Computing the M Most Probable Modes of a Graphical Model

Chao Chen
Rutgers University

Vladimir Kolmogorov
IST Austria

Yan Zhu
Rutgers University

Dimitris Metaxas
Rutgers University

Christoph H. Lampert
IST Austria

Abstract

We introduce the *M-Modes* problem for graphical models: predicting the M label configurations of *highest probability* that are at the same time *local maxima of the probability landscape*. M-Modes have multiple possible applications: because they are intrinsically diverse, they provide a principled alternative to *non-maximum suppression* techniques for structured prediction, they can act as codebook vectors for quantizing the configuration space, or they can form component centers for mixture model approximation.

We present two algorithms for solving the M-Modes problem. The first algorithm solves the problem in polynomial time when the underlying graphical model is a simple chain. The second algorithm solves the problem for junction chains.

In synthetic and real dataset, we demonstrate how M-Modes can improve the performance of prediction. We also use the generated modes as a tool to understand the topography of the probability distribution of configurations, for example with relation to the training set size and amount of noise in the data.

1 Introduction

Discrete graphical models are widely used tools for modelling and solving structured prediction problems. Given a factor graph, one of the most common tasks is to compute the most probable configuration, called *MAP* (for *maximum a-posteriori*), and various algorithms have been developed for it, including (loopy) belief propagation, junction trees, and linear programming relaxations (Wainwright and Jordan, 2008; Nowozin and Lampert, 2010).

Several practical application, however, require access to more than a single configuration, such as multi-label classification (Lampert, 2011), protein design (Yanover and Weiss, 2004; Fromer and Yanover, 2009), or human pose estimation (Park and Ramanan, 2011). A straight-forward way to predict multiple good configurations from a graphical model is *M-best* prediction: instead of just the MAP, one outputs the M configuration of highest probability. Nilsson (1998) proposed an algorithm for this that works with junction trees, while Yanover and Weiss (2004) and Fromer and Globerson (2009) proposed alternative approximate techniques that can handle graphs with larger tree-width. Recently, Batra (2012) also provided a more efficient algorithm for such formulations. The main problem of M-best prediction is its lack of diversity: because the configuration space of graphical models is typically very large and fine-grained, the 2nd best, 3rd best, etc., configurations typically differ only insignificantly from the MAP, so the amount of additional information one obtains from them is limited. To significantly go beyond the MAP, one typically has to choose M very large, with negative consequences on the runtime and the memory footprint.

A more promising strategy is to directly aim for a set of high probability configurations that are also sufficiently *diverse*. Interestingly, few principled approaches to this fundamental structured prediction problem exist, but heuristic and domain-dependent algorithms are prevalent. A common idea is *non-maximum suppression (NMS)*, as it is frequently used, e.g., in computer vision applications (Felzenszwalb et al., 2010; Blaschko, 2011). Instead of returning all M-best prediction they are filtered in a greedy way:

Appearing in Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31.

for each configuration in the output set, one removes all possible configurations of lower probability within a similarity radius. While conceptually simple, NMS provides only a partial solution to the problems mentioned above: one still has to enumerate and evaluate a larger number of solutions. And because of the greedy way that the procedure operates, small changes in parameters can have a big effect on the output set. Alternative approaches include sampling (Stephens et al., 2008), clustering the set of predictions, (Viola and Jones, 2004), adding on-the-fly constraints (Park and Ramanan, 2011) or adding diversity-enforcing penalty terms (Yue and Joachims, 2008; Yadollahpour et al., 2011; Batra et al., 2012).

In this paper, we formalize a more principled approach to predicting diverse subsets of high probability configurations: *M-best modes*. As the name suggests, the idea is to compute the M most probable *modes* of the probability distribution, i.e. configurations that are local maxima of the likelihood function.

We will show in synthetic and real datasets that M-modes can often improve the prediction performance compared to the previously mentioned techniques. Furthermore, M-modes is a natural tool to characterize the topography of the probability distribution of configurations. We provide some preliminary insight into this aspect by characterizing the number of modes of the distribution with respect to factors such as the training set size and the amount of noise in the data. As far as we know, this is the first study of this type.

As algorithmic contribution we propose two different algorithms. The first algorithm solves the M-modes problem for chain graphical models. The algorithm is exact and polynomial in all relative quantities. The second algorithm works on more general graphs, namely, junction chains. Both algorithms are built on relationship between local patterns and global properties. The difference is, the former identifies those local patterns that are essential for a mode, so that one could efficiently search through the space of all modes, to identify the best M. The latter identifies local patterns that are definitely not part of a mode. The algorithm then shrinks the search space by forbidding these patterns.

Related Work. Since it is by far the most frequently used related technique, we briefly describe the algorithm of non-maximum-suppression. It consists of iterative calls to an M-best algorithm (Nilsson, 1998; Yanover and Weiss, 2004; Fromer and Globerson, 2009), which returns the next best configuration that has not been generated. The configuration is compared with all configurations that have been collected so far, and it is discarded (suppressed) if and only if its minimal Hamming distance from the collection is no greater than a given threshold. The process is repeated until M configurations have been collected. Because of its simple structure, generalization are easy to create, e.g. by using a different dissimilarity measure than the Hamming distance (Park and Ramanan, 2011).

