An Incremental Subgradient Algorithm for MAP Estimation in Graphical Models

Jeremy Jancsary **Gerald Matz**[1] **Harald Trost**[2]

jeremy.jancsary@ofai.at

December 8, 2010

[1]Vienna University of Technology
[2]Medical University of Vienna

A Few Things to Take Away From This Talk

- Why you might be interested in our algorithm:
  - It is efficient, both computationally and memory-wise.
  - It finds better solutions than the methods we compared it to.
  - It converges to the global optimum of the first-order LP relaxation of the MAP problem (which is tight in some cases).
  - You get a certificate of optimality w.r.t. the discrete problem.
- How it all works:
  - Start out with tree-reweighted upper bound (Wainwright et al., 2005).
  - The upper bound is developed until it assumes a degenerate form involving a large number of easy problems.
  - The *tightest* bound can then be found very efficiently using incremental methods, solving one easy problem at a time.
  - Equivalent to maximizing the LP relaxation of the MAP problem.

Outline

Outline

Maximum-a-Posteriori (MAP) Estimation

Consider an undirected graphical model (e.g. ) with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ defined over discrete random variables with pairwise interactions. Potential of a particular variable state $\boldsymbol{x} \in \mathcal{X}^n$:

$$P(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{s\in\mathcal{V}} \theta_s(x_s) + \sum_{(s,t)\in\mathcal{E}} \theta_{st}(x_s, x_t).$$

MAP Estimation (Discrete Problem) (OP1)

$$\bar{P}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}\in\mathcal{X}^n} P(\boldsymbol{x};\boldsymbol{\theta}),$$
$$\bar{\boldsymbol{x}} = \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}^n} P(\boldsymbol{x};\boldsymbol{\theta}).$$

Computation of these quantities is NP-hard in general (notable exceptions: trees; binary variables + submodular energies).

Outline

Relaxing the Discrete Problem

Tightest Tree-Reweighted Upper Bound (OP2)

$$\min_{\{\boldsymbol{\theta}^T\}\in\mathcal{C}(\boldsymbol{\theta})} \sum_T \rho^T \bar{P}(\boldsymbol{\theta}^T),$$

$$\mathcal{C}(\boldsymbol{\theta}) = \left\{\{\boldsymbol{\theta}^T\} \,|\, \sum_T \rho^T \boldsymbol{\theta}^T = \boldsymbol{\theta}\right\} \text{ and } \{\rho^T\} \in \left\{\boldsymbol{\rho} \geq \mathbf{o} \,|\, \sum_T \rho^T = 1\right\}.$$

is connected through strong duality to

First-Order LP Relaxation (OP3)

$$\max_{\boldsymbol{\mu}\in\mathcal{L}} \sum_s \boldsymbol{\mu}_s \cdot \boldsymbol{\theta}_s + \sum_{(s,t)} \boldsymbol{\mu}_{st} \cdot \boldsymbol{\theta}_{st},$$

$$\mathcal{L} = \left\{ \boldsymbol{\mu} \geq \mathbf{o} \;\middle|\; \begin{array}{l} \sum_{x_s} \mu_s(x_s) = 1 \text{ for all } s \\ \sum_{x'_s} \mu_{st}(x'_s, x_t) = \mu_t(x_t), \sum_{x'_t} \mu_{st}(x_s, x'_t) = \mu_s(x_s) \end{array} \right\}.$$

Simplifying the Upper Bound (1)

- Curious fact #1 (Wainwright et al., 2005): Choice of $\boldsymbol{\rho}$ irrelevant as long as all edges are covered (otherwise, $\mathcal{C}(\boldsymbol{\theta})$ is empty).

  Minimization of tree-reweighted upper bound $\equiv$ maximization of LP relaxation, which does not depend on $\boldsymbol{\rho}$.
- Can pick small set $S(\mathcal{T})$ of trees needed to cover all edges and set $\rho^T = \rho \stackrel{\text{def}}{=} 1/|S(\mathcal{T})|$ if $T \in S(\mathcal{T})$, and $\rho^T = 0$ otherwise.
- Exploit linearity of $P(\cdot)$, move ρ into params—viz. $\boldsymbol{\lambda}^T = \rho\boldsymbol{\theta}^T$:

"Dual Decomposition"-like Formulation (OP4)

$$\min_{\{\boldsymbol{\lambda}^T\} \in \mathcal{S}(\boldsymbol{\theta})} \sum_{T \in S(\mathcal{T})} \bar{P}(\boldsymbol{\lambda}^T) \text{ with } \mathcal{S}(\boldsymbol{\theta}) = \left\{ \{\boldsymbol{\lambda}^T\} \middle| \textstyle\sum_{T \in S(\mathcal{T})} \boldsymbol{\lambda}^T = \boldsymbol{\theta} \right\}.$$

Simplifying the Upper Bound (2)

- Curious fact #2 (Kolmogorov, 2006): Trees in tree-reweighted upper bound need not be spanning ($\rightarrow$ no impact on *tightest* bound).
- Central idea #1: Choose each tree T as single edge $E = (s, t)$.
- Determines almost all parameters:

$$\boldsymbol{\lambda}_{st}^{E} \stackrel{!}{=} \begin{cases} \boldsymbol{\theta}_{st} & \text{if } E = (s,t) \\ \mathbf{o} & \text{otherwise} \end{cases}, \quad \boldsymbol{\lambda}_{s}^{E} \stackrel{!}{=} \mathbf{o} \text{ if } s \notin E.$$

- Remaining parameters of an edge $E = (s, t)$: $\boldsymbol{\lambda}^{E} = \{\boldsymbol{\lambda}_{s}^{E}, \boldsymbol{\lambda}_{t}^{E}\}$.

Tightest Degenerate Upper Bound (OP5)

$$\min_{\boldsymbol{\lambda} \in \mathcal{Q}(\boldsymbol{\theta})} D(\boldsymbol{\lambda}; \boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{E} \max_{(x_s, x_t)} \{\lambda_{s}^{E}(x_s) + \lambda_{t}^{E}(x_t) + \theta_{st}(x_s, x_t)\},$$

$$\mathcal{Q}(\boldsymbol{\theta}) = \left\{ \{\boldsymbol{\lambda}^{E}\} \mid \sum_{E: s \in E} \boldsymbol{\lambda}_{s}^{E} = \boldsymbol{\theta}_{s} \text{ for all } s \right\}.$$

Tightening the Upper Bound

Subgradient

Objective $D(\boldsymbol{\lambda};\boldsymbol{\theta})$ is non-differentiable, but $\boldsymbol{g} \in \mathbb{R}^{2|\mathcal{X}|\mathcal{E}|}$ is given by:

$g_s^E(x_s) = [x_s = \bar{x}_s^E]$, $g_t^E(x_t) = [x_t = \bar{x}_t^E]$ for all $E = (s,t), x_s, x_t$,

where we use $(\bar{x}_s^E, \bar{x}_t^E)$ to refer to the edge MAP state (cf. OP5).

Projection

We need to solve $\operatorname{argmin}_{\boldsymbol{\lambda} \in \mathcal{Q}(\boldsymbol{\theta})} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2^2$. Solution obtained as:

$$\mathcal{P}_{\boldsymbol{\theta}}(\boldsymbol{\lambda}') = \left\{ \lambda_s^E(x_s) \leftarrow \lambda_s'^E(x_s) - \left(\sum_{E' \in \mathcal{E}_s} \lambda_s'^{E'}(x_s) - \theta_s(x_s) \right) / |\mathcal{E}_s| \right\},$$

which distributes amount of change uniformly over adjacent edges.

Central idea #2: Separable, non-diff. problem, cheap projection $\rightarrow$ use incremental subgradient method (Nedić and Bertsekas, 2001).

