Dual Decomposition for Inference

Yunshu Liu

ASPITRG Research Group

2014-05-06

References:

[1]. D. Sontag, A. Globerson and T. Jaakkola, *Introduction to Dual Decomposition for Inference*, Optimization for Machine Learning, MIT Press, 2011.

[2]. A. Globerson and T. Jaakkola, *Tutorial: Linear Programming Relaxations for Graphical Models*, NIPS, 2011.

[3]. C. Yanover, T. Meltzer and Y. Weiss, *Linear Programming Relaxations and Belief Propagation : An Empirical Study*, JMLR, Volume 7, September 2006, pages 1887-1907.

[4]. A. M. Rush and M. Collins, *A Tutorial on Dual Decomposition and Lagrangian Relaxation for Inference in Natural Language Processing*, JAIR, Volume 45 Issue 1, September 2012, pages 305-362.

[5]. D. Batra, A.C. Gallagher, D. Parikh and T. Chen, *Beyond Trees: MRF Inference via Outer-Planar Decomposition*, CVPR, 2010, pages 2496-2503.

Outline

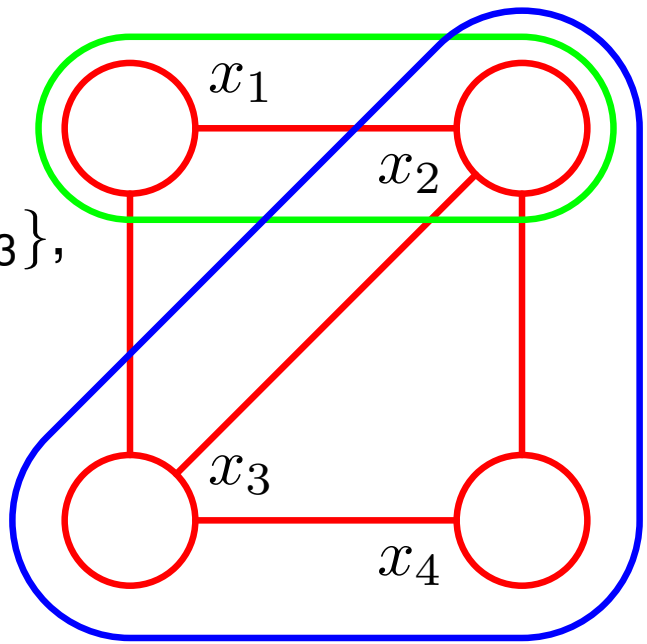
Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
  - Subgradient algorithms
  - Block coordinate descent algorithms
- Discussion

The MAP inference problem: Markov random fields

In the undirected case, the probability distribution factorizes according to functions defined on the clique of the graph.
A clique is a subset of nodes in a graph such that there exist a link between all pairs of nodes in the subset.
A maximal clique is a clique such that it is not possible to include any other nodes from the graph in the set without it ceasing to be a clique.

Example of cliques:
$\{x_1,x_2\},\{x_2,x_3\},\{x_3,x_4\},\{x_2,x_4\},\{x_1,x_3\},$
$\{x_1,x_2,x_3\}, \{x_2,x_3,x_4\}$
Maximal cliques:
$\{x_1,x_2,x_3\}, \{x_2,x_3,x_4\}$



The MAP inference problem: Markov random fields

Markov random fields: Definition

Denote C as a clique, $\mathbf{x}_C$ the set of variables in clique C and $\psi_C(\mathbf{x}_C)$ a nonnegative potential function associated with clique C. Then a Markov random field is a collection of distributions that factorize as a product of potential functions $\psi_C(\mathbf{x}_C)$ over the **maximal cliques** of the graph:

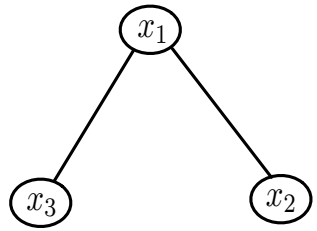
$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

where normalization constant $Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$ sometimes called the partition function.

The MAP inference problem: Markov random fields

Factorization of undirected graphs

Question: how to write the joint distribution for this undirected graph?



$(2 \perp 3|1)$ hold

Answer:

$$p(\mathbf{x}) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)$$

where $\psi_{12}(x_1, x_2)$ and $\psi_{13}(x_1, x_3)$ are the potential functions and Z is the partition function that make sure $p(\mathbf{x})$ satisfy the conditions to be a probability distribution.

The MAP inference problem

MAP: find the maximum probability assignment

$$\arg\max_{\mathbf{x}} p(\mathbf{x}), \quad \textit{where} \quad p(\mathbf{x}) = \frac{1}{Z} \prod_{C} \psi_C(\mathbf{x}_C)$$

Converting to a max-product problem(since Z is constant)

$$\arg\max_{\mathbf{x}} \prod_{C} \psi_C(\mathbf{x}_C)$$

Converting to a max-sum inference task by taking the log and letting $\theta_C(\mathbf{x}_C) = \log \psi_C(\mathbf{x}_C)$

$$\arg\max_{\mathbf{x}} \sum_{C} \theta_C(\mathbf{x}_C)$$

The MAP inference problem

MAP: find the maximum probability assignment

Consider a set of n discrete variables $x_1, \cdots, x_n$, and a set F of subsets on these variables(i.e., $f \in F$ is a subset of $V = \{1, \cdots, n\}$), where each subset corresponds to the domain of one of the factors.
The MAP inference task is to find an assignment $\mathbf{x} = (x_1, \cdots, x_n)$ which maximizes the sum of the factors:

$$MAP(\theta) = \max_{\mathbf{x}} \left(\sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f) \right)$$

Note we also included $\theta_i(x_i)$ on each of the individual variables, which is useful in some applications.