Recently, Batra *et al.* (2012) proposed a new method which is more efficient to generate M diversified solutions. It also works iteratively, starting with the MAP, but instead of enumerating all configuration by their probability score it searches for the next candidate in a more targeted fashion. For this it formulates the problem of computing the m-th solution as computing the optimal configuration under the constraint that it has to be dissimilar by a certain margin from the $m-1$ solutions collected so far. The problem is in general NP-hard, but a Lagrangian relaxation can be solved performing MAP with a modified set of unary potentials. This makes the method very efficient and easy to implement. Furthermore, the algorithm works on loopy graphs using approximation algorithms like α-expansion (Boykov et al., 2001). However because the Lagrangian relaxation is in general not tight, there is no guarantee on the quality of the prediction, or that the dissimilarity constraints will be fulfilled exactly.

Finally, we emphasize that computing the modes of a function, or specifically a probability distribution, has a rich tradition in machine learning, for example in *clustering* (Cheng, 1995; Leung et al., 2000), and it also plays an important role in other branches of mathematics, such as *computational topology* (Edelsbrunner and Harer, 2010). Somewhat surprisingly, we are not aware of any in depth work that follows up on this idea to characterize the probability distribution given by a graphical model. The reason, we assume, is not conceptual but computational: the probability distributions of (discrete) graphical models are defined over a finite but typically very large space, robbing us even of the most basic tools from calculus (gradients, curvature), but at the same time making it impossible to find local maxima by exhaustive search.

2 M-Modes Problem

We start by formalizing the definition of *maxima* and *minima* in a discrete setting. Let $\mathcal{X} = \{x^1, \dots, x^N\}$ be a finite, but potentially large set, for example the set of all L-label configurations of a factor graph with n variables, which has L^n elements. We call the elements of $\mathcal{X}$ *configurations*, or *labelings*. For each $x \in \mathcal{X}$, let $\mathcal{N}(x) \subset \mathcal{X}$ be a *neighborhood*. Let $f : \mathcal{X} \to \mathbb{R}$ be a discrete function. For easier explanation, we follow the tradition of the M-best literature and assume that all configurations have distinct function values.

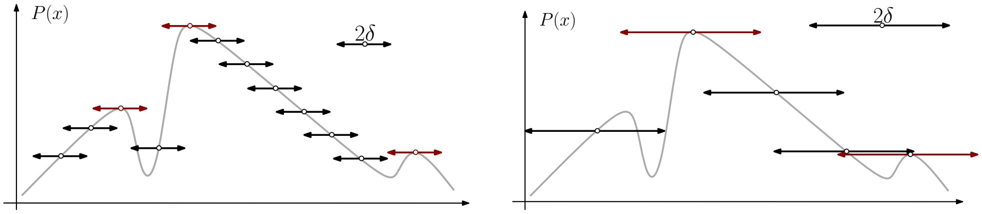

Figure 1: Illustration of M-modes and non-maximum suppression (NMS).
Left: $P(x)$ has 3 local maxima (red). Applying M-modes with δ as indicates returns these in order of decreasing function value. NMS with same δ will first return the global maximum. Subsequently, NMS returns further points on the right slope. Only the 5th returned element will be the left mode, and the 10th the right one.
Right: A larger δ mitigates this effect, but it introduce the danger of suppressing the left mode. M-modes returns the right mode as second result in this case. NMS returns neither the second nor the third mode, but nearby configuration of potentially lower probability.

Definition 1. *We call a labeling $x \in \mathcal{X}$ a* local maximum *of f, iff $f(x) \geq f(x'), \forall x' \in \mathcal{N}(x)$. We define* local minima *analogously by the inverse inequality.*

To define a neighborhood of a labeling, we make use of a distance measure between labelings. The easiest choice in a discrete graphical model framework is the *Hamming distance*, $\mathrm{d_H}$, i.e. the number of variables at which two labelings disagree. For a non-negative integer δ, we define the δ*-neighborhood* of a labeling x to be $\mathcal{N}_\delta(x) = \{x' \in \mathcal{X} \mid \mathrm{d_H}(x, x') \leq \delta\}$, the set of labelings whose distances from x is no more than δ.

Given a graphical model factor graph, its energy f is inversely logarithm proportional to the probability distribution $P(x) = \exp(-f(x) - A)$, where $A = \log \sum_x \exp(-f(x))$ is the log-partition function. Consequently, there is a bijection between the local minima of f and the local maxima of P. We call these labelings *modes* and use both views interchangeably. Given δ, we denote by $\mathcal{M}^\delta$ the set of modes, formally,

$$\mathcal{M}^\delta = \{x \in \mathcal{X} \mid x \text{ is a local minima of } f \text{ under the neighborhood } \mathcal{N}_\delta(x)\}. \quad (1)$$

