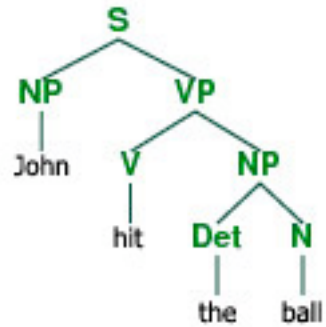
CSAIL

Exponentiated Gradient Algorithms for Structured Classification

Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras (MIT CSAIL)
and Peter Bartlett (UC Berkeley)

Classification Algorithms

- Classify objects $x \in \mathcal{X}$ into labels $y \in \mathcal{Y}$
- First there was binary  $\mathcal{Y} = \{5, 7\}$
- Then multi-class  $\mathcal{Y} = \{4, 5, 7\}$
- Structured labels **John hit the ball** →



- Two popular approaches: CRFs and max-margin models
- Getting harder to do efficiently

CSAIL

Outline

- Log-linear and max-margin models for classification
- Optimizing via the dual
- Exponentiated Gradient (EG) dual updates for log-linear and max-margin models
- Convergence analysis
- Structured Implementation
- Experiments

Log-Linear Models (e.g. Lafferty et al.)

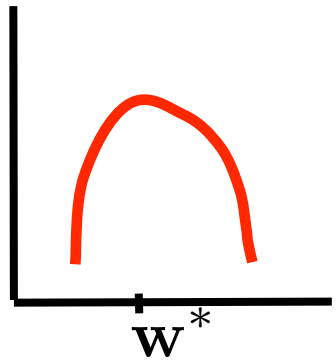
- Assume a function: $\phi(\boldsymbol{x}, y) : \mathcal{X} \otimes \mathcal{Y} \rightarrow \mathbb{R}^d$
- For every **x**, define a distribution over labels:

$$p(y|\boldsymbol{x}; \mathbf{w}) = \frac{1}{Z_{\boldsymbol{x}}} e^{\mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, y)}$$

- Use labeled set $\{(x_i, y_i)\}_{i=1}^n$ to learn **w**
- Maximize likelihood with regularization

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_i \log p(y_i \mid \boldsymbol{x}_i; \mathbf{w}) - \frac{C}{2} ||\mathbf{w}||^2$$

- Concave unconstrained problem.



Max-Margin Models (e.g. Taskar et al.)

- Find parameters that minimize the regularized hinge loss

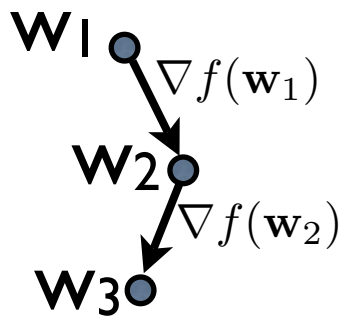
$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} - \sum_i Hinge(\boldsymbol{x}_i, y_i, \mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2$$

- Corresponds to maximization of a "margin"

- Concave unconstrained problem

Gradient Minimization

- Gradient algorithms work by:
  - Calculating objective and gradient by summing over all samples
  - Performing a gradient step



- Each step is O(n) where n is num. of samples.
- Stochastic Gradient (Zhang 04, Le-Cun 98).
  - Estimate gradient via single sample. Takes O(1).
  - No clear measure of improvement, since objective cannot be calculated in O(1)

Desired Properties

- Update model after each sample - O(1)
- Measure improvement in objective in O(1)
- Step size can then be chosen to improve objective with certainty
- Solution: Optimize the dual (Jaakkola & Haussler 99, Platt 98)

Dual Log-Linear Problem

- Recall log-linear loss minimization

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \sum_i \log p(y_i|\boldsymbol{x}_i;\mathbf{w}) - \frac{C}{2}\|\mathbf{w}\|^2$$

- Has an equivalent convex dual (Lebanon & Lafferty,2001)
- Dual variables: $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$ one per training example
- Each $\boldsymbol{\alpha}_i$ is a distribution over $\mathcal{Y}$: $\boldsymbol{\alpha}_i \in \Delta$

$$\Delta = \left\{ \mathbf{p} \in \mathbb{R}^{|\mathcal{Y}|} \; : \; p_y \geq 0 \, , \sum_{y \in \mathcal{Y}} p_y = 1 \right\}$$

Dual Log-Linear Objective

- Define: $\boldsymbol{\psi}_{i,y} = \boldsymbol{\phi}(x_i, y_i) - \boldsymbol{\phi}(x_i, y)$

$$\mathbf{w}(\boldsymbol{\alpha}) = \sum_i \sum_y \alpha_{i,y} \boldsymbol{\psi}_{i,y}$$

- Dual Objective:

$$Q_{\text{LL}}(\boldsymbol{\alpha}) = \sum_i \sum_y \alpha_{i,y} \log \alpha_{i,y} + \frac{1}{2C} \|\mathbf{w}(\boldsymbol{\alpha})\|^2$$

$$Q_{\text{LL}}(\boldsymbol{\alpha}) = -\sum_i H(\boldsymbol{\alpha}_i) + \frac{1}{2C} ||\mathbf{w}(\boldsymbol{\alpha})||^2$$