The MAP inference problem

An example of MAP inference

Consider three binary variables x_1, x_2 and x_3 and two factors $\{1,2\}$ and $\{1,3\}$, for which we know the values of the factors:

	00	01	10	11
$\theta_{12}(\mathbf{x}_{12})$:	2	100	3	0.1
$\theta_{23}(\mathbf{x}_{23})$:	10	5	200	0

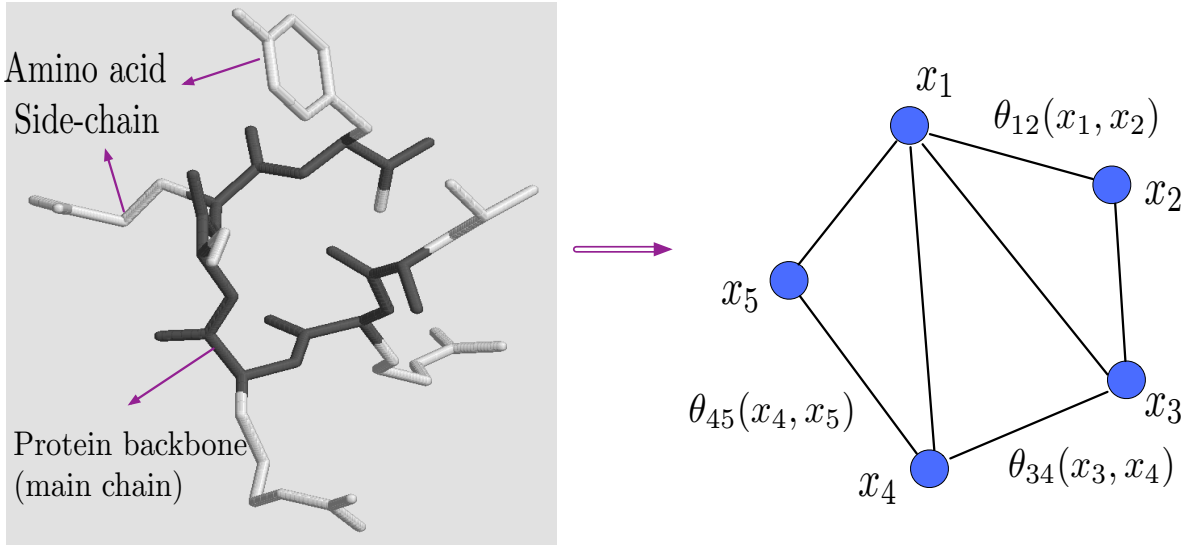
The goal is $MAP(\theta) = \max_{\mathbf{x}}(\theta_{12}(\mathbf{x}_{12}) + \theta_{23}(\mathbf{x}_{23}))$

Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
  - Subgradient algorithms
  - Block coordinate descent algorithms
- Discussion

Motivating Applications

Protein side-chain placement (see reference [3] for detail)

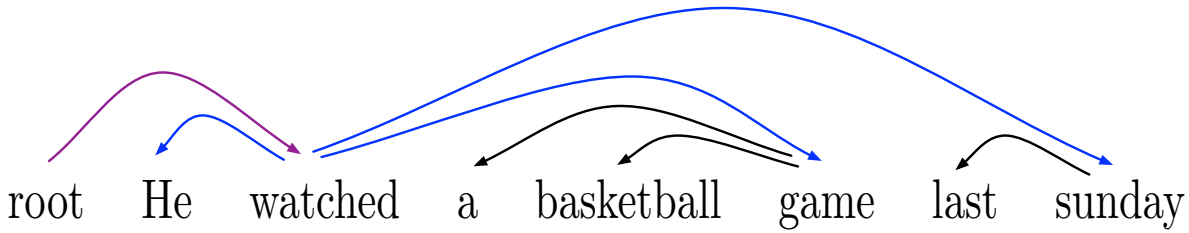


- For fixed protein backbone, find the maximum probability assignment of amino acid side-chains.
- Orientations of the side-chains are represented by discretized angles called rotamers, rotamer choices for nearby amino acids are energetically coupled(attractive or repulsive forces).

Motivating Applications

Dependency parsing (see reference [4] for detail)

- Given a sentence, predict the dependency tree to relates the words in it:



- An arc is drawn from the head word of each phrase to words that modify it.
- We use $\theta_{ij}(x_{ij})$ to denote the weight on the arcs between words i and j, use $\theta_{i|}(x_{i|})$ for higher order interactions between the arc selections for a given word i, where $x_{ij} \in \{0, 1\}$ indicate the existence of arc from i to j and $x_{i|} = \{x_{ij}\}_{j \neq i}$ is the coupled arc selection.
- The task is to maximizing a function involving $\sum_{ij} \theta_{ij}(x_{ij}) + \sum_i \theta_{i|}(x_{i|})$.

Motivating Applications

Other applications

Computational biology

-e.g. protein design etc. (see reference [3] for details)

Natural language processing

-e.g. combined weighted context-free grammar and finite-state tagger etc. (see reference [4] for details)

Computer vision

- e.g. image segmentation, object labeling etc. (see reference [5] for details)

Graphical Model

-e.g. Structure learning of bayesian networks and Markov network.

Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
  - Subgradient algorithms
  - Block coordinate descent algorithms
- Discussion

Dual decomposition and Lagrangian relaxation: Approximate inference for MAP problem

MAP: find the maximum probability assignment

The MAP inference task is to find an assignment $\mathbf{x} = (x_1, \cdots, x_n)$ which maximizes the sum of the factors:

$$MAP(\theta) = \max_{\mathbf{x}} \left(\sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f) \right)$$

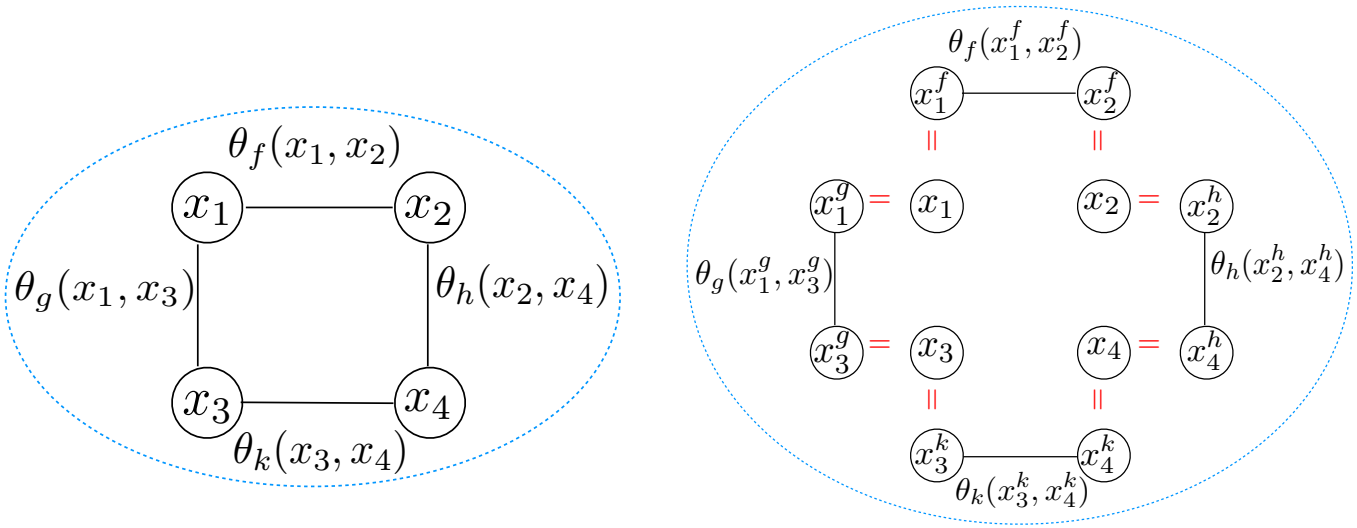
- MAP inference is hard, even for pairwise MRF the problem is NP-hard.
- Dynamic programming provide exact inference for some special cases(e.g. tree-structured graph).
- In general, approximate inference algorithms are used by relax some constraints so that we are able to divide the problem into easy-solving subproblems.

Dual decomposition and Lagrangian relaxation

For the MAP inference problem:

$$MAP(\theta) = \max_{\mathbf{x}}(\sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f))$$

for any given factor f, we make copy of each variables x_i in the factor and denote as x_i^f.



(Note: the figures are adapted from reference [1])

Dual decomposition and Lagrangian relaxation

MAP: find the maximum probability assignment

The original MAP inference problem is equivalent to:

$$\max \sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f^f)$$
$$s.t. \quad x_i^f = x_i \quad \forall f, i \in f$$

since we add the constraints $x_i^f = x_i$ for $\forall f, i \in f$.

Lagrangian

Introduce Lagrange multipliers $\delta = \{\delta_{fi}(x_i) : f \in F, i \in F, x_i\}$, then define the Lagrangian:

$$\begin{aligned} L(\delta, \mathbf{x}, \mathbf{x}^F) &= \sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f^f) \\ &+ \sum_{f \in F} \sum_{i \in f} \sum_{\hat{x}_i} \delta_{fi}(\hat{x}_i)(1[x_i = \hat{x}_i] - 1[x_i^f = \hat{x}_i]) \end{aligned}$$

Dual decomposition and Lagrangian relaxation

Lagrangian

Then the original MAP inference problem is equivalent to:

$$\max_{\mathbf{x},\mathbf{x}^F} L(\delta, \mathbf{x}, \mathbf{x}^F)$$
$$s.t. \quad x_i^f = x_i \quad \forall f, i \in f$$

since if the constraints $x_i^f = x_i$ for $\forall f, i \in f$ hold, the last term in $L(\delta, \mathbf{x}, \mathbf{x}^F)$ is zero.

Meaning of Lagrange multipliers $\delta_{fi}(x_i)$

Consider $\delta_{fi}(x_i)$ as a *message* that factor f sends to variable i about its state x_i, then iteratively updating $\delta_{fi}(x_i)$ is similar to message passing between factor and variables.

Dual decomposition and Lagrangian relaxation

Lagrangian relaxation

Now we discard the condition $x_i^f = x_i \quad \forall f, i \in f$ and consider the relaxed dual problem:

$$L(\delta) = \max_{\mathbf{x}, \mathbf{x}^F} L(\delta, \mathbf{x}, \mathbf{x}^F)$$

$$= \sum_{i \in V} max_{x_i}(\theta_i(x_i) + \sum_{f: i \in f} \delta_{fi}(x_i)) + \sum_{f \in F} max_{\mathbf{x}_f^f}(\theta_f(\mathbf{x}_f^f) - \sum_{i \in f} \delta_{fi}(x_i^f))$$

- Notice that if $x_i^f = x_i$ then the above problem is no longer equivalent to the original one since we changed position of max and sum.
- Each of the subproblem only involved maximization over small set of variables x_i, we use $\delta_{fi}(x_i)$ to help achieve the consensus among the subproblems.

Dual decomposition and Lagrangian relaxation

The dual problem

Without the assumption $x_i^f = x_i \quad \forall f, i \in f$, $L(\delta)$ is maximized over a larger space, thus $MAP(\theta) \leq L(\delta)$, the goal is then to find the tightest upper bound by solving $\min_\delta L(\delta)$, where

$$L(\delta) = \max_{\mathbf{x}, \mathbf{x}^F} L(\delta, \mathbf{x}, \mathbf{x}^F)$$

$$= \sum_{i \in V} max_{x_i}(\theta_i(x_i) + \sum_{f: i \in f} \delta_{fi}(x_i)) + \sum_{f \in F} max_{\mathbf{x}_f}(\theta_f(\mathbf{x}_f) - \sum_{i \in f} \delta_{fi}(x_i))$$

Note that for the optimization problem, the maximizations are independent, thus in the updating process we don't distinguish between x_i and x_i^f.

