Presented at the Workshop on Discrete Optimization in Machine Learning at NIPS 2011.

Diverse M-best Solutions in MRFs

Payman Yadollahpour
TTI Chicago
pyadolla@ttic.edu

Dhruv Batra
TTI Chicago
dbatra@ttic.edu

Gregory Shakhnarovich
TTI Chicago
greg@ttic.edu

Abstract

Current methods for computing the M most probable configurations under a probabilistic model produce solutions that tend to be very similar to the MAP solution and each other. This is often an undesirable property. In this paper we propose an algorithm for the *M-Best Mode* problem, which involves finding a diverse set of highly probable solutions under a discrete probabilistic model. Given a dissimilarity function measuring closeness of two solutions, our formulation maximizes a linear combination of the probability and dissimilarity to previous solutions. Our formulation generalizes the M-Best MAP problem and we show that for certain families of dissimilarity functions we can guarantee that these solutions can be found as easily as the MAP solution.

1 Introduction

The introduction of sophisticated discrete optimization tools for inference in Markov Random Fields (MRFs) over the last two decades has changed the scale of problems in machine learning. We can quickly compute optimal or provably approximate solutions to certain problems that were previously believed to be hopelessly intractable. For instance, since the introduction of max-flow/min-cut methods [4, 16], it is perfectly natural now to *expect* that we will be able to handle a 2-Million-pixel image (binary) segmentation problem, effectively searching $2^{2,000,000} \simeq 10^{600,000}$ possible configurations, and return the optimal solution in a matter of seconds.

Despite this progress, it is important to remember that *optimization error* [2] is only one component of generalization error of a learning algorithm. Specifically, even if we can perform optimal inference in MRFs, the maximum a posteriori (MAP) solution might be very far from the ground-truth. The source of this discrepancy may be *approximation error*, *i.e.* error arising from limitations of the model class (*e.g.* pairwise binary submodular MRFs), and *estimation error*, *i.e.* error made because parameters are learnt from finite training set.

This discrepancy is not merely a theoretical concern. In fact, large-scale empirical studies [20, 32] on a number of computer vision applications have repeatedly found that the MAP solution is of much poorer quality than the ground-truth. Equivalently, the ground-truth has much lower probability than the MAP solution under the model.

We believe that one way to mitigate this problem is to produce not just a single solution, but a diverse set of highly probable solutions. Note that this is different from the M-Best MAP problem [8, 23, 38], which involves finding the M most probable solutions under a probabilistic model. The key difference is the emphasis on "diversity", *i.e.* to produce highly probable solutions that are *qualitatively* different. This is because the literal definition of M-best MAP is not expected to work well in practice. In a large state-space problem ($\sim 2^{2,000,000}$) any reasonable setting of M ($10 - 50$) would return solutions nearly identical to the MAP state. Ideally, we need to perform "mode-hopping"[1], where the next best state is defined as the most probable state at least some minimum measure of distance away from the current best state.

Contributions. In this paper, we present an algorithm for the *M-Best Mode* problem, which involves finding a diverse set of highly probable solutions under a discrete probabilistic model. Our

[1] Throughout this manuscript, we use the word "mode" loosely, without the strict sense of a peak with local optimality, but rather to mean a configuration with a high probability.

formulation assumes access to a dissimilarity function $\Delta(\cdot,\cdot)$ measuring the closeness of two solutions. To extract diverse solutions, we solve the Δ-augmented energy minimization problem, which minimizes a linear combination of the energy and similarity to previous solutions. This approach is similar in flavor to the loss-augmented energy minimization solved in structural SVMs [34]. Our formulation generalizes the M-Best MAP problem in a specific sense – for a particular choice of $\Delta(\cdot,\cdot)$, our formulation reduces to the M-Best MAP. Perhaps our most important contribution is the observation that for certain families of Δ-functions with nice structures, the Δ-augmented energy minimization problem is *as easily solvable* as the original MAP problem. Thus if exact or provably approximate algorithms exist for finding the MAP solution, those *same* algorithms are applicable for finding the M-Best Modes.

