Augmenting Dual Decomposition for MAP Inference

André F. T. Martins[1,2] Noah A. Smith[1] Eric P. Xing[1]
Pedro M. Q. Aguiar[3] Mário A. T. Figueiredo[2]

[1]Language Technologies Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA, USA

[2]Instituto de Telecomunicações
Instituto Superior Técnico, Lisboa, Portugal

[3]Instituto de Sistemas e Robótica
Instituto Superior Técnico, Lisboa, Portugal

NIPS'10 WS: Optimization for Machine Learning
Whistler, Canada, December 10th, 2010

In a Nutshell

Computing the MAP in a graphical model is NP-hard

In a Nutshell

Computing the MAP in a graphical model is NP-hard

LP relaxation: Schlesinger (1976); Wainwright et al. (2005); Globerson and Jaakkola (2008)

In a Nutshell

Computing the MAP in a graphical model is NP-hard

LP relaxation: Schlesinger (1976); Wainwright et al. (2005); Globerson and Jaakkola (2008)

Komodakis et al. (2007) proposed dual decomposition with the subgradient algorithm (Dantzig and Wolfe, 1960)

In a Nutshell

Computing the MAP in a graphical model is NP-hard

LP relaxation: Schlesinger (1976); Wainwright et al. (2005); Globerson and Jaakkola (2008)

Komodakis et al. (2007) proposed dual decomposition with the subgradient algorithm (Dantzig and Wolfe, 1960)

- *Pros:* very simple, provably convergent, parallelizable

In a Nutshell

Computing the MAP in a graphical model is NP-hard

LP relaxation: Schlesinger (1976); Wainwright et al. (2005); Globerson and Jaakkola (2008)

Komodakis et al. (2007) proposed dual decomposition with the subgradient algorithm (Dantzig and Wolfe, 1960)

- *Pros:* very simple, provably convergent, parallelizable
- *Cons:* too slow when there are many slave subproblems

In a Nutshell

Computing the MAP in a graphical model is NP-hard

LP relaxation: Schlesinger (1976); Wainwright et al. (2005); Globerson and Jaakkola (2008)

Komodakis et al. (2007) proposed dual decomposition with the subgradient algorithm (Dantzig and Wolfe, 1960)

- *Pros:* very simple, provably convergent, parallelizable
- *Cons:* too slow when there are many slave subproblems

Our proposal: DD + *augmented Lagrangian* (Hestenes, 1969; Powell, 1969)

In a Nutshell

Computing the MAP in a graphical model is NP-hard

LP relaxation: Schlesinger (1976); Wainwright et al. (2005); Globerson and Jaakkola (2008)

Komodakis et al. (2007) proposed dual decomposition with the subgradient algorithm (Dantzig and Wolfe, 1960)

- *Pros:* very simple, provably convergent, parallelizable
- *Cons:* too slow when there are many slave subproblems

Our proposal: DD + *augmented Lagrangian* (Hestenes, 1969; Powell, 1969)

- Still parallelizable, much faster to reach consensus
- Handles *global* structural constraints efficiently
- Experiments: Ising models and natural language parsing

Outline

1. Problem Formulation
2. Dual Decomposition
3. Augmented Lagrangian Method
4. Experiments
5. Conclusions

Outline

Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a \in \mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a \in \mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

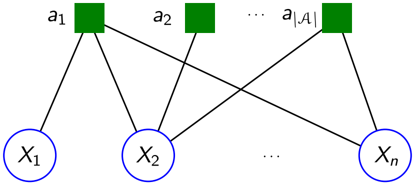
$\mathcal{A}$ is a set of factors, θ_i and ϕ_a are *unary* and *higher-order* log-potentials

Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a \in \mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

$\mathcal{A}$ is a set of factors, θ_i and ϕ_a are *unary* and *higher-order* log-potentials

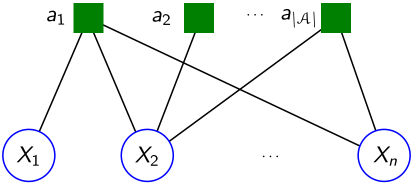


Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

$\mathcal{A}$ is a set of factors, θ_i and ϕ_a are *unary* and *higher-order* log-potentials
Allow hard constraints: $-\infty$ log-potentials

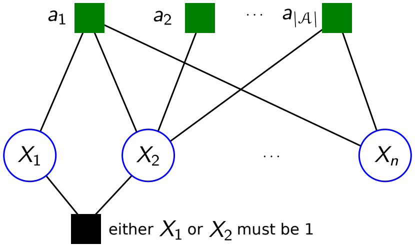


Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a \in \mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

$\mathcal{A}$ is a set of factors, θ_i and ϕ_a are *unary* and *higher-order* log-potentials
Allow hard constraints: $-\infty$ log-potentials



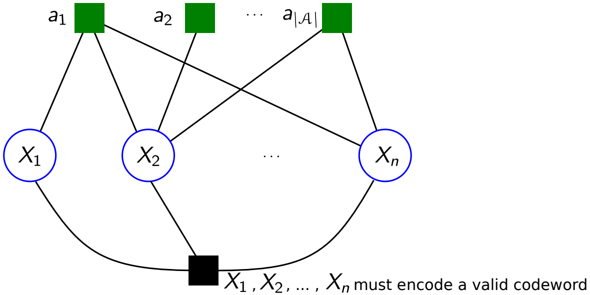
Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a \in \mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