Dual decomposition and Lagrangian relaxation

Reparameterization Interpretation of the Dual

Given a set of dual variables δ, define
$\bar{\theta}_i^{\delta}(x_i) = \theta_i(x_i) + \sum_{f:i\in f} \delta_{fi}(x_i)$ and $\bar{\theta}_f^{\delta}(\mathbf{x}_f) = \theta_f(\mathbf{x}_f) - \sum_{i\in f} \delta_{fi}(x_i)$.
For all assignments $\mathbf{x}$ we have:

$$\sum_{i\in V} \theta_i(x_i) + \sum_{f\in F} \theta_f(\mathbf{x}_f) = \sum_{i\in V} \bar{\theta}_i^{\delta}(x_i) + \sum_{f\in F} \bar{\theta}_f^{\delta}(\mathbf{x}_f)$$

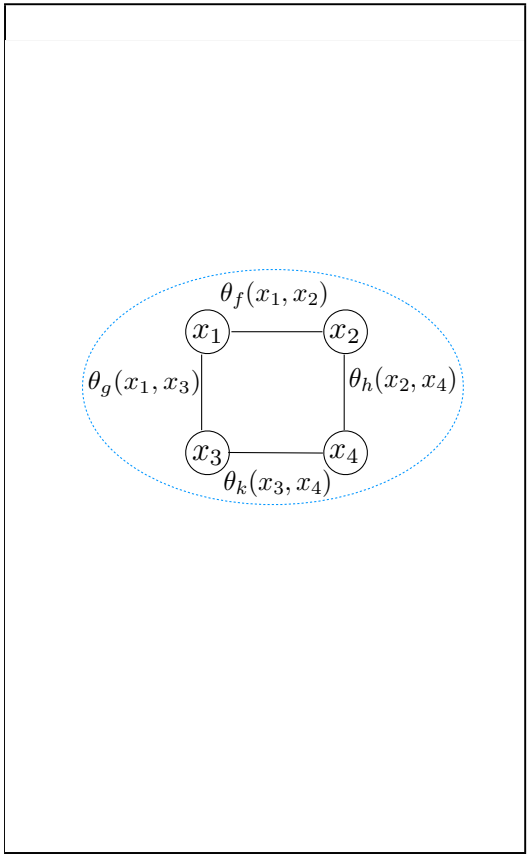
We call $\bar{\theta}$ the reparameterization of the original parameters θ.
Observe

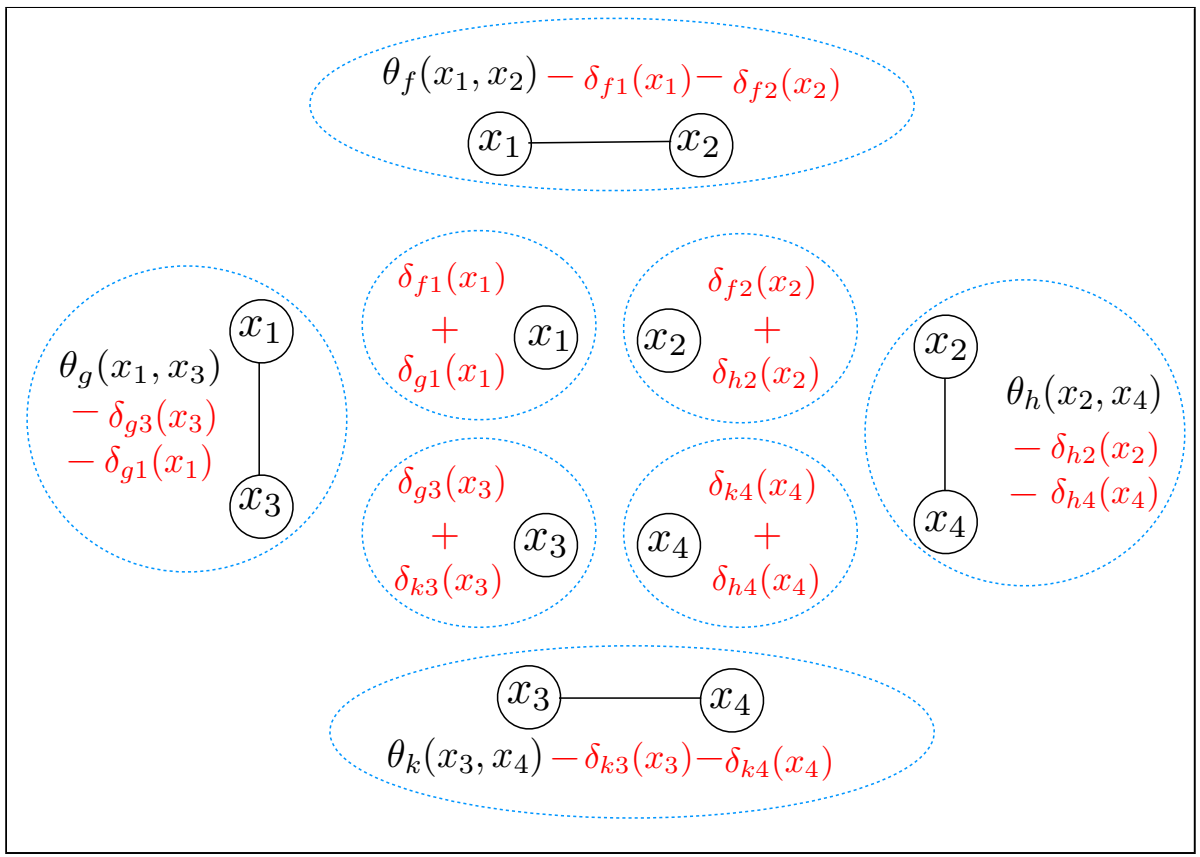
$$L(\delta) = \sum_{i\in V} \max_{x_i} \bar{\theta}_i^{\delta}(x_i) + \sum_{f\in F} \max_{\mathbf{x}_f} \bar{\theta}_f^{\delta}(\mathbf{x}_f)$$

Thus the dual problem can be considered as searching for the best reparameterization $\bar{\theta}$ to minimize $L(\delta)$.

