Abstract

Variants of max-product (MP) belief propagation effectively find modes of many complex graphical models, but are limited to discrete distributions. Diverse particle max-product (D-PMP) robustly approximates max-product updates in continuous MRFs using stochastically sampled particles, but previous work was specialized to tree-structured models. Motivated by the challenging problem of protein side chain prediction, we extend D-PMP in several key ways to create a generic MAP inference algorithm for loopy models. We define a modified diverse particle selection objective that is provably submodular, leading to an efficient greedy algorithm with rigorous optimality guarantees, and corresponding max-marginal error bounds. We further incorporate tree-reweighted variants of the MP algorithm to allow provable verification of global MAP recovery in many models. Our general-purpose MATLAB library is applicable to a wide range of pairwise graphical models, and we validate our approach using optical flow benchmarks. We further demonstrate superior side chain prediction accuracy compared to baseline algorithms from the state-of-the-art Rosetta package.

1. Introduction

Continuous random variables are often used to model complex interactions among objects in the world around us, leading to challenging multi-modal posterior distributions. The maximum a posteriori (MAP) inference objective for such models is typically non-convex, and optimization algorithms become trapped in local optima. Approaches that discretize the latent space and apply max-product (MP) belief propagation (Pearl, 1988; Wainwright et al., 2005; Wainwright & Jordan, 2008) can be effective in few dimensions, but for high-dimensional models only coarse discretizations are feasible. Continuous optimization can be performed via Monte Carlo sampling and simulated annealing (Geman & Geman, 1984; Andrieu et al., 2003), but these methods often require long computation times.

A number of stochastic local search methods have been developed (Trinh & McAllester, 2009; Peng et al., 2011; Besse et al., 2012; Kothapa et al., 2011) that combine the flexibility of sampling-based approaches with the efficiency of MP message passing. This family of particle max-product (PMP) methods share a general framework: at each iteration new hypotheses are sampled from stochastic proposals, evaluated via discrete max-product message updates, and accepted or rejected based on some criterion. PMP algorithms differ primarily in their choice of stochastic proposals and particle selection criteria.

The diverse particle max-product (D-PMP) (Pacheco et al., 2014) algorithm maintains hypotheses near multiple local optima via an optimization-based selection step that minimizes distortions in MP message values. D-PMP has excellent empirical performance on a human pose estimation task, but there is little theoretical justification for its particle selection integer program (IP), and the proposed greedy algorithm has no optimality guarantees. Previous D-PMP formulations also assumed a tree-structured Markov random field (MRF) where MP provides exact max-marginals, and several key assumptions would be violated by a naive generalization to loopy graphical models.

In this paper, we generalize D-PMP to arbitrary pairwise MRFs with cycles by adapting tree-reweighted max-product (RMP) belief propagation (Wainwright et al., 2005). We define an alternative message distortion metric which leads to a submodular particle selection IP. An efficient greedy algorithm is guaranteed to produce message errors within a fraction of \( (1 - \frac{1}{e}) \) of the best achievable, and thus provide provably accurate max-marginal estimates. Our MATLAB library implements the D-PMP algorithm for general pairwise MRFs. For the tasks of optical flow estimation and protein side chain prediction, we demonstrate substantial improvements over previous PMP algorithms, and performance levels that match or exceed state-of-the-art domain-specific inference algorithms.
2. Background

We begin with a brief introduction to max-product inference for discrete MRFs, which forms the basis for our particle-based approximations. To ground these concepts we introduce the protein side chain prediction task, which will be used to motivate and validate our approach.