Primal and Dual Problems

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \begin{array}{ll} \text{argmax} & \sum_i \log p(y_i|\boldsymbol{x}_i;\mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \\ s.t. & \mathbf{w} \in \mathbb{R}^d \end{array}$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i H(\boldsymbol{\alpha}_i) + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

- Primal-Dual mapping: $C\mathbf{w}^* = \mathbf{w}(\boldsymbol{\alpha}^*)$

Primal and Dual Problems

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \begin{array}{ll} \text{argmax} & \sum_i \log p(y_i|\boldsymbol{x}_i;\mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \\ s.t. & \mathbf{w} \in \mathbb{R}^d \end{array}$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i H(\boldsymbol{\alpha}_i) + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

- Primal-Dual mapping: $C\mathbf{w}^* = \mathbf{w}(\boldsymbol{\alpha}^*)$

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \begin{array}{ll} \text{argmax} & -\sum_i Hinge(\boldsymbol{x}_i, y_i, \mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \\ s.t. & \mathbf{w} \in \mathbb{R}^d \end{array}$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i \mathbf{b}_i \cdot \boldsymbol{\alpha}_i + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

Primal and Dual Problems

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \underset{s.t.}{\text{argmax}} \quad \sum_i \log p(y_i|\boldsymbol{x}_i;\mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \quad \mathbf{w} \in \mathbb{R}^d$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \underset{s.t.}{\text{argmin}} \quad -\sum_i H(\boldsymbol{\alpha}_i) + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \quad \boldsymbol{\alpha}_i \in \Delta$$

- Primal-Dual mapping: $C\mathbf{w}^* = \mathbf{w}(\boldsymbol{\alpha}^*)$

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \underset{s.t.}{\text{argmax}} \quad -\sum_i Hinge(\boldsymbol{x}_i, y_i, \mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \quad \mathbf{w} \in \mathbb{R}^d$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \underset{s.t.}{\text{argmin}} \quad -\sum_i \mathbf{b}_i \cdot \boldsymbol{\alpha}_i + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \quad \boldsymbol{\alpha}_i \in \Delta$$

Primal and Dual Problems

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \begin{array}{ll} \text{argmax} & \sum_i \log p(y_i|\boldsymbol{x}_i;\mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \\ s.t. & \mathbf{w} \in \mathbb{R}^d \end{array}$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i H(\boldsymbol{\alpha}_i) + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

- Primal-Dual mapping: $C\mathbf{w}^* = \mathbf{w}(\boldsymbol{\alpha}^*)$

$$\underline{\text{Primal}}: \quad \mathbf{w}^* = \begin{array}{ll} \text{argmax} & -\sum_i Hinge(\boldsymbol{x}_i, y_i, \mathbf{w}) - \frac{C}{2}||\mathbf{w}||^2 \\ s.t. & \mathbf{w} \in \mathbb{R}^d \end{array}$$

$$\underline{\text{Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i \mathbf{b}_i \cdot \boldsymbol{\alpha}_i + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

Dual Optimization

$$\underline{\text{LL-Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i H(\boldsymbol{\alpha}_i) - \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

$$\underline{\text{MM-Dual}}: \quad \boldsymbol{\alpha}^* = \begin{array}{ll} \text{argmin} & -\sum_i \mathbf{b}_i \cdot \boldsymbol{\alpha}_i + \frac{1}{2C}||\mathbf{w}(\boldsymbol{\alpha})||^2 \\ s.t. & \boldsymbol{\alpha}_i \in \Delta \end{array}$$

$$\mathbf{w}(\boldsymbol{\alpha}) = \sum_i \sum_y \alpha_{i,y} \boldsymbol{\psi}_{i,y}$$

- Optimize the dual one $\boldsymbol{\alpha}_i$ at a time (e.g. Platt's SMO)
  - Don't have closed form
  - Need to satisfy constraints

Exponentiated Gradient (EG) Updates

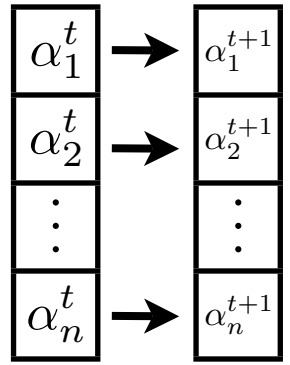
- Introduced by Kivinen and Warmuth (97)
- Apply to minimization of function $Q(\alpha)$ subject to simplex constraints
- Update rule $\alpha_{i,y} \longrightarrow \alpha'_{i,y}$

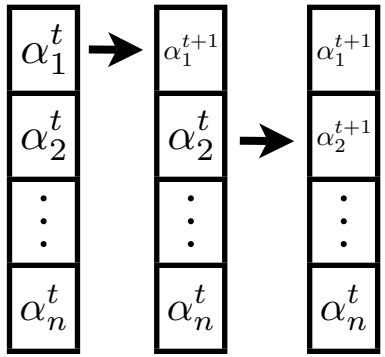
$$\alpha'_{i,y} = \frac{1}{Z_i}\alpha_{i,y}e^{-\eta\frac{\partial Q(\boldsymbol{\alpha})}{\partial \alpha_{i,y}}} \qquad Z_i = \sum_{\hat{y}} \alpha_{i,\hat{y}}e^{-\eta\frac{\partial Q(\boldsymbol{\alpha})}{\partial \alpha_{i,\hat{y}}}}$$