Dual decomposition and Lagrangian relaxation

Reparameterization Interpretation of the Dual





(Note: the figures are adapted from reference [1])

Dual decomposition and Lagrangian relaxation

Existence of duality gap

The difference between the primal optimal value and the dual optimal value is called *duality gap*. For our problem

$$MAP(\theta) = \max_{\mathbf{x}}(\sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f)) \leq \min_{\delta} L(\delta)$$

we don't necessarily have strong duality(i.e., equality hold in the above equation, thus no duality gap).

Strong duality hold only when the subproblems agreeing on a maximizing assignment, that is $\exists \delta^*, \mathbf{x}^*$ such that $x_i^* \in \arg\max_{x_i} \bar{\theta}_i^{\delta^*}(x_i)$ and $\mathbf{x}_f^* \in \arg\max_{\mathbf{x}_f} \bar{\theta}_f^{\delta^*}(\mathbf{x}_f)$, which is not guaranteed. However, in real-world problems, exact solution are frequently found using dual decomposition.

Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
  - Subgradient algorithms
  - Block coordinate descent algorithms
- Discussion

Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
  - Subgradient algorithms
  - Block coordinate descent algorithms
- Discussion

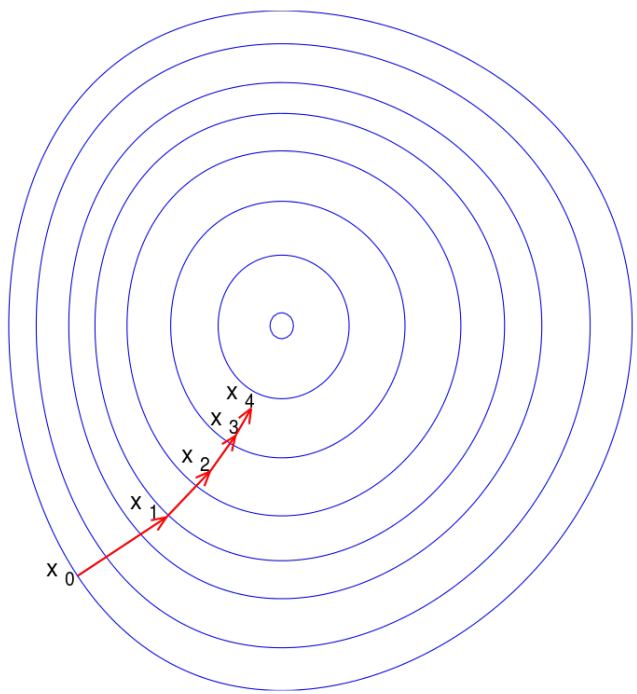
Subgradient algorithms

Gradient descent/steepest descent

If $f(x)$ is differentiable in a neighborhood of a point a, then $F(x)$ decreases fastest if one goes from a in the direction of the negative gradient of f at a, $\nabla f(a)$, which give us the gradient descent algorithm:

$$x_{t+1} = x_t - \alpha_k \nabla f(x_t)$$

where α_t is the step size, recall different ways to choose step size: fixed, decreasing, adapted.
(Note: the figure is adapted from Wikipedia)



Subgradient algorithms

Limitations of gradient descent

- For ill conditioned convex problems, gradient descent perform 'zigzags'
- The function should be differentiable

Our problem of minimizing $L(\delta)$

$$L(\delta) = \sum_{i \in V} \max_{x_i} \bar{\theta}_i^{\delta}(x_i) + \sum_{f \in F} \max_{\mathbf{x}_f} \bar{\theta}_f^{\delta}(\mathbf{x}_f)$$

$L(\delta)$ is continuous and convex, but not differentiable at all points δ where $\bar{\theta}_i^{\delta}(x_i)$ or $\bar{\theta}_f^{\delta}(\mathbf{x}_f)$ have multiple optima.

Subgradient algorithms

Subgradient algorithms

A subgradient of a convex function $L(\delta)$ at δ is a vector g_δ such that for all δ',

$$L(\delta') \geq L(\delta) + g_\delta \cdot (\delta' - \delta)$$

Suppose that the dual variables at iteration t are given by δ^t, then the updating rule is:

$$\delta_{fi}^{t+1}(x_i) = \delta_{fi}^{t}(x_i) - \alpha_t g_{fi}^{t}(x_i)$$

where $\mathbf{g}^t$ is the subgradient of $L(\delta)$ at δ^t.

Subgradient algorithms

Updating subgradient g_δ

Let x_i^s be a maximizing assignment of $\bar{\theta}_i^{\delta^t}(x_i)$, and let $\mathbf{x}_f^f$ be a maximizing assignment of $\bar{\theta}_i^{\delta^t}(\mathbf{x}_f)$, then the subgradient is updated by the following algorithm:
(recall $L(\delta, \mathbf{x}, \mathbf{x}^F) = \sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\mathbf{x}_f^f) +$
$\sum_{f \in F} \sum_{i \in f} \sum_{\hat{x}_i} \delta_{fi}(\hat{x}_i)(1[x_i = \hat{x}_i] - 1[x_i^f = \hat{x}_i]))$

$$g_{fi}^t(x_i) = 0, \qquad \forall f, i \in f, x_i$$

For $f \in F$ and $i \in f$:

If $x_i^f \neq x_i^s$:

$$\begin{aligned} g_{fi}^t(x_i^s) &= +1 \\ g_{fi}^t(x_i^f) &= -1 \end{aligned}$$

(Note: the algorithm is adapted from reference [1])