2.1. Max-Product Belief Propagation

Max-product belief propagation (Pearl, 1988; Aji & McEliece, 2000; Wainwright & Jordan, 2008) performs MAP inference by passing messages along the edges of a graphical model. Consider a pairwise MRF, with edges \((s, t) \in \mathcal{E}\) and nodes \(s \in \mathcal{V}\):

\[
p(x) \propto \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s, t) \in \mathcal{E}} \psi_{st}(x_s, x_t).
\]

(1)

In MRFs with cycles, tree-reweighted max-product (RMP) (Wainwright et al., 2005) approximates MAP inference via a set of spanning trees, with edge appearance probabilities \(\rho_{st}\). The RMP message from node \(t\) to \(s\) is:

\[
\bar{\nu}_s(x_s) \propto \psi_s(x_s) \prod_{u \in \Gamma(t) \setminus s} \bar{m}_{ut}(x_t)^{\rho_{ut}} \bar{m}_{st}(x_t)^{1-\rho_{st}},
\]

(2)

where \(\Gamma(t)\) is the set of nodes neighboring \(t\). Fixed points yield pseudo-max-marginals, which do not necessarily correspond to valid max-marginal distributions:

\[
\bar{\nu}_s(x_s) \propto \psi_s(x_s) \prod_{u \in \Gamma(s)} \bar{m}_{us}(x_u)^{\rho_{us}} \approx \max_{x' \mid x'_s = x_s} p(x').
\]

Via connections to linear programming relaxations, RMP provides a bound on the MAP probability at each iteration, and a certificate of optimality using Lagrange multipliers.

2.2. Particle Max-Product Belief Propagation

For continuous variables \(x \in \mathcal{X}\), the message functions of Eq. (2) cannot be computed in general. Particle max-product (PMP) methods approximate messages by optimizing over a discrete set of \(N\) particles found via stochastic search. Given a current set of \(N\) particles \(x^N\), each PMP iteration has three stages, summarized in Figure 1.

Stochastic Proposals To allow higher-likelihood state configurations to be discovered, at each iteration PMP first creates an augmented particle set \(x^\text{aug} = \mathcal{X} \cup x^\text{prop}\) of size \(\alpha N, \alpha > 1\). New particles are drawn from proposal distributions \(x^\text{prop} \sim q(x)\). In the simplest case, Gaussian random walk proposals \(q^\text{rw}(x_s | x_s, \Sigma)\) sample perturbations of current particle locations \(\bar{x}_s\) (Trinh & McAllester, 2009; Peng et al., 2011). For some models, a more informative neighbor-based proposal is possible that samples from edge potentials \(q^\text{nb}(x_s | \bar{x}_t) \propto \psi_{st}(x_s, \bar{x}_t)\) conditioned on a particle \(\bar{x}_t\) at neighboring node \(t \in \Gamma(s)\) (Besse et al., 2012). Specialized “bottom-up” proposals based on approximations of observation potentials \(\psi(x_s)\) can also be effective (Pacheco et al., 2014).

Max-Product Optimization Standard or reweighted MP message updates are used to approximate the max-marginal distribution of each proposed particle. The \(\alpha N\) values of each discrete message vector satisfy \(m_{ts}(x_s) =\)

\[
\max_{x_t} \nu_t(x_t) \psi_{st}(x_s, x_t) \frac{\prod_{u \in \Gamma(t) \setminus s} \bar{m}_{ut}(x_t)^{\rho_{ut}}}{\bar{m}_{st}(x_t)^{1-\rho_{st}}}.
\]

(2)

Message updates require \(O(N^2)\) operations, and compute the pseudo-max-marginal \(\bar{\nu}_s(x_s)\) for each \(x_s \in x^\text{aug}\).

Particle Selection Particles are accepted or rejected to yield \(N\) new states \(x^\text{new} \subset x^\text{aug}\). Particle selection makes subsequent iterations more computationally efficient.

The simple greedy PMP (G-PMP) method selects the single particle \(x^*_s = \arg \max_{x_s \in x^\text{aug}} \nu_s(x_s)\) with the highest max-marginal, and samples all other particles as Gaussian perturbations of this state (Trinh & McAllester, 2009; Peng et al., 2011). G-PMP updates are efficient, but they cannot preserve multiple modes, and random walk proposals do not effectively explore high-dimensional spaces.

