mixture weight into subblocks. (Each block is the support of a rectangular function in the mixture distribution.) This block subdivision also requires dividing the current marginal distribution estimates as well as the messages. We also describe how to do this.

This chapter is organized as follows. In Section 4.2, we derive the new MF and BP algorithms that can deal with non-uniformly discretized variable space. Section 4.3 presents a method that dynamically discretizes the variable space in a coarse-to-fine manner, which are to be used with the new MF or BP algorithm. Section 4.4 shows the results of the experiments conducted to examine the effectiveness of our approach. Section 4.5 concludes this chapter.

4.2 Algorithms for a non-uniformly discretized variable space

In this section we derive the new MF and BP algorithms that can deal with non-uniformly discretized variable space.

4.2.1 Derivation of a new MF algorithm

The central issue of the variational approach is the choice of the class of the approximating distributions (p's). As we have noted in Section 2.6, The MF algorithm is derived by choosing the following class of p's:

$$q(\mathbf{x}) \equiv \prod_i q_i(x_i). \tag{4.1}$$

This means that the variable of each site is independent of that of any other site. This is in general too restrictive an assumption to accurately approximate the true distribution, whereas it can significantly simplify computation. Substituting Eq. (4.1) into Eq. (2.27), Eq. (2.27) reduces to

$$\begin{aligned}\mathcal{F}[p] = &\sum_i \int p_i(x_i) f_i(x_i) dx_i \\ &+ \sum_{(i,j)\in\mathcal{E}} \iint p_i(x_i) p_j(x_j) f_{ij}(x_i, x_j) dx_i dx_j + \sum_i \int p_i(x_i) \ln p_i(x_i) dx_i, \end{aligned} \tag{4.2}$$

where $\mathcal{E}$ indicates the set of edges in the graph. Note that the first and second terms correspond to $\langle E \rangle_p$ and the third term to $\mathcal{H}[p]$ of Eq. (2.27), respectively.

We wish to find q that minimizes Eq. (4.2) under the constraint of

$$\int p_i(x_i) dx_i = 1 \quad \text{for all } i. \tag{4.3}$$

By introducing a Lagrange multiplier for this constraint and solving the Euler-Lagrange equation, we have the following fixed point equation for the unknown $p_i(x_i)$ $(i = 1, \ldots, N)$:

$$p_i(x_i) \propto \exp\left[-\left(f_i(x_i) + \sum_{j\in\mathcal{N}_i} \int f_{ij}(x_i, x_j) p_j(x_j) dx_i \right) \right], \tag{4.4}$$

where $\mathcal{N}_i$ is the neighboring site of i-th site. The MF algorithm iteratively updates the estimate of $p_i's$ by using this equation; the substitution of the current estimates to the right hand side gives an updated estimate on the left hand side. The distributions $p_i's$ after convergence directly give the estimates of the marginal distributions of q.

The above derivation is valid for both cases of continuous and discrete variables, and from here, different formulations are necessary for the two cases.

When x_i is a discrete variable, $p_i(x_i)$ is naturally a discrete distribution. Letting $[x^1, \ldots, x^S]$ be the discrete values that x_i can take, we denote their probabilities by $[p_i^1, \ldots, p_i^S]$. Then, Eq. (4.2) reduces to

$$\mathcal{F}[\mathbf{p}] = \sum_i \sum_s p_i^s f_i(x^s) + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} p_i^s p_j^t f_{ij}(x^s, x^t) + \sum_i \sum_s p_i^s \ln p_i^s. \tag{4.5}$$

The fixed point equation (4.4) is such that Eq. (4.2) is minimized under the constraint $\int p_i(x_i)dx_i = 1$. In the discrete case, the constraint becomes $\sum_s p_i^s = 1$ (for any i); under this constraint, Eq. (4.4) turns to

$$p_i^s \propto \exp\left[-\left(f_i(x^s) + \sum_{j\in\mathcal{N}_i} \sum_{t=1}^{S} f_{ij}(x^s, x^t) p_j^t\right)\right], \tag{4.6}$$

which gives the updating rule for the probabilities $[p_i^1, \ldots, p_i^S]$ $(i = 1, \ldots, N)$.

When x_i is a continuous variable, we are to represent $p_i(x_i)$ by some parametric function such as a Gaussian distribution; Eq. (4.4) will then give an updating equation for the parameters. Note however that this is possible only for parametric functions such that the right hand side of Eq. (4.4) yields the same parametric function.

Now we present our formulation for discretizing a continuous problem. We wish to make feasible the computation for a problem originally defined in the continuous domain by discretizing the variable space. To do this, sticking to the above continuous formulation, we represent $p_i(x_i)$ by a mixture of S_i rectangular distributions as

$$p_i(x_i) \equiv \sum_{s=1}^{S_i} \alpha_i^s h_i^s(x_i) \quad i = 1, \ldots, N \tag{4.7}$$

where α_i^s is the mixing coefficient to be determined in the minimization; h_i^s is a rectangular function fixed during the minimization, which is defined as follows. Let $\mathcal{X}$ be the variable

space and d be its dimensionality. Also let $\mathcal{X}_i^s$ be a d-dimensional hyperrectangle (i.e., the Cartesian product of intervals) such that $\mathcal{X}_i^s \bigcap \mathcal{X}_i^t = \emptyset$. Then, $h_i^s(x_i)$ is defined to be

$$h_i^s(x_i) = \begin{cases} 1/\mathcal{V}_i^s & \text{if } x \in \mathcal{X}_i^s \\ 0 & \text{otherwise,} \end{cases} \tag{4.8}$$

where $\mathcal{V}_i^s$ is the volume of $\mathcal{X}_i^s$; thus $\int h_i^s(x_i)dx_i = 1$.

Note that the rectangular functions $h_i^s(x_i)$'s may have non-uniform locations and sizes. Thus, one may distribute $h_i^1(x_i), \ldots, h_i^{S_i}(x_i)$ in the variable space $\mathcal{X}$ densely (or sparsely) for particular portions of $\mathcal{X}$ depending on their importance. Note also that their distribution in $\mathcal{X}$ is allowed to be different for each site, so is even S_i. Thus, one may, for example, increase or decrease S_i for particular sites (e.g., an image region) depending on their importance.

Next we derive the updating equation for α_i^s's similar to Eq. (4.4). Unlike the earlier cases, it cannot be obtained by directly substituting Eq. (4.7) into Eq. (4.4), because of the above generality of our mixtures. (The right hand side of Eq. (4.4) cannot generally be represented by the mixture of the (i-th) site.

Thus, we trace back to the free energy of Eq. (4.2). By substituting Eq. (4.7) into $\langle E \rangle_p$ in Eq. (4.2) and introducing new notations, it reduces to

$$\begin{aligned} \langle E \rangle_p &= \sum_i \int p_i(x_i) f_i(x_i) dx_i + \sum_{(i,j)\in\mathcal{E}} \iint p_i(x_i) p_j(x_j) f_{ij}(x_i, x_j) dx_i dx_j \\ &= \sum_i \sum_s \alpha_i^s \int f_i(x_i) h_i^s(x_i) dx_i + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_i^s \alpha_j^t \iint f_{ij}(x_i, x_j) h_i^s(x_i) h_j^t(x_j) dx_i dx_j \\ &= \sum_i \sum_s \alpha_i^s f_i^s + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_i^s \alpha_j^t f_{ij}^{st}, \end{aligned} \tag{4.9}$$

where f_i^s and f_{ij}^{st} are respectively defined by

$$f_i^s = \int f_i(x_i) h_i^s(x_i) dx_i, \tag{4.10}$$

$$f_{ij}^{st} = \iint f_{ij}(x_i, x_j) h_i^s(x_i) h_j^t(x_j) dx_i dx_j, \tag{4.11}$$

which are the expectations of the data term and the smoothness term with respect to $h_i^s(x_i)$ and $h_j^t(x_j)$.

As with the mean energy, we compute the entropy in the free energy. Although the calculation of the entropy of a mixture density is in general intractable, owing to the introduced constraint $\mathcal{X}_i^s \cap \mathcal{X}_i^t = \emptyset$, we can reduce $\mathcal{H}[p]$ in Eq. (4.2) as follows:

$$\begin{aligned}
\mathcal{H}[p] &= -\sum_i \sum_s \alpha_i^s \int h_i^s(x_i) \ln \left(\sum_{s'} \alpha_i^{s'} h_i^{s'}(x_i) \right) dx_i \\
&= -\sum_i \sum_s \alpha_i^s \int h_i^s(x_i) \ln \left(\alpha_i^s h_i^s(x_i) \right) dx_i \\
&= -\sum_i \sum_s \alpha_i^s \ln \alpha_i^s - \sum_i \sum_s \alpha_i^s \int h_i^s(x_i) \ln h_i^s(x_i) dx_i \\
&= -\sum_i \sum_s \alpha_i^s \ln \alpha_i^s + \sum_i \sum_s \alpha_i^s B_i^s,
\end{aligned} \tag{4.12}$$

where B_i^s is defined by

$$B_i^s \equiv -\int h_i^s(x_i) \ln h_i^s(x_i) dx_i. \tag{4.13}$$

Using Eqs.(4.9) and (4.12), Eq.(4.2) is rewritten as

$$\mathcal{F}[\alpha] = \sum_i \sum_s \alpha_i^s (f_i^s - B_i^s) + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_i^s \alpha_j^t f_{ij}^{st} + \sum_i \sum_s \alpha_i^s \ln \alpha_i^s. \tag{4.14}$$

An updating equation for α_i^s's can be directly derived from the similarity between Eq. (4.14) and Eq. (4.5). If we equate the pairs $\alpha_i^s \leftrightarrow p_i^s$, $(f_i^s - B_i^s) \leftrightarrow f_i(x^s)$ (not $f_i^s \leftrightarrow f_i(x^s)$), and $f_{ij}^{st} \leftrightarrow f_{ij}(x^s, x^t)$, the two free energies coincide with each other. Moreover, we have the same constraint for α_i^s's as the one for p_i^s's under which Eq. (4.6) is derived from Eq. (4.5). It is $\sum_s \alpha_i^s = 1$, which is obtained from $\int p_i(x_i) dx_i = 1$ and $\int h_i^s(x_i) dx_i = 1$. Therefore, the fixed point equation for Eq. (4.5) gives the one for α_i^s as

$$\alpha_i^s \propto \exp \left[- \left((f_i^s - B_i^s) + \sum_{j\in\mathcal{N}_i} \sum_{t=1}^{S_j} f_{ij}^{st} \alpha_j^t \right) \right]. \tag{4.15}$$

For this derivation, we need also to assume that $S_i = S$ for any i. However, it is clear that the same equation can be derived for the case where S_i differs for each i.

The updating equation (4.15) has a similar form to Eq. (4.6). Under the natural correspondences $\alpha_i^s \leftrightarrow p_i^s$, $f_i^s \leftrightarrow f_i(x^s)$, and $f_{ij}^{st} \leftrightarrow f_{ij}(x^s, x^t)$, the only difference is the presence of B_i^s. If $h_i^1(x_i), \cdots, h_i^{S_i}(x_i)$ have the same size in $\mathcal{X}$, then B_i^s becomes constant

for any s. If so, it is invalidated in Eq. (4.15) and the above MF algorithm coincides with the conventional one. Therefore, B_i^s can be regarded as a compensating term for the "non-uniformity" of the discretization of the variable space $\mathcal{X}$.

4.2.2 Derivation of a new BP algorithm

For Belief Propagation, the following class of approximating distributions p's is considered.

$$p(\mathbf{x}) = \frac{\prod_{ij} p_{ij}(x_i, x_j)}{\prod_i p_i(x_i)^{z_i - 1}}, \tag{4.16}$$

where z_i is the number of neighboring sites of the i-th site; $p_i(x_i)$ and $p_{ij}(x_i, x_j)$ satisfy

$$\int p_i(x_i) dx_i = 1 \tag{4.17a}$$

$$\iint p_{ij}(x_i, x_j) dx_i dx_j = 1 \tag{4.17b}$$

$$\int p_{ij}(x_i, x_j) dx_i = p_j(x_j). \tag{4.17c}$$

This distribution class has more generality than that for MF (Eq. (4.1)), and thus the marginal distributions estimated by BP tend to be more accurate than MF.

Similarly to the MF algorithm, substituting Eq. (4.16) into Eq. (2.27), we have

$$\begin{aligned} F[p] = &\sum_i \int p_i(x_i) f_i(x_i) dx_i + \sum_{(i,j)\in\mathcal{E}} \iint p_{ij}(x_i, x_j) f_{ij}(x_i, x_j) dx_i dx_j \\ &- \sum_i (z_i - 1) \int p_i(x_i) \ln p_i(x_i) dx_i + \sum_{(i,j)\in\mathcal{E}} \iint p_{ij}(x_i, x_j) \ln p_{ij}(x_i, x_j) dx_i dx_j. \end{aligned} \tag{4.18}$$

In the conventional discrete formulation, the BP algorithm is derived as follows. We denote the discrete values that x_i takes by $[x^1, \ldots, x^S]$ and their probabilities by $[p_i^1, \ldots, p_i^S]$ (i.e., $p_i^s \equiv p(x_i = x^s)$). We also define $p_{ij}^{st} \equiv p_{ij}(x_i = x^s, x_j = x^t)$). Rewriting Eq. (4.18) with the newly defined variables, we have

$$\begin{aligned} \mathcal{F}[\mathbf{p}] = &\sum_i \sum_s p_i^s f_i(x^s) + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} p_{ij}^{st} f_{ij}(x^s, x^t) \\ &- \sum_i (z_i - 1) \sum_s p_i^s \ln p_i^s + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} p_{ij}^{st} \ln p_{ij}^{st}, \end{aligned} \tag{4.19}$$

where $\mathbf{p}$ contain all p_i^s's and p_{ij}^{st}'s. The constraints on $p_i(x_i)$ and $p_{ij}(x_i, x_j)$ reduce to

$$\sum_s p_i^s = 1 \tag{4.20a}$$

$$\sum_{s,t} p_{ij}^{st} = 1 \tag{4.20b}$$

$$\sum_s p_{ij}^{st} = p_j^t. \tag{4.20c}$$

By minimizing $\mathcal{F}[\mathbf{p}]$ under these constraints , we have the discrete BP algorithm that iteratively updates the messages m_{ij}^t according to

$$m_{ij}^t \leftarrow \sum_s \phi_i^s \psi_{ij}^{st} \prod_{k \in \mathcal{N}_i \backslash j} m_{ki}^s, \tag{4.21a}$$

where

$$\phi_i^s = \exp\left[-f_i(x^s)\right], \tag{4.21b}$$

$$\psi_{ij}^{st} = \exp\left[-f_{ij}(x^s, x^t)\right]. \tag{4.21c}$$

In our formulation, we use the same mixture of rectangular distributions for representing $p_i(x_i)$ and $p_{ij}(x_i, x_j)$. To be specific, we represent $p_i(x_i)$ and $p_{ij}(x_i, x_j)$ as

$$p_i(x_i) = \sum_{s=1}^{S_i} \alpha_i^s h_i^s(x_i), \tag{4.22a}$$

$$p_{ij}(x_i, x_j) = \sum_{s=1}^{S_i} \sum_{t=1}^{S_j} \alpha_{ij}^{st} h_i^s(x_i) h_j^t(x_j). \tag{4.22b}$$

By substituting these into Eqs.(4.17), from $\int h_i^s(x_i) dx_i = 1$ we have

$$\sum_s \alpha_i^s = 1 \tag{4.23a}$$

$$\sum_{s,t} \alpha_{ij}^{st} = 1, \tag{4.23b}$$

$$\sum_s \alpha_{ij}^{st} = \alpha_j^t. \tag{4.23c}$$

These coincide with Eqs.(4.20).

