Large Margin Methods for Structured Classification: Exponentiated Gradient Algorithms

Peter Bartlett
Department of Statistics and Division of Computer Science
UC Berkeley

Joint work with: Mike Collins, David McAllester, and Ben Taskar

slides at http://www.stat.berkeley.edu/∼bartlett, under Talks

Optical Character Recognition

$X =$ grey-scale image of a sequence of characters

$Y =$ sequence of characters

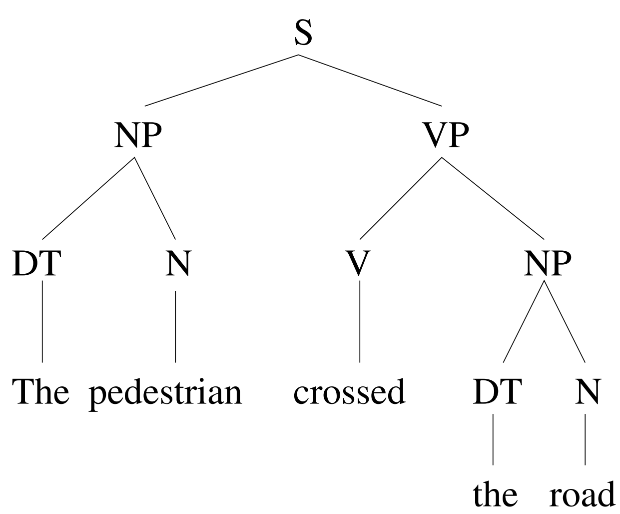
This is an example of

Parsing

$X =$ sentence

$Y =$ parse tree

The pedestrian crossed the road.



Pattern Classification

- Data: i.i.d. $(X, Y), (X_1, Y_1), \ldots, (X_n, Y_n)$ from $\mathcal{X} \times \mathcal{Y}$.
- Loss function: $\ell : \mathcal{Y}^2 \to \mathbb{R}^+$, $\ell(\hat{y}, y) =$ cost of mistake.
- Use data $(X_1, Y_1), \ldots, (X_n, Y_n)$ to choose $f : \mathcal{X} \to \mathcal{Y}$ with small risk,

$$R(f) = \mathbf{E}\ell(f(X), Y).$$

Often choose f from a fixed class $\mathcal{F}$.

Structured Classification Problems

Key issue: $|\mathcal{Y}|$ is very large.

- OCR: exponential in number of characters
- parsing: exponential in sentence length

Large Margin Methods for Structured Classification

- **Generative models**
  - Markov random fields
  - Probabilistic context-free grammars
- Large margin classifiers
- Quadratic program for large margin classifiers
- Exponentiated gradient algorithm
- Convergence analysis

Generative Modelling

Split Y into parts/assume sparse dependencies.

- OCR: graphical model (e.g., HMM)
- parsing: probabilistic context-free grammar.

Structured Classification Problems: OCR

$\Pr(Y|X)$ factorizes into small parts:

$$\begin{aligned}
&\Pr(Y = \text{`}\texttt{This is an example of}\text{'}, X = \text{image}) \\
&= \Pr(Y_1 = \text{`}\texttt{T}\text{'}) \Pr(X_1 = \text{image}_1 | Y_1 = \text{`}\texttt{T}\text{'}) \\
&\qquad \times \Pr(Y_2 = \text{`}\texttt{h}\text{'} | Y_1 = \text{`}\texttt{T}\text{'}) \Pr(X_2 = \text{image}_2 | Y_2 = \text{`}\texttt{h}\text{'}) \cdots \\
&= \prod_{C \in \mathcal{C}} \psi_C(x_C, y_C).
\end{aligned}$$

Graphical model: $\mathcal{C}$ is a set of cliques in a graph; vertices are random variables, lack of an edge denotes conditional independence.

Structured Classification Problems: Parsing

Probabilistic context-free grammar:

S	$\rightarrow$	NP VP	1.0
VP	$\rightarrow$	V NP	0.4
VP	$\rightarrow$	V	0.3
VP	$\rightarrow$	VP PP	0.3
NP	$\rightarrow$	DT N	0.8
	$\vdots$		$\vdots$

DT	$\rightarrow$	the	0.6
DT	$\rightarrow$	a	0.4
N	$\rightarrow$	pedestrian	0.1
N	$\rightarrow$	road	0.2
V	$\rightarrow$	crossed	0.1
	$\vdots$		$\vdots$

$$\Pr(Y = \text{tree} | X = \text{sentence}) = \prod_i \Pr(\alpha_i \rightarrow \beta_i | \alpha_i).$$

Structured Classification Problems

Plug-in estimates:

1. Simple model $\hat{p}(x, y; \theta)$ of $\Pr(Y = y | X = x)$.
2. Use data to estimate parameters θ. (e.g., ML)
3. Compute $\arg\max_{y \in \mathcal{Y}} \hat{p}(x, y; \theta)$.
   (e.g., dynamic programming: Viterbi, inside-outside)

If each factor is a log-linear model, we compute a linear discriminant:

$$
\begin{aligned}
\hat{y} &= \arg\max_{y \in \mathcal{Y}} \log(\hat{p}(x, y; \theta)) \\
&= \arg\max_{y \in \mathcal{Y}} \sum_i \phi_i(x, y)\theta_i.
\end{aligned}
$$

Structured Classification Problems: Sparse Representations

Suppose y naturally decomposes into parts:

$R(x, y)$ denotes the set of "parts" belonging to $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$\Phi(x, y) = \sum_{r \in R(x,y)} \phi(x, r)$$

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \Phi(x, y)'\theta = \arg\max_{y \in \mathcal{Y}} \sum_{r \in R(x,y)} \phi(x, r)'\theta,$$

- e.g. Markov random fields. Parts are configurations for cliques.
- e.g. PCFGs. Parts are rule-location pairs (rules of grammar applied at specific locations in the sentence).

Large Margin Methods for Structured Classification

- Generative models
  - Markov random fields
  - Probabilistic context-free grammars
- **Large margin classifiers**
- Quadratic program for large margin classifiers
- Exponentiated gradient algorithm
- Convergence analysis

Empirical Risk Minimization

- Choose f as maximum of linear functions,

$$f(x) = \arg\max_{y \in \mathcal{Y}} \Phi(x, y)'\theta,$$

to minimize empirical risk,

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i).$$

- Can be preferable to plug-in estimate.

  (e.g., if $\Pr(Y = y|X)$ not too close to uniform)

- Computationally intractable (unless some f has $\hat{R}(f) = 0$).

Large Margin Classifiers

- Convex relaxation:

  Choose f to minimize sample average of a convex criterion.

- e.g., Support Vector Machines:

  $\mathcal{Y} = \{\pm 1\}$, $\ell(\hat{y}, y) = 1[\hat{y} \neq y]$, $\Phi(x, y) = y\Psi(x)$:

  Choose θ to minimize

$$\lambda \|\theta\|^2 + \frac{1}{n} \sum_{i=1}^{n} \left(1 - Y_i \Psi(X_i)'\theta\right)_+ ,$$

  where $(x)_+ = \max\{x, 0\}$.)

  This is a quadratic program (QP).

Large Margin Classifiers

- For $\mathcal{Y} = \{\pm 1\}$, $\ell(\hat{y}, y) = 1[\hat{y} \neq y]$, and $\Phi(x, y) = y\Psi(x)$,

$$
\begin{aligned}
(1 - 2Y_i\Psi(X_i)'\theta)_+ &= \max_{\hat{y}} \left(\ell(\hat{y}, Y_i) - (Y_i - \hat{y})\Psi(X_i)'\theta\right)_+ \\
&= \max_{\hat{y}} \left(\ell(\hat{y}, Y_i) - (\Phi(X_i, Y_i)'\theta - \Phi(X_i, \hat{y})'\theta)\right)_+ .
\end{aligned}
$$

- Think of $\Phi(x, y)'\theta - \Phi(x, \hat{y})'\theta$ as an upper bound on the loss $l(\hat{y}, y)$ that we'll incur when we choose the $\hat{y}$ that maximizes $\Phi(x, \hat{y})'\theta$.

Large Margin Multiclass Classification

Choose θ to minimize

$$\lambda\|\theta\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max_{\hat{y}} \left(\ell(\hat{y}, Y_i) - (\Phi(X_i, Y_i)'\theta - \Phi(X_i, \hat{y})'\theta)\right)_+$$

$$= \lambda\|\theta\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max_{\hat{y}} \left(\ell(\hat{y}, Y_i) - \Phi'_{i,\hat{y}}\theta\right)_+ ,$$

where $(x)_+ = \max\{x, 0\}$ and $\Phi_{i,\hat{y}} = \Phi(X_i, Y_i) - \Phi(X_i, \hat{y})$.

- Suggested by Taskar et al, 2004.
- Quadratic program.