Applications. We see a number of applications of an M-best Mode algorithm. First, using M qualitatively distinct solutions can help speed up cutting-plane methods for training Structural SVMs [12, 34], which otherwise rely only on one MAP solution. Second, interactive applications that work with a user in the loop (e.g. interactive segmentation) can greatly reduce the number of interactions by presenting multiple distinct highly probable solutions, instead of one. Finally, in some applications it is possible to rank multiple generated solutions via a secondary process [5]. Clearly, having a diverse and small solution set is most beneficial for this ranking procedure.

Section 2 discusses and contrasts related work; Section 4 presents our proposed M-Best Mode formulation in detail; Section 5 describes how to solve this formulation; Section 6 presents experiments on the interactive segmentation problem.

2 Related Work

The problem of finding the top M solutions to a general combinatorial optimization problem (not necessarily inference in MRFs) has typically been studied in the context of k-shortest paths [7] in a search graph. Lawler [19] proposed a general algorithm to compute the top M solutions in a large variety of discrete optimization problems. In fact, ideas used in Lawler's algorithm form the basis for most algorithms for M-Best MAP.

M-Best MAP. Most directly relevant to our work is literature on the M-Best MAP problem. The first family of algorithms for M-Best MAP [23, 28] were junction-tree based algorithms, thus feasible only for low-treewidth graphs. Dechter and colleagues [6, 21] have recently provided dynamic-programming algorithms for M-Best MAP, but these are exponential in treewidth as well. Yanover and Weiss [38] proposed an algorithm that requires access only to *max-marginals*. Thus, for certain classes of MRFs that allow efficient exact computation of max-marginals, *e.g.* binary pairwise submodular MRFs [15], M-Best solutions can be found for arbitrary treewidth graphs. Moreover, *approximate* M-Best solutions may be found by approximating the max-marginal computation, *e.g.* via max-product BP. More recently, Fromer and Globerson [8] provided a Linear Programming (LP) view of the M-Best MAP problem by studying the assignment-excluding marginal polytope. The formulation proposed in this paper generalizes the M-Best MAP problem in a specific sense – for a particular choice of $\Delta(\cdot,\cdot)$, our formulation reduces to the M-Best MAP.

A rather different approach is taken by Porway and Zhu in [25], whose C^4 algorithm explores multiple solutions using sampling of connected components in the MRF. This could allow "jumping" between modes of the posterior distribution as the sampling proceeds, however there is no mechanism to require the multiple solutions to be diverse, in contrast to our work.

Diverse Solutions. The need for diverse solutions arises in a number of problems in machine learning. Yu and Joachims [39] studied this problem in the context of document retrieval. They propose to learn a predictor that selects a diverse subsets of documents, where diversity is based on topics covered by the documents. Park and Ramanan [24] applied the max-marginal algorithm of Yanover and Weiss [38] to decode multiple solutions from a deformable parts model, with an added constraint on non-overlapping parts. Their approach works for a fairly strict definition of dissimilarity (Δ) between solutions, one that our formulation contains as a special case. We revisit this issue in Section 4. Interestingly, spectral or eigenvector-based algorithms [22, 29, 36] encode a natural notion of diversity – successive eigenvectors are always orthogonal, and thus maximally dissimilar w.r.t. inner product similarity. However, it is unclear how to incorporate arbitrary dissimilarity functions $\Delta(\cdot,\cdot)$, which is our goal in this paper.

Finally, at a high-level, the data re-weighting rules used in boosting algorithms can also be thought of as searching for qualitatively different weak learners.

3 Preliminaries: MAP Inference in MRFs

Notation. For any positive integer n, let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. We consider a set of discrete random variables $\mathbf{x} = \{x_i \mid i \in [n]\}$, each taking value in a finite label set, $x_i \in X_i$. For a set $A \subseteq [n]$, we use x_A to denote the tuple $\{x_i \mid i \in A\}$, and X_A to be the joint label space $\times_{i \in A} X_i$. For ease of notation, we use x_{ij} as a shorthand for $x_{\{i,j\}}$.