By substituting Eqs.(4.22) into $\langle E \rangle_q$ in Eq. (4.18), we have

$$\begin{aligned}\langle E \rangle_P &= \sum_i \sum_s \alpha_i^s \int f_i(x_i) h_i^s(x_i) dx_i \\ &\quad + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_{ij}^{st} \iint f_{ij}(x_i, x_j) h_i^s(x_i) h_j^t(x_j) dx_i dx_j \\ &= \sum_i \sum_s \alpha_i^s f_i^s + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_{ij}^{st} f_{ij}^{st}. \end{aligned} \tag{4.24}$$

To derive the entropy $\mathcal{H}[q]$, we calculate the entropies of q_i and q_{ij}, respectively. Using Eq.(4.22a), the (negative) entropy of q_i is written as

$$\int p_i(x_i) \ln p_i(x_i) dx_i = \sum_s \alpha_i^s \ln \alpha_i^s - \sum_s \alpha_i^s B_i^s, \tag{4.25}$$

Similarly, Using Eq.(4.22b), that of q_{ij} is written as

$$\begin{aligned} & p_{ij}(x_i, x_j) \ln p_{ij}(x_i, x_j) dx_i dx_j \\ &= \sum_{s,t} \alpha_{ij}^{st} \iint h_i^s(x_i) h_j^t(x_j) \ln \left(\sum_{s',t'} \alpha_{ij}^{s't'} h_i^{s'}(x_i) h_j^{t'}(x_j) \right) dx_i dx_j \\ &= \sum_{s,t} \alpha_{ij}^{st} \iint h_i^s(x_i) h_j^t(x_j) \ln \left(\alpha_{ij}^{st} h_i^s(x_i) h_j^t(x_j) \right) dx_i dx_j \\ &= \sum_{s,t} \alpha_{ij}^{st} \ln \alpha_{ij}^{st} - \sum_{s,t} \alpha_{ij}^{st} (B_i^s + B_j^t). \end{aligned} \tag{4.26}$$

Using these, the entire entropy $\mathcal{H}[q]$ is given as

$$\begin{aligned} \mathcal{H}[\alpha] = \sum_i (z_i - 1) \sum_s \alpha_i^s \ln \alpha_i^s - \sum_i (z_i - 1) \sum_s \alpha_i^s B_i^s \\ - \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_{st}^{ij} \ln \alpha_{st}^{ij} + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_{ij}^{st} (B_i^s + B_j^t). \end{aligned} \tag{4.27}$$

Then, using Eq. (4.24) and Eq. (4.27), $\mathcal{F}[q]$ is rewritten as

$$\begin{aligned} \mathcal{F}_{\mathrm{BP}}[\alpha] = \sum_i \sum_s \alpha_i^s (f_i^s + (z_i - 1) B_i^s) + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_{ij}^{st} (f_{ij}^{st} - B_i^s - B_j^t) \\ - \sum_i (z_i - 1) \sum_s \alpha_i^s \ln \alpha_i^s + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \alpha_{ij}^{st} \ln \alpha_{ij}^{st}. \end{aligned} \tag{4.28}$$

Comparing this with Eq. (4.19), it is seen that the former coincides with the latter if we equate the following four pairs: $\alpha_i^s \leftrightarrow p_i^s$, $\alpha_{ij}^{st} \leftrightarrow p_{ij}^{st}$, $f_i^s + (z_i - 1)B_i^s \leftrightarrow f_i(x^s)$, and $f_{ij}^{st} - B_i^s - B_j^t \leftrightarrow f_{ij}(x^s, x^t)$. Moreover, we have the same constraints for α_i^s and α_{ij}^{st} as those for p_i^s and p_{ij}^{st} given in Eqs.(4.20), under which the message updating rule of Eqs.(4.21) are derived from Eq. (4.19). Therefore, by performing the above four substitution on Eqs.(4.21) , we have the new message updating rule for our formulation, which is given by (the same as Eq.(4.21a))

$$m_{ij}^t \leftarrow \sum_s \phi_i^s \psi_{ij}^{st} \prod_{k \in \mathcal{N}_i \backslash j} m_{ki}^s, \tag{4.29a}$$

where ϕ_i^s and ψ_{ij}^{st} are differently calculated as

$$\phi_i^s = \exp\left[-(f_i^s + (z_i - 1)B_i^s)\right], \tag{4.29b}$$

$$\psi_{ij}^{st} = \exp\left[-(f_{ij}^{st} - B_i^s - B_j^t)\right]. \tag{4.29c}$$

From these, the mixture weights are computed as

$$\alpha_i^s \propto \phi_i^s \prod_{k \in \mathcal{N}_i} m_{ki}^s. \tag{4.30}$$

As mentioned earlier, if $h_i^1(x_i), \cdots, h_i^{S_i}(x_i)$ have the same size in $\mathcal{X}$, then B_i^s becomes constant. If so, all the terms associated with B_i^s are invalidated in the above updating equations and then the above BP algorithm coincides with the conventional one. Therefore, similarly to MF, B_i^s can be regarded as a compensating factor for the non-uniformity of the discretization.

4.3 Dynamic discretization of the variable space

4.3.1 Usefulness of non-uniform discretization

The new MF and BP algorithms can deal with non-uniformly discretized variable space. By densely discretizing important portion of the space and sparsely discretizing the rest and then using these algorithms, we will be able to achieve higher balance between computational efficiency and accuracy of marginal distribution estimates. When the variable

space is of two or higher dimensions, this effect will be significant; it is particularly so for the BP algorithms, in which computational cost is mainly determined by the number of labels. (When the dimensionality of the variable space is D and the number of labels is L, the computational cost of BP is proportional to L^{2D}.)

The next question is how to obtain such an effective discretization of the variable space. If we have a prior knowledge about where is more important in the variable space, it will be possible to use it to obtain a good discretization. For the case where no such knowledge is available, we present a method for dynamically discretizing the variable space to have an effective discretization.

4.3.2 Coarse-to-fine block subdivision

We assume here that behind the estimation of the marginal distribution, there is a motivation to accurately know its shape around its maximum, e.g., to determine the position of its maximum as accurately as possible. Then, this will be made possible by more densely discretizing the space around the maximum of the marginal distribution.

As it is in general impossible to know the maximum of the marginal distribution beforehand, we consider dynamically dividing the variable space, as shown in Fig. 4.1. We start with initial coarse discretization of the variable space, that is, the variable space is divided into a small number of blocks. The rectangular function whose support is each block composes the mixture distribution approximating the true marginal distribution. For this discretization, the MF or BP algorithm is run for a certain iterations. Then, for each site (i), identifying the block (s) whose mixture weight α_i^s is the largest, we divide this block into a number of subblocks. (Multiple blocks with the largest weights may be divided simultaneously.) Then, integrating these new blocks with the blocks that are not divided, we consider a new mixture of rectangular functions whose supports are given by them. We repeatedly perform these three procedures for a desired number of iterations: updating the mixture weights for the current discretization by our MF or BP algorithm, identifying the block(s) with the largest weight(s), and dividing them into subblocks to obtain a new discretization.

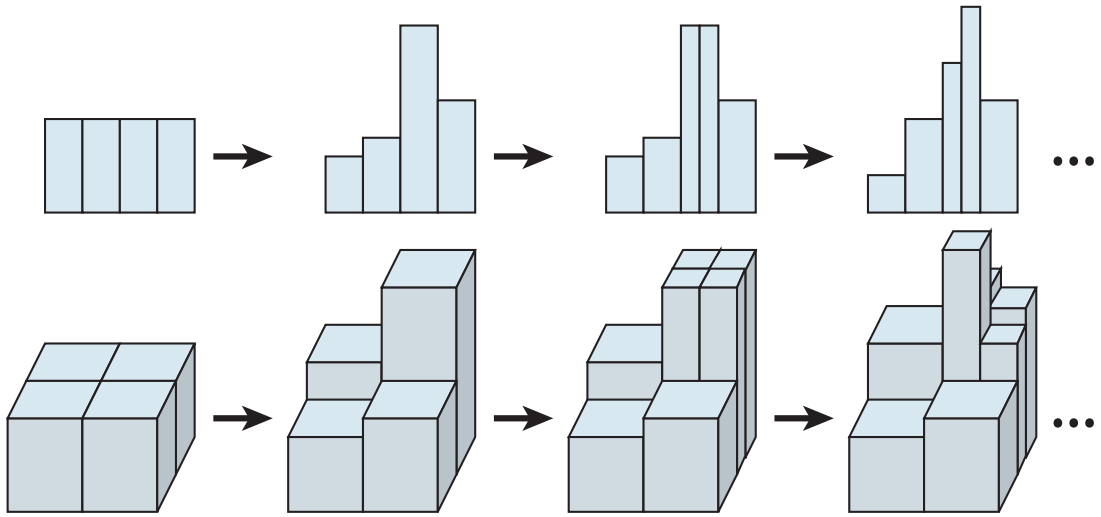

Figure 4.1: Dynamic discretization of the variable space. Each block indicates the support of a rectangular distribution composing the mixture approximating the true marginal distribution.The block having the largest weight is divided into subblocks.

4.3.3 Dividing a rectangular distribution

The subdivision of a block means dividing the corresponding rectangular distribution into multiple rectangular distributions, as shown in Fig. 4.1. Thus, the mixture of rectangular distributions after the subdivision has a different representation from the one before it. Corresponding to this representation change, we need to update α_i^s's in MF and m_{ij}^t's in BP. The principle of updating these parameters is that the mixtures before and after the subdivision should be the same distribution regardless of their difference in representation. Based on this principle, these parameters before the subdivision are processed and transferred to those after the subdivision. Different procedures are necessary for MF and BP.

The procedure for MF is as follows. Suppose that a rectangular distribution is divided into K rectangular distributions of an identical size. Following the above principle, the mixture weights of the new distributions are given by the weight of the original distribution divided by K. This ensures that the mixtures before and after the subdivision have the same shape. Suppose, for example, that the s-th block of the i-th site is divided into

two blocks. Denoting S_i pre-division weights by $[\alpha_i^1, \ldots, \alpha_i^{s-1}, \alpha_i^s, \alpha_i^{s+1}, \ldots, \alpha_i^{S_i}]$, the new weights are of $S_i + 1$ long and is given by $[\alpha_i^1, \ldots, \alpha_i^{s-1}, \alpha_i^s/2, \alpha_i^s/2, \alpha_i^{s+1}, \ldots, \alpha_i^{S_i}]$.

The procedure for BP is as follows. Suppose that we want to perform the subdivision at the i-th site. To do this, we first compute α_i^s's based on Eq. (4.30) by using all the messages passed to this site, i.e., $\{m_{ki}^s \mid k \in \mathcal{N}_i\}$. Using these weights, we then perform the block subdivision of the variable space as described above. In the case of MF, the mixture weights $[\alpha_i^1, \ldots, \alpha_i^{S_i}]$ are manipulated so as to reflect the subdivision. We apply the same manipulation to $[m_{ki}^1, \ldots, m_{ki}^{S_i}]$ for each $k \in \mathcal{N}_i$.

4.4 Experimental results

To examine the effectiveness of the proposed methods, we conducted several experiments.

4.4.1 Effect of non-uniform discretization on marginal distribution estimates

To compare the behaviours of the conventional and proposed algorithms when the variable space is non-uniformly discretized, we consider a simple Gaussian MRF for which the exact marginal distributions can be analytically obtained. To be specific, we consider a MRF model defined on a 5×5 grid graph that has the following energy:

$$E(\mathbf{x}) = \sum_i x_i^2 + \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2. \tag{4.31}$$

Clearly, its marginal distributions are Gaussian distributions having zero mean.

For this MRF model, we divide the variable space in an asymmetric way with respect to the origin $x = 0$ of the space. To be specific, considering only the range of $[-2, 2]$, we discretize its negative part $[-2, 0]$ into 64 blocks and the positive one $[0, 2]$ into 16 blocks. Thus, the continuous MRF is converted into a discrete MRF with 80 labels in total.

Then we apply the conventional MF and BP algorithms and the proposed MF and BP algorithms to this discrete MRF. Figs. 4.2 and 4.3 show the results of the MF and BP

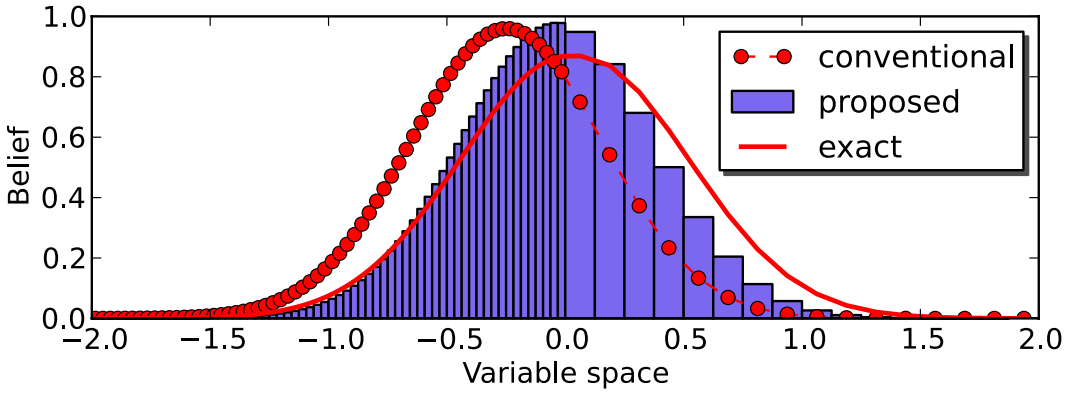

Figure 4.2: The results of the conventional and proposed MF algorithms. The red dots indicate the marginal distribution estimate by the conventional MF; the blue histogram indicates those by the proposed MF; the continuous red curve indicates the exact marginal distribution.

algorithms, respectively. They show the estimates of the marginal distribution at the site in the upper-left corner of the 5×5 grid graph. The estimates by the conventional algorithms are shown by red dots; those by the proposed algorithms are shown by blue histograms; the exact marginal distributions are shown by a continuous red curve. In the conventional algorithms, a marginal distribution is represented as a discrete distribution, i.e., $[p_i^1, \ldots, p_i^S]$. In the plots, to enable direct comparison with distributions in the continuous domain, its scale (i.e., the heights of the red dots) is appropriately adjusted. In the proposed algorithms, the marginal distributions are represented as the mixtures of rectangular distributions, which are shown in the plots.