The Algorithm

Input : Graph G, target parameters $\boldsymbol{\theta}$, initial feasible point $\boldsymbol{\lambda}$
Output: Feasible primal solution $\tilde{\boldsymbol{x}}$ that is an approximation to $\bar{\boldsymbol{x}}$
choose initial feasible primal solution $\tilde{\boldsymbol{x}}$ arbitarily ;
repeat
  pick next step size α and shuffle the set of edges $\mathcal{E}$;
  foreach $E = (s, t) \in \mathcal{E}$ **do**
    find MAP state: $(\bar{x}_s^E, \bar{x}_t^E)$;
    subtract subgradient: $\lambda_s^E(\bar{x}_s^E) \leftarrow \lambda_s^E(\bar{x}_s^E) - \alpha$, $\lambda_t^E(\bar{x}_t^E) \leftarrow \lambda_t^E(\bar{x}_t^E) - \alpha$;
    foreach $E' \in \mathcal{E}_s$ **do** project: $\lambda_s^{E'}(\bar{x}_s^E) \leftarrow \lambda_s^{E'}(\bar{x}_s^E) + \alpha/|\mathcal{E}_s|$;
    foreach $E' \in \mathcal{E}_t$ **do** project: $\lambda_t^{E'}(\bar{x}_t^E) \leftarrow \lambda_t^{E'}(\bar{x}_t^E) + \alpha/|\mathcal{E}_t|$;
  foreach $s \in \mathcal{V}$ **do** build candidate $\tilde{\boldsymbol{c}}$: $\tilde{c}_s \leftarrow$ at random from $\{\bar{x}_s^E \mid E \in \mathcal{E}_s\}$;
  if $P(\tilde{\boldsymbol{c}}; \boldsymbol{\theta}) > P(\tilde{\boldsymbol{x}}; \boldsymbol{\theta})$ **then**
    accept best primal solution so far: $\tilde{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{c}}$;
  if $D(\boldsymbol{\lambda}; \boldsymbol{\theta}) = P(\tilde{\boldsymbol{x}}; \boldsymbol{\theta})$ **then**
    optimal primal solution found: **return** $\tilde{\boldsymbol{x}}$;
until converged;
approximate primal solution found: **return** $\tilde{\boldsymbol{x}}$;

Outline

Formal Guarantees

Proposition (Global Convergence)

For an appropriately chosen sequence of step sizes $\{\alpha^{(k)}\}$*, convergence to the global optimum of (OP5) is guaranteed as* $k \to \infty$.

The choice of $\{\alpha^{(k)}\}$ is discussed in detail in the paper.

Proposition (Optimality of Primal Solutions)

Assume that at an outer iteration, $P(\tilde{\boldsymbol{x}}; \boldsymbol{\theta}) = D(\boldsymbol{\lambda}; \boldsymbol{\theta})$*. It follows that* $\tilde{\boldsymbol{x}}$ *maximizes* $P(\cdot)$ *and* $\boldsymbol{\lambda}$ *minimizes* $D(\cdot)$*. This happens precisely if for each node, the edge MAP states agree on a common node MAP state.*

We thus obtain a certificate of optimality for our primal solution.

Comparison to Related Approaches

Several methods have been devised that all aim at solving the first-order linear programming relaxation (OP3).

Method	Converg.	Global	Rate	Memory
IncMP (Our Method)	yes	yes	sublinear	$\mathcal{O}(\lvert\mathcal{X}\rvert)$
DDSub (Komodakis et al., 2007)	yes	yes	sublinear	$\mathcal{O}(\lvert\mathcal{X}\rvert^2)$
TrwMP (Wainwright et al., 2005)	no	no	?	$\mathcal{O}(\lvert\mathcal{X}\rvert)$
Trw-S (Kolmogorov, 2006)	yes	no	?	$\mathcal{O}(\lvert\mathcal{X}\rvert)$
MPLP (Globerson et al., 2007)	yes	no	?	$\mathcal{O}(\lvert\mathcal{X}\rvert)$
ProxLP (Ravikumar et al., 2010)	yes	yes	superlinear	$\mathcal{O}(\lvert\mathcal{X}\rvert^2)$
DDProx (Jojic et al., 2010)	yes	yes	linear	$\mathcal{O}(\lvert\mathcal{X}\rvert^2)$

The convergence rates and working memory requirements listed above are asymptotic and do not necessarily reveal a lot about real-world performance (cost of an iteration is crucial).

Outline

Experimental Setup

We compared three solvers (IncMP, DDSub and TrwMP) on three different types of graphs, averaged over 20 runs.

GridIsingUni: A 50×50 grid with binary variables ($\mathcal{X} = \{-1, +1\}$) and potentials given by $\theta_s(x_s) = \gamma x_s$ and $\theta_{st}(x_s, x_t) = \gamma x_s x_t$ with $\gamma \sim \mathcal{U}(-1, +1)$ drawn independently for each node and edge.