A less greedy selection method retains the \(N\) particles with highest estimated max-marginal probability. This top-\(N\) PMP (T-PMP) (Pacheco et al., 2014) generalizes PatchMatch BP (Besse et al., 2012), a method specialized to low-level vision tasks which utilizes top-\(N\) particle selection and neighbor proposals. T-PMP finds high probability solutions quickly, but the top-\(N\) particles are often slight perturbations of the same solution, reducing the number of effective particles and causing sensitivity to initialization.
To avoid the particle degeneracy common to G-PMP and T-PMP, the diverse PMP (D-PMP) (Pacheco et al., 2014) method selects particles via combinatorial optimization. An IP favors particles that minimally distort the current MP messages, and thus implicitly encourages diversity. By preserving solutions near multiple local optima, D-PMP reasons more globally and is less sensitive to initialization.

2.3. Protein Side Chain Prediction

Predicting protein structure, which is governed by pairwise energetic interactions, is a natural target for PMP algorithms (Peng et al., 2011; Soltan Ghorai et al., 2013). Proteins are formed by chains of amino acids which consist of a backbone and a side chain unique to each amino acid type. Each protein assumes a 3D structure, or conformation, relating to its function. Given an amino acid sequence and a set of observed dihedral angles. More details are in Sec. 4.2.

Here atoms do not interact. Log-likelihoods encode interactions between nearby side chains via the attractive and repulsive components of the “6-12 Lennard-Jones” log-potential:

\[
\log \psi_s(x_s, x_t) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} 4\epsilon \left( \frac{\sigma}{r_{ij}} \right)^6 - \left( \frac{\sigma}{r_{ij}} \right)^{12}.
\]

Here \(N_s\) is the number of atoms in the \(s^{th}\) amino acid, \(\epsilon\) controls the strength of attraction and \(\sigma\) the cutoff distance where atoms do not interact. Log-likelihoods are given by a Gaussian mixture fit to the marginal statistics of observed dihedral angles. More details are in Sec. 4.2.

3. Loopy Diverse Particle Max-Product

The D-PMP message updates of Pacheco et al. (2014) can be directly applied to loopy MRFs, since each step decomposes into local computations on the graph. But, a naive extension may have convergence problems like those served for loopy MP in many discrete models. Using RMP message passing, combined with our method for resolving ties, we can verify that global optimality is achieved and ensure that the MAP estimate is nondecreasing. We also introduce a new IP objective in the particle selection step which is a monotonic submodular maximization. This IP allows us to use a standard greedy algorithm for particle selection, and attain a multiplicative optimality bound.

3.1. Submodular Particle Selection

For each node \(t \in \mathcal{V}\) we select particles to minimize the distortion between two message vectors. Specifically, we choose a subset of particles which minimizes the \(L_1\) norm,

\[
\min_{z} \sum_{s \in \Gamma(t)} \|m_{ts} - \hat{m}_{ts}(z)\|_1 \quad (4)
\]

subject to \(\|z\|_1 \leq N\), \(z \in \{0, 1\}^{\alpha N}\).

The message vector \(m_{ts}\) is computed over the augmented particles \(X_t^{\text{aug}} = \{x_t^{(1)}, \ldots, x_t^{(\alpha N)}\}\), with \(\alpha > 1\). The message vector \(\hat{m}_{ts}(z)\) is computed over any subset of at most \(N\) particles \(X_t^{\text{new}} \subset X_t^{\text{aug}}\) indexed by the indicator vector, \(z\).

\[
m_{ts}(a) = \max_{b \in \{1, \ldots, \alpha N\}} M_{st}(a, b), \quad (5)
\]

\[
\hat{m}_{ts}(a; z) = \max_{b \in \{1, \ldots, \alpha N\}} z(b) M_{st}(a, b). \quad (6)
\]

Here we have accumulated the terms needed for RMP message updates in a message foundation matrix \(M_{st}(a, b) = \frac{\psi_t(x_t^{(b)}) \psi_s(x_s^{(a)}, x_t^{(b)})}{\prod_{u \in \Gamma(t) \setminus s} m_{ut}(b)^{\rho_{st}} m_{st}(b)^{1 - \rho_{st}}} \). (7)

Pseudo-Max-Marginal Bounds

Particles are chosen to minimize message distortions, but our primary goal is to maintain approximations of the pseudo-max-marginals:

\[
\nu_s(x_s) \propto \psi_s(x_s) \prod_{u \in \Gamma(s)} m_{us}(x_s)^{\rho_{su}},
\]

and analogously for pseudo-max-marginals on the selected particles \(\hat{\nu}\). If the potentials are bounded above and normalized so that \(0 \leq \psi \leq 1\), then the sum of message distortions bounds the pseudo-max-marginal error.