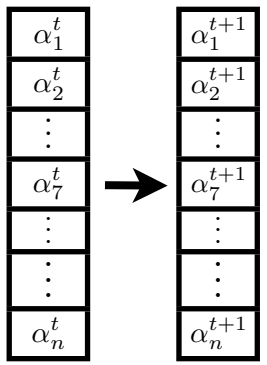
- Clearly $\boldsymbol{\alpha}'_i \in \Delta$

Batch and Online Variants

- Several possible update schemes:
  - Batch: At each iteration update all α_i
  - Online:
    - Deterministic: Update α_1, then α_2 ...
    - Stochastic: Pick i uniformly at random and update α_i







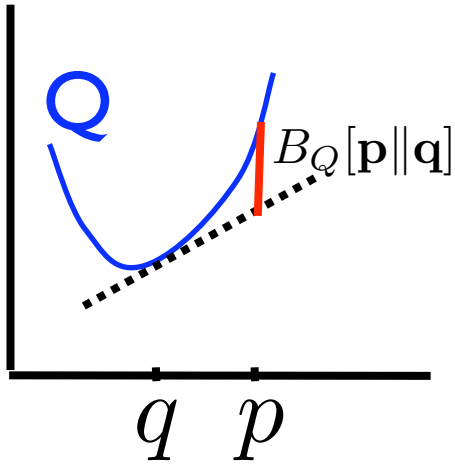
Convergence Properties

- There exists a fixed step size $\eta > 0$ for which:
  - All algorithms converge to α^*
    - For stochastic online, with probability $\delta > 0$
  - Every update improves the objective
  - $O(\log(1/\epsilon))$ rate of convergence for log-linear
  - $O(1/\epsilon)$ rate of convergence for max-margin
  - Stochastic online is n times faster than batch !
- Proofs involve Bregman divergences

Bregman Divergence

- For every convex Q, we can define

$$B_Q[\mathbf{p}\|\mathbf{q}] = Q(\mathbf{p}) - Q(\mathbf{q}) - \nabla Q(\mathbf{q}) \cdot (\mathbf{p} - \mathbf{q})$$



- Non-negative for all **p**,**q**
- Examples:

$Q(\mathbf{p})$	$B_Q[\mathbf{p}\|\mathbf{q}]$	
$\sum_i p_i \log p_i$	$D_{\mathrm{KL}}[p\|q] = \sum_i p_i \log \frac{p_i}{q_i}$	KL Diverg.
$\frac{1}{2}\mathbf{p}^T A \mathbf{p}$	$\frac{1}{2}(\mathbf{p} - \mathbf{q})^T A (\mathbf{p} - \mathbf{q})$	Mahalanobis

Monotone Updates

- A function $Q(\boldsymbol{\alpha})$ is τ-*upper-bounded* if

$$B_Q[\mathbf{p}\|\mathbf{q}] \leq \tau D_{\mathrm{KL}}[\mathbf{p}\|\mathbf{q}]$$

- If $Q(\boldsymbol{\alpha})$ is τ-*upper-bounded* and $\eta \leq 1/\tau$

$$Q(\boldsymbol{\alpha}^t) - Q(\boldsymbol{\alpha}^{t+1}) \geq \frac{1}{\eta} D_{\mathrm{KL}}[\boldsymbol{\alpha}^t \| \boldsymbol{\alpha}^{t+1}]$$

- For $\eta \leq 1/\tau$ update always improves $Q(\boldsymbol{\alpha})$!
- $|Q(\boldsymbol{\alpha}) - Q(\boldsymbol{\alpha}^*)| \leq \epsilon$ after $O(1/\epsilon)$ updates

Better Bounds

- A function $Q(\boldsymbol{\alpha})$ is (μ,τ) bounded if

$$\mu D_{\mathrm{KL}}[\mathbf{p}\|\mathbf{q}] \leq B_Q[\mathbf{p}\|\mathbf{q}] \leq \tau D_{\mathrm{KL}}[\mathbf{p}\|\mathbf{q}]$$

- For some $0 < \mu \leq \tau$

- $|Q(\boldsymbol{\alpha}) - Q(\boldsymbol{\alpha}^*)| \leq \epsilon$ after $O(\log(1/\epsilon))$ updates

- Similar to results for normal gradient descent

Rates of Convergence

- Log-linear dual objective satisfies

$$D_{\mathrm{KL}}[\mathbf{p}||\mathbf{q}] \leq B_{Q_{\mathrm{LL}}}[\mathbf{p}||\mathbf{q}] \leq (1 + nR^2/C) D_{\mathrm{KL}}[\mathbf{p}||\mathbf{q}]$$

- Rate of convergence is $O(\log(1/\epsilon))$

- Max-margin dual objective satisfies

$$B_{Q_{\mathrm{MM}}}[\mathbf{p}||\mathbf{q}] \leq (nR^2/C) D_{\mathrm{KL}}[\mathbf{p}||\mathbf{q}]$$