Note that through the choice of the threshold δ in the above definition we influence the *smoothness* or *scale* of the topography, on which we characterize the energy. When $\delta = 0$, $\mathcal{N}_\delta(x) = \{x\}$, so every point is a mode. When $\delta = \infty$, $\mathcal{N}_\delta(x) = \mathcal{X}$, so MAP is the only mode. As δ increases from zero to infinity, the δ-neighborhood of x monotonically grows and the set of modes $\mathcal{M}^\delta$ monotonically decreases. Because modes only disappear in this process except at M^0, we obtain that the $\mathcal{M}^\delta$ form a nested sequence,

$$\mathcal{X} = \mathcal{M}^0 \supseteq \mathcal{M}^1 \supseteq \cdots \supseteq \mathcal{M}^\infty = \{\text{MAP}\}, \quad (2)$$

which one can view as a multiscale description of the probability landscape. With the above notation we formalize the problem that we study in this paper as

Problem 1 (M-modes)**.** *Compute the M labelings with minimal energies in $\mathcal{M}^\delta$.*

Relation to non-maximum suppression. While based on a similar intuition and language, there are significant differences between M-modes and the concept of non-maximum suppression (NMS). NMS is typically defined in an algorithmic way: starting from the MAP prediction one goes through all labelings according to an increasing order of the energy. A labeling becomes part of the predicted set if and only if it is more than δ away from the ones chosen before. NMS solutions are typically not local extrema of the probability distribution, and no nested sequence of the type of Equation (2) exists. Using a same parameter δ, both M-modes and NMS guarantee the solutions to be δ apart from each other. However, because M-modes are at the same time local extrema, they can naturally achieve larger diversity with smaller δ, see Figure 1 for an illustration. The difference in the neighborhood size is relevant, since NMS for graphical model probability functions quickly becomes inefficient with large δ: the iterative algorithm might have to go through $O(|\mathcal{N}_\delta|)$ labelings before getting the next solution, and the size of $\mathcal{N}_\delta$ grows exponentially with δ.

3 Algorithm for Simple Chains

In this section, we present the algorithm for simple chains. We first study local behavior of modes. Theorem 2 reveals that a labeling is a mode if and only if it behaves like a “local mode” everywhere. Inspired by this observation, we construct a new chain, and reduce M-modes problem into the M-best problem of the new chain. Throughout this section, we assume that $\delta < n$ is fixed and function f is given by

$$f(x) = \sum_{i=1}^{n} \sum_{i=1}^{n-1} f_{i,i+1}(x_i, x_{i+1}) \quad (3)$$

(For brevity, we do not use unary terms since that can be merged to pairwise terms). For a labeling $x = (x_1, \ldots, x_n) \in \mathcal{X}$ it is convenient to define $x_0 = x_{n+1} = *$. This can be thought of as appending nodes 0 and $n+1$ to the chain with one allowed label "$*$". For a labeling x and an interval $[i,j] \subseteq [0, n+1]$, denote by $x_{i:j}$ the *partial labeling* of x on the interval $[i,j]$. The cost $f(x_{i:j})$ of a partial labeling is the sum of those terms in (3) that lie inside $[i,j]$. We call $x_{i:j}$ a *partial mode* if its cost is smaller than the cost of any other partial labeling $y_{i:j}$ with the same labels on i and j, and $\mathrm{d_H}(x_{i:j}, y_{i:j}) \le \delta$.

Lemma 1 (Uniqueness of Partial Modes)**.** *Consider interval $[i,j] \subseteq [0, n+1]$ of length $j - i + 1 \le \delta + 2$ and a pair of labels (ℓ_i, ℓ_j) for nodes i and j. There exists exactly one partial mode on $[i,j]$, called $x^{opt}_{i:j}(\ell_i, \ell_j)$.*

The lemma holds because the set of all partial labelings $x_{i:j}$ with both ends fixed are at most δ from each other in the Hamming distance.

Corollary 1. *On any interval $[i,j] \subseteq [0, n+1]$ of length $\delta{+}2$, there are exactly L^2 partial modes if $[i,j] \subseteq [1,n]$, and L partial modes otherwise.*

Theorem 2. *A labeling $x \in \mathcal{X}$ is a mode iff for any interval $[i,j] \subseteq [0, n+1]$ of length $\delta + 2$, the partial labeling $x_{i:j}$ is a partial mode, i.e. $x_{i:j} = x^{opt}_{i:j}(x_i, x_j)$.*

A proof is given in Appendix A. The theorem suggests the following algorithm for solving the modes problem on a chain. Let us create a new energy minimization instance on a chain whose set of nodes $\widehat{V}$ is the set of intervals $[i,j] \subseteq [0, n+1]$ of length $\delta + 2$. We can write $\widehat{V} = \{v_0, \ldots, v_{n-\delta}\}$ where $v_i = [i,j]$. Each vertex $v_i = [v_i, v_j] \in \widehat{V}$ is allowed to have L^2 states (ℓ_i, ℓ_j) (except for the first and last vertices, which have only L states)[1]. Consider two consecutive vertices $v_i = [i,j]$ and $v_{i+1} = [i+1, j+1]$. We say that their states $\alpha_i = (\ell_i, \ell_j)$ and $\alpha_{i+1} = (\ell_{i+1}, \ell_{j+1})$ are *consistent* if partial labelings $x^{opt}_{i:j}(\ell_i, \ell_j)$ and $x^{opt}_{i+1:j+1}(\ell_{i+1}, \ell_{j+1})$ agree on the overlap $[i+1, j]$. We say that configuration $\alpha = (\alpha_0, \ldots, \alpha_{n-\delta})$ is *consistent* if α_i and α_{i+1} are consistent for all $i \in [0, n-\delta-1]$. Using an induction on i, we get the following fact.