Proposition 1. Let \(0 \leq \hat{m} \leq m \leq 1\) and edge appearance probabilities \(\rho_{st} \in [0, 1]\). For all nodes \(s \in \mathcal{V}\) we have:

\[
\|\nu_s - \hat{\nu}_s\|_1 \leq \sum_{t \in \Gamma(s)} \|m_{ts} - \hat{m}_{ts}\|_1^{\alpha_s}. \quad (9)
\]

We provide a proof in the Appendix. Note that we do not bound the difference between the D-PMP max-marginals \(\nu_s(x_s)\) and the continuous max-marginals \(\hat{\nu}_s(x_s)\); such results typically require strong and unrealistic assumptions (Peng et al., 2011). Instead, Eq. (9) shows that if we succeed in producing small message errors, the particle selection step will not significantly distort the pseudo-max-marginals, nor will it discard important hypotheses.
Submodularity The particle selection IP we propose minimizes the absolute sum of message differences (the $L_1$ norm). In contrast, the selection objective proposed by Pacheco et al. (2014) minimizes the maximum message difference (the $L_{\infty}$ norm). With this modified objective metric, minimizing Eq. (4) is equivalent to maximizing a monotonic submodular function.

**Definition 1** (Submodularity). A set function $f : 2^Z \rightarrow \mathbb{R}$ defined over subsets of $Z$ is submodular iff for any subsets $Y \subseteq X \subseteq Z$ and an element $e \notin X$ the function $f$ satisfies,

$$f(Y \cup \{e\}) - f(Y) \geq f(X \cup \{e\}) - f(X).$$

The quantity $\Delta(Y, e) \triangleq f(Y \cup \{e\}) - f(Y)$ is the margin, and Def. (1) states that for any $e \notin X$ the margin is non-increasing in $|Y|$. This property of diminishing marginal returns allows us to use efficient greedy algorithms.

**Proposition 2.** The optimization of Equation (4) is equivalent to maximizing a monotonic submodular objective subject to cardinality constraints.

**Proof.** We focus on a single node and drop subscripts. Dropping constants, we can minimize Eq. (4) as follows:

$$\arg \max_{z\mid \|z\|_1 \leq N} \sum_a F_a(z) = \sum_a \left[ \max_{1 \leq b \leq N} z(b) M(a, b) \right].$$

Let $y, z \in \{0, 1\}^{aN}$ be particle selections and $y \subseteq z$ such that $y(b) = 1 \Rightarrow z(b) = 1$. For some candidate particle $\tilde{b}$:

$$\overline{\gamma}(b) = \begin{cases} 1, & \text{if } b = \tilde{b} \\ y(b), & \text{otherwise} \end{cases} \quad \overline{\tau}(b) = \begin{cases} 1, & \text{if } b = \tilde{b} \\ z(b), & \text{otherwise} \end{cases}$$

The margins are given by direct calculation:

$$F_a(\overline{\gamma}) - F_a(y) = \max(0, M(a, \tilde{b}) - \tilde{m}(a; y))$$

$$F_a(\overline{\tau}) - F_a(z) = \max(0, M(a, \tilde{b}) - \tilde{m}(a; z)).$$

Since $\gamma \subseteq z$ we have that $F_a$ is submodular,

$$F_a(\overline{\gamma}) - F_a(y) \geq F_a(\overline{\tau}) - F_a(z).$$

A sum of submodular functions is submodular, and monotonicity holds since $\tilde{m}(y) \leq \tilde{m}(z)$.

### 3.2. LazyGreedy Particle Selection

The LazyGreedy algorithm exploits diminishing marginal returns to avoid redundant computations (Mionux, 1978; Leskovec et al., 2007). Each iteration updates and sorts the largest margin until a stable maximizer is found. The algorithm terminates when the particle budget is exhausted, or the maximum margin is zero. Surprisingly, this greedy approach yields solutions within a factor $(1 - \frac{1}{e}) \approx 0.63$ of optimal (Nemhauser et al., 1978).