Large Margin Multiclass Classification

Primal problem:

$$\min_{\theta,\epsilon} \left(\tfrac{1}{2}\lambda\|\theta\|^2 + \tfrac{1}{n}\textstyle\sum_i \epsilon_i\right)$$

Subject to the constraints:

$$\forall i, y \in \mathcal{Y}(X_i),$$

$$\theta' \Phi_{i,y} \geq \ell(y, Y_i) - \epsilon_i$$

$$\forall i,\ \epsilon_i \geq 0$$

Dual problem:

$$\max_{\alpha} \left(C \sum_{i,y} \alpha_{i,y} \ell(y, Y_i) - \frac{C^2}{2} \sum_{i,y,j,z} \alpha_{i,y}\alpha_{j,z}\Phi'_{i,y}\Phi_{j,z} \right)$$

Subject to the constraints:

$$\forall i,\ \sum_y \alpha_{i,y} = 1$$

$$\forall i, y,\ \alpha_{i,y} \geq 0$$

Large Margin Multiclass Classification

Some observations:

- Quadratic program over $\alpha = (\alpha_{i,y})$, restricted to (n copies of) the probability simplex:

$$\begin{aligned} \max_{\alpha} \quad & Q(\alpha) \\ \text{s.t.} \quad & \alpha_i \in \Delta. \end{aligned}$$

- Number of variables is sum over data of number of possible labels. Very large: $n|\mathcal{Y}|$.

Large Margin Methods for Structured Classification

- Generative models
  - Markov random fields
  - Probabilistic context-free grammars
- Large margin classifiers
- Quadratic program for large margin classifiers
- **Exponentiated gradient algorithm**
- Convergence analysis

Exponentiated Gradient Algorithm

Exponentiated gradients:

$$\alpha^{(t+1)} = \arg\min_{\alpha} \left(D\left(\alpha, \alpha^{(t)}\right) + \eta \alpha' \nabla Q \left(\alpha^{(t)}\right)\right).$$

- D is Kullback-Liebler divergence.
- ∇Q term moves α in direction of decreasing Q.
- KL term constrains it to be close to $\alpha^{(t)}$.

Solution is

$$\alpha_{i,y}^{(t)} = \frac{\exp(\theta_{i,y}^{(t)})}{\sum_z \exp(\theta_{i,z}^{(t)})},$$

with $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla Q(\alpha^{(t)})$.

Exponentiated Gradient Algorithm: Convergence

Theorem: For all $u \in \Delta$,

$$\frac{1}{T}\sum_{t=1}^{T} Q(\alpha^{(t)}) \leq Q(u) + \frac{D(u, \alpha^{(1)})}{\eta T} + c_{\eta,Q}\frac{Q(\alpha^{(1)})}{T}.$$

Exponentiated Gradient Algorithm with Parts

Suppose y naturally decomposes into parts:

$R(x, y)$ denotes the set of "parts" belonging to $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$\Phi(x, y) = \sum_{r \in R(x,y)} \phi(x, r)$$

$$\ell(\hat{y}, y) = \sum_{r \in R(x,\hat{y})} L(r, y).$$

- e.g. Markov random fields. Parts are configurations for cliques.
- e.g. PCFGs. Parts are rule-location pairs (rules of grammar applied at specific locations in the sentence).

Exponentiated Gradient Algorithm with Parts

$$\Phi(x,y) = \sum_{r \in R(x,y)} \phi(x,r)$$

$$\ell(\hat{y},y) = \sum_{r \in R(x,\hat{y})} L(r,y).$$

- Like a factorization of $\Pr(Y|X)$, where log probabilities decompose as sums over parts.
- We require that loss decomposes in the same way.
  - e.g., Markov random field: $\ell(\hat{y},y) = \sum_c L(\hat{y}_c, y_c)$.
  - e.g., PCFG: $\ell(\hat{y},y) = \sum_r 1[r \text{ in } \hat{y}, \text{not in } y]$.

Exponentiated Gradient Algorithm with Parts

In this case, Q can be expressed as a function of the "marginal" variables, $Q(\alpha) = \tilde{Q}(\mu)$, with

$$\mu_{i,r} = \sum_y \alpha_{i,y} 1[r \in R(x_i, y)].$$

Exponentiated gradient algorithm:

$$\mu_{i,r}^{(t)} = \sum_y \alpha_{i,y}^{(t)} \; 1[r \in R(x_i, y)]$$

$$\alpha_{i,y}^{(t)} = \frac{\exp(\sum_{r \in R(x_i,y)} \theta_{i,r}^{(t)})}{\sum_y \exp(\sum_{r \in R(x_i,y)} \theta_{i,r}^{(t)})}$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\mu \tilde{Q}(\mu^{(t)}).$$

Exponentiated Gradient Algorithm: Sparse Representations

Efficiently computing μ from θ:

- Markov random field: Computing clique marginals from exponential family parameters.
- PCFG: Computing rule probabilities from exponential family parameters.