Theorem 3. *For any consistent configuration α there exists a unique labeling $x \in \mathcal{X}$ that is consistent with α, i.e. for any interval $v_i = [i,j] \in \widehat{V}$ with label $\alpha_i = (\ell_i, \ell_j)$ there holds $x_{i:j} = x^{opt}_{i:j}(\ell_i, \ell_j)$.*

Clearly, we can define energy

$$\widehat{f}(\alpha) = \sum_{i=0}^{n-\delta-1} \widehat{f}_{i,i+1}(\alpha_i, \alpha_{i+1}) \tag{4}$$

in such a way that (i) cost $\widehat{f}_{i,i+1}(\alpha_i, \alpha_{i+1})$ is finite iff α_i and α_{i+1} are consistent, and (ii) if α is a consistent configuration corresponding to labeling $x \in \mathcal{X}$ then $\widehat{f}(\alpha) = f(x)$. By Theorem 2 and 3, there is a one-to-one cost-preserving correspondence between consistent configurations α and the set of modes $\mathcal{M}^\delta$.

We reduced the problem of computing the M best modes to the problem of computing the M best configurations in the new chain. The latter problem can be solved using the M-best algorithm by Nilsson (1998).

Complexity. To construct the new chain instance, we need to compute partial labelings $x^{opt}_{i:j}$. This can be done by calling dynamic programming L times for each interval $[i,j] \in \widehat{V}$; these computations take $O(nL^2\delta)$ time. Let us now discuss the complexity of the second step (running Nilsson's algorithm on the new instance). We need the following observation.

Lemma 4. *For any $i \in [0, n-\delta-1]$ there exist at most L^3 consistent pairs of states (α_i, α_{i+1}).*

Proof. If states (ℓ_i, ℓ_j) and (ℓ_{i+1}, ℓ_{j+1}) agree then $\ell_{i+1} = [x^{opt}_{i:j}(\ell_i, \ell_j)]_{i+1}$. Thus, (ℓ_i, ℓ_j) completely determines ℓ_{i+1}. This implies the lemma. □

Using this fact and the complexity stated in (Nilsson, 1998), we get that the second step takes $O(nL^3 + MnL^2 + nM\log(nM))$ time. Together with the complexity of computing partial labelings, we showed that M best modes of energy (3) can be computed in $O(nL^2(L + M + \delta) + nM\log(nM))$ time.

4 Algorithm for Junction Chains

We now consider the problem of computing modes in a junction chain. Clearly, the energy of the junction chain can be written in the form of eq. (3) where nodes $1, \ldots, n$ correspond to separators of the junction chain and states ℓ_i correspond to labelings of these separators. If states ℓ_i and ℓ_{i+1} correspond to inconsistent labelings then the cost $f_{i,i+1}(\ell_i, \ell_{i+1})$ equals ∞.

There is a one-to-one correspondence between labelings X of the original junction chain and consistent labelings x of the chain in eq. (3). Let us define the distance function $d(\cdot,\cdot)$ between labelings of the new chain via $d(x,y) = \mathrm{d_H}(X,Y)$ where x corresponds to X and y corresponds to Y. It is not difficult to see that d can be decomposed as follows:

$$d(x, x') = \sum_{i=1}^{n} d_i(x_i, x'_i) \tag{5}$$

where functions $d_i(\cdot,\cdot)$ for $i \in [1,n]$ take non-negative integer values. We thus focus on the following problem.

[1] For clarity, we use vertex/states/configurations for the new chain and nodes/labels/labelings for the original chain.

Problem 2. *Find M best modes of energy* (3) *under the distance function* (5)*.*

Unfortunately, since $d(\cdot,\cdot)$ is not necessarily a Hamming distance, we cannot reuse our previous algorithm on simple chains. In particular, Theorem 2 does not hold anymore. We thus need a different approach.

We will use a technique which is somewhat similar to the algorithm for non-maximum suppression described in the introduction. Namely, we find the best available labeling and check whether it is a mode. If so, we output this labeling. Otherwise, we suppress this labeling and look again. One key difference is, instead of suppressing one labeling, we identify some local pattern that stops this labeling from being a mode. Then we suppress all (could be exponentially many) labelings sharing the same pattern.

To implement this strategy, we use the *pattern-based CRF* on a chain (Ye et al., 2009), which is defined by the following energy:

$$F(x) = \sum_{(\alpha,i,j)\in\Lambda} f_{ij}(\alpha)\cdot\delta(x_{i:j}=\alpha) \qquad (6)$$

Here Λ is a set of triplets (α,i,j) such that $[i,j]\subseteq[1,n]$ and α is a partial labeling of length $j-i+1$. Function $\delta(\cdot)$ in eq. (6) equals 1 if the argument is true, and 0 otherwise.