- Rate of convergence is $O(1/\epsilon)$

Stochastic Online Analysis

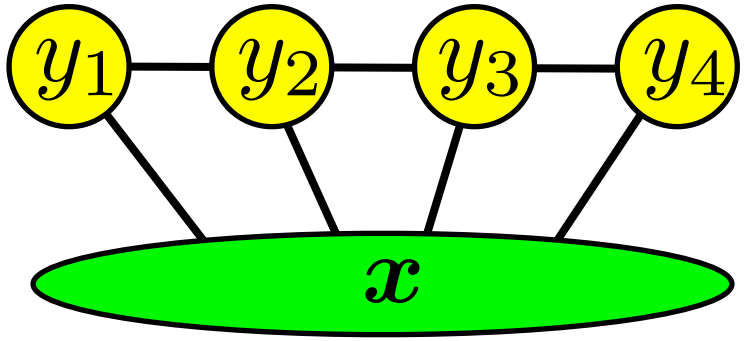
- Proof is more involved
- Basic intuition: take expectations over all possible random update sequences
- Rates are still $O(\log(1/\epsilon))$ and $O(1/\epsilon)$ for log-linear and max-margin
- Can use a larger step size (n times larger)

Outline

- Log-linear and max-margin models for classification
- Optimizing via the dual
- Exponentiated Gradient (EG) dual updates for log-linear and max-margin models
- Convergence analysis
- Structured Implementation
- Experiments

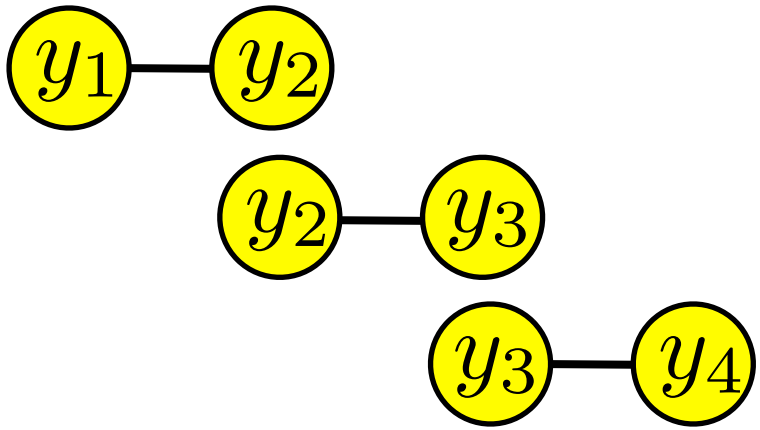
Structured Implementation

- When $\mathcal{Y}$ is large, can't use standard EG
- E.g., sequential pairwise CRF



- Assume factorization into "parts" (e.g., cliques)

$$\phi(x, y) = \sum_{jk \in E} \phi_{jk}(x, y_j, y_k)$$



Structured Implementation

- Define part-based dual variables $\boldsymbol{\theta}_i$

$$\boldsymbol{\theta}_i = \begin{cases} \theta_{i,y_1,y_2} & : \quad y_1 - y_2 \\ \theta_{i,y_2,y_3} & : \quad y_2 - y_3 \\ \theta_{i,y_3,y_4} & : \quad y_3 - y_4 \\ \ldots & \end{cases}$$

- "Normal" duals $\boldsymbol{\alpha}_i$ implicitly defined via

$$\alpha_{i,y}(\boldsymbol{\theta}_i) = \frac{1}{Z_i} \prod_{(y_j,y_k) \in y} e^{\theta_{i,y_j,y_k}}$$

Structured Implementation

- Factored update rule

$$\alpha'_{i,y} = \frac{1}{Z_i} \alpha_{i,y} e^{-\eta \frac{\partial Q(\boldsymbol{\alpha})}{\partial \alpha_{i,y}}}$$

$$\theta'_{i,y_j,y_k} = (1-\eta)\theta_{i,y_j,y_k} + \frac{\eta}{C} \mathbf{w}(\boldsymbol{\alpha}) \cdot \boldsymbol{\phi}_{jk}(\boldsymbol{x}_i, y_j, y_k)$$

Structured Implementation

- Factored $\mathbf{w}(\boldsymbol{\alpha})$

$$\mathbf{w}(\boldsymbol{\alpha}) = \sum_i \sum_y \alpha_{i,y} \boldsymbol{\psi}_{i,y}$$

$$\mathbf{w}(\boldsymbol{\alpha}) = \sum_i \sum_{(y_j, y_k)} \mu_{i,y_j,y_k} \boldsymbol{\psi}_{i,y_j,y_k}$$

- Requires computation of marginals

$$\mu_{i,y_j,y_k} = \sum_{y:(y_j,y_k)\in y} \alpha_{i,y}(\boldsymbol{\theta}_i)$$

