Majorization for CRFs and Latent Likelihoods

Tony Jebara and Anna Choromanska
Columbia University

February 18, 2013

Machine Learning

- Unsupervised
  - Given $\mathcal{D} = (y_t)_{t=1}^{T}$ sampled *iid* from unknown $P(y)$
  - Given family of functions $p_{\boldsymbol{\theta}}(y)$ parametrized by $\boldsymbol{\theta}$
  - Find $\boldsymbol{\theta}$ by minimizing some cost (e.g. partition function)
  - Output $p_{\boldsymbol{\theta}}(y)$
- Supervised
  - Given $\mathcal{D} = (x_t, y_t)_{t=1}^{T}$ sampled *iid* from unknown $P(x, y)$
  - Given family of functions $p_{\boldsymbol{\theta}}(y|x)$ parametrized by $\boldsymbol{\theta}$
  - Find $\boldsymbol{\theta}$ by minimizing some cost (e.g. partition function)
  - Predict using $\hat{y} = \arg\max_y p_{\boldsymbol{\theta}}(y|x)$
- Today: a new simple bound for such optimizations

Optimization in Learning: Three Schools of Thought

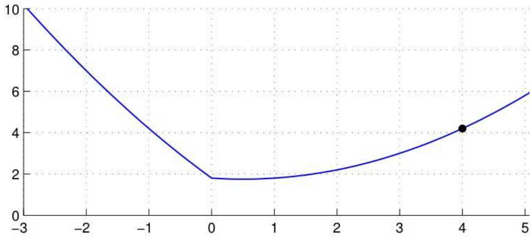
Ways to optimize parameters to data

- First order methods
  - Steepest descent
  - Conjugate gradient
  - Stochastic gradient descent
  - Bundle methods
- Second order methods
  - Newton
  - BFGS [Broyden; Fletcher; Goldfarb; Shanno '70]
  - Limited memory BFGS [Liu & Nocedal '89]
- Majorization and bounding methods
  - Expectation-Maximization [Baum 1970, Dempster '77]
  - Generalized iterative scaling [Darroch & Ratcliff '72]
  - Majorization or majorize/minimize [deLeeuw & Heiser '77]
  - Quadratic lower bound principle [Bohning & Lindsay '88]
  - Improved iterative scaling [Berger et al. '97]
  - Extended Baum-Welch [Kanevsky et al. '08]

Majorization

If cost function $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$ has no closed form solution
Majorization uses with a surrogate Q with closed form update
to monotonically minimize the cost from an initial $\boldsymbol{\theta}_0$

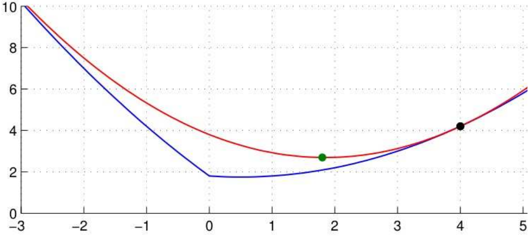
- Find bound $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_i) \geq C(\boldsymbol{\theta})$ where $Q(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) = C(\boldsymbol{\theta}_i)$
- Update $\boldsymbol{\theta}_{i+1} = \arg\min_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_i)$
- Repeat until converged



Majorization

If cost function $\theta^* = \arg\min_\theta C(\theta)$ has no closed form solution Majorization uses with a surrogate Q with closed form update to monotonically minimize the cost from an initial θ_0

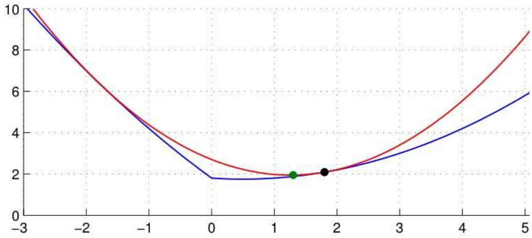
- Find bound $Q(\theta, \theta_i) \geq C(\theta)$ where $Q(\theta_i, \theta_i) = C(\theta_i)$
- Update $\theta_{i+1} = \arg\min_\theta Q(\theta, \theta_i)$
- Repeat until converged