We now formally describe the algorithm for solving Problem 2. We start with the energy (6) that is equivalent to the energy (3). (Thus, set Λ contains all possible patterns of length 2). We then iterate the following steps: (i) find labeling x that minimizes energy (6); (ii) modify the energy by adding a "forbidden" pattern to Λ with an infinite cost. This pattern is determined as follows. First, we check whether x is a mode using dynamic programming (details are given below). If yes, then we output x as the next mode and add $(x,1,n)$ as a forbidden pattern. If x is not a mode then there exists an interval $[i,j]\subseteq[1,n]$ and labeling y such that (a) $f(y)<f(x)$; (b) y agrees with x on all nodes except for nodes in the interval $[i,j]$; (c) $d(x,y)\le\delta$, or equivalently $\sum_{k=i}^{j} d_k(x_k,y_k)\le\delta$. In that case we know that a labeling $z\in\mathcal{X}$ with $z_{i':j'}=x_{i':j'}$ (where $i'=\max\{i-1,1\}$ and $j'=\min\{j+1,n\}$) cannot be a mode. Accordingly, we add $(x_{i':j'},i',j')$ as a forbidden pattern and repeat.

It is clear that this procedure will output modes of function f in the order of increasing cost; let us discuss the complexity. The minimum of function (6) can be computed in $O(\sum_{(\alpha,i,j)\in\Lambda}|\alpha|\cdot L)$ time where $|\alpha|=j-i+1$ is the length of α and L is the maximum number of states of a node (Ye et al., 2009). In a parallel submission we developed an alternative algorithm whose complexity is $O(\sum_{(\alpha,i,j)\in\Lambda}|\alpha|\cdot\log\ell_{\max})$ where $\ell_{\max}$ is the maximum length of a pattern in Λ. The factor $\log\ell_{\max}$ can also be replaced with $m+\log\ell'_{\max}$ where m is the number of patterns of size n and $\ell'_{\max}$ is the maximum length of remaining patterns in Λ.

The complexity of the iterative procedure is thus determined by the number of forbidden "local" patterns that needs to be added before the next mode is found. Unfortunately, we do not have a good bound on that.

In our preliminary implementation we were able to improve the running time using the idea of Lemma 1 for junction chains. Namely, we first construct a new chain instance whose nodes correspond to short intervals. Next, we compute allowed labelings for these intervals (i.e. labelings that are local modes); they will be the allowed states in the new chain. Finally, we apply the algorithm in this section to the obtained chain instance.

The idea can be extended to general graphical models, given an efficient algorithm to minimize an energy function with forbidden patterns. But we are not aware of any polynomial time algorithm for cases other than junction chains. In fact, via a reduction from the vertex cover problem, we can show that it is NP-hard to minimize a submodular function of binary variables with pairwise terms with forbidden patterns of size 2 (such patterns can be non-submodular).

Checking a mode. It remains to specify how to check whether a given labeling $x\in\mathcal{X}$ is a mode of function (3) under distance (5). We can do it by computing *messages* $m_{i-1,i}(\ell_i,\gamma)$ which have the following interpretation: it is the minimum cost of partial labeling $z_{1:i}$ that satisfies $z_i=\ell_i$ and $d(x_{1:i},z_{1:i})=\gamma$. We can use the recursion

$$m_{i,i+1}(\ell_{i+1},\gamma)=\min_{\ell_i}\left[m_{i-1,i}(\ell_i,\gamma')+f_{i,i+1}(\ell_i,\ell_{i+1})\right]$$

where we denoted $\gamma'=\gamma-d_{i+1}(\ell_{i+1},x_{i+1})$, and assumed that $m_{i-1,i}(\ell_i,\gamma')=\infty$ if $\gamma'<0$. It suffices to compute messages for all labels ℓ_i and all integers $\gamma\in[0,\delta]$.

5 Experiments

To motivate that M-Modes solves a problem of practical relevance we performed experiments on synthetic and real data. We used only simple chain models.

A first goal of the experiments is to provide a quantitative comparison of M-Modes with two other methods for predicting diverse subsets: non-maximum-suppression (M-NMS) and M-Diverse (Batra et al., 2012). For this we let the methods predict up to M

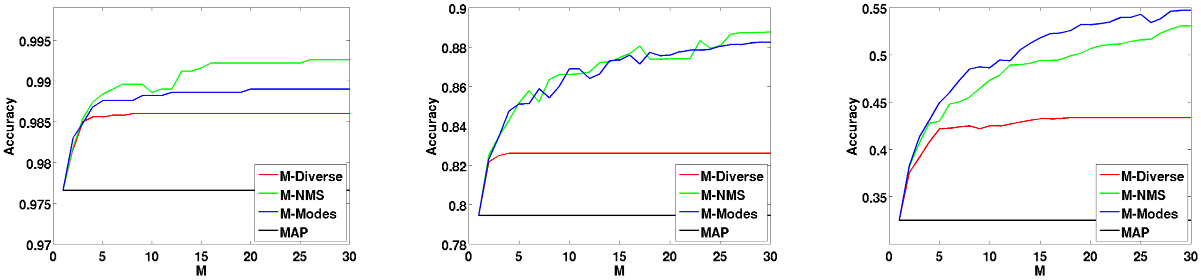

Figure 2: Comparison of the three methods in synthetic experiment. From left to right: $\sigma = 0.5$, 1.0, and 2.5.