- Marginals are also required by e.g. L-BFGS

CSAIL

Experiments

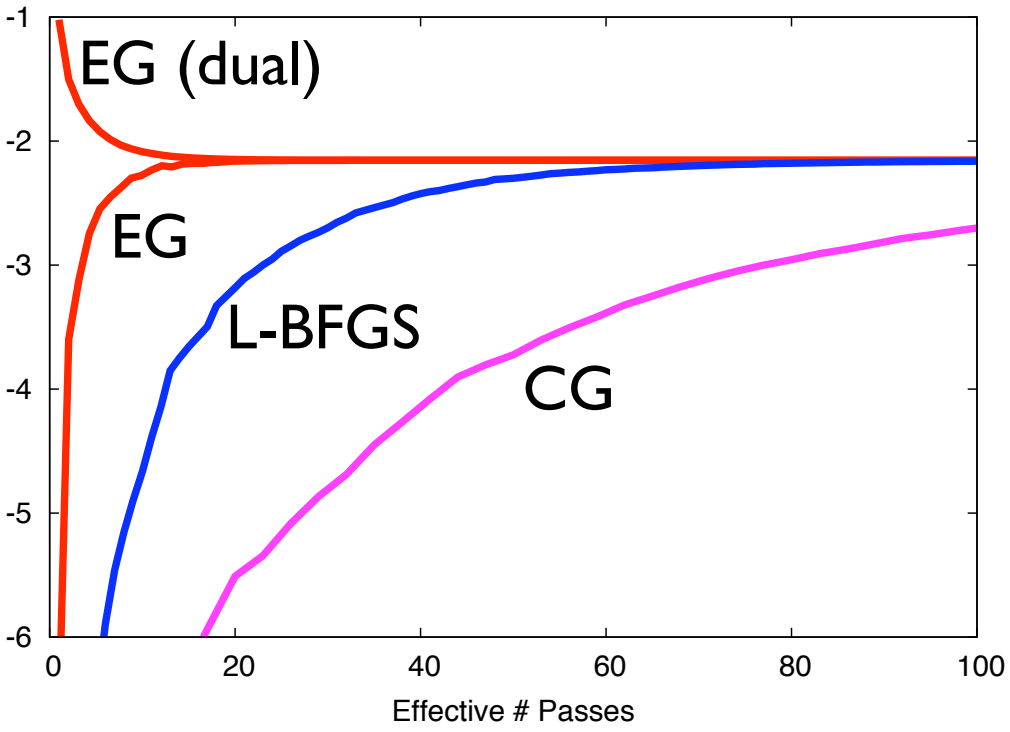
- Compare log-linear EG to CG, L-BFGS, SGD
- Compare max-margin EG to SVM-struct
- Used stochastic online EG
- Crude line search instead of fixed learning rate

Dependency Parsing

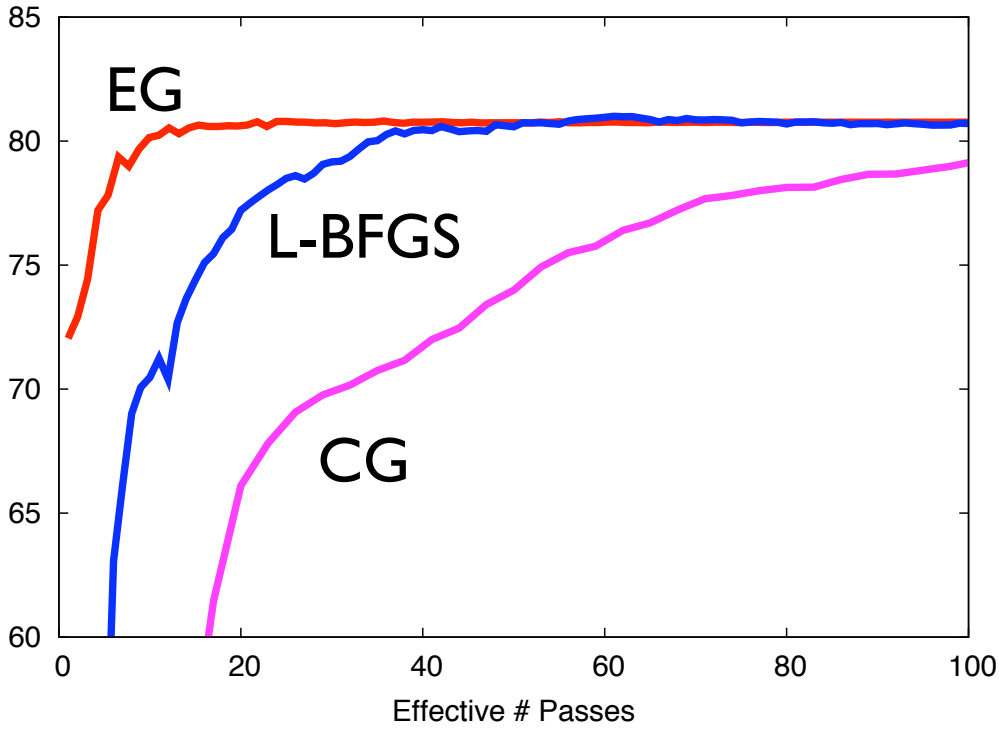
- Map sentences into trees
- Marginals can be calculated by inside-outside (Koo et al. EMNLP, 07).
- Used the Spanish language data (CoNLL-X)
- 2300 sentences, 59000 tokens
- 2.4 million features