$\mathcal{A}$ is a set of factors, θ_i and ϕ_a are *unary* and *higher-order* log-potentials

Allow hard constraints: $-\infty$ *log-potentials*



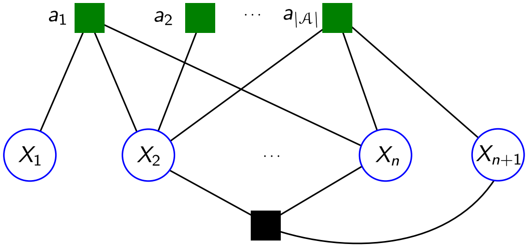
Factor Graphs

Let $\mathbf{X} \triangleq (X_1, \ldots, X_n) \in \mathcal{X}$ be a vector of *discrete* random variables

$$P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) \propto \exp\left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a \in \mathcal{A}} \phi_a(\mathbf{x}_a)\right)$$

$\mathcal{A}$ is a set of factors, θ_i and ϕ_a are *unary* and *higher-order* log-potentials

Allow hard constraints: $-\infty$ *log-potentials*



$X_{n+1} \triangleq X_2 \wedge X_n$ (conjunction)

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})}$$

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} = \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{M}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} = \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{M}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

- $\mathcal{M}(\mathcal{G})$ is the marginal polytope of $\mathcal{G}$ (Wainwright and Jordan, 2008)

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} = \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{M}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

- $\mathcal{M}(\mathcal{G})$ is the marginal polytope of $\mathcal{G}$ (Wainwright and Jordan, 2008)
- Problem: $\mathcal{M}(\mathcal{G})$ not concise in general (e.g., if $\mathcal{G}$ has loops)

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} = \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{M}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

- $\mathcal{M}(\mathcal{G})$ is the marginal polytope of $\mathcal{G}$ (Wainwright and Jordan, 2008)
- Problem: $\mathcal{M}(\mathcal{G})$ not concise in general (e.g., if $\mathcal{G}$ has loops)
- Workaround: use a *local* approximation:

$$\mathcal{L}(\mathcal{G}) = \left\{(\boldsymbol{\mu},\boldsymbol{\nu}) \mid (\boldsymbol{\mu}_{N(a)},\boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A}\right\}$$

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} = \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{M}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

- $\mathcal{M}(\mathcal{G})$ is the marginal polytope of $\mathcal{G}$ (Wainwright and Jordan, 2008)
- Problem: $\mathcal{M}(\mathcal{G})$ not concise in general (e.g., if $\mathcal{G}$ has loops)
- Workaround: use a *local* approximation:

$$\mathcal{L}(\mathcal{G}) = \left\{(\boldsymbol{\mu},\boldsymbol{\nu}) \mid (\boldsymbol{\mu}_{N(a)},\boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A}\right\}$$

- Remark: $\mathcal{L}(\mathcal{G}) \supseteq \mathcal{M}(\mathcal{G}) = \mathrm{conv}(\mathcal{L}(\mathcal{G}) \cap \mathbb{Z})$

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} = \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{M}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

- $\mathcal{M}(\mathcal{G})$ is the marginal polytope of $\mathcal{G}$ (Wainwright and Jordan, 2008)
- Problem: $\mathcal{M}(\mathcal{G})$ not concise in general (e.g., if $\mathcal{G}$ has loops)
- Workaround: use a *local* approximation:

$$\mathcal{L}(\mathcal{G}) = \left\{(\boldsymbol{\mu},\boldsymbol{\nu}) \mid (\boldsymbol{\mu}_{N(a)},\boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A}\right\}$$

- Remark: $\mathcal{L}(\mathcal{G}) \supseteq \mathcal{M}(\mathcal{G}) = \mathrm{conv}(\mathcal{L}(\mathcal{G}) \cap \mathbb{Z})$
- LP-relaxation (LP-MAP): just replace $\mathcal{M}(\mathcal{G})$ by $\mathcal{L}(\mathcal{G})$

MAP and LP-MAP Inference

Goal: compute $\hat{\mathbf{x}}$ which maximizes $P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$

$$\max_{\mathbf{x}\in\mathcal{X}} \overbrace{\sum_{i=1}^{n} \theta_i(x_i) + \sum_{a\in\mathcal{A}} \phi_a(\mathbf{x}_a)}^{\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})} \leq \max_{(\boldsymbol{\mu},\boldsymbol{\nu})\in\mathcal{L}(\mathcal{G})} \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a$$

- $\mathcal{M}(\mathcal{G})$ is the marginal polytope of $\mathcal{G}$ (Wainwright and Jordan, 2008)
- Problem: $\mathcal{M}(\mathcal{G})$ not concise in general (e.g., if $\mathcal{G}$ has loops)
- Workaround: use a *local* approximation:

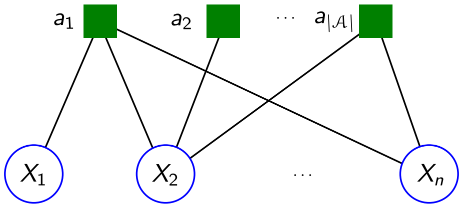
$$\mathcal{L}(\mathcal{G}) = \left\{(\boldsymbol{\mu},\boldsymbol{\nu}) \mid (\boldsymbol{\mu}_{N(a)},\boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A}\right\}$$