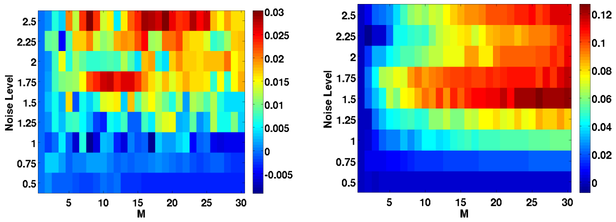

Figure 3: Synthetic data: accuracy difference between M-Modes and M-NMS (left), and between M-Modes and M-Diverse (right). The x-axis is M (small to big). The y-axis is the noise level σ (small to large). Positive values indicate M-Modes outperforming others.

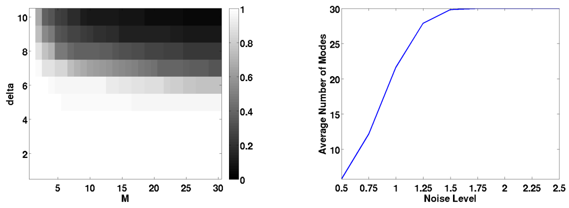

Figure 4: Synthetic data: mode statistics. Left: map plot of which fraction of test samples has at least M modes with respect to a δ-neighborhood (noise level $\sigma = 1$). Right: average number of modes (cropped at 30) for varying noise levels.

solutions and evaluate those by their *oracle accuracy*, i.e. the smallest distance between one of the M solutions and the ground truth. [2]

A second goal of the evaluation is to highlight how M-Modes can serve as a tool to gain insight into the probability landscape. As an initial step in this direction we report the total number of modes that the learned probability distributions have. To obtain this number we simply run M-Modes until it reports that no further modes exist. Note that M-Diverse and M-NMS cannot be used for a similar evaluation, since they cannot guarantee to detect all modes for a given neighborhood size (see Figure 1), nor are their solutions guaranteed to be actual local maxima of the probability landscape.

5.1 Synthetic Experiments

We first illustrate the characteristics of M-Modes by performing experiments on synthetic data: a chain-CRF of $n = 100$ variables, each of which takes one out of $L = 8$ labels. We first create a ground truth set of 100 randomly generated labelings, see the supplemental material for details of the procedure. We then define feature functions for the CRF, which are indicator vectors of the labels plus i.i.d. Gaussian noise with standard deviation σ, the value of which we vary between experiments, and we train using the UGM package (Schmidt). We use 6-fold cross-validation to select free parameters, of which each method has one: M-Modes has $\delta \in [1, 10]$, which defines the neighborhood radius of a labeling. M-NMS has $\gamma \in [1, 6]$, which is the radius of the suppression. M-Diverse has $\lambda \in [0, 2.9]$, which is the Lagrange multiplier of the suppression constraint and added to unary potentials to encourage diversity. For our experiments we select the optimal parameters for each method and for each M by cross-validation. (Different M could lead to different optimal parameters, even for a same method.) The reason for choosing γ in a smaller interval than δ is purely computational. Because the number of solutions M-NMS must explore grows exponentially in γ, often the algorithm did not terminal in reasonable time when γ was bigger than 6.

As test data, we create 100 test examples using the same procedure as the training data, and we run the three methods, M-Modes, M-NMS, and M-Diverse to predict set with up to M labelings for varying number of M. Figure 3 reports the result of the three methods for different values of σ in terms of their oracle accuracy. Figure 2 illustrates the differences for different σ. One can see that for larger noise levels, M-Modes predict subsets of higher accuracy better than

[2] The task of *selecting* one labeling out of M candidates is also a problem of active research, but not specific to the problem of M-best modes and therefore orthogonal to our studies.

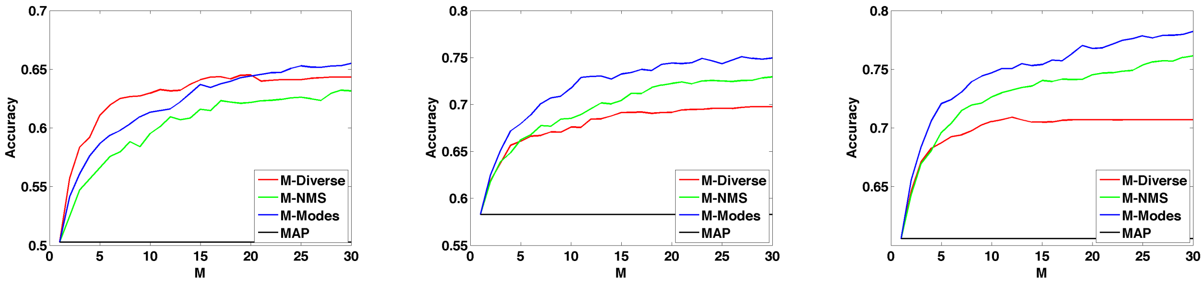

Figure 5: Results on the gesture dataset. From left to right the training set size is 10, 20 and 28 examples.