EG and CG/L-BFGS

Objective Value



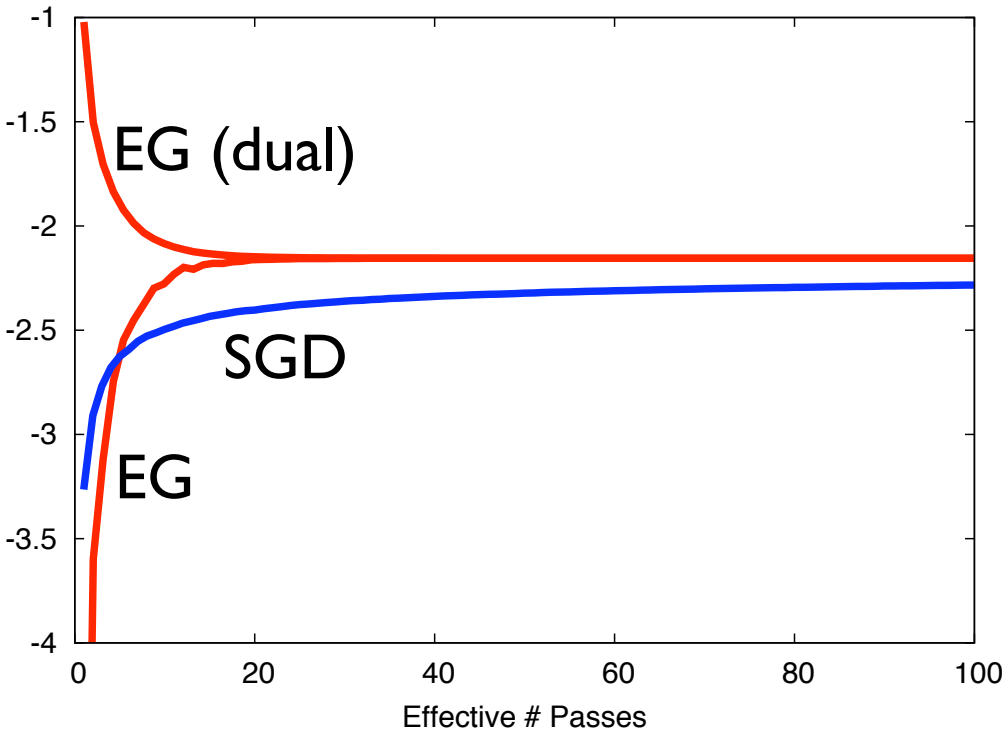
Parsing Accuracy



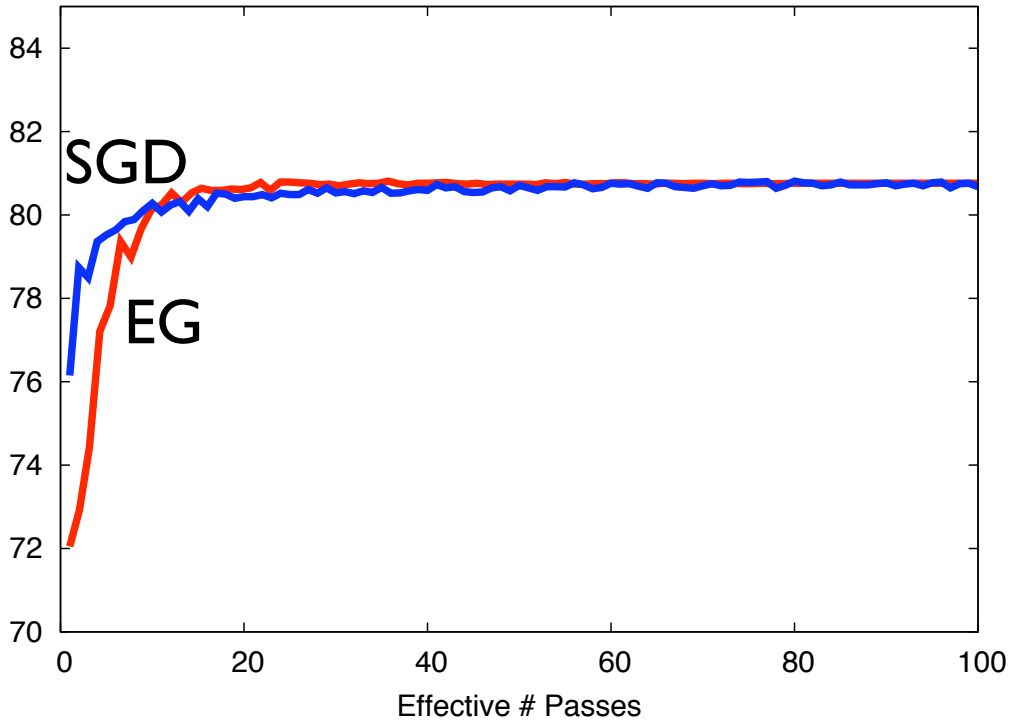
- Obtain primal vector via $\frac{1}{C}\mathbf{w}(\boldsymbol{\alpha})$