Majorization

If cost function $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$ has no closed form solution Majorization uses with a surrogate Q with closed form update to monotonically minimize the cost from an initial $\boldsymbol{\theta}_0$

- Find bound $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_i) \geq C(\boldsymbol{\theta})$ where $Q(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) = C(\boldsymbol{\theta}_i)$
- Update $\boldsymbol{\theta}_{i+1} = \arg\min_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_i)$
- Repeat until converged



Partition Function

- Central quantity to optimize in
  - Maximum likelihood and e-family [Pitman & Wishart '36]
  - Maximum entropy [Jaynes '57]
  - Conditional random fields [Lafferty, et al. '01]
  - Log-linear models [Darroch & Ratcliff '72]
  - Graphical models, HMMs [Jordan, et al. '99]
- Majorization preferred until [Wallach '03, Andrew & Gao '07]

Method	Iterations	LL Evaluations	Time (s)
IIS	≥ 150	≥ 150	≥ 188.65
Conjugate gradient (FR)	19	99	124.67
Conjugate gradient (PRP)	27	140	176.55
L-BFGS	22	22	29.72

The problem: loose & complicated bounds. Let's fix this!

Partition Function

Consider log-linear model over discrete $y \in \Omega$ where $|\Omega| = n$

$$p(y|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} h(y) \exp\left(\boldsymbol{\theta}^\top \mathbf{f}(y)\right)$$

- Parameters are vector $\boldsymbol{\theta} \in \mathbb{R}^d$
- Features are $\mathbf{f} : \Omega \mapsto \mathbb{R}^d$ mapping each y to some vector
- Prior is $h : \Omega \mapsto \mathbb{R}^+$ a fixed non-negative measure
- Partition function ensures that $p(y|\boldsymbol{\theta})$ normalizes

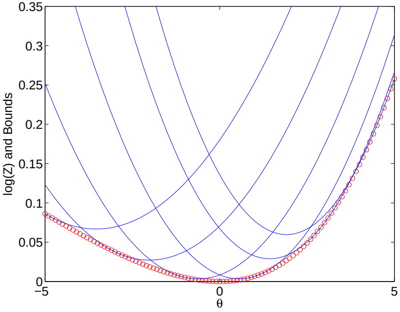
$$Z(\boldsymbol{\theta}) = \sum_y h(y) \exp(\boldsymbol{\theta}^\top \mathbf{f}(y))$$

Problem: it's ugly to minimize, we much prefer quadratics

Partition Function Bound

The bound $\ln Z(\boldsymbol{\theta}) \leq \ln z + \frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top \boldsymbol{\mu}$ is tight at $\tilde{\boldsymbol{\theta}}$ and holds for parameters given by

Input $\tilde{\boldsymbol{\theta}}, \mathbf{f}(y), h(y)\ \forall y \in \Omega$
Init $z \rightarrow 0^+, \boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = z\mathbf{I}$ For each $y \in \Omega$ { $\alpha = h(y)\exp(\tilde{\boldsymbol{\theta}}^\top \mathbf{f}(y))$ $\mathbf{l} = \mathbf{f}(y) - \boldsymbol{\mu}$ $\boldsymbol{\Sigma} += \frac{\tanh(\frac{1}{2}\ln(\alpha/z))}{2\ln(\alpha/z)} \mathbf{l}\mathbf{l}^\top$ $\boldsymbol{\mu} += \frac{\alpha}{z+\alpha}\mathbf{l}$ $z += \alpha$ }
Output $z, \boldsymbol{\mu}, \boldsymbol{\Sigma}$



Bound Proof (Sketch)

Proof.