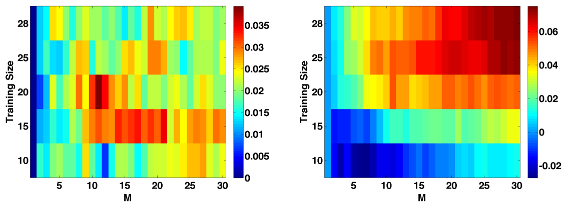

Figure 6: Gesture data: accuracy difference between M-Modes and M-NMS (left), and between M-Modes and M-Diverse (right) for different training set sizes (x-axis).

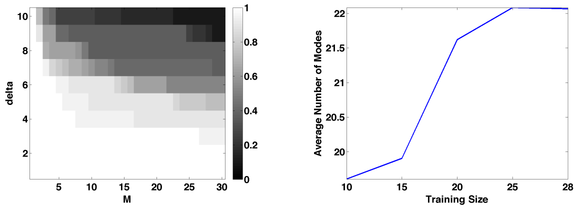

Figure 7: Gesture data: mode statistics. Left: map plot of which fraction of test samples has at least M modes with respect to a δ-neighborhood (trained with 20 examples). Right: average number of modes (cropped at 30) for varying training size.

M-NMS and M-Diverse do. When the noise level becomes smaller, the gap shrinks, and for very small σ, M-NMS achieves slightly higher accuracy values than M-Modes. However, all three methods have very similar and high accuracy in this regime, so it is not clear how significant this result is.

Figure 4 illustrates the second aspect of interest in our experiments: the number of local maxima that the probability distribution exhibits. It shows that with increasing noise level, the number of modes increases quickly. In other words, a noise-free model tend to have only a few modes, whereas with stronger noise, more modes exist. Note that the role of local minima is ambivalent: on the one hand they pose problems for local optimization techniques for MAP prediction, in particular ICM and variants (Besag, 1986; Andres et al., 2012). On the other hand, as we saw in the previous section, modes can also be beneficial, as promising candidates for creating diverse prediction sets.

5.2 Gesture Recognition Data

In a second set of experiments, we apply M-Modes to probability distributions obtained from the development part of the ChaLearn gesture recognition dataset (ges). The data in this case consists of video sequences of an actor making certain gestures, where each video frame is given one label corresponding to a gesture. We represent each frame of the video by a 30-dimensional feature vector obtained using standard computer vision features, HOG/HOF (Laptev et al., 2008). The dataset has 20 batches of 47 data samples each. We randomly select 19 of them as test data and use the rest to train a chain CRF as in the synthetic case and use cross-validation for model selection. The label space size varies for different batches, between 2 and 13. The sequences have an average length 86, and maximum length 305.

Figures 6 and 5 show the results using the same evaluation and visualization as for the synthetic experiments. As one can see M-Modes achieves better accuracy than the other two methods and this effect is stronger the more data is used for training. When reducing the training set size to 20 per batch, or even 10 per patch, M-Modes still outperforms M-NMS, but M-Diverse's performance is relatively increased. One explanation for this is visible from Figure 7, which shows the modes statistics for the different situations: with less training data, the number of modes decreases.

5.3 Other Sequential Data

We also apply our algorithm on two classical sequential dataset OCR (Kassel, 1995) and chunking (chu). In these cases, we observed that regardless of δ the probability distributions learned had very few modes. As a consequence, M-Modes could typically only return a solution set of size below M, and its accuracy did not improve significantly over the MAP state. We found this an unexpected behavior, and we plan to analyze it in more detail in future work.

5.4 Discussion of the Result

In both the synthetic and the real data experiments, we see that if the probability distribution has sufficiently many modes, M-Modes achieves equal or better accuracy than M-NMS and M-Diverse. At the same time it offers insight into the topography of the probability landscape, such as that the number of modes is affected by the noise level and training data size. For the OCR and chunking dataset, hardly any modes besides the MAP state formed during training. Since a common factor among the latter two dataset is that their features all have largely discrete values, we believe that the characteristic of the feature function might play a role here, and we plan to study this aspect in the future. Overall, the experiments give us hope that by studying the interaction of M-Modes and these factors, one may gain insight of what makes a model unimodal, and therefore easy to optimize, or multi-modal, and therefore is more suitable for diverse predictions.

An aspect of secondary interest to us is the prediction speed. Nevertheless, we performed measurements of average runtime in the synthetic data case, and report the result in Figure 8. One can see that for simple chains, where both M-Diverse and M-Modes are polynomial algorithms, their computation takes only fractions of a second. M-Diverse is even faster, since for each predicted labeling it only require a single call to a MAP predictor, and only the potential values differ between calls. Interestingly, the speed of M-Modes increases with larger neighborhood size. However, this is easily explained by the fact that with a larger neighborhood larger regions of the set of all labelings can be suppressed efficiently between detections. This is in contrast to M-NMS, which scales exponentially in the size of the neighborhood, and therefore exhibits runtime behavior that depends exponentially on the neighborhood size. Even though not visible in the plot, we also observed its runtime to varies strongly depending on the data itself.

On loopy graph, the junction chain variant of M-Modes is still guaranteed to yield complete and correct results, but it is applicable only for relatively small problem. For large problem, M-Diverse seems currently the only possibility to use, since it can readily be combined with off-the-shelf techniques for approximate inference. A second direction of future work for us will therefore be to re-examine the situation of mode prediction for loopy graph.