- Remark: $\mathcal{L}(\mathcal{G}) \supseteq \mathcal{M}(\mathcal{G}) = \mathrm{conv}(\mathcal{L}(\mathcal{G}) \cap \mathbb{Z})$
- LP-relaxation (LP-MAP): just replace $\mathcal{M}(\mathcal{G})$ by $\mathcal{L}(\mathcal{G})$

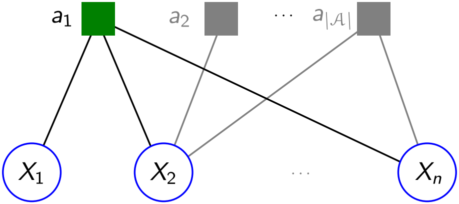
Outline

1. Problem Formulation
2. Dual Decomposition
3. Augmented Lagrangian Method
4. Experiments
5. Conclusions

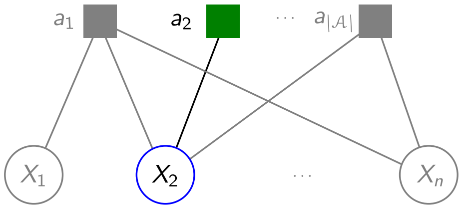
Dual *Decomposition* (Komodakis et al., 2007)



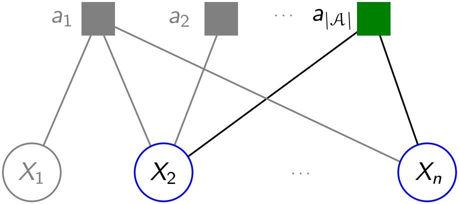
Dual *Decomposition* (Komodakis et al., 2007)



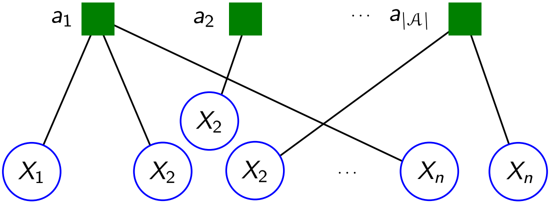
Dual *Decomposition* (Komodakis et al., 2007)



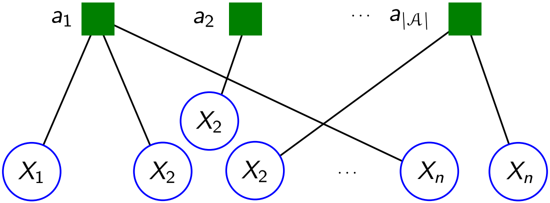
Dual *Decomposition* (Komodakis et al., 2007)



Dual *Decomposition* (Komodakis et al., 2007)

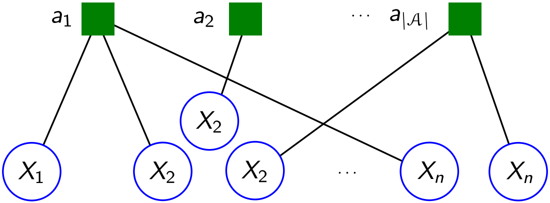


Dual *Decomposition* (Komodakis et al., 2007)



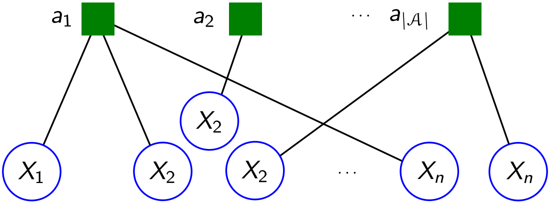
$$
\begin{aligned}
\max_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad & \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\mu}_i + \sum_{a \in \mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \\
\text{s.t.} \quad & (\boldsymbol{\mu}_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A}
\end{aligned}
$$

Dual *Decomposition* (Komodakis et al., 2007)



$$
\begin{aligned}
\max_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad & \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \sum_{a \in \mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \\
\text{s.t.} \quad & (\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A} \\
& \boldsymbol{\nu}_i^a = \boldsymbol{\mu}_i, \quad \forall a \in \mathcal{A}, i \in N(a)
\end{aligned}
$$

Dual *Decomposition* (Komodakis et al., 2007)

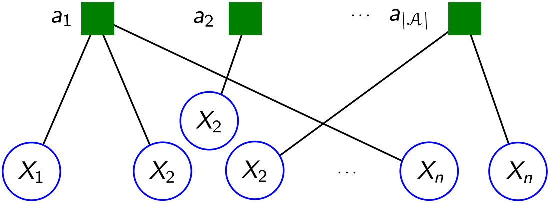


$$\max_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a = \sum_{a\in\mathcal{A}} \left(\sum_{i\in N(a)} d_i^{-1} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right)$$

$$\text{s.t.} \quad (\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A}$$

$$\boldsymbol{\nu}_i^a = \boldsymbol{\mu}_i, \quad \forall a \in \mathcal{A}, i \in N(a)$$

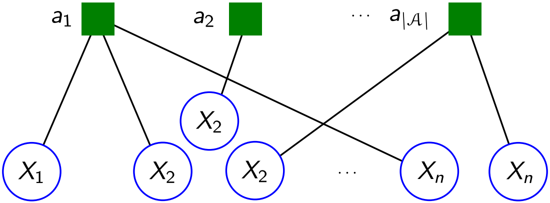
Dual *Decomposition* (Komodakis et al., 2007)



$$
\begin{aligned}
\max_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad & \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \sum_{a\in\mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a = \sum_{a\in\mathcal{A}} \left(\sum_{i\in N(a)} d_i^{-1} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) \\
\text{s.t.} \quad & (\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A} \\
& \boldsymbol{\nu}_i^a = \boldsymbol{\mu}_i, \quad \forall a \in \mathcal{A}, i \in N(a)
\end{aligned}
$$