1) Start with bound $\log(e^{\theta} + e^{-\theta}) \leq c\theta^2$ [Jaakkola & Jordan '99]
2) Prove scalar bound via Fenchel dual using $\theta = \sqrt{\vartheta}$
3) Make bound multivariate $\log(e^{\boldsymbol{\theta}^\top \mathbf{1}} + e^{-\boldsymbol{\theta}^\top \mathbf{1}})$
4) Handle scaling of exponentials $\log(h_1 e^{\boldsymbol{\theta}^\top \mathbf{f}_1} + h_2 e^{-\boldsymbol{\theta}^\top \mathbf{f}_2})$
5) Add one term $\log(h_1 e^{\boldsymbol{\theta}^\top \mathbf{f}_1} + h_2 e^{-\boldsymbol{\theta}^\top \mathbf{f}_2} + h_3 e^{-\boldsymbol{\theta}^\top \mathbf{f}_3})$
6) Repeat extension for n terms $\square$

The Bound as a Variation of Newton

$$
\boxed{\begin{array}{l}
\text{Input } \tilde{\boldsymbol{\theta}}, \mathbf{f}(y), h(y)\ \forall y \in \Omega \\
\hline
\text{Init } z \to 0^+, \boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = z\mathbf{I} \\
\text{For each } y \in \Omega\ \{ \\
\quad \alpha \quad = h(y)\exp(\tilde{\boldsymbol{\theta}}^\top \mathbf{f}(y)) \\
\quad \mathbf{l} \quad = \mathbf{f}(y) - \boldsymbol{\mu} \\
\quad \boldsymbol{\Sigma} += \frac{\tanh(\frac{1}{2}\ln(\alpha/z))}{2\ln(\alpha/z)} \mathbf{l}\mathbf{l}^\top \\
\quad \boldsymbol{\mu} += \frac{\alpha}{z+\alpha}\mathbf{l} \\
\quad z \ += \alpha \qquad \} \\
\hline
\text{Output } z, \boldsymbol{\mu}, \boldsymbol{\Sigma}
\end{array}}
$$

$$\boldsymbol{\Sigma} += \frac{z\alpha}{(z+\alpha)^2}\mathbf{l}\mathbf{l}^\top - \frac{\alpha}{z+\alpha}\boldsymbol{\Sigma}$$

Computing the bound (left) and Newton's approximation (right)
Both take $\mathcal{O}(nd^2)$ and update via $\boldsymbol{\theta} \leftarrow \tilde{\boldsymbol{\theta}} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ in $\mathcal{O}(d^3)$

Maximum Entropy

Maximum entropy (or generally) minimum relative entropy $\mathcal{RE}(p\|h) = \sum_y p(y) \ln \frac{p(y)}{h(y)}$ subject to linear constraints

$$\min_p \mathcal{RE}(p\|h) \text{ s.t. } \sum_y p(y)\mathbf{f}(y) = \mathbf{0}, \sum_y p(y)\mathbf{g}(y) \geq \mathbf{0}$$

Its dual is the negative log-partition function (to be maximized):

$$-\ln Z(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = -\ln \sum_y h(y) \exp\left(\boldsymbol{\theta}^\top \mathbf{f}(y) + \boldsymbol{\vartheta}^\top \mathbf{g}(y)\right)$$

Maximum entropy is a natural application of the bound!

Conditional Random Fields (CRFs)

- Conditional random fields generalize maximum entropy
- Trained on *iid* data $\{(x_1, y_1), \ldots, (x_t, y_t)\}$
- Each CRF is a log-linear model

$$p(y|x_j, \boldsymbol{\theta}) \quad = \quad \frac{1}{Z_{x_j}(\boldsymbol{\theta})} h_{x_j}(y) \exp(\boldsymbol{\theta}^\top \mathbf{f}_{x_j}(y))$$

- Regularized maximum likelihood objective function is

$$J(\boldsymbol{\theta}) = \sum_{j=1}^{t} \ln \frac{h_{x_j}(y_j)}{Z_{x_j}(\boldsymbol{\theta})} + \boldsymbol{\theta}^\top \mathbf{f}_{x_j}(y_j) - \tfrac{t\lambda}{2} \|\boldsymbol{\theta}\|^2 \qquad (1)$$