MAP. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph defined over these variables, *i.e.* $\mathcal{V} = [n]$, $\mathcal{E} \subseteq \binom{\mathcal{V}}{2}$, and let $\theta_A : X_A \to \mathbb{R}$, $\forall A \in \mathcal{V} \cup \mathcal{E}$ be functions defining the energy at each node and edge for the labeling of variables in scope. Let $\vec{\mathcal{E}} = \{(i \to j), (j \to i) \mid \{i, j\} \in \mathcal{E}\}$ be a set holding directed edges for each undirected edge. The goal of MAP inference is to find the labeling $\mathbf{x}$ of the variables that minimize this real-valued energy function:

$$\min_{\mathbf{x} \in X_{\mathcal{V}}} \sum_{A \in \mathcal{V} \cup \mathcal{E}} \theta_A(x_A) = \min_{\mathbf{x} \in X_{\mathcal{V}}} \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j). \tag{1}$$

MAP Integer Program MAP inference is typically set up as an integer programming problem over the boolean variables. Let $\boldsymbol{\mu}_A = \{\mu_A(s) \mid \mu_A(s) \in \{0, 1\}, s \in X_A\}$, be a vector of indicator variables for all possible configurations of $x_A, \forall A \in \mathcal{V} \cup \mathcal{E}$. Thus, $\{\mu_A(s) = 1 \Leftrightarrow x_A = s\}$. Moreover, let $\boldsymbol{\theta}_A = \{\theta_A(s) \mid s \in X_A\}$, be a vector holding energies for all possible configurations of x_A, and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_A \mid A \in \mathcal{V} \cup \mathcal{E}\}$ be a vector holding the entire configuration. Using this notation, the MAP inference integer program can be written as:

$$\min_{\boldsymbol{\mu}} \quad \sum_{A \in \mathcal{V} \cup \mathcal{E}} \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A = \min_{\boldsymbol{\mu}_i, \boldsymbol{\mu}_{ij}} \quad \sum_{i \in \mathcal{V}} \boldsymbol{\theta}_i \cdot \boldsymbol{\mu}_i + \sum_{(i,j) \in \mathcal{E}} \boldsymbol{\theta}_{ij} \cdot \boldsymbol{\mu}_{ij} \tag{2a}$$

$$s.t. \quad \sum_{s \in X_i} \mu_i(s) = 1 \qquad \forall i \in \mathcal{V} \tag{2b}$$

$$\sum_{s \in X_i} \mu_{ij}(s, t) = \mu_j(t) \qquad \forall (i \to j) \in \vec{\mathcal{E}} \tag{2c}$$

$$\mu_i(s), \mu_{ij}(s, t) \in \{0, 1\}. \tag{2d}$$

To be concise, we will use $\mathcal{L}(G)$ to denote the linear constraints in 2b,2c, *i.e.* $\mathcal{L}(G) \doteq \{\boldsymbol{\mu}_A, \mid A \in \mathcal{V} \cup \mathcal{E}, \sum_{s \in X_i} \mu_i(s) = 1, \sum_{s \in X_i} \mu_{ij}(s, t) = \mu_j(t)\ \forall (i \to j) \in \vec{\mathcal{E}}\}$. Thus, the above problem (2) can be written concisely as:

$$\min_{\boldsymbol{\mu}} \quad \sum_{A \in \mathcal{V} \cup \mathcal{E}} \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A \tag{3a}$$

$$s.t. \quad \boldsymbol{\mu} \in \mathcal{L}(G) \tag{3b}$$

$$\mu_A(s) \in \{0, 1\}. \tag{3c}$$

Problem (2) is known to be NP-hard in general [30]. A number of techniques [9, 17, 18, 35] solve a Linear Programming (LP) relaxation of this problem, also known as Schlesinger's bound [27, 37], which is given by relaxing the boolean constraints (2d) to the unit interval, *i.e.* $\mu_i(s), \mu_{ij}(s, t) \geq 0$.

4 M-Best Mode: Formulation

We now describe our proposed M-Best Mode formulation. Recall that the goal is to produce a diverse set of low-energy solutions. We approach this problem with an iterative algorithm, where the next best state is defined as the lowest energy state at least some minimum measure of dissimilarity away from the current best state. To do so, we assume access to a dissimilarity function $\Delta(\boldsymbol{\mu}^1, \boldsymbol{\mu}^2)$ measuring how far two solutions are from each other. Let $\boldsymbol{\mu}^1$ denote the MAP, and let us first search for the second mode. We propose the following straightforward yet fairly general formulation:

$$\min_{\boldsymbol{\mu}} \quad \sum_{A \in \mathcal{V} \cup \mathcal{E}} \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A \tag{4a}$$

$$s.t. \quad \boldsymbol{\mu} \in \mathcal{L}(G) \tag{4b}$$

$$\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) \geq k \tag{4c}$$

$$\mu_A(s) \in \{0, 1\}. \tag{4d}$$

We refer to the above formulation as $MModes(\Delta, k)$, since it is parametrized by the two design choices. Intuitively, we can see that the above formulation searches for the lowest energy solution

such that they are at least k-units dissimilar to the MAP solution. The extension from 2nd-Best Mode to Mth-Best Mode is fairly simple: we search for the lowest energy solution at least k-units distance away from all previously found (M-1) solutions, *i.e.* $\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^j) \geq k, \quad \forall j \in \{1, \ldots, M-1\}$.