Large Margin Methods for Structured Classification

- Generative models
  - Markov random fields
  - Probabilistic context-free grammars
- Large margin classifiers
- Quadratic program for large margin classifiers
- Exponentiated gradient algorithm
- **Convergence analysis**

Exponentiated Gradient Algorithm: Convergence

Theorem: For all $u \in \Delta$,

$$\frac{1}{T}\sum_{t=1}^{T} Q(\alpha^{(t)}) \leq Q(u) + \frac{D(u,\alpha^{(1)})}{\eta T} + c_{\eta,Q}\frac{Q(\alpha^{(1)})}{T}.$$

Step 1:

For any $u \in \Delta$,

$$\eta Q(\alpha^{(t)}) - \eta Q(u) \leq D(u,\alpha^{(t)}) - D(u,\alpha^{(t+1)}) + D(\alpha^{(t)},\alpha^{(t+1)}).$$

Follows from convexity of Q, definition of updates.

(Standard in analysis of online prediction algorithms.)

Exponentiated Gradient Algorithm: Convergence

Step 2:

$$D(\alpha^{(t)}, \alpha^{(t+1)}) = \sum_{i=1}^{n} \log \mathbf{E}\left[e^{\eta\left(X_i^{(t)} - \mathbf{E}X_i^{(t)}\right)}\right]$$

$$\leq \left(\frac{e^{\eta B} - 1 - \eta B}{B^2}\right) \sum_{i=1}^{n} \mathrm{var}(X_i^{(t)}),$$

where $\Pr\left(X_i^{(t)} = -\left(\nabla Q(\alpha^{(t)})\right)_{i,y}\right) = \alpha_{i,y}^{(t)}$.

Follows from definition of updates, Bernstein's inequality.

Exponentiated Gradient Algorithm: Convergence

Step 3a: For some $\theta \in [\theta^{(t)}, \theta^{(t+1)}]$,

$$\eta \sum_{i=1}^{n} \mathrm{var}(X_i^{(t)}) - \eta^2 (B + \lambda) \sum_{i=1}^{n} \mathrm{var}(X_{i,\theta}^{(t)}) \leq Q(\alpha^{(t)}) - Q(\alpha^{(t+1)}),$$

where $\Pr\left(X_{i,\theta}^{(t)} = -\left(\nabla Q(\alpha^{(t)})\right)_{i,y}\right) = \alpha(\theta)_{i,y}$.

- Variance of $X_i^{(t)}$ is first order term in Taylor series expansion *(in θ)* for Q.
- Variance of $X_{i,\theta}^{(t)}$ is second order term.
- B is infinity norm of centered version of ∇Q
- λ is largest eigenvalue of $\nabla^2 Q$.

Exponentiated Gradient Algorithm: Convergence

Step 3b: For all $\theta \in [\theta^{(t)}, \theta^{(t+1)}]$,

$$\mathrm{var}(X_{i,\theta}^{(t)}) \leq e^{\eta B}\,\mathrm{var}(X_i^{(t)}).$$

Hence,

$$\sum_{i=1}^{n} \mathrm{var}(X_i^{(t)}) \leq \frac{1}{\eta\left(1 - \eta(B+\lambda)e^{2\eta B}\right)} \left(Q(\alpha^{(t)}) - Q(\alpha^{(t+1)})\right).$$

Exponentiated Gradient Algorithm: Convergence

$$
\begin{aligned}
&\eta Q(\alpha^{(t)}) - \eta Q(u) \\
&\leq D(u, \alpha^{(t)}) - D(u, \alpha^{(t+1)}) + D(\alpha^{(t)}, \alpha^{(t+1)}) \\
&\leq D(u, \alpha^{(t)}) - D(u, \alpha^{(t+1)}) + \left(\frac{e^{\eta B} - 1 - \eta B}{B^2} \right) \sum_{i=1}^{n} \mathrm{var}(X_i^{(t)}) \\
&\leq D(u, \alpha^{(t)}) - D(u, \alpha^{(t+1)}) + c'_{\eta,Q} \left(Q(\alpha^{(t)}) - Q(\alpha^{(t+1)}) \right).
\end{aligned}
$$

Theorem: For all $u \in \Delta$,

$$
\frac{1}{T} \sum_{t=1}^{T} Q(\alpha^{(t)}) \leq Q(u) + \frac{D(u, \alpha^{(1)})}{\eta T} + c_{\eta,Q} \frac{Q(\alpha^{(1)})}{T}.
$$

Large Margin Methods for Structured Classification

- Generative models
  - Markov random fields
  - Probabilistic context-free grammars
- Large margin classifiers
- Quadratic program for large margin classifiers
- Exponentiated gradient algorithm
- Convergence analysis
- **Experiments**
- **Risk bounds**