- Can even constrain the allowable $\boldsymbol{\theta}$ inside convex $\boldsymbol{\Lambda}$
- Permits ℓ_1 regularized CRFs, and other variants

Maximum Likelihood Algorithm for CRFs

Input x_j, y_j and functions $h_{x_j}, \mathbf{f}_{x_j}$ for $j = 1, \ldots, t$ Input regularizer $\lambda \in \mathbb{R}^+$ and convex hull $\boldsymbol{\Lambda} \subseteq \mathbb{R}^d$
Initialize $\boldsymbol{\theta}_0$ anywhere inside $\boldsymbol{\Lambda}$ and set $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_0$ While not converged For $j = 1$ to t compute bound for $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$ from $h_{x_j}, \mathbf{f}_{x_j}, \tilde{\boldsymbol{\theta}}$ Set $\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Lambda}} \sum_j \frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^\top (\boldsymbol{\Sigma}_j + \lambda \mathbf{I})(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})$ $+ \sum_j \boldsymbol{\theta}^\top (\boldsymbol{\mu}_j - \mathbf{f}_{x_j}(y_j) + \lambda \tilde{\boldsymbol{\theta}})$
Output $\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}$

Theorem

If $\|\mathbf{f}_{x_j}(y)\| \leq r$ *get* $J(\hat{\boldsymbol{\theta}}) - J(\boldsymbol{\theta}_0) \geq (1 - \epsilon) \max_{\boldsymbol{\theta} \in \boldsymbol{\Lambda}} (J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_0))$ *within* $\left\lceil \ln(1/\epsilon) / \ln\left(1 + \frac{\lambda \log n}{2r^2 n}\right) \right\rceil$ *steps*

Convergence Proof

Proof.

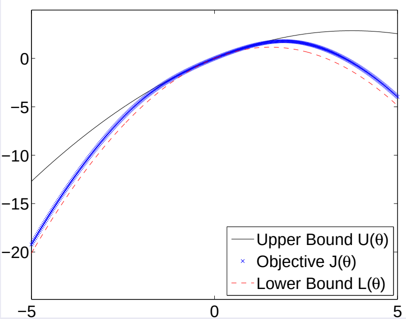
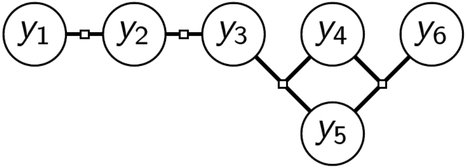


Figure: Quadratic bounding sandwich. Compare upper and lower bound curvatures to bound maximum # of iterations.

□

Bounding Graphical Models with Large n

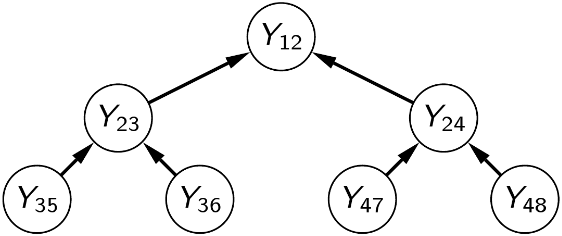


- Each iteration is $\mathcal{O}(tnd^2)$, but what if n is large?
- Graphical model: a bipartite factor graph G representing a distribution $p(Y)$ where $Y = \{y_1, \ldots, y_n\}$ and $y_i \in \mathbb{Z}$
- $p(Y)$ factorizes as product of $\{\psi_1, \ldots, \psi_C\}$ functions (squares) over $\{Y_1, \ldots, Y_C\}$ subsets of variables (circles)

$$p(y_1, \ldots, y_n) \quad = \quad \frac{1}{Z} \prod_{c \in C} \psi_c(Y_c)$$

- E.g. $p(y_1, \ldots, y_6) = \psi(y_1, y_2)\psi(y_2, y_3)\psi(y_3, y_4, y_5)\psi(y_4, y_5, y_6)$