We now show that this formulation is general enough to contain existing ones as special cases.

Special Case: M-Best MAP. The M-Best MAP problem can be seen as a special case of this formulation ($MModes(\Delta, k)$), where Δ is a $0-1$ dissimilarity (*i.e.* $\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) = [\![\boldsymbol{\mu} \neq \boldsymbol{\mu}^1]\!]$, where $[\![\cdot]\!]$ is an indicator function), and $k = 1$. Thus the dissimilarity constraint in $MModes(\Delta, k)$ simply forces the next best solution to not be identical to MAP.

Special Case: N-Best Maximal Decoding of Park and Ramanan [24]. The recently proposed approach of [24] uses the following dissimilarity function: $\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) = \max_{i \in \mathcal{V}} \Delta_i(\boldsymbol{\mu}_i, \boldsymbol{\mu}_i^1)$ and $k = 1$. Thus, their approach defines local dissimilarity functions at each node, and forces at least one node to be 1-unit away from the corresponding MAP label at that node.

5 M-Best Mode: LP Relaxation

Having presented a general formulation for M-Best Mode, we now address the issue of solving the optimization problem (4). We note that in general, $MModes(\Delta, k)$ is at least as hard to solve the MAP inference problem, which is NP-hard. However, similar to techniques for MAP inference, we study the continuous relaxation of (4) where we replace (4d) with $\mu_A(s) \geq 0$. Moreover, we optimize a specific form of the continuous relaxation, formed by dualizing the dissimilarity constraint $\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) \geq k$:

$$\min_{\boldsymbol{\mu}} \quad \sum_{A \in \mathcal{V} \cup \mathcal{E}} \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A - \lambda \, \Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) \tag{5a}$$

$$s.t. \quad \boldsymbol{\mu} \in \mathcal{L}(G) \tag{5b}$$

$$\mu_A(s) \geq 0. \tag{5c}$$

Intuitively, we can see that this program $MModes(\Delta, \lambda)$ minimizes a linear combination of the energy of the MRF and similarity (negative dissimilarity) to the MAP solution.

The reason for working with (5) instead of the continuous relaxation of (4) is that for some classes of dissimilarity functions, we can find M-Best Modes simply by reusing the *same algorithms* used for finding the MAP solution, by modifying the energy of the MRF fed into the algorithm. This allows all the developments in the MAP inference literature to be directly translated to the M-Best Mode problem, without any changes.

Example: Dot-Product Dissimilarity. Consider $\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) = -\boldsymbol{\mu} \cdot \boldsymbol{\mu}^1$, *i.e.* the negative dot-product between the two solutions. For discrete solutions $\boldsymbol{\mu}(s), \boldsymbol{\mu}^1(s) \in \{0, 1\}$, this dissimilarity function is equivalent to the Hamming Distance between the two solutions. Moreover, note that $\sum_{A \in \mathcal{V} \cup \mathcal{E}} \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A - \lambda \, \Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) = \sum_{i \in \mathcal{V}} (\boldsymbol{\theta}_i - \lambda \boldsymbol{\mu}_i^1) \cdot \boldsymbol{\mu}_i + \sum_{(i,j) \in \mathcal{E}} \boldsymbol{\theta}_{ij} \cdot \boldsymbol{\mu}_{ij}$. Thus, the $MModes(\Delta, \lambda)$ problem becomes the same as the MAP problem with modified unary energies (that are biased away from the current solution). Thus, we can use any existing MAP inference algorithm to solve this problem. Perhaps the most attractive feature of this formulation is that the edge-energies are left unaffected. Thus, if they were submodular, they continue to be submodular; if they were metric energies, they continue to be metrics. This allows us to use efficient max-flow based algorithms [16, 18].