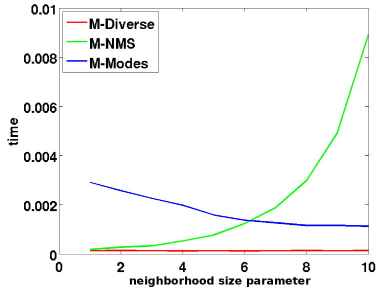

Figure 8: Average running time of the three methods in the synthetic data experiments.

6 Conclusion

In this paper, we formulated the problem of computing the M best modes. Two algorithms for simple chains and junction chains are presented. Experimental results show that modes could be used to improve prediction accuracy, and could be used as tools to characterize the probabilistic distribution of labelings.

In the future, we will study the possibility of extending these algorithms to more general graphs. Extending to trees is already nontrivial. It seems unavoidable to have the complexity exponential to some parameter, e.g. the tree degree. We would also be interested in exploring the possibility of reducing the problem into a sequence of MAP inferences, and use the state-of-the-art approximation algorithms, e.g. α-expansion (Boykov et al., 2001), like M-Diverse.

Appendix A: Proof of Theorem 2

We first prove the necessity. Suppose x is a mode and there is an interval $[i, j]$ on which $x_{i:j} \neq x^{opt}_{i:j}(x_i, x_j)$. Consider labeling $y = (x_{1:i-1}, x^{opt}_{i:j}(x_i, x_j), x_{j+1:N})$. We have $f(x^{opt}_{i:j}(x_i, x_j)) < f(x_{i:j})$ and thus $f(y) < f(x)$. Also, y is at most δ away from x in the Hamming distance. This contradicts the fact that x is a mode.

Now suppose that x is not a mode. Then there exists another labeling y with $f(y) < f(x)$ such that $\mathrm{d_H}(x, y) \leq \delta$. Let $[a, b]$ be a maximal interval within which x and y have different labels on every node. This interval is at most δ long. We can find a length $\delta+2$ interval $[i, j]$ containing $[a, b]$ such that x and y have the same label on i and j. Condition $f(y) < f(x)$ implies that $f(y_{i:j}) < f(x_{i:j})$, and therefore $x_{i:j} \neq x^{opt}_{i:j}(x_i, x_j)$. This concludes the proof of the theorem.

Acknowledgements

We thank reviewers for helpful comments. We thank Dhruv Batra and Filip Korč for discussions.

References

CoNLL 2000 shared task: Chunking. http://www.cnts.ua.ac.be/conll2000/chunking/. Accessed: Nov. 2012.

ChaLearn gesture challenge. https://sites.google.com/a/chalearn.org/gesturechallenge/. Accessed: Nov. 2012.

B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. The lazy flipper: Efficient depth-limited exhaustive search in discrete graphical models. In *ECCV*, 2012.

D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-best solutions in Markov random fields. In *ECCV*, 2012.

Dhruv Batra. An efficient message-passing algorithm for the M-best MAP problem. In *UAI*, 2012.

J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

M. Blaschko. Branch and bound strategies for non-maximal suppression in object detection. In *EMMCVPR*, 2011.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23 (11):1222–1239, 2001.

Y. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 17(8):790–799, 1995.

H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. AMS, 2010.

P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9): 1627–1645, 2010.

M. Fromer and A. Globerson. An LP view of the M-best MAP problem. In *NIPS*, 2009.

M. Fromer and C. Yanover. Accurate prediction for atomic-level protein design and its application in diversifying the near-optimal sequence space. *Proteins: Structure, Function, and Bioinformatics*, 75 (3):682–705, 2009.

R.H. Kassel. *A comparison of approaches to on-line handwritten character recognition*. PhD thesis, Massachusetts Institute of Technology, 1995.

C.H. Lampert. Maximum margin multi-label structured prediction. *NIPS*, 2011.

I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

Y. Leung, J.S. Zhang, and Z.B. Xu. Clustering by scale-space filtering. *PAMI*, 22(12):1396–1410, 2000.

D. Nilsson. An efficient algorithm for finding the M most probable configurationsin probabilistic expert systems. *Statistics and Computing*, 8(2):159–173, 1998.

S. Nowozin and C.H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4): 185–365, 2010.

D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011.

M. Schmidt. UGM: matlab code for undirected graphical models. http://www.di.ens.fr/~mschmidt/Software/UGM.html. Accessed: Nov. 2012.

G.J. Stephens, T. Mora, G. Tkacik, and W. Bialek. Thermodynamics of natural images. *arXiv:0806.2694 [q-bio.NC]*, 2008. URL http://arxiv.org/abs/0806.2694.

P. Viola and M.J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

P. Yadollahpour, D. Batra, and G. Shakhnarovich. Diverse M-best solutions in MRFs. In *Workshop on Discrete Optimization in Machine Learning, NIPS*, 2011.

C. Yanover and Y. Weiss. Finding the M most probable configurations using loopy belief propagation. In *NIPS*, 2004.

N. Ye, W.S. Lee, H.L. Chieu, and D. Wu. Conditional random fields with high-order features for sequence labeling. In *NIPS*, 2009.

Yisong Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *ICML*, 2008.