- Problem would be separable, if not for the *coupling constraints*

Dual *Decomposition* (Komodakis et al., 2007)



$$\begin{aligned} \max_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad & \sum_{i=1}^{n} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \sum_{a \in \mathcal{A}} \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a = \sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} d_i^{-1} \boldsymbol{\theta}_i^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) \\ \text{s.t.} \quad & (\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a), \quad \forall a \in \mathcal{A} \\ & \boldsymbol{\nu}_i^a = \boldsymbol{\mu}_i, \quad \forall a \in \mathcal{A}, i \in N(a) \end{aligned}$$

- Problem would be separable, if not for the *coupling constraints*
- Dualize them out by adding Lagrange multipliers $\boldsymbol{\lambda}_i^a$

Dual Decomposition

Dual formulation:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & L(\boldsymbol{\lambda}) \triangleq \sum_{a \in \mathcal{A}} \overbrace{\max_{(\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a\right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a}^{\text{Slave subproblem at factor } a} \\ \text{s.t.} \quad & \boldsymbol{\lambda} \in \Lambda \triangleq \left\{ \boldsymbol{\lambda} \;\middle|\; \sum_{a \in N(i)} \boldsymbol{\lambda}_i^a = 0, \ \forall i \right\} \end{aligned}$$

Dual Decomposition

Dual formulation:

$$
\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & L(\boldsymbol{\lambda}) \triangleq \sum_{a \in \mathcal{A}} \overbrace{\max_{(\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}^a_i\right)^\top \boldsymbol{\nu}^a_i + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a}^{\text{Slave subproblem at factor } a} \\
\text{s.t.} \quad & \boldsymbol{\lambda} \in \Lambda \triangleq \left\{ \boldsymbol{\lambda} \;\middle|\; \sum_{a \in N(i)} \boldsymbol{\lambda}^a_i = 0, \;\forall i \right\}
\end{aligned}
$$

Slave subproblems: one per each factor $a \in \mathcal{A}$—parallelizable!

Dual Decomposition

Dual formulation:

$$
\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & L(\boldsymbol{\lambda}) \triangleq \sum_{a \in \mathcal{A}} \overbrace{\max_{(\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a\right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a}^{\text{Slave subproblem at factor } a} \\
\text{s.t.} \quad & \boldsymbol{\lambda} \in \Lambda \triangleq \left\{ \boldsymbol{\lambda} \;\middle|\; \sum_{a \in N(i)} \boldsymbol{\lambda}_i^a = 0, \ \forall i \right\}
\end{aligned}
$$

Slave subproblems: one per each factor $a \in \mathcal{A}$—parallelizable!

Master problem: minimize $L(\boldsymbol{\lambda})$ via the *projected subgradient algorithm*

Dual Decomposition

Dual formulation:

$$
\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & L(\boldsymbol{\lambda}) \triangleq \sum_{a \in \mathcal{A}} \overbrace{\max_{(\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}^a_i\right)^\top \boldsymbol{\nu}^a_i + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a}^{\text{Slave subproblem at factor } a} \\
\text{s.t.} \quad & \boldsymbol{\lambda} \in \Lambda \triangleq \left\{ \boldsymbol{\lambda} \;\middle|\; \sum_{a \in N(i)} \boldsymbol{\lambda}^a_i = 0, \ \forall i \right\}
\end{aligned}
$$

Slave subproblems: one per each factor $a \in \mathcal{A}$—parallelizable!

Master problem: minimize $L(\boldsymbol{\lambda})$ via the *projected subgradient algorithm*

- Subgradient given by the slaves:

$$\nabla_{\boldsymbol{\lambda}^a_i} L(\boldsymbol{\lambda}) = \hat{\boldsymbol{\nu}}^a_i \quad \text{(solution of a } \textit{local} \text{ MAP subproblem at } \mathcal{G}_a\text{)}$$

Dual Decomposition

Dual formulation:

$$\begin{array}{ll} \min_{\boldsymbol{\lambda}} & L(\boldsymbol{\lambda}) \triangleq \sum_{a \in \mathcal{A}} \overbrace{\max_{(\boldsymbol{\nu}^a_{N(a)}, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \sum_{i \in N(a)} \left(d_i^{-1}\boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a\right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a}^{\text{Slave subproblem at factor } a} \\ \text{s.t.} & \boldsymbol{\lambda} \in \Lambda \triangleq \left\{ \boldsymbol{\lambda} \;\middle|\; \sum_{a \in N(i)} \boldsymbol{\lambda}_i^a = 0, \; \forall i \right\} \end{array}$$

Slave subproblems: one per each factor $a \in \mathcal{A}$—parallelizable!