EG and SGD

Objective Value



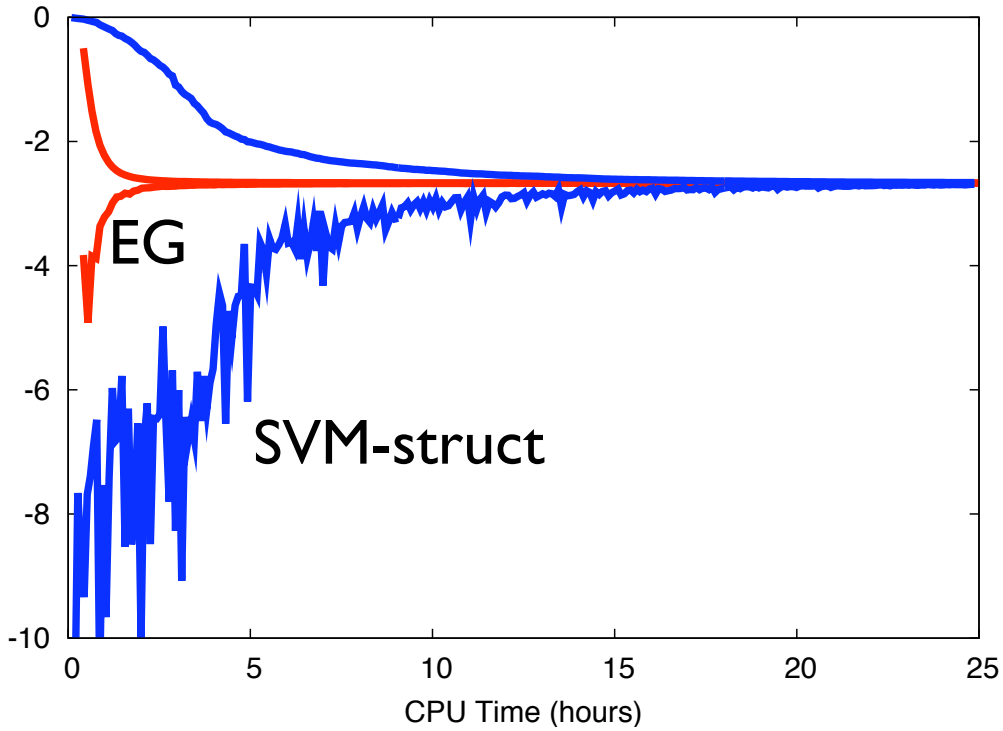
Parsing Accuracy



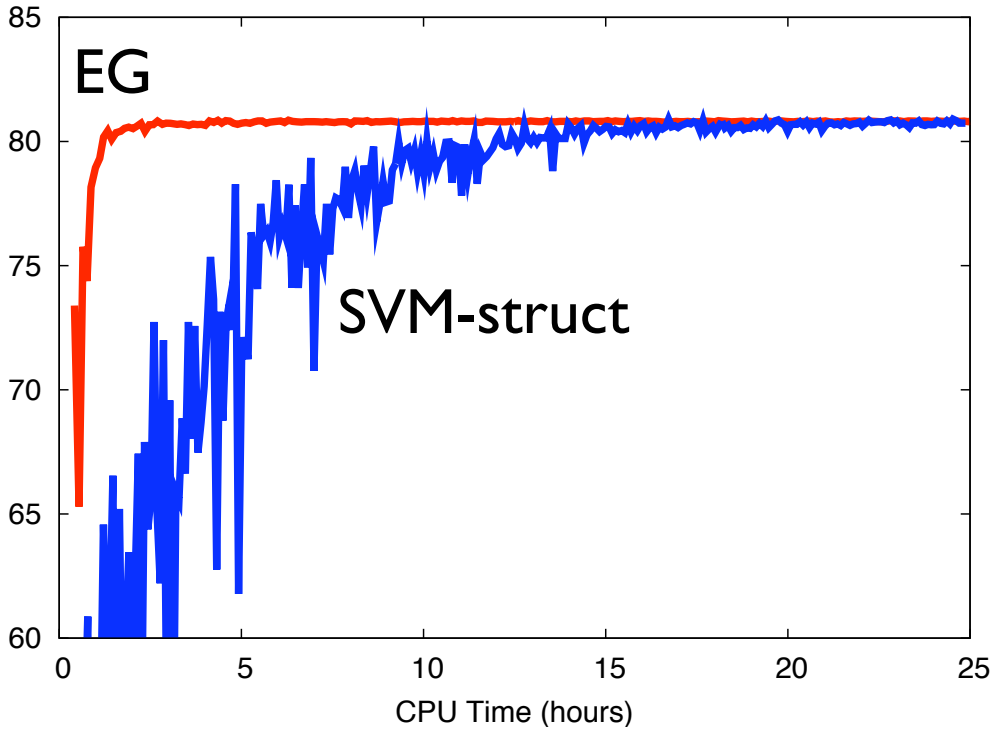
- Explored 5 learning rates for SGD

EG and SVM-struct

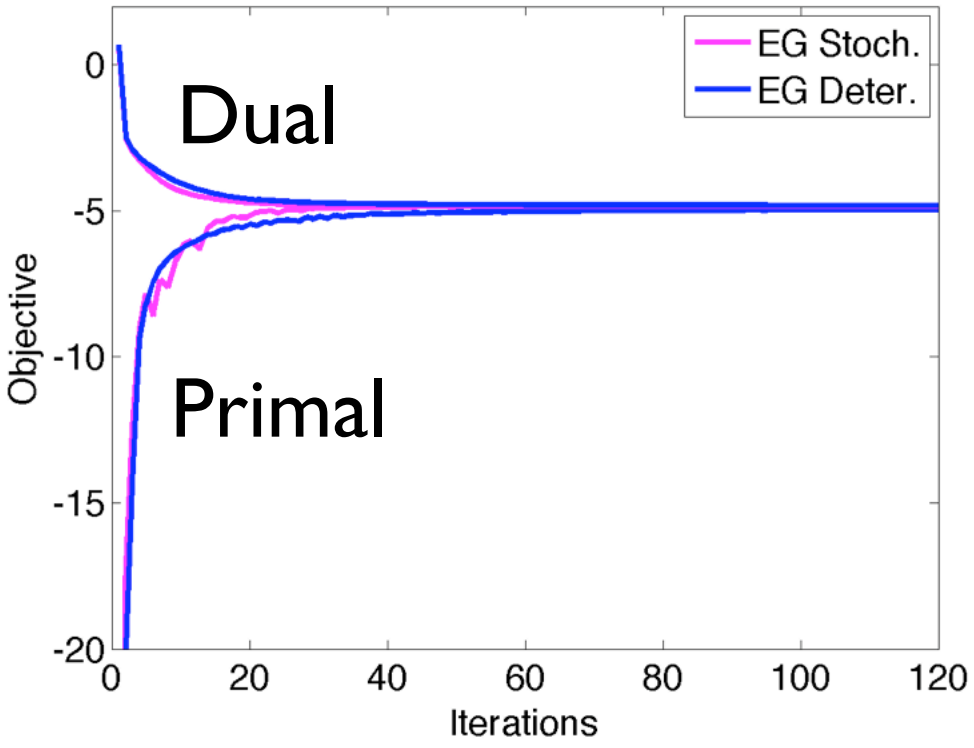
Objective Value

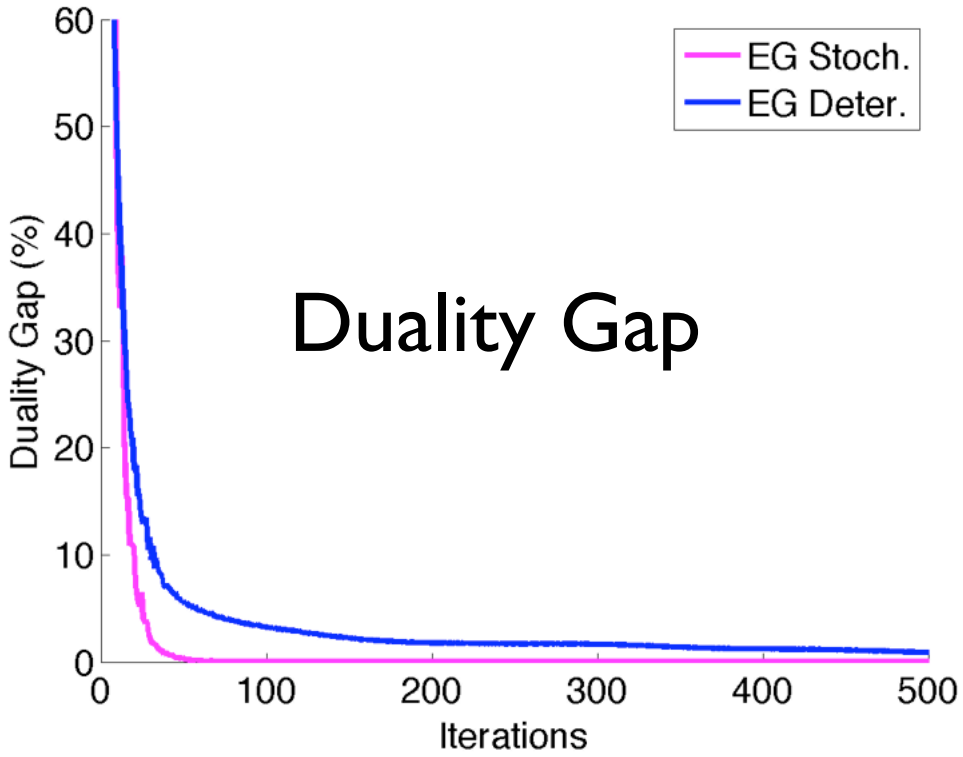


Parsing Accuracy



Deterministic vs. Stochastic Online EG





CSAIL

Conclusions

- EG algorithm for minimizing convex function subject to simplex constraints
- Applicable to log-linear and max-margin models
- Analysis of convergence
- Theory predicts that online is n times faster
- Randomization is good!
- Performs well in practice
- Are log-linear models "easier"?