**Initialize:** For each node $t$ let $M = [M_{s_1t}, \ldots, M_{s_Tt}]^T$ be the message foundations of neighbors $\Gamma(t) = \{s_1, \ldots, s_T\}$. Initialize the selection vector $z$ and margins:

$$\Delta(b) = \sum_{a=1}^{\alpha N} M(a, b), \quad z(b) = 0 \forall b \in \{1, \alpha N\}. \quad (11)$$

**First Iteration:** Ensure that the current MAP estimate $z^*$ is never discarded by setting $z(b^*) = 1$, where $b^*$ is the index of $x^*_t$ in the augmented particle set $\mathcal{X}_t$ (see Sec. 3.3). Update the message approximation $\tilde{m}(a) = M(a, b^*)$.

**Iterations 2 to $N$:** Choose the largest margin to update,

$$\tilde{b} = \arg \max_{\{b|z(b)=0\}} \Delta(b).$$

If $\Delta(\tilde{b}) = 0$ then terminate prematurely, the message can be perfectly reconstructed with a subset of particles. If $\Delta(\tilde{b})$ has already been updated on the current iteration then set $z(\tilde{b}) = 1$ and update the message approximation,

$$\tilde{m}(a) = \max(\tilde{m}(a), M_t(a, \tilde{b})).$$

Otherwise, update the margin and repeat,

$$\Delta(\tilde{b}) \triangleq \sum_a \left[ \max(\tilde{m}(a), M(a, \tilde{b})) - \tilde{m}(a) \right].$$

Selections are performed in parallel and updates at one node do not affect the selection at neighboring nodes. Figure 3 graphically demonstrates LazyGreedy selection on the small toy protein of Fig. 2.
3.3. Resolving ties

In PMP we resolve ties using an approach similar to one proposed for discrete MRFs (Weiss et al., 2007). For discrete models the RMP pseudo-max-marginals \( \nu \) admit a provably MAP solution \( x^\star \) if a consistent labeling exists in the set of maxima (Wainwright et al., 2005):

\[
x^\star_s \in \arg \max_{x_s} \nu_s(x_s), \quad (x^\star_s, x^\star_t) \in \arg \max_{x_s,x_t} \nu_{st}(x_s, x_t).
\]

For continuous distributions exact ties rarely exist, but small numerical errors in the estimated pseudo-max-marginals can perturb the particle that is inferred to be most likely, and lead to joint states with low probability due to “conflicted” edges. This problem is common in the side chain model, and as illustrated in Fig. 4, the diversity in the D-PMP particles makes conflicts more likely. To address this we relax the set of optima to be states with pseudo-max-marginals within tolerance \( \epsilon \) of the maximum:

\[
\mathrm{OPT}(\nu_s) \triangleq \{ x^\star_s : |\nu_s(x^\star_s) - \arg \max_{x_s} \nu_s(x_s) | \leq \epsilon \}.
\]

Let \( V_T \) be the set of tied nodes with more than one near-maximal state, and \( E_T \equiv E \cap (V_T \otimes V_T) \) the edges joining them. Let \( x^\star_{V_T} \) be the unique assignments for non-tied nodes. Construct an MRF over the remaining tied nodes as

\[
\begin{align*}
p_T(x_T) \propto \prod_{s \in V_T} \tilde{\psi}_s(x_s) \prod_{(s,t) \in E_T} \psi_{st}(x_s, x_t), \quad (12)
\end{align*}
\]

with the conditioned node potentials

\[
\tilde{\psi}_s(x_s) = \psi_s(x_s) \prod_{t \in P(s) \setminus V_T} \psi_{st}(x_s, x_t^\star). \quad (13)
\]

We label the remaining nodes \( x^\star_T = \arg \max_{x_T} p_T(x_T) \) using the junction tree algorithm. If the junction tree contains a unique maximizer, then \( x^\star = (x^\star_T, x^\star_{NT}) \) is the global MAP over the particles \( X \). This guarantee follows from the reparameterization property of pseudo-max-marginals and Theorem 2 of Weiss et al. (2007). Clique size is reduced by eliminating non-tied nodes, and by constraining labels to the set of tied states \( x^\star_T \in \mathrm{OPT}(\nu_s) \).