Master problem: minimize $L(\boldsymbol{\lambda})$ via the *projected subgradient algorithm*

- Subgradient given by the slaves:

$$\nabla_{\boldsymbol{\lambda}_i^a} L(\boldsymbol{\lambda}) = \hat{\boldsymbol{\nu}}_i^a \quad \text{(solution of a } \textit{local} \text{ MAP subproblem at } \mathcal{G}_a\text{)}$$

- Projection onto Λ: a simple *centering* operation

DD-Subgradient Algorithm

input: $\mathcal{G}$, $\boldsymbol{\theta}, \boldsymbol{\phi}$, number of iterations T, sequence $(\eta_t)_{t=1}^T$
Initialize $\boldsymbol{\lambda} = \mathbf{0}$
for $t = 1$ **to** T **do**
  for each factor $a \in \mathcal{A}$ **do**
    Set unary potentials $\boldsymbol{\omega}_i^a = d_i^{-1}\boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a$, for $i \in N(a)$
    Compute $(\hat{\boldsymbol{\nu}}_{N(a)}^a, \hat{\boldsymbol{\nu}}_a) = \mathrm{MAP}\left(\boldsymbol{\omega}_{N(a)}^a, \boldsymbol{\phi}_a\right)$
  end for
  Compute average $\bar{\boldsymbol{\nu}}_i = d_i^{-1} \sum_{a:i\in N(a)} \hat{\boldsymbol{\nu}}_i^a$
  Update $\boldsymbol{\lambda}_i^a \leftarrow \boldsymbol{\lambda}_i^a - \eta_t \left(\hat{\boldsymbol{\nu}}_i^a - \bar{\boldsymbol{\nu}}_i\right)$
end for
output: $\boldsymbol{\lambda}$

- Converges for a suitable stepsize sequence $(\eta_t)_{t\in T}$
- *Slow* when the number of slaves is large

Outline

Augmented Lagrangian and ADMM

$$
\begin{aligned}
A_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \quad \triangleq \quad & \overbrace{\sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a \right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) - \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \boldsymbol{\lambda}_i^{a\top} \boldsymbol{\mu}_i}^{\text{Lagrangian function}} \\
& \underbrace{- \frac{\eta}{2} \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \| \boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i \|^2}_{\text{Residual Term}}
\end{aligned}
$$

Augmented Lagrangian and ADMM

$$A_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \triangleq \overbrace{\sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a \right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) - \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \boldsymbol{\lambda}_i^{a\top} \boldsymbol{\mu}_i}^{\text{Lagrangian function}}$$

$$-\frac{\eta}{2} \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \|\boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i\|^2$$

Residual Term

We want to maximize A_η wrt $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$

Augmented Lagrangian and ADMM

$$A_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \triangleq \overbrace{\sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a \right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) - \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \boldsymbol{\lambda}_i^{a\top} \boldsymbol{\mu}_i}^{\text{Lagrangian function}}$$
$$- \frac{\eta}{2} \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \| \boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i \|^2$$

Residual Term

We want to maximize A_η wrt $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$

- Problem: the quadratic term breaks the separability!

Augmented Lagrangian and ADMM

$$A_\eta(\boldsymbol{\mu},\boldsymbol{\nu},\boldsymbol{\lambda}) \triangleq \overbrace{\sum_{a\in\mathcal{A}}\left(\sum_{i\in N(a)}\left(d_i^{-1}\boldsymbol{\theta}_i+\boldsymbol{\lambda}_i^a\right)^\top\boldsymbol{\nu}_i^a+\boldsymbol{\phi}_a^\top\boldsymbol{\nu}_a\right)-\sum_{a\in\mathcal{A}}\sum_{i\in N(a)}\boldsymbol{\lambda}_i^{a\top}\boldsymbol{\mu}_i}^{\text{Lagrangian function}}$$
$$-\frac{\eta}{2}\sum_{a\in\mathcal{A}}\sum_{i\in N(a)}\|\boldsymbol{\nu}_i^a-\boldsymbol{\mu}_i\|^2$$

Residual Term

We want to maximize A_η wrt $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$

- Problem: the quadratic term breaks the separability!
- *Alternating Direction Method of Multipliers* (Gabay and Mercier, 1976; Glowinski and Marroco, 1975):

Augmented Lagrangian and ADMM

$$A_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \triangleq \overbrace{\sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a \right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) - \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \boldsymbol{\lambda}_i^{a\top} \boldsymbol{\mu}_i}^{\text{Lagrangian function}}$$

$$\underbrace{- \frac{\eta}{2} \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \| \boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i \|^2}_{\text{Residual Term}}$$

We want to maximize A_η wrt $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$

- Problem: the quadratic term breaks the separability!
- *Alternating Direction Method of Multipliers* (Gabay and Mercier, 1976; Glowinski and Marroco, 1975):

1. Maximize w.r.t. $\boldsymbol{\mu}$ (closed form)

Augmented Lagrangian and ADMM

$$A_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \triangleq \overbrace{\sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a \right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) - \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \boldsymbol{\lambda}_i^{a\top} \boldsymbol{\mu}_i}^{\text{Lagrangian function}}$$
$$\underbrace{- \frac{\eta}{2} \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \| \boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i \|^2}_{\text{Residual Term}}$$

We want to maximize A_η wrt $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$

- Problem: the quadratic term breaks the separability!
- *Alternating Direction Method of Multipliers* (Gabay and Mercier, 1976; Glowinski and Marroco, 1975):

1. Maximize w.r.t. $\boldsymbol{\mu}$ (closed form)
2. Maximize w.r.t. $\boldsymbol{\nu}$ (can be carried out in parallel at each factor)