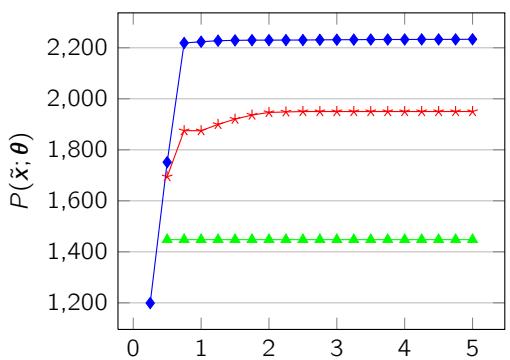
GridMultiGauss: A 20×20 grid with variables of arity $|\mathcal{X}| = 16$ and potentials chosen as $\theta_s(x_s) = 0$ and $\theta_{st}(x_s, x_t) \sim \mathcal{N}(0, 15)$ independently.

CompIsingUni: A complete graph of 50 binary variables with potentials chosen akin to GridIsingUni.

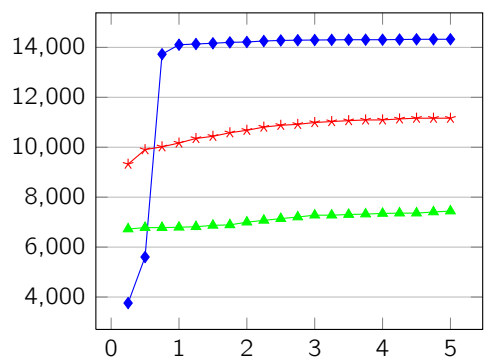
Results

- Measured the score $P(\tilde{\boldsymbol{x}};\boldsymbol{\theta})$ of the best primal solution $\tilde{\boldsymbol{x}}$ found so far, as a function of running time (seconds).
- For IncMP and DDSub, constructed $\tilde{\boldsymbol{x}}$ randomly from the edge and tree MAP states, respectively (at each iteration).
- For TrwMP, used the maximizers of the node beliefs.

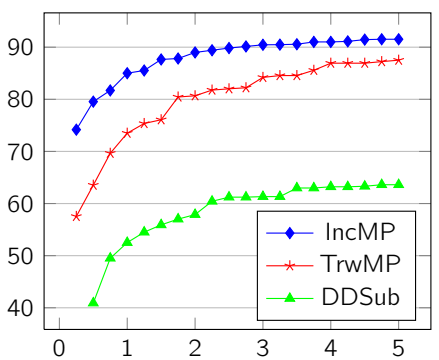
GridIsingUni 



GridMultiGauss 



CompIsingUni 



Outline

Future Work

- Step size α can be determined analytically so as not to increase the dual objective $P(\boldsymbol{\lambda}, \boldsymbol{\theta})$.
  - Turns algorithm into a "dual descent method" (Bertsekas, 1999).
  - Open question: Can global convergence still be guaranteed?
  - Most likely, can get stuck in a "corner" (akin to MPLP).
- Use as computational core in a branch-and-bound scheme.
  - Low working memory requirements, ideal for parallelization.
  - Constraints like "$X_s \stackrel{!}{=} x_s$" can easily be added, warm-starting should work rather well.
- Release source code as part of the PhiWeave package for approximate training of discriminative graphical models.

Some References

M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky.

MAP estimation via agreement on (hyper)trees: Message passing and linear-programming approaches.
IEEE Transactions on Information Theory, Vol. 51(11), 2005.

V. Kolmogorov.

Convergent tree-reweighted message passing for energy minimization.
Pattern Analysis and Machine Intelligence, Vol. 28(10), 2006.

A. Nedić and D. P. Bertsekas.

Incremental subgradient methods for nondifferentiable optimization.
SIAM Journal on Optimization, Vol. 12, 2001.

N. Komodakis, N. Paragios, and G. Tziritas.

MRF optimization via dual decomposition: Message-passing revisited.
IEEE 11th International Conference on Computer Vision, 2007.

A. Globerson and T. Jaakkola.

Fixing max-product: Convergent message passing algorithms for MAP-LP relaxations.
Advances in Neural Information Processing Systems, 2007.

P. Ravikumar, A. Argarwal, and M. J. Wainwright.

Message-passing for graph-structured linear programs: proximal methods and rounding schemes.
Journal of Machine Learning Research, Vol. 11, 2010.

V. Jojic, S. Gould, and D. Koller.

Accelerated dual decomposition for MAP inference.
27th International Conference on Machine Learning, 2010.