4. Experimental Results

We consider two tasks that demonstrate the effectiveness and flexibility of D-PMP inference. We begin with optical flow estimation, a low-level vision task which recovers 2D pixel motion in an image sequence. Optical flow is a well-studied problem where specialized inference methods are thought to be near-optimal for the model and dataset we consider, and so provide a good comparison. We then revisit our running example of protein side chain prediction, which is more challenging due to increased dimensionality and complex potentials. Many methods for side chain prediction make coarse discrete approximations to speed up computation, and we show significant improvement using D-PMP to optimize the continuous energy model.

4.1. Optical Flow

Given a pair of (grayscale) images \( I_1 \) and \( I_2 \) in \( \mathbb{R}^{M \times N} \), we estimate the motion of each pixel \( s \) from one image to the next. This flow vector \( x_s \) is decomposed into horizontal \( u \) and vertical \( v \) scalar components. The model presented below is based on the Classic-C method (Sun et al., 2014). To reduce the number of edges we model flow at the superpixel level, holding flow constant over each superpixel. Edges are given by the immediate neighbors in \( I_1 \).

The pairwise log-potential imposes a smoothness prior on flow vectors. We use the robust Charbonnier penalty, a differentiable approximation to \( L_1 \), which is approximately quadratic in the range \( [-\sigma, \sigma] \) and smoothly transitions to a linear function outside this range. The potential decomposes additively as \( \log \psi_{st} = \phi^\text{ver}_{st} + \phi^\text{hor}_{st} \) into vertical and horizontal components, defined as follows:

\[
\phi^\text{ver}_{st}(u_s, u_t) = -\lambda_s \sqrt{\sigma^2 + (u_s - u_t)^2}, \quad (14)
\]

The spatial smoothness depends on scaling parameter \( \lambda_s \),
Figure 6. Preserving multiple hypotheses. Top Row: Final flow estimate of each method for the “Rubber Whale” sequence. The color key (top-right) encodes flow vector orientation, color saturation denotes magnitude. Bottom Row: Detail of highlighted region showing selected flow particles as vectors (black) and the MAP label (red). The MAP estimates of D-PMP and T-PMP have higher probability than ground truth, but D-PMP preserves the correct flow in the particle set.

Figure 7. Optical flow results. Left: Log-probability quantiles showing median (solid) and best/worst (dashed) MAP estimates versus PMP iteration for 11 random initializations on the Middlebury training set. Left-Center: Oracle AEPE over the training set. Right-Center: Log-probability quantiles on the test set (G-PMP omitted due to poor performance on training). Right: Log-probability of the MAP estimates at the pixel-level model obtained by initializing L-BFGS at the D-PMP solution.

Results We evaluate on the Middlebury optical flow benchmark (Baker et al., 2011) using 11 random initializations. D-PMP and T-PMP utilize the same set of proposals (75% neighbor, 25% random walk). We compute SLIC superpixels (Achanta et al., 2012) with region size 5 and regularizer 0.1; about 5,000 to 15,000 per image. We use the Charbonnier widths $\sigma = 0.001$ recommended for this model (Sun et al., 2014), but learn different scaling parameters ($\lambda_s = 16$, $\lambda_d = 1$) to compensate for our superpixel representation.

The Middlebury training set contains 8 images with ground truth flow, and we report log-probability quantiles over this set (Fig. 7 (left)). To demonstrate diversity in the particle sets we report average endpoint error (AEPE) of the oracle solution—we choose the flow particle closest to ground truth in the order given by the particle selection step.