Augmented Lagrangian and ADMM

$$A_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \triangleq \overbrace{\sum_{a \in \mathcal{A}} \left(\sum_{i \in N(a)} \left(d_i^{-1} \boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a \right)^\top \boldsymbol{\nu}_i^a + \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a \right) - \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \boldsymbol{\lambda}_i^{a\top} \boldsymbol{\mu}_i}^{\text{Lagrangian function}}$$
$$- \frac{\eta}{2} \sum_{a \in \mathcal{A}} \sum_{i \in N(a)} \| \boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i \|^2$$

Residual Term

We want to maximize A_η wrt $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$

- Problem: the quadratic term breaks the separability!
- *Alternating Direction Method of Multipliers* (Gabay and Mercier, 1976; Glowinski and Marroco, 1975):

1. Maximize w.r.t. $\boldsymbol{\mu}$ (closed form)
2. Maximize w.r.t. $\boldsymbol{\nu}$ (can be carried out in parallel at each factor)
3. Adjust the Lagrange multipliers $\boldsymbol{\lambda}_i^a \leftarrow \boldsymbol{\lambda}_i^a - \tau\eta \left(\boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i \right)$

DD-ADMM Algorithm

input: $\mathcal{G}$, $\boldsymbol{\theta}, \boldsymbol{\phi}$, number of iterations T, sequence $(\eta_t)_{t=1}^{T}$, parameter τ
Initialize $\boldsymbol{\mu}$ uniformly, $\boldsymbol{\lambda} = \mathbf{0}$
for $t = 1$ **to** T **do**
  for each factor $a \in \mathcal{A}$ **do**
    Set unary potentials $\boldsymbol{\omega}_i^a = d_i^{-1}\boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a + \eta_t \boldsymbol{\mu}_i$, for $i \in N(a)$
    Update $(\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \leftarrow \mathrm{QUAD}_{\eta_t}(\boldsymbol{\omega}_{N(a)}^a, \boldsymbol{\phi}_a)$
  end for
  Update $\boldsymbol{\mu}_i \leftarrow d_i^{-1} \sum_{a:i\in N(a)} (\boldsymbol{\nu}_i^a - \eta_t^{-1}\boldsymbol{\lambda}_i^a)$
  Update $\boldsymbol{\lambda}_i^a \leftarrow \boldsymbol{\lambda}_i^a - \tau\eta_t (\boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i)$
end for
output: $\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}$

DD-ADMM Algorithm

input: $\mathcal{G}$, $\boldsymbol{\theta}, \boldsymbol{\phi}$, number of iterations T, sequence $(\eta_t)_{t=1}^{T}$, parameter τ
Initialize $\boldsymbol{\mu}$ uniformly, $\boldsymbol{\lambda} = \mathbf{0}$
for $t = 1$ **to** T **do**
  for each factor $a \in \mathcal{A}$ **do**
    Set unary potentials $\boldsymbol{\omega}_i^a = d_i^{-1}\boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a + \eta_t \boldsymbol{\mu}_i$, for $i \in N(a)$
    Update $(\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \leftarrow \mathrm{QUAD}_{\eta_t}(\boldsymbol{\omega}_{N(a)}^a, \boldsymbol{\phi}_a)$
  end for
  Update $\boldsymbol{\mu}_i \leftarrow d_i^{-1} \sum_{a:i \in N(a)} (\boldsymbol{\nu}_i^a - \eta_t^{-1}\boldsymbol{\lambda}_i^a)$
  Update $\boldsymbol{\lambda}_i^a \leftarrow \boldsymbol{\lambda}_i^a - \tau\eta_t (\boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i)$
end for
output: $\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}$

- Instead of MAP, we solve a quadratic problem QUAD at each factor:

$$\min_{(\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \frac{\eta_t}{2} \sum_{i \in a} \|\boldsymbol{\nu}_i^a - \eta_t^{-1}\boldsymbol{\omega}_i^a\|^2 - \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a,$$

DD-ADMM Algorithm

input: $\mathcal{G}$, $\boldsymbol{\theta}, \boldsymbol{\phi}$, number of iterations T, sequence $(\eta_t)_{t=1}^{T}$, parameter τ
Initialize $\boldsymbol{\mu}$ uniformly, $\boldsymbol{\lambda} = \mathbf{0}$
for $t = 1$ **to** T **do**
  for each factor $a \in \mathcal{A}$ **do**
    Set unary potentials $\boldsymbol{\omega}_i^a = d_i^{-1}\boldsymbol{\theta}_i + \boldsymbol{\lambda}_i^a + \eta_t \boldsymbol{\mu}_i$, for $i \in N(a)$
    Update $(\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \leftarrow \mathrm{QUAD}_{\eta_t}(\boldsymbol{\omega}_{N(a)}^a, \boldsymbol{\phi}_a)$
  end for
  Update $\boldsymbol{\mu}_i \leftarrow d_i^{-1} \sum_{a:i \in N(a)} (\boldsymbol{\nu}_i^a - \eta_t^{-1}\boldsymbol{\lambda}_i^a)$
  Update $\boldsymbol{\lambda}_i^a \leftarrow \boldsymbol{\lambda}_i^a - \tau\eta_t (\boldsymbol{\nu}_i^a - \boldsymbol{\mu}_i)$
end for
output: $\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}$

- Instead of MAP, we solve a quadratic problem QUAD at each factor:

$$\min_{(\boldsymbol{\nu}_{N(a)}^a, \boldsymbol{\nu}_a) \in \mathcal{M}(\mathcal{G}_a)} \frac{\eta_t}{2} \sum_{i \in a} \|\boldsymbol{\nu}_i^a - \eta_t^{-1}\boldsymbol{\omega}_i^a\|^2 - \boldsymbol{\phi}_a^\top \boldsymbol{\nu}_a,$$

- Under certain conditions, converges even when QUAD is *approximately* solved (Eckstein and Bertsekas, 1992).