Bounding Graphical Models with Large n



- Instead of enumerating over all n, exploit graphical model
- Build junction tree and run a *Collect* algorithm
- Already used for computing $Z(\boldsymbol{\theta})$ and $Z'(\boldsymbol{\theta})$ efficiently
- Bound needs $\mathcal{O}(td^2 \sum_c |Y_c|)$ rather than $\mathcal{O}(td^2 n)$
- For an HMM, this is $\mathcal{O}(TM^2)$ instead of $\mathcal{O}(M^T)$

Bounding Graphical Models with Large n

for $c = 1, \ldots, m$ {
$Y_{both} = Y_c \cap Y_{pa(c)}$; $Y_{solo} = Y_c \setminus Y_{pa(c)}$
for each $u \in Y_{both}$ {
initialize $z_{c|x} \leftarrow 0^+$, $\boldsymbol{\mu}_{c|x} = \mathbf{0}$, $\boldsymbol{\Sigma}_{c|x} = z_{c|x}\mathbf{I}$
for each $v \in Y_{solo}${

$$w = u \otimes v; \quad \alpha_w = h_c(w) e^{\tilde{\boldsymbol{\theta}}^\top \mathbf{f}_c(w)} \prod_{b \in ch(c)} z_{b|w}$$

$$\mathbf{l}_w = \mathbf{f}_c(w) - \boldsymbol{\mu}_{c|u} + \sum_{b \in ch(c)} \boldsymbol{\mu}_{b|w}$$

$$\boldsymbol{\Sigma}_{c|u} += \sum_{b \in ch(c)} \boldsymbol{\Sigma}_{b|w} + \frac{\tanh(\frac{1}{2}\ln(\frac{\alpha_w}{z_{c|u}}))}{2\ln(\frac{\alpha_w}{z_{c|u}})} \mathbf{l}_w \mathbf{l}_w^\top$$

$$\boldsymbol{\mu}_{c|u} += \frac{\alpha_w}{z_{c|u} + \alpha_w} \mathbf{l}_w; \quad z_{c|u} += \alpha_w \ \}\}\}$$

Low Rank Bound for Large d

- Naive bounding takes $\mathcal{O}(tnd^2)$, inverting takes $\mathcal{O}(d^3)$
- To match gradient methods and LBFGS, need $\mathcal{O}(tnd)$
- Consider a rank 1 update: $\boldsymbol{\Sigma} += \frac{\tanh(\frac{1}{2}\ln(\alpha/z))}{2\ln(\alpha/z)}\mathbf{l}\mathbf{l}^\top$
- As in LBFGS, use rank-k storage $\boldsymbol{\Sigma} = \mathbf{V}\mathbf{S}\mathbf{V}^\top + \mathbf{D}$
- Each rank 1 update on $\boldsymbol{\Sigma}$ is projected on $\mathbf{V}$
- Top k eigenvectors are kept with updated eigenvalues in $\mathbf{S}$
- Remaining residual is absorbed into diagonal $\mathbf{D}$
- By Jensen inequality on diagonal $\mathbf{D}$, low-rank is *still a bound*
- Avoid $\mathcal{O}(d^3)$ inversion in $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$: use Woodbury formula, $\boldsymbol{\Sigma}^{-1} = \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{V}^\top(\mathbf{S}^{-1} + \mathbf{V}\mathbf{D}^{-1}\mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{D}^{-1}$ with only $\mathcal{O}(k^3)$ work