![Diagram](image_url)

Table 1. Optical flow MAP estimates. Average log-probability and AEPE over 11 random initializations on the Middlebury training set. Reported $p$ values are compared to D-PMP using a Wilcoxon signed rank test, we consider $p < 0.05$ significant.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. Log-Prob. ($p$ value)</th>
<th>Avg. EPE ($p$ value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMP</td>
<td>-2.446E6 (0.008)</td>
<td>1.623 (0.008)</td>
</tr>
<tr>
<td>G-PMP</td>
<td>-1.408E6 (0.008)</td>
<td>0.699 (0.008)</td>
</tr>
<tr>
<td>T-PMP</td>
<td>-1.212E6 (0.008)</td>
<td>0.382 (0.727)</td>
</tr>
<tr>
<td>D-PMP</td>
<td>-1.209E6 (−)</td>
<td>0.362 (−)</td>
</tr>
<tr>
<td>Classic-C</td>
<td>−</td>
<td>0.349 (0.727)</td>
</tr>
</tbody>
</table>
Proteins, Particles, and Pseudo-Max-Marginals: A Submodular Approach

Figure 8. Side chain prediction. We compare each method and both $L_1$ and $L_{\infty}$ diverse selection methods. Left: Total log-probability over 20 proteins. Median (solid) and best/worst (dashed) results on 11 random initializations. Left-Center: Total log-probability for 370 proteins. Right-Center: RMSD (in angstroms Å) of the oracle solution on larger set. Right: Log-probability of all 370 proteins versus the fixed rotamer discretization with RMP inference.

(Fig. 7 (left-center)). D-PMP shows a large reduction in AEPE after just a few particles. T-PMP remains nearly flat, suggesting little diversity. In just two dimensions the Gaussian spread of G-PMP particles naturally leads to an error reduction, although higher. The benefit of particle diversity is best visualized near object boundaries (see Fig. 6).

We compare to a specialized coarse-to-fine, multiscale inference algorithm for Classic-C, using default settings and with the median filter disabled. We also compare to RMP on a fixed regular discretization of 200 flow vectors. As shown in Table 1, D-PMP yields significantly higher probability solutions, but is equivalent to T-PMP in AEPE. D-PMP also achieves equivalent results to Classic-C optimization, which is highly tuned to the Middlebury dataset.

We cannot directly compare probability of the Classic-C and D-PMP solutions, because the former models flow at the pixel level. Instead, using L-BFGS initialized from the D-PMP solution, we optimize the pixel level model and compare log-probability of the result with Classic-C for both training and test sequences (Fig. 7 (right)). Again, even compared to a highly-tuned specialized optimization method, D-PMP achieves statistically equivalent results.

4.2. Protein Side Chain Prediction

Most computational approaches optimize side chain placement over a standard discretization, known as a rotamer library (Bower et al., 1997; Fromer et al., 2010). Rotamer configurations are learned from the marginal statistics of experimentally validated side chains and generally allocate three states {60°, 180°, 300°} for each dihedral angle, resulting in up to 81 possible states per node. This is a coarse discretization which can fail to capture important details of the side chain placement. Applying D-PMP we optimize the continuous energy function, allowing estimation of non-rotameric side chains which do not obey the standard discretization (see Fig. 9). Log-likelihoods are the so-called Dunbrack probabilities—Gaussian mixtures with

1. http://people.seas.harvard.edu/~dqsun

Experiments use code accessed on 06 February 2015.

Results We evaluate the energy function with Rosetta (Rohl et al., 2004), a state-of-the-art molecular modeling package. We configure the Rosetta energy functions using three terms: the Lennard-Jones attractive and repulsive (fa_atr, fa_rep) terms and the Dunbrack probabilities (fa_dun), each with unit weight. We run PMP with 50 particles for 50 iterations. D-PMP and T-PMP proposals are 50% random walks from Gaussians wrapped to account for angular discontinuities, and 50% samples from the rotamer marginals. Neighbor-based proposals are not used, due to the complex transformation between dihedral angles and atom locations. We compare to Rosetta’s implementation of simulated annealing using Metropolis-Hastings proposals from the discrete rotamers, followed by local continuous optimization.

We experiment on two sets of proteins selected from the Protein Data Bank1, resolved using X-ray diffraction below 1.5-Å resolution, and less than 1000 amino acids. We run each method from 11 random initialization on a small set (20 proteins) and report quantiles of total log-probability (Fig. 8 (left)). Both D-PMP and T-PMP outperform G-PMP, due to their ability to explore the model likelihood through rotamer proposals, with D-PMP showing the tightest confidence intervals. The second set is larger (370 proteins) and we report the total log-probability of a single run.