It is seen from Figs.4.2 and 4.3 that the estimated marginal distributions by the conventional MF and BP algorithms both have bias; their means deviate from the true mean (i.e., $x = 0$) toward the side of $x < 0$. This is because of the asymmetric discretization; the energy tends to have a lower value when the marginal distribution estimates are in the side of denser discretization. On the other hand, the proposed MF and BP algorithms both yield more accurate estimates. The result of MF still has a bias but it is much smaller than the conventional one. The result of BP is even more accurate. Although there appears to exist small bias in the variances of the marginal distribution estimates, this is a fundamental limitation of these algorithms; even in the case of symmetric discretization,

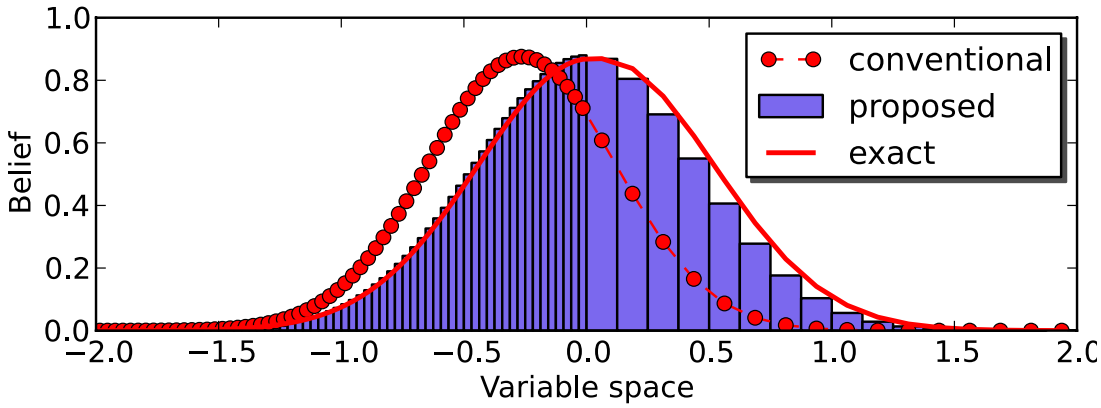

Figure 4.3: The results of the conventional and proposed BP algorithms. Legends are the same as Fig. 4.2.

the MF and BP algorithms cannot estimate the exact value of the variance.

4.4.2 Stereo matching

We applied the proposed dynamic discretization method to stereo matching and examined its effectiveness. To generate an energy function, we used the Middlebury MRF library [81]. We set $|L| = 128$, $\lambda = 2$, `smoothmax`=20, and `truncated` $= 2$. We multiply the values of the data and smoothness terms given by the library by $1/10$, as otherwise, the marginal distributions will have very sharp peaks, which is not fit for the purpose of this experiment.

The dynamic discretization method is applied to the data as follows. Initially dividing the variable space into eight blocks of an identical width, we iterate the following three steps for eight times: performing the MF or BP algorithm, identifying the block of the largest weight, and dividing the block into two blocks. At each of the eight iterations, the MF or BP update is iterated for 100 times. In the experiment, we set the lower bound of the block size to be 1 for ease of implementation. Thus, if the block with the largest weight has reached this lower bound, we divide the one with the second largest weight. If it has reached the bound, then we divide the next largest one, and so on.

The above recursive subdivision increases the number of blocks from initial eight to

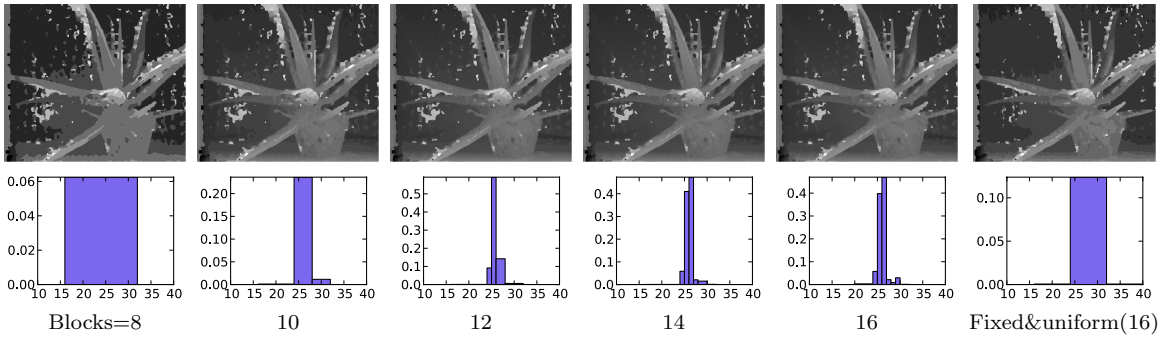

Figure 4.4: Results for *Aloe* of the MF algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

sixteen $(= 8 + 8)$. Figs.4.4 and 4.5 show the initial, intermediate, and final results when the number of blocks is 8, 10, 12, 14, and 16, for the *Aloe* dataset (641×555 pixels). For the sake of comparison, each figure also shows the result obtained with a fixed, uniform discretization; it is obtained by our MF or BP algorithm after 1000 iterations, when the variable space of the range $[0, 128]$ is divided into 16 blocks.

For both results, it is seen that the mixture distribution depicts the marginal distribution in a finer way with the increasing number of blocks. Note that the horizontal axes correspond to a portion of the full range $[0, 128]$. (The block sizes in the case of eight and sixteen block divisions are $128/8 = 16$ and $128/16 = 8$, respectively.) As compared with the mixture distributions of the fixed discretization, those of the dynamic discretization draw much finer details not only at the same number of blocks (i.e., 16) but even at the smaller number of blocks. As a result, the maxima of the marginal distributions can be determined much more accurately, and thus the dynamic discretization yields smoother disparity maps. Clearly, suffering from the insufficient number of divisions, the disparity maps for the fixed discretization are not smooth.

We show here additional results of stereo matching obtained by the proposed method. Choosing three images from the Middlebury MRF library, *Cloth1*, *Rocks1*, and *Flowerpots*, we applied the proposed method to them in a similar manner to *Aloe* shown in Figs.4.4 and 4.5. Fig. 4.6 shows their input images (including *Aloe*) along with their ground truths; the results obtained by the MAP inference (the α-expansion algorithm [81]) are also shown

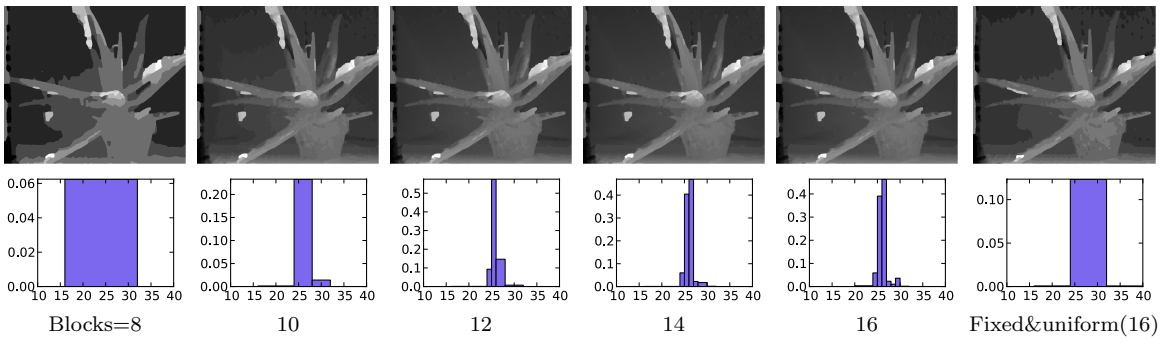

Figure 4.5: Results for *Aloe* of the BP algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

for comparison with our results.

Figs.4.7–4.12 show the results formatted in the same way as *Aloe* in our main paper. Similar to *Aloe*, it is seen that the mixture densities depict the marginal densities in a finer way with the increasing number of blocks; as compared with the results of the fixed discretization, those of the dynamic discretization draw much finer details even for smaller number of blocks. It is seen from the estimated disparity maps that those of the dynamic discretization tend to be smoother than the fixed discretization, which well agrees with the observation on the estimated marginal densities.

4.5 Summary

We have described a novel formulation of continuous-discrete conversion for the inference of marginal distributions based on MRF models. In the formulation, the marginal distributions are estimated in the continuous domain by approximating them with mixtures of rectangular distributions. Based on this formulation, we have derived the MF and BP algorithms, which can correctly deal with the non-uniform discretizaion of variable space. We have also shown the method for dynamically discretizing the variable space in a coarse-to-fine manner in the course of the computation. This enables to improve the accuracy of marginal distribution estimates without sacrificing computational efficiency. We have shown several experimental results proving the effectiveness of our approach.

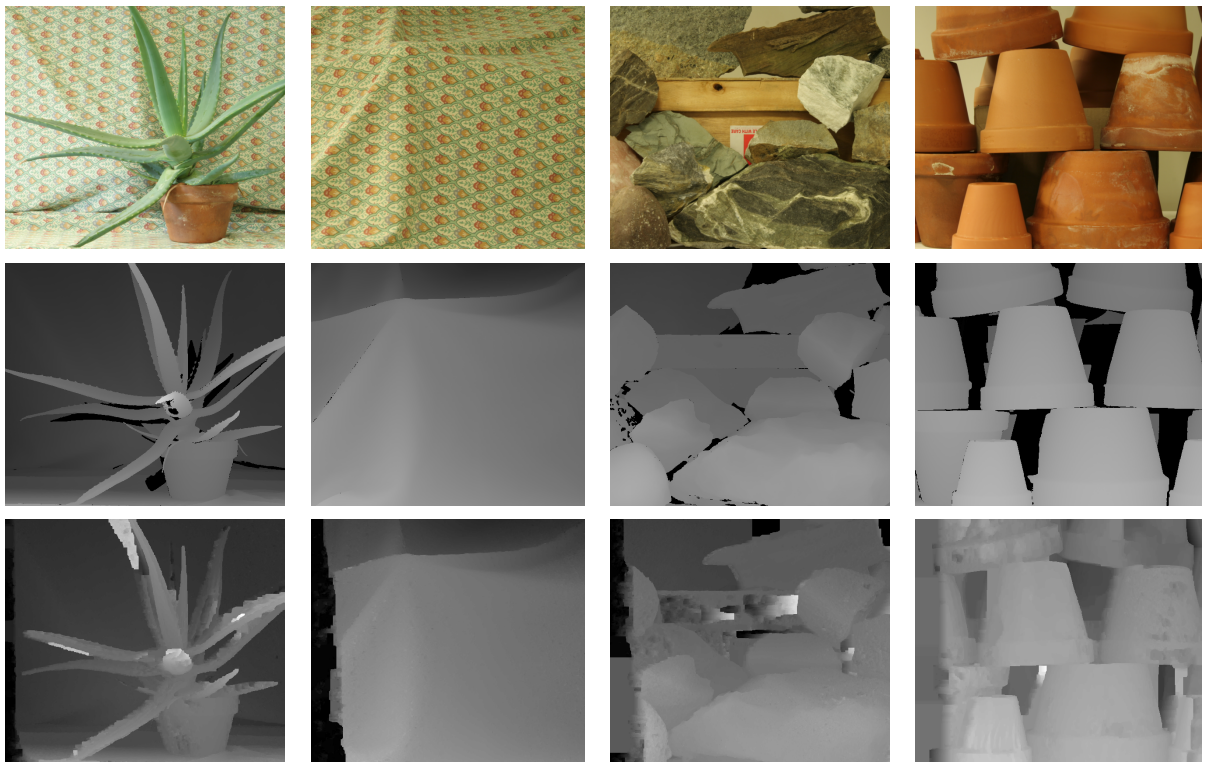

Figure 4.6: Four datasets used in our experiments: *Aloe*, *Cloth1*, *Rocks1*, and *Flowerpots*. Upper row: Input left images. Middle row: Ground truths. Lower row: Disparity maps estimated by the α-expansion algorithm.

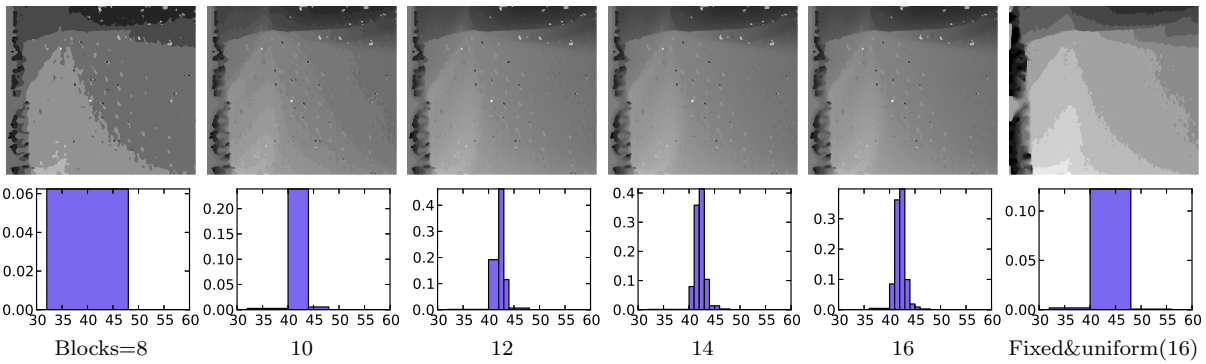

Figure 4.7: Results for *Cloth1* of the MF algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

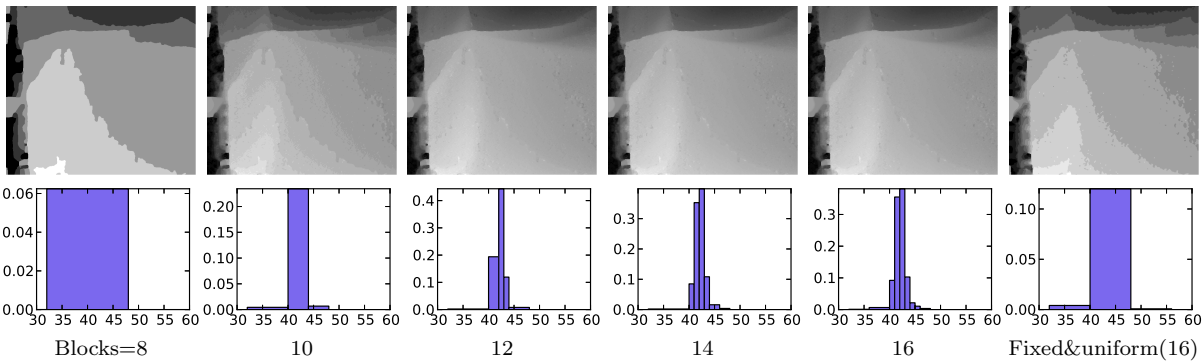

Figure 4.8: Results for *Cloth1* of the BP algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

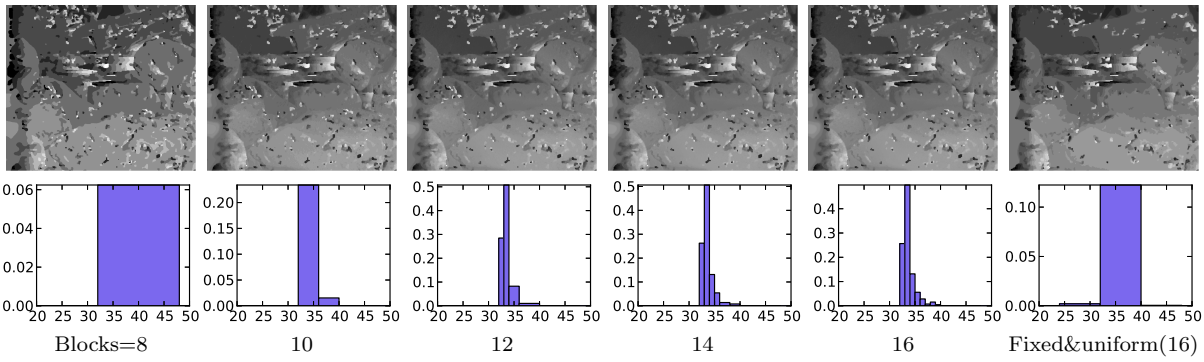

Figure 4.9: Results for *Rocks1* of the MF algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