Subgradient algorithms

Updating subgradient g_δ

Recall

$$\delta_{fi}^{t+1}(x_i) = \delta_{fi}^{t}(x_i) - \alpha_t g_{fi}^{t}(x_i)$$
$$\bar{\theta}_i^{\delta}(x_i) = \theta_i(x_i) + \sum_{f: i \in f} \delta_{fi}(x_i)$$
$$\bar{\theta}_f^{\delta}(\mathbf{x}_f) = \theta_f(\mathbf{x}_f) - \sum_{i \in f} \delta_{fi}(x_i)$$

In the updating process, when ever $x_i^f \neq x_i^s$, $\bar{\theta}_i^{\delta^t}(x_i^s)$ will be decreasing and $\bar{\theta}_i^{\delta^t}(x_i^f)$ will be increasing; similarly, $\bar{\theta}_i^{\delta^t}(x_i^f, \mathbf{x}_{f\setminus i})$ will be decreasing and $\bar{\theta}_i^{\delta^t}(x_i^s, \mathbf{x}_{f\setminus i})$ will be increasing.

Subgradient algorithms

Notes on subgradient algorithm

- If at any iteration k we find that $\mathbf{g}^t = 0$, then $x_i^f = x_i^s$ for all $f \in F$, $i \in f$, there is no duality gap and we exactly solved the problem.
- The subgradient algorithm will solve the dual to optimality whenever the step-sizes are chosen such that $\lim_{t\to\infty} \alpha_t = 0$ and $\sum_{t=0}^{\infty} \alpha_t = \infty$, e.g. $\alpha_t = \frac{1}{t}$.

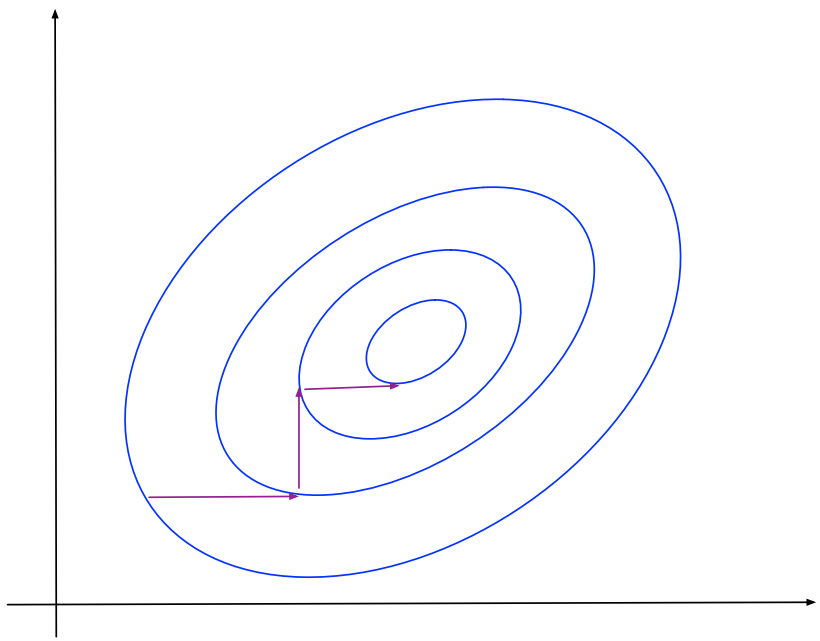
Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
  - Subgradient algorithms
  - Block coordinate descent algorithms
- Discussion

Block coordinate descent algorithms

Coordinate descent

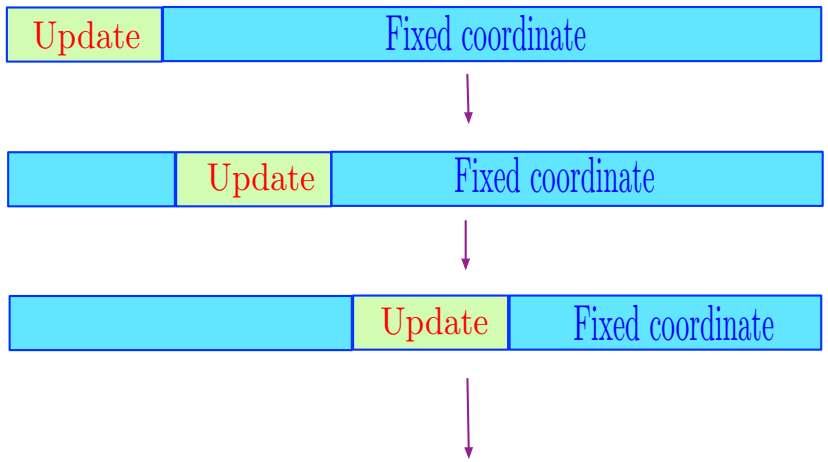
Fixed all other coordinates, do a line search along the chosen coordinate direction, then change the update coordinate cyclically throughout the procedure.



Block coordinate descent algorithms

Block coordinate descent

Coordinates are divided into blocks, where each block contains a subset of coordinates, the update rule is to only update one block at a time(fix other blocks):



Difficult of designing block coordinate descent algorithms:

- How to choose coordinate block to update.
- How to design the update rule for given coordinate block.