Diversity is important in structure prediction, since proteins are known to alternate between many stable configurations (Ma et al., 1999). Fig. 10 shows a qualitative comparison of diversity between D-PMP and T-PMP for a single protein. To measure diversity we report RMSD of the oracle solution (Fig. 8); D-PMP shows a substantial improvement in accuracy after only a few particles. We also compare the submodular particle selection (L₁) with the minimax formulation (L∞); both preserve diversity similarly, but the former offers stronger theoretical justification.

5. Discussion

We have generalized previous PMP algorithms in several substantial ways. Our proposed extensions to D-PMP not only allow inference in loopy MRFs, but our reformulation of the particle selection IP allows for greedy optimization within a guaranteed optimality bound. We demonstrate effectiveness in protein structure prediction, where we are substantially more accurate than the G-PMP algorithm that Peng et al. (2011) applied to a broader structure prediction task, and the state-of-the-art Rosetta package. The same general-purpose D-PMP algorithm is also competitive with standard inference algorithms for a very loopy optical flow model. A MATLAB library, built on UGM (Schmidt, 2007), implementing these methods is available.

Acknowledgements We thank Silvia Zuffi for her advice about connections between submodularity and the particle selection problem. This research supported in part by ONR Award No. N00014-13-1-0644. J. Pacheco supported in part by funding from the Naval Undersea Warfare Center, Division Newport, Rhode Island.

Appendix. Proof of Proposition 1

To simplify we ignore normalization terms and drop dependence on z so \( \hat{m}(z) = \hat{m} \). The proof is by induction on the number of neighbors, for the base case let \( \Gamma(s) = \{i, j\} \):

\[
\|\nu_s - \hat{\nu}_s\|_1 \leq \sum_{x_s} \left( (m_{is}(x_s)^{\rho_{is}} - \hat{m}_{is}(x_s)^{\rho_{is}}) m_{js}(x_s)^{\rho_{js}} + (m_{js}(x_s)^{\rho_{js}} - \hat{m}_{js}(x_s)^{\rho_{js}}) \hat{m}_{is}(x_s)^{\rho_{is}} \right)
\]

\[
\leq \sum_{x_s} \left( (m_{is}(x_s)^{\rho_{is}} - \hat{m}_{is}(x_s)^{\rho_{is}}) \right) + \sum_{x_s} \left( (m_{js}(x_s)^{\rho_{js}} - \hat{m}_{js}(x_s)^{\rho_{js}}) \right)
\]

\[
\leq \|m_{is} - \hat{m}_{is}\|_{1\rho_{is}} + \|m_{js} - \hat{m}_{js}\|_{1\rho_{js}}
\]

The first inequality drops \( \psi_s \in [0, 1] \), and \( |\cdot| \) since \( \hat{m}_{is} \leq m_{is} \leq \hat{m}_{is} \). The second inequality holds since \( m, \hat{m} \in [0, 1] \). The last follows from the triangle inequality since, \( |x-y| \) is a metric (though not a norm for \( \rho \in (0, 1) \)). For the inductive step let \( \Gamma(s) = \{t_1, \ldots, t_n\} \) and assume:

\[
\|\nu_s^{\cap n} - \hat{\nu}_s^{\cap n}\|_1 \leq \sum_{i \neq n} \|m_{t_is} - \hat{m}_{t_is}\|_{1\rho_{t_is}}
\]

where \( \nu_s^{\cap n}(x_s) \) is the product of all messages except \( m_{t_is} \):

\[
\|\nu_s - \hat{\nu}_s\|_1 \leq \|m_{t_is} - \hat{m}_{t_is}\|_{1\rho_{t_is}} + \|\nu_s^{\cap n} - \hat{\nu}_s^{\cap n}\|_1
\]

\[
\leq \sum_{i=1}^n \|m_{t_is} - \hat{m}_{t_is}\|_{1\rho_{t_is}}.
\]
References