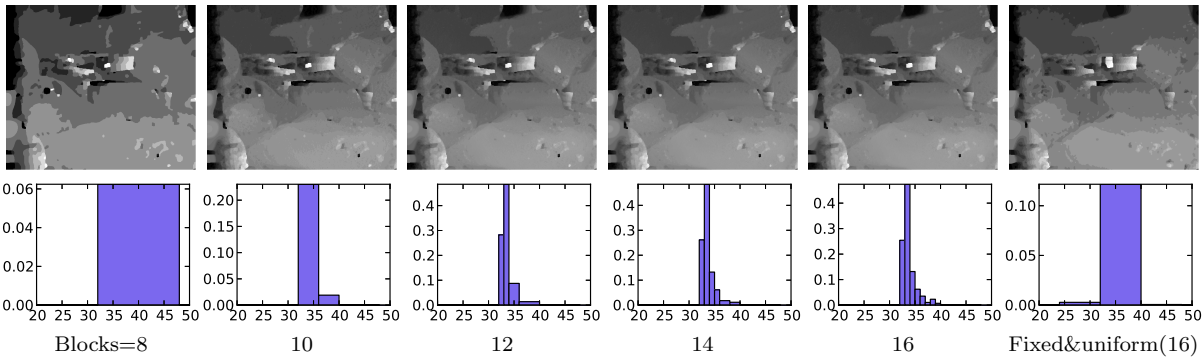

Figure 4.10: Results for *Rocks1* of the BP algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

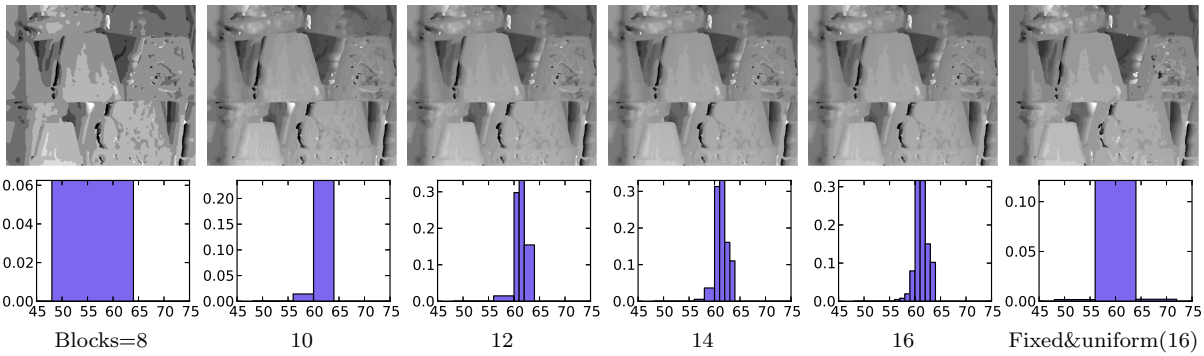

Figure 4.11: Results for *Flowerpots* of the MF algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

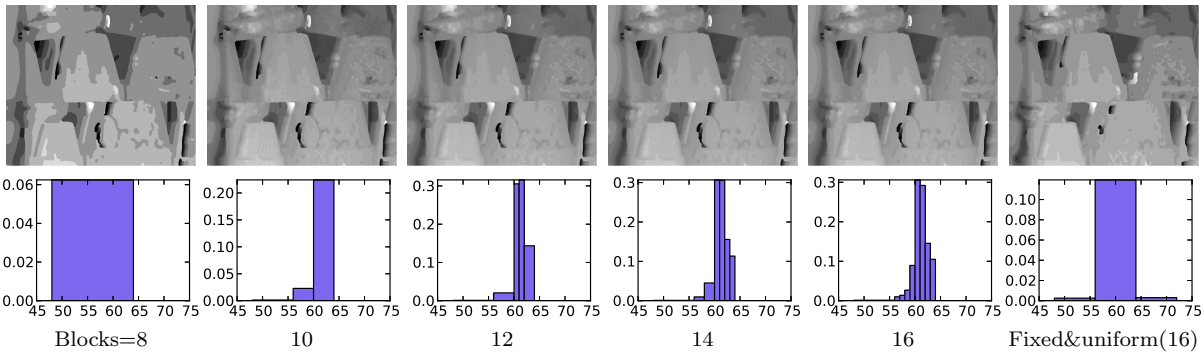

Figure 4.12: Results for *Flowerpots* of the BP algorithm with the dynamic discretization. Upper row: Disparity maps. Lower row: The mixture of rectangular densities approximating the marginal density at the site of the image pixel $(100, 100)$.

Chapter 5

Transformation of Markov random fields for marginal distribution estimation

This chapter presents a generic method for transforming MRFs for the marginal inference problem. Its major application is to downsize MRFs to speed up the computation. Unlike the MAP inference, there are only classical algorithms for the marginal inference problem such as BP etc. that require large computational cost. Although downsizing MRFs should directly reduce the computational cost, there is no systematic way of doing this, since it is unclear how to obtain the MRF energy for the downsized MRFs and also how to translate the estimates of their marginal distributions to those of the original MRFs.

The proposed method resolves these issues by a novel probabilistic formulation of MRF transformation. The key idea is to represent the joint distribution of an MRF with that of the transformed one, in which the variables of the latter are treated as latent variables. We also show that the proposed method can be applied to discretization of variable space of continuous MRFs and can be used with Markov chain Monte Carlo methods. The experimental results demonstrate the effectiveness of the proposed method.

5.1 Introduction

Markov Random Fields (MRFs) have been used for a wide range of problems in computer vision, such as optical flow estimation [19, 95, 93], image restoration [65], bundle adjustment [13, 80], object segmentation [34, 43] etc. There are two types of inference problems for MRFs. One is the MAP (Maximum a Posteriori) inference and the other is the marginal inference problem. In this study we consider the latter, which is to estimate the marginal distributions of MRF variables.

As for the MAP inference problem, there exists many sophisticated algorithms such as sequential tree-reweighted message passing (TRW-S) [40] and FastPD [42]. On the other hand, there are only classical methods for the marginal inference problem, such as mean field (MF) approximation and belief propagation (BP), which usually require a large computational cost. The marginal inference problem is nevertheless important, as it needs to be solved for MPM (maximum posterior marginal) inference [49, 38, 43], learning parameters of conditional random fields (CRFs) [78], and Boltzmann machines [68, 18].

The goal of this study is to provide methods for solving the marginal inference problem more efficiently. As for the MAP inference, a mainstream approach to reduce computational cost is to transform an MRF into a smaller, simpler one. The energy function of the MRF is transformed accordingly and is minimized to find the MAP solution. This approach has been successful in practice, resulting in a number of efficient algorithms. However, the same approach cannot be directly used for the marginal inference problem. In this problem, we are interested in the probabilistic structure (given by the Boltzmann distribution) of the MRF, which needs to be preserved as much as possible before and after transforming the MRF. Otherwise, there is no guarantee that the estimates of the marginal distributions obtained for the transformed MRF well approximate those of the original MRF. Furthermore, it is even unclear how the estimates of the marginal distributions of the transformed MRF can be translated to those of the original MRF. Suppose an image segmentation problem for example. How can we obtain pixel-level marginal distributions from the estimates of the marginal distributions at superpixels? Note that these are not the case with the MAP inference, as it is basically point estimation that can

be performed using the energy function alone.

To deal with these difficulties, we propose a novel generic method for transforming MRFs. The key idea is to use the variables of the transformed MRF as latent variables and then represent the joint distribution of the target MRF with them. To be specific, the representation consists of a conditional distribution of the original variables conditioned on the latent variables and their joint distribution. The former conditional distribution is determined by the selected MRF transformation. This formulation enables the direct computation of the energy function of the transformed MRF, which we call the *augmented energy*; this new energy gives the joint distribution of the transformed MRF as its Boltzmann distribution. Then, the marginal distributions of the transformed MRF are estimated from this joint distribution using any regular algorithm such as BP etc. Finally, the marginal distributions of the original MRF are directly calculated from them. This method is based on the variational principle and has a firm theoretical foundation.

This chapter is organized as follows. Section 5.2 briefly summarizes the related work of our study. After that, we present our generic method for MRF transformations in Section 5.3. We then show three practical applications of the proposed method in Section 5.4, which are i) discretizing variable space of continuous MRFs, ii) grouping discrete labels of MRFs to reduce the number of labels, and iii) coarse graining of MRFs by grouping multiple sites. In Section 5.5, we show how some of these MRF transformations are combined to perform coarse-to-fine inference, and also how our MRF transformation approach is applied to Markov chain Monte Carlo methods. Section 5.6 presents experimental results of our proposed method. Section 5.7 concludes this chapter.

5.2 Related work

5.2.1 Discretization of continuous MRF

Continuous MRFs whose site variables are continuous have only a limited applicability, as the marginal distributions of their variables need to be represented by a limited set of pdfs (e.g., Gaussian distribution). As there is no such limitation for discrete MRFs,

it is quite common to formulate problems in discrete domain, even if they are more natural to formulate in continuous domain. However, as is pointed out in Chapter 4, a naive discretization of variable space can cause a problem; the estimates can have errors, when the discretization is non-uniform. They extend MF and BP algorithms to be able to properly deal with this. In the present study, we reformulate the discretization as MRF transformation. This enables to deal with a wider class of algorithms, which contains practically any algorithm derived by the variational-principle such as TRW and generalized BP, and also higher-order MRFs [65, 34], both of which cannot be dealt with by their approach.

5.2.2 Grouping of discrete labels

The number of labels in discrete MRFs directly affects computational cost. For example, in the case of second-order MRFs, the complexity of BP per one iteration is proportional to $O(KL^2)$, where K is the number of neighboring sites and L is the number of labels. Thus, if we can reduce the number of labels, so does the computational cost. The problem is how we can reduce them while minimizing the loss of accuracy. As far as the MAP inference is concerned, there exist some related studies. Veksler [89] and Wang et al. [92] both proposed heuristic algorithms for reducing the search space of variables for the problem of stereo matching. Yang et. al. [96] also proposed a sophisticated BP algorithm that makes the computational cost independent of L by selecting a few labels having small data cost. However, to the authors' knowledge, there is no study of reducing the number of labels for the marginal inference problem. The above methods cannot be directly applied to the marginal inference problem.

5.2.3 Coarse-graining of MRFs

As computational cost also depends on the number of sites and edges between them, it is also effective to apply coarse-graining to MRFs, i.e., transforming their graphs into smaller ones in such a way that a number of connected sites are grouped into a single site [23] . As with the label grouping mentioned above, existing studies are limited for

the MAP inference. They are targeted at specific problems such as stereo matching [19, 46, 96] and object segmentation [33]. Although Conejo et. al. [11] proposed a general method for speeding-up MRF optimization by using the coarse-graining and the label pruning methods, their method is only targeted at the MAP inference. As for the marginal inference problem, the only study the authors are aware of is that of Ferriera et al. [20], which considers only Gaussian MRFs, though. There are a few difficulties with using coarse-graining of MRFs for the marginal inference problem. One is how the marginal distributions of the original MRF can be obtained from the estimates of those of a coarse MRF. Another is how the joint distribution (or the energy function) of the coarse MRF can be obtained.

5.3 General-purpose method for transformation of MRFs

This section presents a general-purpose method for transforming MRFs. Its applications to specific problems will be presented in Section 5.4.

Note that for brevity, we focus on pairwise MRFs described in Section 2.4. It is however that our method is applicable to any graphical models including directed models.

5.3.1 Minimization of free energy

As we have mentioned in Section 2.6, a variety of algorithms for the estimation of marginal distribution, such as MF, BP, and TRW, can be derived by the same procedure, in which a free energy is minimized based on the variational principle. For consistency, we summarize Section 2.6 by using another notation.

We suppose that the probability distribution of a MRF $\mathcal{G}$ is given by

$$q_0(\mathbf{x}) = \frac{1}{Z_0} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c), \tag{5.1}$$

where Z_0 is a normalization constant called a partition function, ϕ_c is the function of the factor c, and $\mathbf{x}_c$ is the site variables included in c. Letting $f_c(\mathbf{x}_c)$ be the negative logarithm of $\phi_c(\mathbf{x}_c)$, i.e., $f_c(\mathbf{x}_c) = -\ln \phi_c(\mathbf{x}_c)$, we may rewrite p_0 into

$$q_0(\mathbf{x}) = \frac{1}{Z_0} \exp(-E_0(\mathbf{x})), \tag{5.2}$$

$$E_0(\mathbf{x}) = \sum_{c \in \mathcal{C}} f_c(\mathbf{x}_c). \tag{5.3}$$

As it is generally intractable to directly compute the marginal distributions of the site variables using p_0 defined as above, an arbitrary distribution $q_0(\mathbf{x})$ is introduced that approximates $p_0(\mathbf{x})$, using which the marginal distributions are approximately computed.

The distribution $q_0(\mathbf{x})$ has a certain degree of freedom, within which we search for $q_0(\mathbf{x})$ the best approximating $p_0(\mathbf{x})$. This is done by minimizing the KL distance between the two:

$$\mathcal{D}[p_0 \| q_0] = \sum_{\mathbf{x}} p_0(\mathbf{x}) \ln \frac{p_0(\mathbf{x})}{q_0(\mathbf{x})}. \tag{5.4}$$

The substitution of Eq. (5.2) into Eq. (5.4) yields

$$\mathcal{D}[p_0 \| q_0] = \langle E_0(\mathbf{x}) \rangle_{p_0} - \mathcal{H}[p_0] + \ln Z_0, \tag{5.5}$$

where $\langle E_0(\mathbf{x}) \rangle_{p_0} = \sum_{\mathbf{x}} E_0(\mathbf{x}) p_0(\mathbf{x})$ is the expectation of the energy $E_0(\mathbf{x})$ with respect to $p_0(\mathbf{x})$, and $\mathcal{H}[p_0] = -\sum_{\mathbf{x}} p_0(\mathbf{x}) \ln p_0(\mathbf{x})$ is the entropy of $p_0(\mathbf{x})$. As the third term of Eq. (5.5) is independent of $p_0(\mathbf{x})$, the minimization of Eq. (5.5) is equivalent to that of the following *free energy*:

$$\mathcal{F}[p_0] = \langle E_0(\mathbf{x}) \rangle_{p_0} - \mathcal{H}[p_0]. \tag{5.6}$$

Many algorithms including MF, BP, and TRW are derived by minimizing this free energy for some selected class of p_0. For example, the generalized BP algorithm is derived when p_0 is chosen as

$$p_0(\mathbf{x}) = \frac{\prod_{c \in \mathcal{C}} p_c(\mathbf{x}_c)}{\prod_i p_i(x_i)^{z_i - 1}}, \tag{5.7}$$

where z_i is the number of factors that include the i-th site.

5.3.2 MRF transformation

We now present our method for transforming MRFs. As we have described in Section 2.6, A variety of algorithms for the estimation of marginal distribution, such as MF, BP, and TRW, can be derived by the same procedure, in which a free energy is minimized based on the variational principle.

It is often the case that depending on the structure of MRFs, the algorithms of MF, BP etc. are impossible to derive, or the derived ones are computationally costly. To cope with such difficulties, we consider transforming the MRF and its associated objective function $\mathcal{F}[p_0]$ into another one, for which the resulting minimization is easier to perform.

Toward this end, introducing a new variable $\mathbf{z}_1$, we consider an approximate distribution $p_0(\mathbf{x})$ defined in the form of

$$p_0(\mathbf{x}) = \sum_{\mathbf{z}_1} p_{0,1}(\mathbf{x}|\mathbf{z}_1) p_1(\mathbf{z}_1), \tag{5.8}$$

where $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$ is a conditional distribution that we arbitrarily choose for our purpose and $p_1(\mathbf{z}_1)$ is a unknown distribution that we are to determine. By using Eq. (5.8) we wish to transform the optimization of $p_0(\mathbf{x})$ into that of $p_1(\mathbf{z}_1)$ that will be easier to perform. For example, it is often effective to use $\mathbf{z}_1$ having a lower-dimensionality than $\mathbf{x}$, or to use discrete $\mathbf{z}_1$ when $\mathbf{x}$ is continuous. An obvious issue is how to choose $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$. We choose it differently for different purposes, which will be described in the subsequent sections.