Low Rank Bound for Large d in $\mathcal{O}(tndk)$

for each t { set $z \to 0^+$; $\boldsymbol{\mu} = \mathbf{0}$;
for each y {

$$\alpha = h_t(y) e^{\tilde{\boldsymbol{\theta}}^\top \mathbf{f}_t(y)}; \ \mathbf{r} = \frac{\sqrt{\tanh(\frac{1}{2}\log(\frac{\alpha}{z}))}}{\sqrt{2\log(\frac{\alpha}{z})}} (\mathbf{f}_t(y) - \boldsymbol{\mu});$$

$$\text{For } i = 1, \ldots, k : \mathbf{p}(i) = \mathbf{r}^\top \mathbf{V}(i, \cdot); \mathbf{r} = \mathbf{r} - \mathbf{p}(i)\mathbf{V}(i, \cdot);$$

$$\text{For } i = 1, \ldots, k : \text{For } j = 1, \ldots, k : \mathbf{S}(i, j) = \mathbf{S}(i, j) + \mathbf{p}(i)\mathbf{p}(j);$$

$$\mathbf{Q}^\top \mathbf{A} \mathbf{Q} = \mathrm{svd}(\mathbf{S}); \mathbf{S} \leftarrow \mathbf{A}; \mathbf{V} \leftarrow \mathbf{Q}\mathbf{V};$$

$$\mathbf{s} = [\mathbf{S}(1,1), \ldots, \mathbf{S}(k,k), \|\mathbf{r}\|^2]^\top; \ \tilde{k} = \arg\min_{i=1,\ldots,k+1} \mathbf{s}(i);$$

$$\text{if } (\tilde{k} \leq k) \{ \ \mathbf{D} = \mathbf{D} + \mathbf{S}(\tilde{k}, \tilde{k}) \mathbf{1}^\top |\mathbf{V}(j, \cdot)| \ \mathrm{diag}(|\mathbf{V}(k, \cdot)|);$$

$$\mathbf{S}(\tilde{k}, \tilde{k}) = \|\mathbf{r}\|^2; \ \mathbf{r} = \|\mathbf{r}\|^{-1}\mathbf{r}; \ \mathbf{V}(k, \cdot) = \mathbf{r}; \ \}$$

$$\text{else} \quad \{ \ \mathbf{D} = \mathbf{D} + \mathbf{1}^\top |\mathbf{r}| \mathrm{diag}(|\mathbf{r}|); \ \} \ \}$$

$$\boldsymbol{\mu} += \frac{\alpha}{z+\alpha}(\mathbf{f}_t(y) - \boldsymbol{\mu}); \ z += \alpha;$$

} }

Mixture Models and Latent Likelihood

- Bounding also simplifies mixture models with hidden variables
- Allows mixtures of Gaussians, HMMs, latent graphical models
- Assume data-set is generated by a conditional distribution

$$p(y|x,\Theta) = \frac{\sum_m p(x,y,m|\Theta)}{\sum_{y,m} p(x,y,m|\Theta)}$$

- It is natural to maximize incomplete likelihood

$$L(\Theta) = \prod_{j=1}^{t} p(y_j|x_j,\Theta) = \prod_{j=1}^{t} \frac{\sum_m p(x_j,y_j,m|\Theta)}{\sum_{y,m} p(x_j,y,m|\Theta)}$$

- Assume exponential family mixture components (Gaussian, multinomial, Poisson, Laplace)

$$p(x|y,m,\Theta) = h(x)\exp\left(\boldsymbol{\theta}_{y,m}^{\top}\phi_{y,m}(x) - a_{y,m}(\boldsymbol{\theta}_{y,m})\right)$$

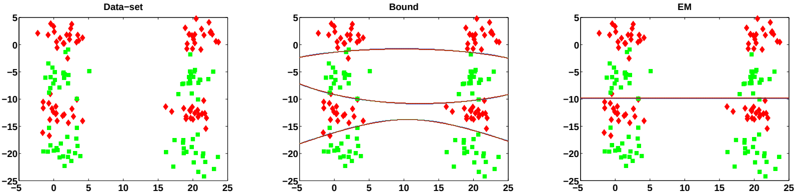
Mixture Models and Latent Likelihood

- Latent CRFs are just log-linear mixtures [Quattoni '07]
- When a CRF has hidden variable m, the latent likelihood is

$$L(\boldsymbol{\theta}) = \prod_{j=1}^{t} \frac{\sum_m \exp\left(\boldsymbol{\theta}^\top \mathbf{f}_{j,y_j,m}\right)}{\sum_{y,m} \exp\left(\boldsymbol{\theta}^\top \mathbf{f}_{j,y,m}\right)}.$$