Example: Higher-order Dissimilarity. Another example of a useful dissimilarity function is one that decomposes into functions of subsets of variables, *i.e.* $\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^1) = \sum_{A \in [n]} \Delta_A(\boldsymbol{\mu}_A, \boldsymbol{\mu}_A^1)$. If each of the these terms $\Delta_A(\cdot, \cdot)$ has some structure, *e.g.* cardinality potentials [10,33] or lower linear envelope potentials [13] or sparse higher-order potentials [26], that allows messages to be efficiently computed, this Δ-augmented energy minimization can be performed via message-passing based coordinate descent algorithms. We refer the reader to Tarlow *et al.* [33] for more details about message computation with high-order terms. For lack of space, we do not describe this example in detail.

6 Experimental Setup

We apply the M-best mode formulation to the problem of interactive segmentation. In this setting the user is interested in segmenting an image into foreground and background regions. Initially the

user provides annotations (via scribbles) for a small subset of pixels. The goal then is to provide the M-best image segmentations for the user to choose from. Clearly, these segmentations must be highly probable yet qualitatively different. The user selects the most accurate image segmentation and applies corrections to regions by adding scribbles at which point the process is repeated until a satisfactory segmentation is achieved. From the perspective of quantitative evaluation, we are interested in determining the accuracy of M-best image segmentations returned by various methods.

Energy. Consider an image-scribble pair $(\mathbf{x}, \mathcal{S})$, where each image is a collection of n superpixels to be labelled as either foreground or background, *i.e.* $\mathbf{x} = \{x_i | i \in [n]\}$ where each $x_i \in \{fg, bg\}$, and $\mathcal{S} \subset [n]$ is a subset of superpixels for whom labels are known. We chose to use superpixels as opposed to pixels for computational efficiency and in order to produce segmentations that were better aligned to boundaries in the image. We build a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, over the superpixels with edges connecting adjacent superpixels. The pairwise MRF energy for this application is given by:

$$E(\mathbf{x} : \mathcal{A}) = \sum_{i \in \mathcal{V}} \theta_i(x_i : \mathcal{A}) + \lambda \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j), \tag{6}$$

where the first (data) term is the cost for assigning a superpixel to the foreground or background classes, while the second (smoothness) term is the penalization for having different labelings of neighboring superpixels. The data term depends on an appearance model $\mathcal{A}$ learned from the user scribbles ($\mathcal{S}$) and is defined below.

Data Term. The unary appearance model is based on the output of a linear Transductive SVM (TSVM). Specifically, we extract feature vectors $\phi(x_i)$ from both labeled and unlabeled superpixels and train a TSVM [31] to learn the appearance model. We extract colour features (C1-C4 as proposed by [11]), a histogram of gradients (HOG features), and a histogram over SIFT codewords. Let $\mathbf{w}$ be the learnt weight vector from the TSVM and $s_i = \mathbf{w}^T \phi(x_i)$ be the score for each superpixel. The resulting data term is of the following form:

$$\theta_i(x_i = fg) = \begin{cases} \eta, & \text{if } s_i \geq 0 \\ 1 - \eta, & \text{otherwise} \end{cases}, \quad \theta_i(x_i = bg) = \begin{cases} 1 - \eta, & \text{if } s_i \geq 0 \\ \eta, & \text{otherwise} \end{cases}, \tag{7}$$

where, and $\eta = 0.5\, e^{\frac{-|s_i|^2}{\alpha \sigma^2}}$. We set $\sigma^2 = var\left(\{s_j \mid j \in \mathcal{S}\}\right)$, and α is set to a hand tuned constant fixed for all images.

Smoothness Term. The pairwise smoothness term we used is the contrast sensitive Potts model:

$$\theta_{ij}(x_i, x_j) = [\![x_i \neq x_j]\!] \cdot \beta_1 \cdot e^{-\beta_2 d_{ij}}, \tag{8}$$

where $[\![\cdot]\!]$ is an indicator function that is 1 when its argument evaluates to true and 0 otherwise, d_{ij} is the distance between feature vectors at superpixels i and j, and β_1, β_2[2] are scale parameters. The effect of this smoothness term is to penalize for label transitions between neighboring superpixels, but the penalization decreases as the distance in feature space between the corresponding feature vectors increases.