Using Eq. (5.8), the free energy of p_0 given in Eq. (5.6) is rewritten as follows:

$$\mathcal{F}[p_0] = \langle E_1(\mathbf{z}_1) \rangle_{p_1} - \mathcal{H}[p_1] + \langle S_1(\mathbf{x}) \rangle_{p_0(\mathbf{x})}, \tag{5.9}$$

where $E_1(\mathbf{z}_1)$ and $S_1(\mathbf{x})$ are defined as follows:

$$E_1(\mathbf{z}_1) = \sum_{\mathbf{x}} p_{0,1}(\mathbf{x}|\mathbf{z}_1) \left\{ E_0(\mathbf{x}) + \ln p_{0,1}(\mathbf{x}|\mathbf{z}_1) \right\}, \tag{5.10}$$

$$S_1(\mathbf{x}) = -\sum_{\mathbf{z}} p_{0,1}(\mathbf{z}_1|\mathbf{x}) \ln p_{0,1}(\mathbf{z}_1|\mathbf{x}). \tag{5.11}$$

In the above, we used $p_{0,1}(\mathbf{z}_1|\mathbf{x}) = p_{0,1}(\mathbf{x}|\mathbf{z}_1) p_1(\mathbf{z}_1)/p_0(\mathbf{x})$. The right hand side of Eq. (5.9) has a similar form to a free energy (defined as in Eq. (5.6) for p_0) except for the third

term. To be specific, if we neglect the third term, we may think of Eq. (5.9) as the free energy of $p_1(\mathbf{z}_1)$ for the MRF whose energy is given by Eq. (5.10).

The third term of Eq. (5.9) does vanish when a condition is met as follows.

Lemma 5.3.1. (Erasure of S_1) *Let $\delta(\mathbf{z}_1)$ be the delta function. It holds that $S_1(\mathbf{x}) = 0$ if there exists a unique mapping function $\zeta_1 : \mathcal{X} \mapsto \mathcal{Z}_1$ that satisfies*

$$p_{0,1}(\mathbf{z}_1|\mathbf{x}) = \delta(\zeta_1(\mathbf{x}) - \mathbf{z}_1), \tag{5.12}$$

for any $\mathbf{x} \in \mathcal{X}$ and for any distribution $p_1(\mathbf{z}_1)$.

Proof. we rewrite Eq. (5.6) by using Eq. (5.8). The first term of Eq. (5.6) is rewritten as

$$\langle E_0(\mathbf{x}) \rangle_{p_0} = \sum_{\mathbf{z}_1} p_1(\mathbf{z}_1) \sum_{\mathbf{x}} p_{0,1}(\mathbf{x}|\mathbf{z}_1) E_0(\mathbf{x}) = \left\langle \sum_{\mathbf{x}} p_{0,1}(\mathbf{x}|\mathbf{z}_1) E_0(\mathbf{x}) \right\rangle_{p_1}. \tag{5.13}$$

To rewrite the second term of Eq. (5.6), we use $p_{0,1}(\mathbf{z}_1|\mathbf{x}) = p_{0,1}(\mathbf{x}|\mathbf{z}_1)p_1(\mathbf{z}_1)/p_0(\mathbf{x})$, which can be rewritten as

$$\ln p_0(\mathbf{x}) = \ln p(\mathbf{z}_1) + \ln p_{0,1}(\mathbf{x}|\mathbf{z}_1) - \ln p_{0,1}(\mathbf{z}_1|\mathbf{x}). \tag{5.14}$$

Note that this equation holds true for any $\mathbf{z}_1 \in \mathcal{Z}_1$, where $\mathcal{Z}_1$ is an appropriately defined variable space of $\mathbf{z}_1$. Using Eq. (5.14), we can rewrite $\mathcal{H}[p_0]$ in Eq. (5.6) as

$$\begin{aligned} \mathcal{H}[p_0] = &- \sum_{\mathbf{z}_1} p_1(\mathbf{z}_1) \ln p_1(\mathbf{z}_1) - \sum_{\mathbf{z}_1} p_1(\mathbf{z}_1) \sum_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}_1) \ln p(\mathbf{x}|\mathbf{z}_1) \\ &+ \sum_{\mathbf{x}} p_0(\mathbf{x}_0) \sum_{\mathbf{z}_1} p_{0,1}(\mathbf{z}_1|\mathbf{x}) \ln p_{0,1}(\mathbf{z}_1|\mathbf{x}). \end{aligned} \tag{5.15}$$

The substitution of Eq. (5.13) and Eq. (5.15) into Eq. (5.6) yields

$$\begin{aligned} \mathcal{F}[p_0] = &\left\langle \sum_{\mathbf{x}} p_{0,1}(\mathbf{x}|\mathbf{z}_1)(E_0(\mathbf{x}) + \ln p_{0,1}(\mathbf{x}|\mathbf{z}_1)) \right\rangle_{p_1} \\ &+ \sum_{\mathbf{z}_1} p_1(\mathbf{z}_1) \ln p_1(\mathbf{z}_1) + \left\langle \sum_{\mathbf{z}_1} p_{0,1}(\mathbf{z}_1|\mathbf{x}) \ln p_{0,1}(\mathbf{z}_1|\mathbf{x}) \right\rangle_{p_0}. \end{aligned} \tag{5.16}$$

The second term of the right hand side is the entropy of p_1, which we write as $\mathcal{H}[p_1]$. Defining $E_1(\mathbf{z}_1)$ and $S_1(\mathbf{x})$ as Eq. (5.10) and Eq. (5.11), respectively, we can rewrite $\mathcal{F}[p_0]$ as

$$\mathcal{F}[p_0] = \langle E_1(\mathbf{z}_1) \rangle_{p_1} - \mathcal{H}[p_1] + \langle S_1(\mathbf{x}) \rangle_{p_0}. \tag{5.17}$$

Lemma 3.1 states that the third term vanishes when the condition given in Lemma 3.1 is met. This is self-evident from Eq. (5.11). □

Thus, under the condition of this lemma, we can regard Eq. (5.9) as the free energy of the MRF model with a new energy $E_1(\mathbf{z}_1)$. As this energy includes the original energy $E_0(\mathbf{x})$ as well as additional terms as in Eq. (5.10), we call this the *augmented energy*. The results are summarized as follows:

Theorem 5.3.2. (MRF transformation) *Suppose a MRF specified by the distribution $q_0(\mathbf{x})$. When its approximation $p_0(\mathbf{x})$ is specified by Eq. (5.8) with $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$ satisfying the condition of Lemma 5.3.1, the variational solution to the marginal inference problem with this MRF (which searches for $p_0(\mathbf{x})$ that minimizes $D[p_0\|q_0]$) reduces to that with the MRF specified by $q_1(\mathbf{z}_1)$ defined as*

$$q_1(\mathbf{z}_1) = \frac{1}{Z_1}\exp(-E_1(\mathbf{z}_1)), \tag{5.18}$$

where $E_1(\mathbf{z}_1)$ is the augmented energy defined by Eq. (5.10).

When the marginal inference problem with a MRF is intractable or computationally costly (even with the variational approach), we may transform the MRF into another one using the above method. As the transformed MRF is a regular MRF, many existing algorithms including MF, BP, and TRW can be used for its marginal inference. The outline of the proposed method is summarized as follows.

1. Choose $p_{0,1}(\mathbf{x}|\mathbf{z})$ that implements the target transformation of the MRF.
2. Compute the augmented energy $E_1(\mathbf{z}_1)$ as in Eq. (5.10).
3. Compute the marginal distributions for the transformed MRF (having $E_1(\mathbf{z}_1)$ as the energy) by using a selected algorithm (e.g., BP, TRW, etc.).

The marginal distributions of $p_0(\mathbf{x})$ may sometimes be necessary. In that case, they are to be computed from those of $p_1(\mathbf{z}_1)$. Although there is no automatic method, it will be easy to do so in some cases, as will be shown in the next section.

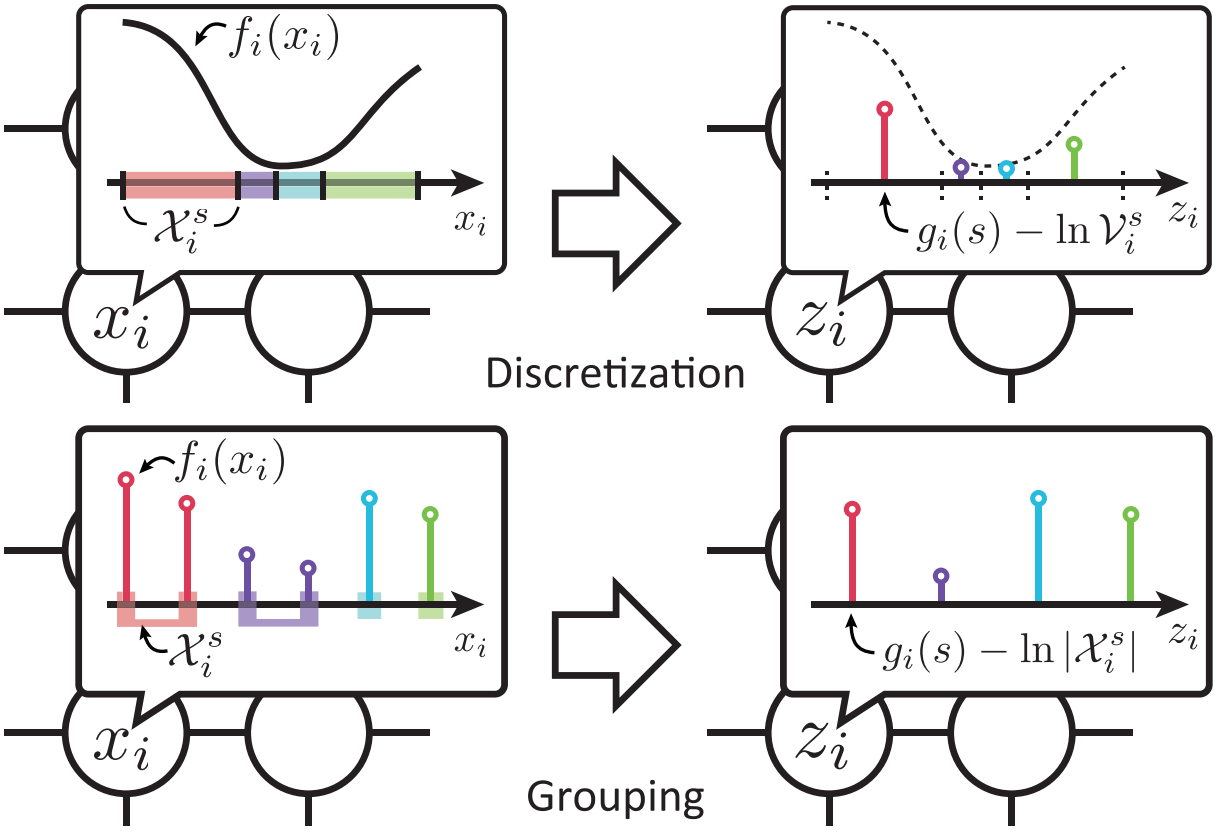

Figure 5.1: Top: Discretization of variable space. Bottom: Grouping of discrete labels. $f_i(x_i)$ is the unary term in the site i. $\mathcal{X}_i^s$ is the support of a label and is a set of labels to be grouped into a label.

5.4 Applications

This section shows how the above method for MRF transformation can be applied to real problems. We consider three problems, the discretization of variable space, the grouping of discrete labels, and the coarse graining of MRFs.

5.4.1 Discretization of variable space

As described earlier, the discrete formulation of MRFs has a wider applicability than the continuous formulation. Thus, it is a common approach to discretize the variable space of a continuous problem and then apply some algorithm designed for discrete variables.

However, as was pointed out in Chapter 4, if the discretization is non-uniform, the regular algorithms that do not consider the non-uniformity could yield inaccurate results. The method presented in the last section can derive algorithms that better handle such non-uniformity.

To do so, the method transforms the target MRF in the following way. Suppose an MRF having N sites with continuous variables $\mathbf{x} = [x_1, \ldots, x_N]$. We define $\mathbf{z}_1 = [z_1, \ldots, z_N]$, where z_i is the discrete variable of the i-th site that takes one of S_i discrete values, i.e., $z_i \in \mathcal{Z}_i \equiv \{1, \cdots, S_i\}$. We then choose $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$ of Eq. (5.8) as

$$p_{0,1}(\mathbf{x}|\mathbf{z}_1) = \prod_{i=1}^{N} p_i(x_i|z_i), \tag{5.19}$$

where $p(x_i|z_i)$ is a rectangular density such that the position of the rectangle varies depending on z_i. To be specific, when z_i takes a discrete value $s \in \mathcal{Z}_i$, it is given as

$$p_i(x_i|z_i = s) \equiv h_i^s(x_i), \tag{5.20}$$

where $h_i^s(x_i)$ is defined to be

$$h_i^s(x_i) = \begin{cases} 1/\mathcal{V}_i^s & \text{if } x_i \in \mathcal{X}_i^s \\ 0 & \text{otherwise,} \end{cases} \tag{5.21}$$

where $\mathcal{X}_i^s$ is the support of $h_i^s(x_i)$ in $\mathcal{X}$ and $\mathcal{V}_i^s$ is its volume; see Fig. 5.1.

By choosing $\mathcal{X}_i^s$ appropriately, the requirement of the proposed method is met.

Proposition 5.4.1. *If $\mathcal{X}_i^s \cap \mathcal{X}_i^t = \emptyset$ for any $s \neq t$, then $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$ of Eq. (5.19) satisfies the condition of Lemma 5.3.1.*

The augmented energy $E_1(\mathbf{z}_1)$ is calculated in a straightforward manner. Let $\mathcal{X}(\mathbf{z}_c) = \bigotimes_{i \in c} \mathcal{X}_i^{z_i}$ and $\mathcal{V}(\mathbf{z}_c)$ be the volume of $\mathcal{X}(\mathbf{z}_c)$. (Recall c is a factor of the graph.) From Eqs.(5.10), (5.19), and (5.20), $E_1(\mathbf{z}_1)$ is calculated as follows:

$$E_1(\mathbf{z}_1) = \sum_{c \in \mathcal{C}} g_c(\mathbf{z}_c) - \sum_{i=1}^{N} \ln \mathcal{V}_i^{z_i}, \tag{5.22}$$

where $g_c(\mathbf{z}_c)$ is given by

$$g_c(\mathbf{z}_c) = \frac{1}{\mathcal{V}(\mathbf{z}_c)} \sum_{\mathbf{x}_c \in \mathcal{X}(\mathbf{z}_c)} f_c(\mathbf{x}_c). \tag{5.23}$$

Note that the first term in the augmented energy is the regular energy of discrete MRFs. The second term is the additional term that accounts for the non-uniform discretization. In fact, when the discretization is uniform, $\mathcal{X}_i^{z_i}$'s will have the same shape and thus $\mathcal{V}_i^{z_i}$'s will be constant for different z_i's. Then we may neglect the term $-\ln \mathcal{V}_i^{z_i}$, resulting in the regular energy. If the discretization is non-uniform, we need to consider the second term. We can use any discrete algorithm for the marginal inference of the transformed MRF. We have only to replace the regular energy with the augmented energy derived as above.