Block coordinate descent algorithms

Block coordinate descent for MAP inferene

The updating rule of subgradient method only involved x_i^s,$\mathbf{x}_f^f$ corresponding to the maximizing assignment, where in block coordinate descent algorithms we update $\delta_{fi}(x_i)$ for all configurations of x_i. Two common choices:

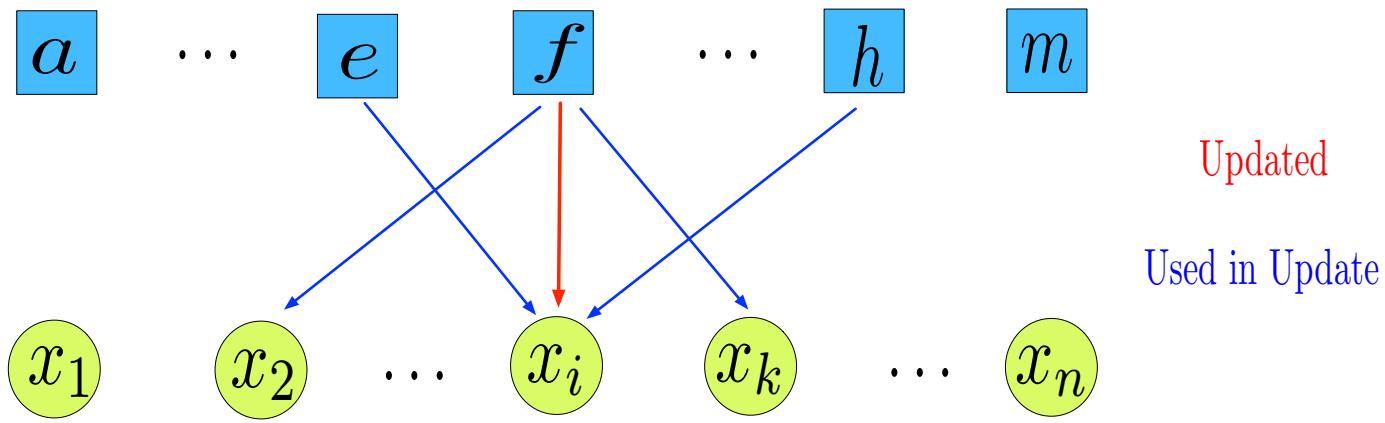
- Fix f and i, update $\delta_{fi}(x_i)$ for $\forall x_i$
  Max-Sum Diffusion(MSD) (Schlesinger 76, Werner 07)
- Fix f, update $\delta_{fi}(x_i)$ for $\forall\ i$, x_i
  Max-Product Linear Programming(MPLP) (Globerson & Jaakkola 07)

Block coordinate descent algorithms

The Max-Sum Diffusion algorithm

$$\delta_{fi}(x_i) = -\frac{1}{2}\delta_i^{-f}(x_i) + \frac{1}{2}\max_{\mathbf{x}_{f\setminus i}}\{\theta_f(\mathbf{x}_f) - \sum_{\hat{i}\in f\setminus i}\delta_{f\hat{i}}(x_{\hat{i}})\}$$

where $\delta_i^{-f}(x_i) = \theta_i(x_i) + \sum_{\hat{f}\neq f}\delta_{\hat{f}i}(x_i)$.
For fixed f and i, $\sum_{\hat{i}\in f\setminus i}\delta_{f\hat{i}}(x_{\hat{i}})$ is the message from f to all the children of f except for i (i.e. $f\setminus i$); $\sum_{\hat{f}\neq f}\delta_{\hat{f}i}(x_i)$ is the message from all of i's parents other than f to i.



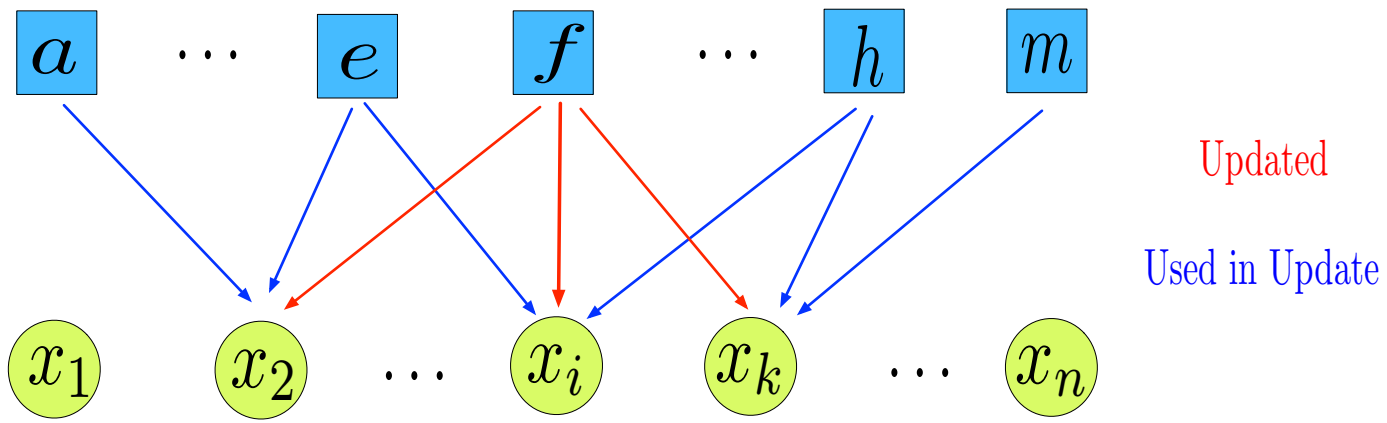
Block coordinate descent algorithms

The Max-Product Linear Programming algorithm

$$\delta_{fi}(x_i) = -\delta_i^{-f}(x_i) + \frac{1}{|f|} \max_{\mathbf{x}_{f\setminus i}} \{\theta_f(\mathbf{x}_f) + \sum_{\hat{i}\in f} \delta_i^{-f}(x_i)\}$$

where $\delta_i^{-f}(x_i) = \theta_i(x_i) + \sum_{\hat{f}\neq f} \delta_{\hat{f}i}(x_i)$.

For fixed f, $\sum_{\hat{i}\in f} \delta_i^{-f}(x_i)$ is the message from f's co-parents to all their children; $\sum_{\hat{f}\neq f} \delta_{\hat{f}i}(x_i)$ is the message from all of i's parents other than f to i.