Inference. The contrast-sensitive Potts model results in a submodular energy function for a two-label problem so we can efficiently compute the MAP solution using the publicly available graph-cut implementation of [14], [3]. Moreover, with negative dot-product dissimilarity, we can efficiently and optimally compute the top M-Modes using graph-cuts as well.

6.1 Evaluation and Results

For each image-scribble pair in our dataset we ran interactive segmentation to get the MAP solution along with the 5 next-best solutions returned by the $MModes$ method. We compared the highest scoring of these next-best solutions, in terms of pixel accuracy, with the top-scoring solutions we got from the baseline methods we describe next.

Baselines. We compared against three baselines. As a first baseline we implemented the M-best MAP algorithm of Yanover and Weiss [38]. This M-Best MAP algorithm requires repeated computation of min-marginals, which we computed using the publicly available implementation of Kohli and Torr [15]. The goal of including this baseline is to show that M-Best MAP solutions lack the diversity offered by M-Best Modes. For the second baseline we flipped the labels of m randomly selected superpixels, where m is the number of superpixels that changed labels from the MAP to our $MModes$ solution. For the third baseline we flipped the label of the top m *uncertain* superpixels,

[2] In the experiments $\beta_1 = 2$ and $\beta_2 = \sqrt{.05 \max\{d_{ij}\}}$.

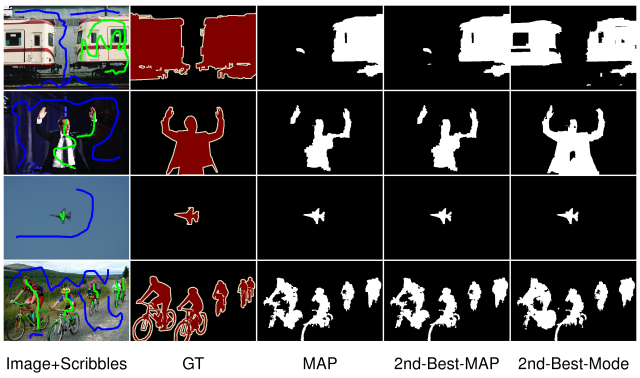

Figure 1: *(bottom-to-top)* Images where the pixel accuracies of $MModes$ are slightly less than, same as, slightly better than, and significantly better than MAP solution.

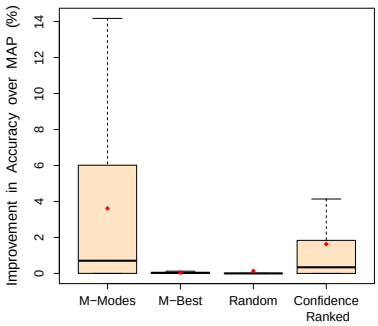

Figure 2: Red points indicate the average over 50 training images.

	Map	M-Modes	M-Best	Random	Confidence Ranked
Mean	91.542	**95.157**	91.590	91.676	93.174

Table 1: Absolute pixel accuracies averaged over 50 test images.

where we use the entropy of normalized min-marginals as a measure of uncertainty. The purpose of the last two baselines is to show that our M-Best Mode formulation is better than methods that introduce random perturbations to achieve diversity.

The dataset consisted of 100 images from Pascal VOC2010, with corresponding scribble locations indicating foreground/background superpixels[3] in each image. Fifty of the images were used for tuning the regularization weight, λ, and the other 50 for testing. We ran grid search over values of λ in the range $[0, 1]$. The best results on the training images was achieved with $\lambda = .18$, which we fixed for experiments on the test set. Summaries of the improvement in pixel accuracies relative to the MAP solution are reported in Figure 2 along with the absolute accuracies in Table 1. A selection of the final segmentations is shown in Figure 1. Note that the 2^{nd}-best MAP solution is nearly identical to the MAP solution whereas the solution from $MModes$ is closer to the ground-truth labeling. In fact, on average the solution from $MModes$ is more accurate compared to ground-truth than the other baseline methods as well as M-best MAP. The first two rows of Figure 1 illustrate that it is sometimes highly beneficial to find solutions that are further away from the MAP solution.