5.4.2 Grouping of discrete labels

A similar method to the above one for dividing continuous variable space $\mathcal{X}_i$ into a discrete set of $\mathcal{X}_i^s$'s can be used to dividing discrete variable space, by which we can reduce the number of labels. To be specific, we divide the discrete variable space $\mathcal{X}_i$ into several subsets $\mathcal{X}_i^s \subset \mathcal{X}_i$ such that $\mathcal{X}_i^s \bigcap \mathcal{X}_i^t = \emptyset$; see Fig. 5.1. This grouping of the labels is represented by making a few modifications to the above continuous-discrete transformation. We replace $h_i^s(x_i)$ of Eq. (5.21) with

$$h_i^s(x_i) = \begin{cases} 1/|\mathcal{X}_i^s| & \text{if } x_i \in \mathcal{X}_i^s \\ 0 & \text{otherwise,} \end{cases} \tag{5.24}$$

where $|\mathcal{X}_i^{z_i}|$ is the number of elements in $\mathcal{X}_i^{z_i}$. Then the augmented energy will be

$$E_1(\mathbf{z}_1) = \sum_{c \in \mathcal{C}} g_c(\mathbf{z}_c) - \sum_{i=1}^{N} \ln |\mathcal{X}_i^{z_i}|, \tag{5.25}$$

where $g_c(\mathbf{z}_c)$ is equivalent to the one in Eq. (5.23) except that $\mathcal{V}(\mathbf{z}_c)$ is replaced with $|\mathcal{X}(\mathbf{z}_c)|$.

As with the above continuous-discrete transformation, the additional term $-\ln|\mathcal{X}_i^{z_i}|$ compensates for the non-uniformity of the grouping of labels. Its effect will be large when each group $\mathcal{X}_i^{z_i}$ contains a different number of labels.

5.4.3 Coarse graining of MRFs

The proposed method can also be applied to coarse graining of MRFs. After downsizing the graph of an MRF, it is then required to transform the energy $E_0(\mathbf{x})$ accordingly. Our method provides a systematic way for this transformation, which was missing in the literature.

Our method assumes that it is already determined how to modify the graph. Suppose that N sites of the graph are grouped into K blocks ($K < N$). Each block becomes a single site of the new graph. Let $\mathcal{C}(k)$ be the set of the sites grouped into the k-th block ($k = 1, \ldots, K$), such that $\mathcal{C}(k) \neq \emptyset$ for any k and also $\mathcal{C}(k) \cap \mathcal{C}(k') = \emptyset$ for any $k \neq k'$. We then consider a new variable z_k for each block k, which shares the same variable space as x_i; thus, if x_i is discrete, so is z_i.

We choose $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$ of Eq. (5.8) as

$$p_{0,1}(\mathbf{x}|\mathbf{z}_1) = \prod_{k=1}^{M} p_k(\mathbf{x}_k|z_k), \tag{5.26}$$

where $\mathbf{x}_k$ indicates a vector containing all the site variables of the k-th block, and further choose $q(\mathbf{x}_k|z_k)$ as

$$p_k(\mathbf{x}_k|z_k) = \prod_{i \in \mathcal{C}(k)} \delta(x_i - z_k), \tag{5.27}$$

where $\delta(x)$ is Dirac's delta function if the site x_i is continuous and is Kronecker's delta function if x_i is discrete. Although there are other possibilities, the above choice of $p_{0,1}(\mathbf{x}|\mathbf{z})$ is natural, as it enforces that the sites of the original MRF belonging to each group will have the same value as the corresponding site of the coarse grained MRF. It also satisfies the requirement of the proposed method.

Proposition 5.4.2. *The conditional distribution $p_{0,1}(\mathbf{x}|\mathbf{z}_1)$ defined by Eqs.(5.26) and (5.27) satisfies the condition of Lemma 5.3.1.*

The augmented energy $E_1(\mathbf{z}_1)$ can be calculated as above, but unlike earlier MRF transformations, the results will vary depending on the structure of MRFs. For lack of space, we show here only the derivation for second-order MRFs. The energy $E_0(\mathbf{x})$ of a second-order MRF is given as

$$E_0(\mathbf{x}) = \sum_i f_i(x_i) + \sum_{(i,j)\in\mathcal{E}} f(x_i, x_j), \tag{5.28}$$

where $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ are the unary and the pairwise terms, respectively; $\mathcal{E}$ is the set of edges in $\mathcal{G}$. Using Eq. (5.10) and Eqs. (5.26) - (5.28), $E_1(\mathbf{z}_1)$ is calculated as

$$E_1(\mathbf{z}_1) = \sum_k \left(\sum_{i\in\mathcal{C}(k)} f_i(z_k) + \sum_{(i,j)\in\mathrm{In}(k)} f_{ij}(z_k, z_k) \right) + \sum_{(k,l)\in\mathcal{E}_{\mathrm{Ex}}} \sum_{(i,j)\in\mathrm{Ex}(k,l)} f_{ij}(z_k, z_l), \tag{5.29}$$

where $\mathrm{In}(k)$ indicates the set of the edges contained in the k-th block (i.e., the edges between any pair of the sites in the k-th block); $\mathcal{E}_{\mathrm{Ex}}$ is the set of pairs of any neighboring blocks; $\mathrm{Ex}(k, l)$ indicates the set of the edges crossing the boundary between the neighboring (k-th and l-th) blocks.

For notational simplicity, we rewrite Eq. (5.29) as

$$E_1(\mathbf{z}_1) = \sum_k g_k(z_k) + \sum_{(k,l)\in\mathcal{E}_{\mathrm{Ex}}} g_{kl}(z_k, z_l), \tag{5.30}$$

where

$$g_k(z_k) = \sum_{i\in\mathcal{C}(k)} f_i(z_k) + \sum_{(i,j)\in\mathrm{In}(k)} f_{ij}(z_k, z_k), \tag{5.31a}$$

$$g_{kl}(z_k, z_l) = \sum_{(i,j)\in\mathrm{Ex}(k,l)} f_{ij}(z_k, z_l). \tag{5.31b}$$

The second term of (5.31a) expresses the interaction occurring within each block, and constitutes the unary term of the augmented energy. The term $g_{kl}(z_k, z_l)$ of (5.31b) expresses the interaction between the blocks and serves as the pairwise term.

These are illustrated in Fig. 5.2. As in the earlier MRF transformations, we may use any algorithm for the transformed, coarse grained MRF. One can use the derived augmented energy as if it is a regular energy of a regular MRF.

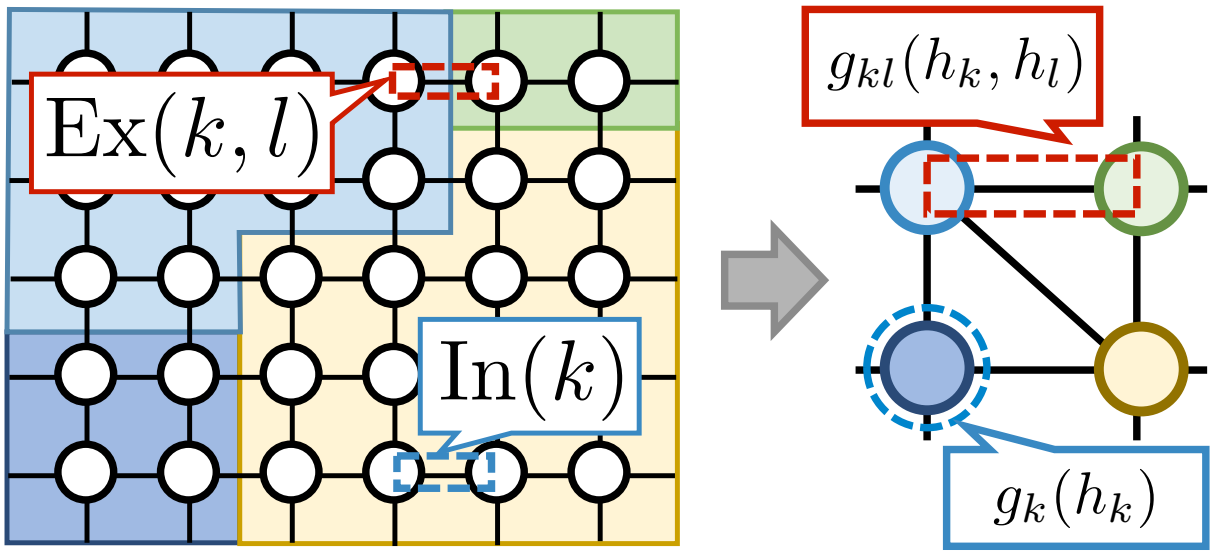

Figure 5.2: Illustration of coarse graining of an MRF graph and how the interactions between the sites in the original graph are transformed to unary and pairwise terms of the coarse-grained MRF.

Although it is omitted here, higher-order MRFs can be treated in a similar way, and the results are similar, too. For any energy term having only the site variables contained in a single block, it reduces to the form of (5.31a). For any term having site variables split to different blocks, it will reduce to the form of (5.31b).

5.5 Other applications

5.5.1 Coarse-to-fine inference

We have shown how the proposed method is used to transform MRFs for different purposes. Although it is not explicitly mentioned so, the discussion so far mostly considers MRF transformations in the direction of downsizing them. This is the case with the grouping of discrete labels and the MRF coarse graining. However, the proposed method can be used to *upsizing* MRFs, i.e., transforming MRFs into those having more sites or more labels. This is useful when we employ the coarse-to-fine strategy for the inference

with large-size MRFs.

Such coarse-to-fine inference can be implemented as follows. For a given MRF, we first transform it into a smaller one by one (or a combination) of the above techniques and perform the marginal inference with the transformed MRF. We then consider another transformation of the original MRF that has an intermediate size between the first and the original MRFs. Let the approximate distributions for the first and second MRFs be $p(\mathbf{x}) = \sum_{\mathbf{z}_1} p_{0,1}(\mathbf{x}|\mathbf{z}_1)p_1(\mathbf{z}_1)$ and $p'(\mathbf{x}) = \sum_{\mathbf{z}_2} p_{0,2}(\mathbf{x}|\mathbf{z}_2)p_2(\mathbf{z}_2)$, respectively. By appropriately designing the second transformation such that the space of $p'(\mathbf{x})$ include that of $p(\mathbf{x})$, there always exists $p_2(\mathbf{z}_2)$ such that $p(\mathbf{x}) = p'(\mathbf{x})$. Therefore, we can transfer the result obtained with the first MRF (i.e., $p_1(\mathbf{z}_1)$) to the second MRF, which gives an estimate of $p_2(\mathbf{z}_2)$. Using this as an initial value, we perform the marginal inference with the second MRF, which is expected to yield more accurate estimate of $q_0(\mathbf{x})$ due to the increased degrees of freedom. We may iterate this process until we reach the original MRF.

As good initial values are given at each step, the inference in this coarse-to-fine manner is expected to reduce the total computational cost as compared with performing the marginal inference with the original MRF once. The proposed method provides a smooth connection between two MRFs in consecutive steps. Thus, it is also possible to employ the coarse graining and the label grouping at the same time at each step.

5.5.2 Markov chain Monte Carlo (MCMC)

As mentioned above, the proposed method can be used with any algorithm derived from the variational principle, such as MF, BP, TRW etc. The method can also be used with MCMC-based algorithms such as Gibbs Sampling and Slice Sampling. It is similarly expected to reduce computational cost by downsizing MRFs.

MCMC-based methods estimate marginal distributions by generating a lot of samples from the target distribution $p_0(\mathbf{x})$. From the viewpoint of the variational principle, it is equivalent to defining the approximate density $p_0(\mathbf{x})$ as

$$p_0(\mathbf{x}) = \frac{1}{M} \sum_m \delta(\mathbf{x} - \mathbf{x}^m), \tag{5.32}$$

where M is the number of samples, $\mathbf{x}^m$ is the sample from the distribution $p_0(\mathbf{x})$, and δ is the delta function. It is easy to calculate (the estimate of) the marginal distribution of x_i from $p_0(\mathbf{x})$, which is merely the histogram of the generated samples, i.e., $(\sum_m \delta(x_i - x_i^m))/M$.

An advantage of using MCMC methods for marginal inference is that the estimates can be more accurate than those of MF, BP etc., provided that we can generate a large number of samples. However, this prohibitively increases computational cost in most cases, which is the reason why MF, BP etc. are preferred. The computational cost of MCMC methods depend on the size of the MRF, rigorously, the number of sites and either the dimensionality of the variable space in continuous cases or the number of labels in discrete cases. Therefore, it is attractive to downsize the MRF and reduce the computational cost by the proposed method.

To do so, we transform the target MRF with $p_0(\mathbf{x})$ into a smaller one with $p_1(\mathbf{z}_1)$ by one or a combination of the individual methods described in Section 4. We then apply a regular MCMC method to the transformed MRF, generating samples $z_1^1, \ldots, z_1^M$ from $p_1(\mathbf{z}_1)$. (Note that this is expected to be computationally less costly than sampling $p_0(\mathbf{x})$.) The approximate distribution of $p_0(\mathbf{x})$ is given from these samples as

$$p_0'(\mathbf{x}) = \frac{1}{M} \sum_m \sum_{\mathbf{z}_1} p_{0,1}(\mathbf{x}|\mathbf{z}_1)\delta(\mathbf{z}_1 - \mathbf{z}_1^m) = \frac{1}{M} \sum_m p_{0,1}(\mathbf{x}|\mathbf{z}_1^m). \tag{5.33}$$

It differs from the original $p_0(\mathbf{x})$ of Eq. (5.32) in that it consists of a set of *distributions* $p_{0,1}(\mathbf{x}|\mathbf{z}^m)$ not of *samples* x^m. It is nevertheless still easy to calculate the marginal densities of $\mathbf{x}$ using $p_0'(\mathbf{x})$.

A caveat is that unlike $p_0(\mathbf{x})$, $p_0'(\mathbf{x})$ will never coincide with the true density $p_0(\mathbf{x})$ even if we generate an infinite number of samples from $p_1(\mathbf{z}_1)$. Our experiments show that this might not be a serious issue in reality, although this is not a rigorous proof. Even when it is really a problem, the above approach will still be useful when used with the coarse-to-fine strategy, in which starting with a small-size MRF, we gradually increase the MRF size until reaching the original MRF. In that case, Eq. (5.33) gives a smooth connection in the transition from an MRF to another.

5.6 Experimental results

5.6.1 Discretization of variable space

If the variable space of a MRF is discretized in a uniform manner and nevertheless an ordinary algorithms is naively used for it, the results will be inaccurate. We have already pointed out in Chapter 4, in which only MF and BP are considered. The proposed method can handle any algorithm derived from the variational principle as well as methods of MCMC, yielding their extensions that can properly deal with non-uniform discretization. To demonstrate these, we show here the results for TRW and Gibbs sampling. We used OpenGM [3] for their implementation.

For the sake of comparison, we use the same experimental setting as Section 4.4. That is, we consider a simple Gaussian MRF of a 5×5 grid graph with pairwise 4-neighbor connections:

$$E(\mathbf{x}) = \sum_i x_i^2 + \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2. \tag{5.34}$$

For this MRF, we divide the variable space in an asymmetric way that the negative and positive parts in the range $[-2 : 2]$ are discretized by 64 and 16 points, respectively, as shown in Figs. 5.3 and 5.4. We then applied the ordinary and extended versions of TRW and Gibbs sampling to this MRF. For Gibbs sampling, we generated 10^7 samples, from which we calculate marginal distributions either naively (by Eq. (5.22) without the second term) or by our method. We set the burn-in period to 1000 steps.