Solving the Quadratic Problem

Binary pairwise factors: closed form solution

Solving the Quadratic Problem

Binary pairwise factors: closed form solution

Hard constraint factors (Tarlow et al., 2010; Martins et al., 2010):

$$\phi_a(\mathbf{x}_a) = \begin{cases} 0 & \text{if } \mathbf{x}_a \in \mathcal{S}_a \\ -\infty & \text{otherwise} \end{cases}$$

Solving the Quadratic Problem

Binary pairwise factors: closed form solution

Hard constraint factors (Tarlow et al., 2010; Martins et al., 2010):

$$\phi_a(\mathbf{x}_a) = \begin{cases} 0 & \text{if } \mathbf{x}_a \in \mathcal{S}_a \\ -\infty & \text{otherwise} \end{cases}$$

- Binary variables: $\textsc{quad}$ = project onto $\operatorname{conv} \mathcal{S}_a$

Solving the Quadratic Problem

Binary pairwise factors: closed form solution

Hard constraint factors (Tarlow et al., 2010; Martins et al., 2010):

$$\phi_a(\mathbf{x}_a) = \begin{cases} 0 & \text{if } \mathbf{x}_a \in \mathcal{S}_a \\ -\infty & \text{otherwise} \end{cases}$$

- Binary variables: QUAD = project onto $\operatorname{conv} \mathcal{S}_a$
- For many hard factors imposing logical constraints, can be done efficiently with *sort* operations

Solving the Quadratic Problem

Binary pairwise factors: closed form solution

Hard constraint factors (Tarlow et al., 2010; Martins et al., 2010):

$$\phi_a(\mathbf{x}_a) = \begin{cases} 0 & \text{if } \mathbf{x}_a \in \mathcal{S}_a \\ -\infty & \text{otherwise} \end{cases}$$

- Binary variables: QUAD = project onto $\operatorname{conv} \mathcal{S}_a$
- For many hard factors imposing logical constraints, can be done efficiently with *sort* operations

Larger slaves (sequences, trees): solved approximately with primal-dual cyclic projection algorithms (work in progress)

Outline

Binary Pairwise MRFs

Ising model on a random 30×30 toroidal grid; DD-ADMM against:

Binary Pairwise MRFs

Ising model on a random 30×30 toroidal grid; DD-ADMM against:

- *DD-Subgradient* (Komodakis et al., 2007)

Binary Pairwise MRFs

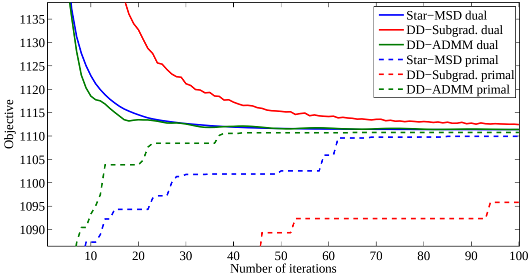
Ising model on a random 30×30 toroidal grid; DD-ADMM against:

- *DD-Subgradient* (Komodakis et al., 2007)
- *Max-sum diffusion with star updates* (Meshi et al., 2010): LP-MAP inference via dual block coordinate descent message-passing

Binary Pairwise MRFs

Ising model on a random 30 $\times$ 30 toroidal grid; DD-ADMM against:

- *DD-Subgradient* (Komodakis et al., 2007)
- *Max-sum diffusion with star updates* (Meshi et al., 2010): LP-MAP inference via dual block coordinate descent message-passing

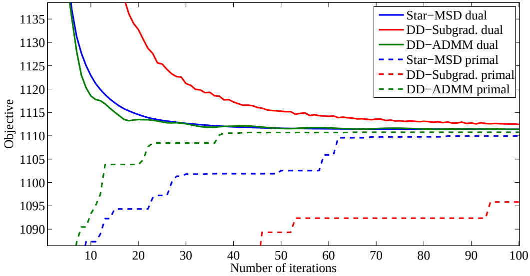


- DD-subgradient is the slowest (many slaves!)

Binary Pairwise MRFs

Ising model on a random 30 × 30 toroidal grid; DD-ADMM against:

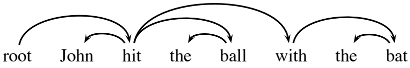
- *DD-Subgradient* (Komodakis et al., 2007)
- *Max-sum diffusion with star updates* (Meshi et al., 2010): LP-MAP inference via dual block coordinate descent message-passing



- DD-subgradient is the slowest (many slaves!)
- DD-ADMM outperforms the others: it approaches a near optimal primal-dual solution in a few tens of iterations

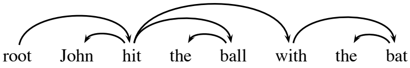
Non-projective Dependency Parsing

A problem with heavily constrained outputs, to which DD has recently been applied (Rush et al., 2010; Koo et al., 2010)

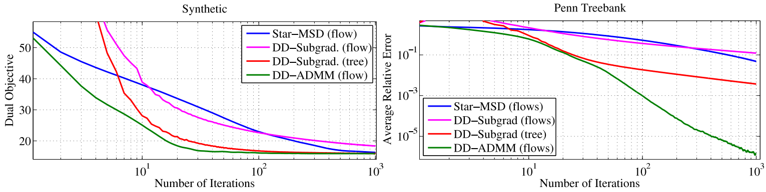


Non-projective Dependency Parsing

A problem with heavily constrained outputs, to which DD has recently been applied (Rush et al., 2010; Koo et al., 2010)