7 Conclusion and Future Work

In summary, we present the first algorithm for the M-Best Mode problem, which involves finding a diverse set of highly probable solutions under a discrete probabilistic model. Our formulation solves the Δ-augmented energy minimization problem, which minimizes a linear combination of the energy and similarity to previous solutions. We showed that this framework is a generalization of the M-best MAP formulation and that for certain classes of the Δ-function, the proposed $MModes$ algorithm finds solutions that can be computed using the same algorithms for computing the MAP solution. Our experiments show that our $MModes$ algorithm produces significant improvement in pixel accuracies on the interactive segmentation problem as compared to the M-best MAP algorithm.

As future work we would like to investigate the performance and implications of other Δ-functions, apply them to higher-order energy functions and also apply this method to speed up cutting-plane methods for training Structural SVMs.

Acknowledgements. We thank Pushmeet Kohli for helpful discussions and suggesting applications.

[3]We used SLIC [1] to extract superpixels, with the desired number of superpixels in an image set to 3000. The images contained roughly 150K-200K pixels.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels. Technical report, EPFL, 2010. 6

[2] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Adv. in NIPS*, pages 161–168, 2008. 1

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2001. 5

[4] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001. 1

[5] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, pages 3241 –3248, june 2010. 2

[6] N. F. Emma Rollon and R. Dechter. Inference schemes for m best solutions for soft csps. In *Proceedings of Workshop on Preferences and Soft Constraints*, 2011. 2

[7] D. Eppstein. Finding the k shortest paths. *SIAM J. Computing*, 28(2):652–673, 1998. 2

[8] M. Fromer and A. Globerson. An LP view of the m-best MAP problem. In *NIPS*, 2009. 1, 2

[9] A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, 2007. 3

[10] R. Gupta, S. Sarawagi, and A. A. Diwan. Collective inference for extraction mrfs coupled with symmetric clique potentials. *J. Mach. Learn. Res.*, 11:3097–3135, 2010. 4

[11] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *In ICCV*, pages 654–661, 2005. 5

[12] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009. 2

[13] P. Kohli and M. P. Kumar. Energy minimization for linear envelope mrfs. In *CVPR*, pages 1863–1870, 2010. 4

[14] P. Kohli and P. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 922 –929 Vol. 2, oct. 2005. 5

[15] P. Kohli and P. H. S. Torr. Measuring uncertainty in graph cut solutions. *CVIU*, 112(1):30–38, 2008. 2, 5

[16] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 1, 4

[17] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007. 3

[18] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29, 2008. 3, 4

[19] E. L. Lawler. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science*, 18:401–405, 1972. 2

[20] T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *ICCV*, pages 428–435, 2005. 1

[21] E. R. Natalia Flerova and R. Dechter. Bucket and mini-bucket schemes for m best solutions over graphical models. In *IJCAI Workshop on Graph Structures for Knowledge Representation and Reasoning*, 2011. 2

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001. 2

[23] D. Nilsson. An efficient algorithm for finding the m most probable configurations in probabilistic expert systems. *Statistics and Computing*, 8:159–173, 1998. 10.1023/A:1008990218483. 1, 2

[24] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011. 2, 4

[25] J. Porway and S.-C. Zhu. c^4: Exploring multiple solutions in graphical models vy cluster sampling. 33(9), 2011. 2

[26] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, pages 1382–1389, 2009. 4

[27] M. I. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions (in Russian). *Kibernetika*, 4:113–130, 1976. 3

[28] B. Seroussi and J. Golmard. An algorithm directly finding the k most probable configurations in bayesian networks. *Int. J. of Approx. Reasoning*, 11(3):205 – 233, 1994. 2

[29] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 2

[30] S. E. Shimony. Finding MAPs for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, August 1994. 3

[31] V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear svms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 477–484, New York, NY, USA, 2006. ACM. 5

[32] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *PAMI*, 30(6):1068–1080, 2008. 1

[33] D. Tarlow, I. E. Givoni, and R. S. Zemel. Hop-map: Efficient message passing with high order potentials. In *AISTATS*, pages 812–819, 2010. 4

[34] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005. 2

[35] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *Trans. Inf. Th.*, 51(11):3697–3717, 2005. 3

[36] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV*, 1999. 2

[37] T. Werner. A linear programming approach to max-sum problem: A review. *PAMI*, 29(7):1165–1179, 2007. 3

[38] C. Yanover and Y. Weiss. Finding the m most probable configurations using loopy belief propagation. In *NIPS*, 2003. 1, 2, 5

[39] Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *ICML*, pages 271–278, 2008. 2