Figures 5.3 and 5.4 shows the results. They are the estimates of the marginal distribution of the site at the upper-left corner of the 5×5 graph. The white dots are the results of the naive TRW and Gibbs sampling, whereas the blue histograms are those of their extended counterparts that are obtained by the proposed method. Note that the former are purely discrete distributions and we adjusted the vertical scale properly for comparison. The red curves are the exact distributions. (As it is a Gaussian MRF, its marginal distributions can be computed analytically in the continuous domain.) It is observed that while the distributions estimated by the naive methods have some biases, those of the extended methods do not. Although they are less significant, the variances

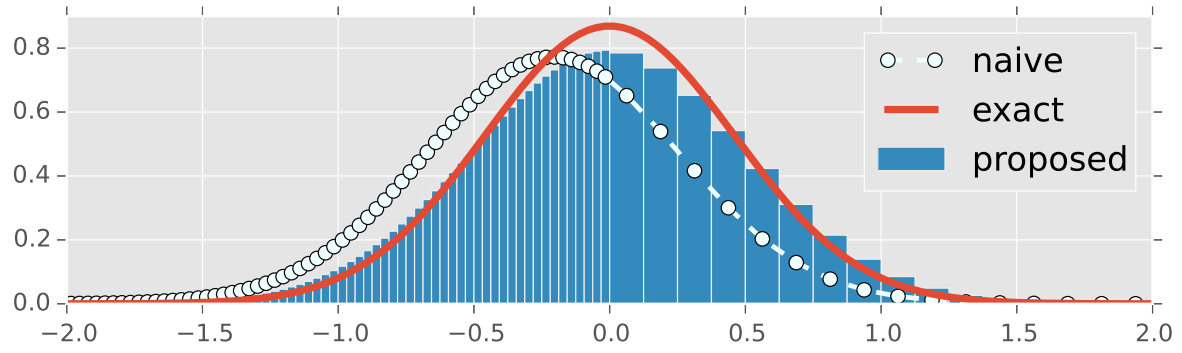

Figure 5.3: Result for non-uniformly discretized variable space with regard to Tree-reweighted Belief Progation (TRW). The marginal distribution of the site at the upper-left corner of a 5×5 grid is estimated by the naive and extended versions of TRW. See text for details.

are more accurate for the extended methods, too.

5.6.2 Downsizing CRFs

We next examine how the proposed method works for downsizing a discrete CRF. As an example problem, we chose a CRF-based formulation of semantic labeling. To be specific, we consider its learning step for determining CRF parameters, to which we applied coarse graining and label grouping. Owing to its theoretical foundation, the proposed method is expected to minimize inaccuracy caused by the downsizing. Therefore we evaluated computational efficiency as well as estimation accuracy. We used the MSRC-21 dataset [73] for the experiments. It consists of images of 320×213 pixels, each of which is given one of 21 discrete object labels. We used the "accurate ground truth" introduced by [43] for the evaluation of results.

We consider a grid CRF whose energy is given by

$$E(\mathbf{x}|\mathcal{I};\theta) = \sum_i f_i(x_i|\mathcal{I}) + \sum_{(i,j)\in\mathcal{E}} \sum_{s,t} \theta_{st}\delta(x_i - s)\delta(x_j - t), \tag{5.35}$$

where $\mathcal{I}$ is the input image; x_i is the variable of the i-th site taking one of the 21 labels; $f_i(x_i|\mathcal{I})$ is the unary term; and θ_{st} is the parameter representing the interaction between

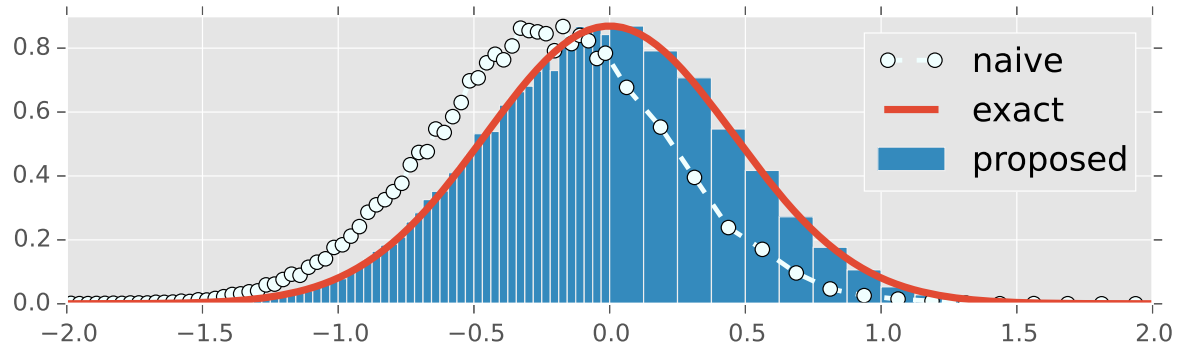

Figure 5.4: Result for non-uniformly discretized variable space with regard to Gibbs sampling. The marginal distribution of the site at the upper-left corner of a 5×5 grid is estimated by the naive and extended versions of Gibbs sampling. See text for details.

the label s and t. In the learning step, θ_{st} is determined from the training data consisting of the pairs of an image and its true label. This is performed by maximizing the likelihood calculated from the (estimates of) marginal distributions at the sites. Note that it is equivalent to determine $\theta = \{\theta_{st}\}$ by minimizing the negative log likelihood:

$$\mathcal{J}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \ln p(\mathbf{x}^m | \mathcal{I}^m; \theta). \tag{5.36}$$

Their estimation requires to use BP or similar methods, which is the bottleneck in the entire process of learning. This can be resolved or mitigated by downsizing the MRF.

We used the (stochastic) gradient descent method for the minimization. The gradient of $\mathcal{J}(\theta)$ is given as

$$\frac{\partial \mathcal{J}}{\partial \theta_{st}} = \frac{1}{M} \sum_{m} \sum_{(i,j) \in \mathcal{E}} \delta(x_i^m - s)\delta(x_j^m - t) - \frac{1}{M} \sum_{m} \sum_{(i,j) \in \mathcal{E}} p_{ij}(s, t | \mathcal{I}^m; \theta), \tag{5.37}$$

where $p_{ij}(x_i, x_j | \mathcal{I}^m; \theta)$ is the marginal distribution between the i-th and j-th sites.

grouping of discrete labels

We used the following two methods for the downsizing. The first is grouping the discrete labels, where we reduced the number of labels to K for each pixel. (We fixed it throughout

the learning.) To be specific, we selected $K-1$ labels having the smallest values of the unary terms and grouped the other labels into one label. Note that the selection was performed independently at each pixel and thus the resulting grouping may be different for different pixels.

We describe the computation of original marginal distributions. Let $p^1_{ij}(z_i, z_j|\mathcal{I}^m;\theta)$ be the marginal distribution of the transformed CRF, which is estimated by using the augmented energy of Eq. (5.22), and let $p^0_{ij}(x_i, x_j|\mathcal{I}^m;\theta)$ be the marginal distribution of the original energy function. Using Eqs.(5.19), (5.21), and (5.22), these two are related as

$$p^0_{ij}(x_i, x_j|\mathcal{I}^m;\theta) = \frac{1}{|\mathcal{X}^u_i||\mathcal{X}^v_j|} p^1_{ij}(u, v|\mathcal{I}^m;\theta), \tag{5.38}$$

where u and v on the right hand side are the labels (indeces) of the supports $\mathcal{X}^u_i$ and $\mathcal{X}^v_j$ within which x_i and x_j lie, respectively; that is, $x_i \in \mathcal{X}^u_i$ and $x_j \in \mathcal{X}^v_j$.

Coarse graining of MRFs

The second is coarse graining of the MRF, where we downsized the original grid MRF by grouping the pixels in $b \times b$ square blocks into a single *superpixel*. Note that in spite of the downsizing, we do estimate the marginal distributions of the *original* MRF. They are used to calculate the likelihood, which is to be minimized.

As with grouping of discrete labels, using Eqs.(5.26), (5.29), and (5.30), we can express p^0_{ij} with p^1_{ij}. If $(i, j) \in \mathrm{In}(k)$, p^0_{ij} can be expressed as

$$p^0_{ij}(x_i, x_j|\mathcal{I}^m;\theta) = \delta(x_i - x_j) p^1_k(x_i|\mathcal{I}^m;\theta), \tag{5.39}$$

where $p^1_k(x_i|\mathcal{I}^m;\theta)$ is the marginal distribution of the i-th site estimated from $p^1(\mathbf{z}_1)$. If $(i, j) \in \mathrm{Ex}(k, l)$, p^0_{ij} is expressed as

$$p^0_{ij}(x_i, x_j|\mathcal{I}^m;\theta) = p^1_{k,l}(x_i, x_j|\mathcal{I}^m;\theta), \tag{5.40}$$

where $p^1_{k,l}(x_i, x_j)$ is the marginal distribution estimated from $p^1(\mathbf{z}_1)$. Thus, we can regard $p^1_{k,l}$ as p^0_{ij} in this case.

Table 5.1: Quantitative results on the MSRC-21 dataset.

	time [h]	speedup	disparity	accuracy
full MRF	9.5	-	0.0	81.6
2 labels	0.89	10.6×	0.01235	77.8
3 labels	0.95	10.0×	0.00526	80.7
4 labels	1.0	9.2×	0.00496	81.3
5 labels	1.1	9.0×	0.00473	81.3
4×4 grid	0.66	14.4×	0.215	81.5
3×3 grid	1.1	8.5×	0.236	81.6
2×2 grid	2.5	3.9×	0.237	81.7

Settings

We divided the MSRC-21 dataset into 276, 59, and 256 images for training, validation, and test, which is the same as [43, 73]. We used BP [56] with the damping factor 0.5 and 50 iteration counts for estimating marginal distributions for each MRF. We multiplied the unary term of [43] by 1/10 to stabilize the computation. We employ the stochastic gradient descent (SGD) method for minimizing the negative log likelihood to determine θ_{st}'s. We set the learning rate to 1.5×10^{-5}, the batch size to 8, and the number of epochs to 5. In the testing step, we used the α-expansion [10] to obtain the MAP estimates for the MRF, which was used to measure the accuracy of the learned parameters. We used OpenGM [3] for the implementation on a PC with Intel Core i7-2600 having eight CPU cores clocked at 3.40GHz.

Experimental results

Table 5.1 shows quantitative results. The "disparity" column shows the mean differences of the parameter θ_{st} between the full MRF and its downsized versions. The "accuracy" column shows the percentage of correctly labeled pixels. The rows of "L labels" show the results of different label grouping, and those of "$b \times b$ grid" show the results of differently coarse-grained MRFs. It is observed that both methods for downsizing achieve significant

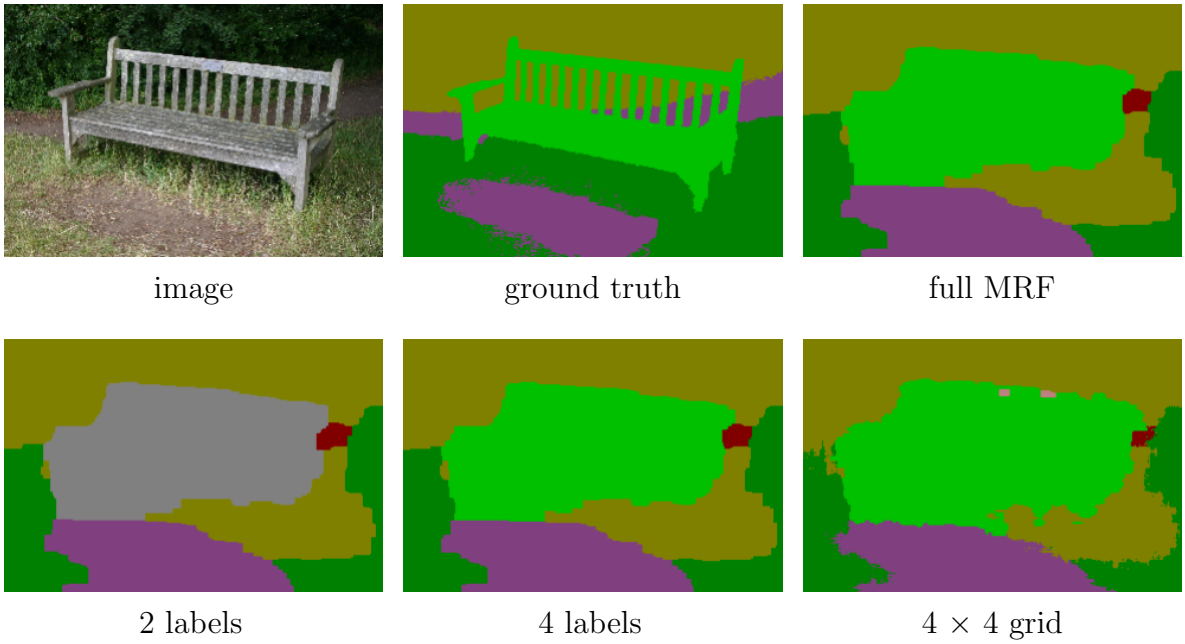

Figure 5.5: Qualitative results on the MRRC-21 dataset.

speed ups at a small expense of inaccuracy. An exception is two-label grouping, which shows considerably lower accuracy. This indicates that the reduction from 21 to only two labels is excessive. An interesting remark is that the label grouping yields much smaller disparity than the coarse graining, and nevertheless their labeling accuracy are almost the same or the latter is even slightly better. An implication of this is that label grouping is more "accurate" in the sense that it is more close to the results of full MRFs. However, there is no guarantee that full MRFs are better at learning better parameters. The coarse grained MRFs could avoid local maxima.

Figure 5.5 shows a qualitative comparison of the results. It is observed that the two-label grouping yields very inaccurate labeling and the four-label grouping and the 4×4 coarse graining both yield similar results to the original MRF. We have checked that this is the case with the other images.

5.7 Summary

We have described a novel generic formulation for transforming MRF into a smaller, simpler one. As applications of our proposed formulation, we have derived the three methods: (1) the discretization of variable space, (2) the grouping of discrete labels, and (3) coarse graining of MRFs. We have also described that these MRF transformations are combined to perform coarse-to-fine inference, and can be used with MCMC methods. Through several experiments, we have confirmed the effectiveness of the proposed method.

Chapter 6

Conclusions

We tackled the problem of optimization algorithms for Markov Random Fields. MRF is one of the most fundamental probabilistic model in the field of computer vision, and have been used to solve many problems such as image restoration, super resolution, stereo matching, and optical flow estimation. In MRFs, these inference algorithms are roughly classified into the two groups: the MAP inference and the marginal inference. In this thesis, we focused on the latter. Although there exists a number of sophisticated algorithms for the MAP inference, there are only classical methods for the marginal inference.

We conducted three studies to deal with this problem. Firstly, we described the basis of MRFs and introduced the two inference algorithms for marginal distribution, such as mean field approximation (MF) and belief propagation (BP). As our three studies use these algorithms, we also introduced the variational principle, which is a basis of MF and BP, and derived them.

The first study is for improving the accuracy of the MF approximation with the TAP equation, which was developed in the field of solid state physics. Despite many studies revealed that the TAP equation outperform the MF, it has not been so popular in other fields. This may be because of the limitation of the original TAP equation that is applicable only to binary MRFs and not to more general MRFs.

To eliminate this limitation, we first generalize the conventional TAP equation and

derive a general-purpose expression of the second-order TAP equation that can be applied to more general MRFs. As examples of its application, we then derive the specific TAP equations for binary-label MRFs, multi-label MRFs, and for Boltzmann Machines having softmax units. We show the results of several experiments with discrete multi-label MRFs for stereo matching and with DBMs for supervised learning and unsupervised learning using the MNIST and NORB datasets. They demonstrate the effectiveness of our approach.

The second study is for continuous-discrete conversion for the inference of marginal distribution. Specifically, we have proposed a novel formulation for handling such discretization in non-uniform manner. Based on this formulation, we have derived the MF and BP algorithms, which can correctly deal with the non-uniform discretization of variable space. In addition to this, we have also shown the method for dynamically discretizaing the variable space in a coarse-to-fine manner in the course of the computation. It enables to improve the accuracy of the marginal distribution without sacrificing computational efficiency. Through several experiments, we have confirmed the effectiveness of our approach.

The third study is for reducing a computational cost of marginal distribution by transforming a MRF into a smaller, simpler one. The key idea is to use the variables of the transformed MRF as latent variables and then represent the joint distribution of the target MRF with them. Using this representation, we derived the transformed MRF which is easy to compute their marginal distributions. As applications of our formulation, we proposed the three practical applications, which are discretizing variable space of continuous MRFs, grouping discrete labels of MRFs to reduce the number of labels, and coarse graining of MRFs by grouping multiple sites. We also showed how some of these transformations are combined to perform coarse-to-fine inference, and how our MRF transformation approach is applied to Markov chain Monte Carlo methods. The experimental results demonstrated the effectiveness of our formulation.

6.1 Future work

For future work, we will tackle following the remaining tasks.

Generalization of TAP equation and their applications Our generalized TAP equation can be extended to handle a wider range of probabilistic models such as Gaussian distribution and directed graphical models. We will derive the TAP equations for dealing with these models, and confirmed their effectiveness. Moreover, the variational bound of the first-order TAP equation (i.e., the mean field approximation) is well-studied, whereas the higher-order TAP equation is not studied well. We will confirm this bound and its convergence property from a viewpoint of the variational principle.

Discrete inference of Markov random fields for non-uniformly discretized variable space Currently we have performed the experiments with regard to our method with limited problems such as stereo matching. In order to claim that our method is applicable to a wide range of problems, we will perform additional experiments with problems using continuous variables, such as Structure from Motion and image reconstruction.

Transformation of Markov random fields for marginal distribution estimation As with the transformation of MRFs, we will perform further experiments to confirm that our formulation is useful for several practical applications. We consider that the transformation regarding the grouping of discrete variables is especially useful in many problems, since many applications such as semantic segmentation regard all the variables as discrete ones. Currently there exists the problem that the number of labels in discrete MRFs needs to be a small due to the computational cost. We will overcome this problem by our proposed formulation, and demonstrate its effectiveness.

Bibliography

[1] S. Alchatzidis, A. Sotiras, and N. Paragios. Efficient parallel message computation for MAP inference. In *ICCV*, 2011.

[2] S. Amari, S. Ikeda, and H. Shimokawa. Information geometry of α-projection in mean field approximation. In *Advanced Mean Field Methods: Theory and Practice*, chapter 16. The MIT Press, 2001.

[3] B. Andres, T. Beier, and J. H. Kappes. OpenGM: A C++ Library for Discrete Graphical Models. *CoRR*, abs/1206.0, 2012.

[4] A. Barbu. Learning Real-Time MRF Inference for Image Denoising. In *CVPR*, 2009.

[5] J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society. Series B(Methodological)*, 36:192–236, 1974.

[6] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.

[7] H. A. Bethe. Statistical Theory of Superlattices. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 150(871):552–575, 1935.

[8] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[9] Y. Boykov and O. Veksler. Graph Cuts in Vision and Graphics: Theories and Applications. In *Handbook of Mathematical Models in Computer Vision*, chapter 5, pages 79–96. Springer, 2006.

[10] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *PAMI*, 23(11), 2001.

[11] B. Conejo, N. Komodakis, S. Leprince, and J. P. Avouac. Speeding-up Graphical Model Optimization via a Coarse-to-fine Cascade of Pruning Classifiers. In *NIPS*, 2014.

[12] T. Cour and J. Shi. Solving Markov Random Fields with Spectral Relaxation. In *AISTATS*, 2007.

[13] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011.

[14] R. C. Dubes, A. K. Jain, S. G. Nadabar, and C. C. Chen. MRF model-based algorithms for image segmentation. In *ICPR*, volume 1, pages 808–814, 1990.

[15] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[16] G. E. Hinton. A Practical Guide to Training Restricted Boltzmann Machines. Technical report, University of Toronto, 2010.

[17] G. Elidan. Nonparanormal Belief Propagation (NPNBP). In *NIPS*, 2012.

[18] S. M. A. Eslami, N. Heess, and J. Winn. The Shape Boltzmann Machine : a Strong Model of Object Shape. In *CVPR*, 2012.

[19] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Belief Propagation for Early Vision. *IJCV*, 70(1):41–54, 2006.

[20] M. A. R. Ferreira and H. K. H. Lee. *Multiscale Modeling: A Bayesian Perspective*. Springer, 2007.

[21] A. E. Gelfand and A. F. M. Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[22] D. Geman. Random Fields and Inverse Problems in Imaging. *In Saint-Flour Lectures 1988, Lecture Notes in Mathematics*, pages 113–193, 1990.

[23] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary Detection by Constrained Optimization. *PAMI*, 12(7):609–628, 1990.

[24] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6:721–741, 1984.

[25] A. Georges and J. S. Yedidia. How to expand around mean-field theory using high-temperature expansions. *Journal of Physics A: Mathematical and General*, 24(9):2173, 1991.

[26] A. Ihler and D. McAllester. Particle Belief Propagation. In D. van Dyk and M. Welling, editors, *AISTATS*, pages 256–263, Clearwater Beach, Florida, 2009. JMLR: W&CP 5.

[27] M. Isard. Pampas: Real-Valued Graphical Models for Computer Vision. In *CVPR*, 2003.

[28] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925.

[29] Y. Kabashima and D. Saad. The TAP approach to intensive and extensive connectivity systems. In *Advanced Mean Field Methods: Theory and Practice*, chapter 5. The MIT Press, 2001.

[30] H. J. Kappen and F. B. Rodriguez. Boltzmann Machine learning using mean field theory and linear response correction. In *NIPS*, pages 280–286. The MIT Press, 1998.

[31] H. J. Kappen, F. B. Rodríguez, and F. B. Rodr'iguez. Efficient learning in Boltzmann Machines using linear response theory. *Neural Computation*, 10:1137–1156, 1997.

[32] H. J. Kappen and W. J. Wiegerinck. Mean Field Theory for Graphical Models. In *Advanced Mean Field Methods: Theory and Practice*, chapter 4. The MIT Press, 2001.

[33] T. Kim, S. Nowozin, P. Kohli, and C. D. Yoo. Variable Grouping for Energy Minimization. In *CVPR*, 2011.

[34] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *IJCV*, 82(3), 2009.

[35] P. Kohli and P. H. S. Torr. Measuring uncertainty in graph cut solutions. *Computer Vision and Image Understanding*, 112(1):30–38, 2008.

[36] D. Koller and N. Friedman. Inference as Optimization. In *Probabilistic Graphical Models: Principles and Techniques*, chapter 11, pages 381–485. The MIT Press, 2009.

[37] D. Koller and N. Friedman. MAP Inference. In *Probabilistic Graphical Models: Principles and Techniques*, chapter 13, pages 551–604. The MIT Press, 2009.

[38] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[39] D. Koller and N. Friedman. Undirected Graphical Models. In *Probabilistic Graphical Models: Principles and Techniques*, chapter 4, pages 103–156. The MIT Press, 2009.

[40] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *PAMI*, 28(10):1568–1583, 2006.

[41] N. Komodakis, N. Paragios, and G. Tziritas. MRF Energy Minimization and Beyond via Dual Decomposition. *PAMI*, 33(3):531–552, 2011.

[42] N. Komodakis and G. Tziritas. Approximate Labeling via Graph Cuts Based on Linear Programming. *PAMI*, 29(8):1436–1453, 2007.

[43] P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS*. 2011.

[44] H. Larochelle and Y. Bengio. Classification using discriminative restricted boltzmann machines. In *ICML*, 2008.

[45] Y. LeCun, F. J. Huang, and L. Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In *CVPR*, 2004.

[46] C. Lei and Y.-H. Yang. Optical Flow Estimation on Coarse-to-Fine Region-Trees using Discrete Optimization. In *ICCV*, 2009.

[47] M. A. R. Leisink and H. J. Kappen. Learning in Higher Order Boltzmann Machines using Linear Response. *Neural Networks*, 13:2000, 1999.

[48] W. Lenz. Beiträge zum Verständnis der magnetischen Eigenschaften in festen Körpern. *Physikalische Zeitschrift*, 21:613–615, 1920.

[49] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009.

[50] M. Malfait and D. Roose. Wavelet-based image denoising using a Markov random field a priori model. *IEEE Transactions on Image Processing*, 6(4):549–565, 1997.

[51] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, pages 1–7, 2002.

[52] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001.

[53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[54] T. Morita and T. Horiguchi. Exactly solvable model of a spin glass. *Solid State Communications*, 19(9):833–835, 1976.

[55] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge, MA, 2012.

[56] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

[57] A. Nemirovski and D. Yudin. On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions. *Soviet Mathematics-Doklady*, 19(2):239, 1978.

[58] H. Nishimori and K. Y. M. Wong. Statistical Mechanics of Image Restoration and Error-Correcting Codes. *Physical Review E*, 60(1):132, 1999.

[59] A. Noma, A. B. V. Graciano, R. M. Cesar Jr, L. A. Consularo, and I. Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(3):1159–1179, 2012.

[60] M. Opper and D. Saad. From Naive Mean Field Theory to the TAP Equations. In *Advanced Mean Field Methods: Theory and Practice*, chapter 2. The MIT Press, 2001.

[61] G. Parisi and M. Potters. Mean-Field Equations for Spin Models with Orthogonal Interaction Matrices. *Matrices, J. Phys. A (Math. Gen*, 28:5267, 1995.

[62] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *American Association of Artificial Intelligence National Conference on AI*, pages 133–136, 1982.

[63] T. Plefka. Convergence condition of the TAP equation for the infinite-ranged Ising spin glass model. *Journal of Physics A: Mathematical and General*, 15(6):1971, 1982.

[64] D. Rajan and S. Chaudhuri. An MRF-Based Approach to Generation of Super-Resolution Images from Blurred Observations. *Journal of Mathematical Imaging and Vision*, 16(1):5–15, 2002.

[65] S. Roth and M. J. Black. Fields of Experts. *IJCV*, 82(2):205–229, 2009.

[66] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[67] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Trans. Graph*, 23(3):309–314, 2004.

[68] R. Salakhutdinov and G. Hinton. Deep Boltzmann Machines. In *AISTATS*, 2009.

[69] R. Salakhutdinov and G. E. Hinton. Replicated Softmax: an Undirected Topic Model. In *NIPS*, 2009.

[70] R. Salakhutdinov and G. E. Hinton. A Better Way to Pretrain Deep Boltzmann Machines. In *NIPS*, 2013.

[71] Shamir M and S. H. Thouless-anderson-palmer equations for neural networks. *Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 61(2):1839–1844, 2000.

[72] D. Sherrington and S. Kirkpatrick. Solvable Model of a Spin-Glass. *Physical Review Letters*, 35:1792, 1975.

[73] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *IJCV*, 81(1), 2009.

[74] Solomon Eyal Shimony. Finding MAPs for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, 1994.

[75] N. Srivastava, R. Salakhutdinov, and G. E. Hinton. Modeling Documents with a Deep Boltzmann Machine. In *NIPS*, 2013.

[76] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric Belief Propagation. In *CVPR*, pages 605–612. IEEE Comput. Soc, 2003.

[77] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo Matching Using Belief Propagation. *PAMI*, 25(7):787–800, 2003.

[78] C. Sutton and A. McCallum. *An Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2007.

[79] R. H. Swendsen and J.-S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2), 1987.

[80] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.

[81] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, A. Agarwala, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *ECCV*, 2006.

[82] T. Tanaka. Information geometry of mean-field approximation. *Neural Computation*, 12(8):1951–1968, 2000.

[83] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, 2003.

[84] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the Sparse Derivative Prior for Super-Resolution and Image Demosaicing. In *IEEE Workshop on Statistical and Computational Theories of Vision at ICCV*, pages 900–907, 2003.

[85] D. Tarlow and R. P. Adams. Revisiting Uncertainty in Graph Cut Solutions. In *CVPR*, pages 2440–2447, 2012.

[86] D. J. Thouless, P. W. Anderson, and R. G. Palmer. Solution of solvable model of a spin glass. *Philosophical Magazine*, 35:593–601, 1977.

[87] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[88] P. H. Torr. Solving Markov Random Fields using Semi Definite Programming. In *AISTATS*, 2003.

[89] O. Veksler. Reducing Search Space for Stereo Correspondence with Graph Cuts. In *BMVC*, 2006.

[90] N. Vinod and G. E. Hinton. Implicit Mixtures of Restricted Boltzmann Machines. In *NIPS*, 2008.

[91] A. Viterbi. A personal history of the Viterbi algorithm. *IEEE Signal Processing Magazine*, 23(4):120–142, 2006.

[92] L. Wang, H. Jin, and R. Yang. Search Space Reduction for MRF Stereo. In *ECCV*, 2008.

[93] L. Wang, G. Zhao, L. Cheng, and M. Pietikainen. *Machine Learning for Vision-Based Motion Analysis*. Springer, 2011.

[94] M. Welling and Y. W. Teh. Approximate inference in Boltzmann machines. *Artificial Intelligence*, 143(1):19–50, 2003.

[95] L. Xu, J. Jia, and Y. Matsushita. Motion Detail Preserving Optical Flow Estimation. In *CVPR*, 2010.

[96] Q. Yang, L. Wang, and N. Ahuja. A constant-space belief propagation algorithm for stereo matching. In *CVPR*, 2010.

[97] M. Yasuda and K. Tanaka. TAP equation for non-negative Boltzmann machine. *Philosophical Magazine*, 92(1):192–209, 2012.

[98] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding Belief Propagation and Its Generalizations. In *Exploring artificial intelligence in the new millennium*, chapter 8, pages 239–269. Morgan Kaufmann, 2003.

[99] J. S. Yedidia and A. Georges. The fully frustrated Ising model in infinite dimensions. *Journal of Physics A: Mathematical and General*, 23(11):2165, 1990.

[100] C. Zach, T. Pock, and H. Bischof. A Duality Based Approach for Realtime TV-L1 Optical Flow. In *German Association for Pattern Recognition(DAGM)*, number 1, pages 214–223, 2007.

Acknowledgement

I am very grateful to my supervisor Professor Takayuki Okatani. I would like to thank him for supporting my study with his passion, motivation, and immense knowledge. I could not have imagined having a better advisor for my study, and I would not go to a doctoral course if I decided to join another laboratory. I would also like to thank assistant professor Kota Yamaguchi for discussing my research and giving great ideas.

I have spend very satisfying five years because of the members in my laboratory. I would like to thank everyone including (but not limited to) Dr. Mete Ozay, Eisuke Ito, Makoto Ozeki, Liu Xing, Hiroto Date, Sumadianto Eka Putra, Ken Sakurada, and Takashi Abe. Especially, I would like to thank Dr. Mete Ozay for reviewing my Ph.D. thesis and giving a lot of advices.

Finally, I would like to thank my family for their years of support, motivation, and belief.