- Vectors $\mathbf{f}$ are concatenations of $\phi_{y,m}(x)$ sufficient statistics
- Apply Jensen to numerator and our bound to denominator
- Get an auxiliary function $L(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})$
- Maximizing $Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})$ is just a matrix inverse
- Like CEM or conditional variant of EM [J & Pentland '00]
- If $|m| = 1$, reduces back to usual conditional random field

Mixture Models and Latent Likelihood



(a) Data-set (b) Discriminative Bound (c) Generative EM

Figure: Mixture model conditional estimation.

Two Gaussian mixture model trained for maximum conditional likelihood (left) and maximum likelihood (right)

Experiments - Classification and Structured Prediction

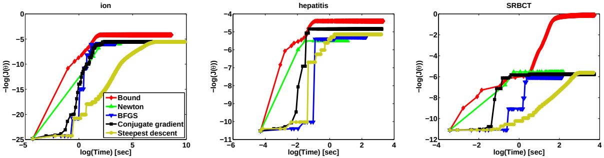
Data-set	SRBCT		Tumors		Text		SecStr		CoNLL		PennTree	
Size	$n=4$ $t=83$ $d=9236$ $\lambda=10^1$		$n=26$ $t=308$ $d=390260$ $\lambda=10^1$		$n=2$ $t=1500$ $d=23922$ $\lambda=10^2$		$n=2$ $t=83679$ $d=632$ $\lambda=10^1$		$m=9$ $t=1000$ $d=33615$ $\lambda=10^1$		$m=45$ $t=1000$ $d=14175$ $\lambda=10^1$	
Algorithm	time	iter	time	iter	time	iter	time	iter	time	iter	time	iter
LBFGS	6.10	42	3246.83	8	15.54	7	881.31	47	25661.54	17	62848.08	7
SD	7.27	43	18749.15	53	153.10	69	1490.51	79	93821.72	12	156319.31	12
CG	40.61	100	14840.66	42	57.30	23	667.67	36	88973.93	23	76332.39	18
Bound	**3.67**	**8**	**1639.93**	**4**	**6.18**	**3**	**27.97**	**9**	**16445.93**	**4**	**27073.42**	**2**

Table: Time in seconds and iterations to match LBFGS solution for logistic regression (on SRBCT, Tumors, Text and SecStr data-sets where n is the number of classes) and Markov CRFs (on CoNLL and PennTree data-sets, where m is the number of classes). Here, t is the total number of samples (training and testing), d is the dimensionality of the feature vector and λ is the cross-validated regularization setting.

Experiments - Testing Latent Likelihood

Data-set	ion	bupa	hepatitis	wine	SRBCT
Algorithm	m = 3	m = 2	m = 2	m = 3	m = 4
BFGS	-5.88	-21.78	-5.28	-1.79	-6.06
SD	-5.56	-21.74	-5.14	-1.37	-5.61
CG	-5.57	-21.81	-4.84	-0.95	-5.76
Newton	-5.95	-21.85	-5.50	-0.71	-5.54
Bound	**-4.18**	**-19.95**	**-4.40**	**-0.48**	**-0.11**

Table: Test log-likelihood at convergence for ion, bupa, hepatitis, wine and SRBCT data-sets.



(a) Ion (b) Hepatitis (c) SRBCT

Figure: Convergence of latent likelihood over time for several data-sets.

Conclusions

- Majorization was non-competitive due to slow & loose bounds
- We derived *simple* quadratic bound on the partition function
- Makes majorization competitive with state-of-the-art
- Bound is efficient for graphical models and large n
- Low-rank bound is efficient for large dimensionality d
- Yields fast and monotonically convergent majorization
- Used for maximum entropy, CRFs and latent likelihood
- Current work: HMMs, stochastic bounds, loopy graphs, deep belief networks, distributed optimization