Block coordinate descent algorithms

Comparison of Subgradient, MSD and MPLP

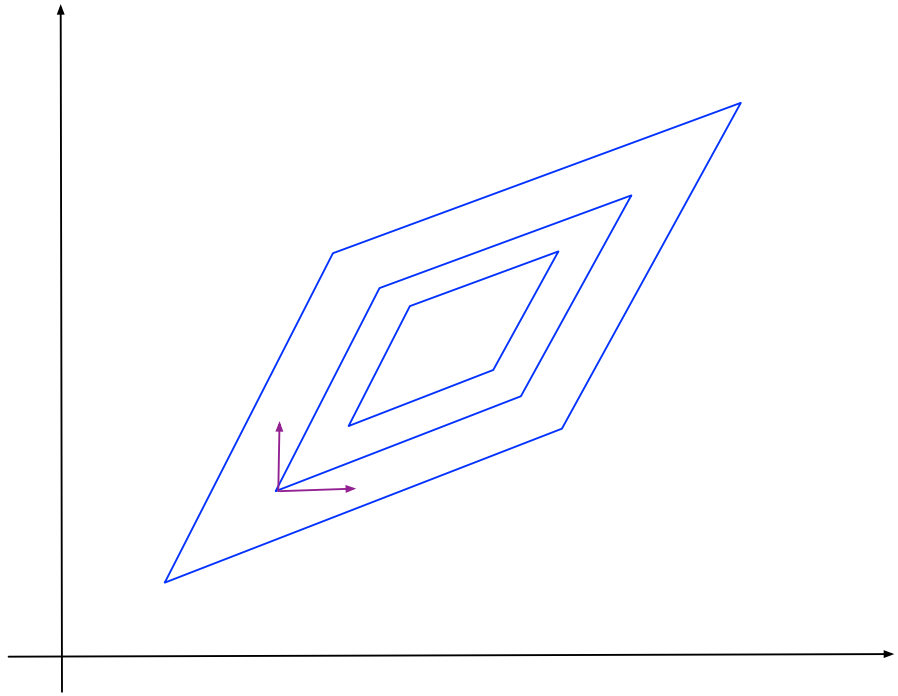
- In subgradient algorithm, only x_i^s,$\mathbf{x}_f^f$ corresponding to the maximizing assignment are used.
- In MSD, we fix f,i and update δ_{fi} for all x_i simultaneously.
- In MPLP, we fix f and update δ_{fi} for $\forall$ i and x_i simultaneously.

Larger blocks are used in MPLP; More communication are involved between subproblems in MPLP.

Block coordinate descent algorithms

Limitations of coordinate descent

For non-smooth problem, coordinate descent may stuck at the 'corners'!



Limitations of coordinate descent

For non-smooth problem, coordinate descent may stuck at the 'corners'!

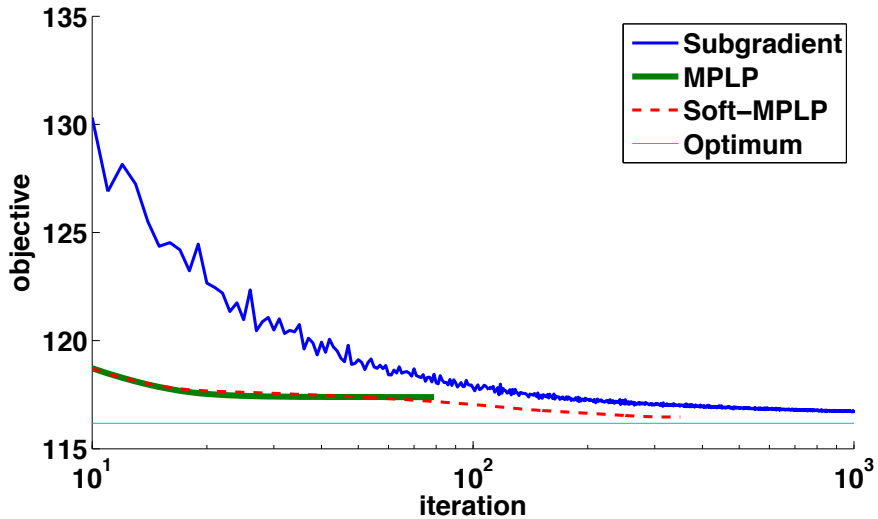
Block coordinate descent algorithms

Possible solution

Use soft-max(convex and differentiable) instead of max:

$$\max_i f_i \leq \epsilon \log \sum_i \exp^{\frac{f_i}{\epsilon}}$$

where soft-max converge to max as $\epsilon \to 0$.
Simulation result of using soft-max when MPLP stucked:



(Note: the figure is adapted from reference [2])

Outline

- The MAP inference problem
- Motivating Applications
- Dual decomposition and Lagrangian relaxation
- Algorithms for solving the dual problem
- Discussion

Discussion

Decoding the MAP assignment

- After locally decoding the single node reparameterizations:

$$x_i \quad \leftarrow \quad \arg\max_{\hat{x}_i} \bar{\theta}_i^{\delta}(\hat{x}_i)$$

  If the node terms $\bar{\theta}_i^{\delta}(x_i)$ have a unique maximum, we say that δ is locally decodable to $\mathbf{x}$.
- For subgradient algorithms, when the dual decomposition has a unique solution and it is integral, then the MAP assignment is guaranteed to be find from the dual solution.

Discussion

Decoding the MAP assignment

For coordinate descent algorithm, when the dual decomposition has a unique solution and it is integral, then

- For binary pairwise MRFs, fixed points of the coordinate descent algorithms are locally decodable to true MAP assignment.
- For pairwise tree-structured MRFs and MRFs with a single cycle, fixed points of the coordinate descent algorithms are locally decodable to true MAP assignment.
- There exist non-binary pairwise MRFs with cycles and dual optimal fixed points of the coordinate descent algorithms that are not locally decodable.

Discussion

Combine subgradient and coordinate descent

Subgradient is slow but guaranteed to converge, coordinate descent is faster but may not converge to optimal, but we can combine the two algorithms:

- First use coordinate descent, after it is close to converge, use subgradient to reach the global optimal.
- Alternating between subgradient and coordinate descent.

Thanks!

Questions!