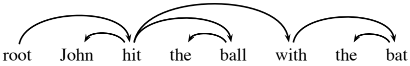


Our model: 2nd-order with $O(n^3)$ slaves (distinct from Koo et al. 2010)

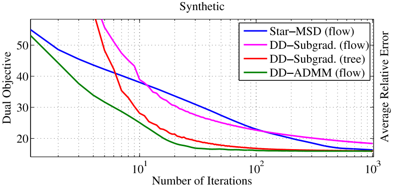


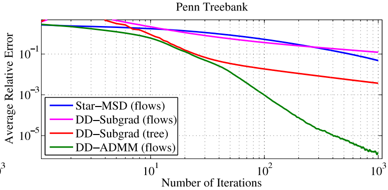
Non-projective Dependency Parsing

A problem with heavily constrained outputs, to which DD has recently been applied (Rush et al., 2010; Koo et al., 2010)



Our model: 2nd-order with $O(n^3)$ slaves (distinct from Koo et al. 2010)

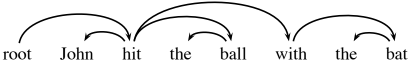




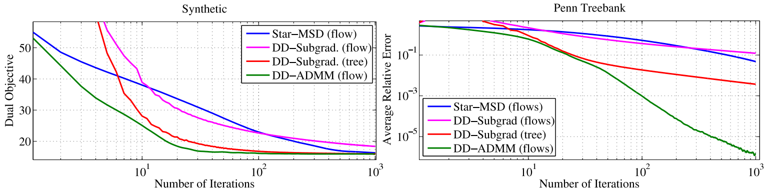
- DD-ADMM wins with both synthetic (10 words) and real sentences

Non-projective Dependency Parsing

A problem with heavily constrained outputs, to which DD has recently been applied (Rush et al., 2010; Koo et al., 2010)



Our model: 2nd-order with $O(n^3)$ slaves (distinct from Koo et al. 2010)



- DD-ADMM wins with both synthetic (10 words) and real sentences
- Also outperforms DD-Subgradient with a TREE model (fewer slaves)

Outline

Conclusions

- DD-ADMM: a new algorithm for LP-MAP inference
- Dual decomposable, hence the slaves can all be solved in parallel
- Allies the simplicity of DD with the effectiveness of AL methods
- Suitable for problems with many slaves, outperforming Komodakis et al. (2007)
- Optimality certificates for LP-MAP (not just MAP)
- A significant amount of computation can be saved by *caching* and *warm-starting* the subproblems
- Related work in accelerating DD and in quadratic projections: Jojic et al. (2010); Ravikumar et al. (2010)
- Future work: larger slaves and *approximate* ADMM steps

Thank you!

- Questions?

References I

Dantzig, G. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations research*, 8(1):101–111.

Eckstein, J. and Bertsekas, D. (1992). On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318.

Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17–40.

Globerson, A. and Jaakkola, T. (2008). Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. *NIPS*, 20.

Glowinski, R. and Marroco, A. (1975). Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité d'une classe de problemes de Dirichlet non lineares. *Revue Française d'Automatique, Informatique et Recherche Opérationelle*, 9:41–76.

Hestenes, M. (1969). Multiplier and gradient methods. *Jour. Optim. Theory and Applic.*, 4:302–320.

Jojic, V., Gould, S., and Koller, D. (2010). Accelerated dual decomposition for MAP inference. In *Proc. of ICML*.

Komodakis, N., Paragios, N., and Tziritas, G. (2007). MRF optimization via dual decomposition: Message-passing revisited. In *International Conference on Computer Vision*. Citeseer.

Koo, T., Rush, A. M., Collins, M., Jaakkola, T., and Sontag, D. (2010). Dual decomposition for parsing with non-projective head automata. In *Proc. of EMNLP*.

Martins, A. F. T., Smith, N. A., Xing, E. P., Figueiredo, M. A. T., and Aguiar, P. M. Q. (2010). Turbo parsers: Dependency parsing by approximate variational inference. In *Proc. of EMNLP*.

Meshi, O., Sontag, D., Jaakkola, T., and Globerson, A. (2010). Learning Efficiently with Approximate Inference via Dual Losses. In *Proc. ICML*. Citeseer.

Powell, M. (1969). A method for nonlinear constraints in minimization problems. In Fletcher, R., editor, *Optimization*, pages 283–298. Academic Press.

Ravikumar, P., Agarwal, A., and Wainwright, M. (2010). Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *JMLR*, 11:1043–1080.

Rush, A. M., Sontag, D., Collins, M., and Jaakkola, T. (2010). On dual decomposition and linear programming relaxations for natural language processing. In *Proc. of EMNLP*.

References II

Schlesinger, M. (1976). Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika, Kiev*, 4:113–130.

Tarlow, D., Givoni, I. E., and Zemel, R. S. (2010). HOP-MAP: Efficient message passing with high order potentials. In *Proc. of AISTATS*.

Wainwright, M., Jaakkola, T., and Willsky, A. (2005). MAP estimation via agreement on trees: message-passing and linear programming. *Information Theory, IEEE Transactions on*, 51(11):3697–3717.

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers.