Boosting with Incomplete Information

Gholamreza Haffari\textsuperscript{1} Yang Wang\textsuperscript{1} Shaojun Wang\textsuperscript{2}
Greg Mori\textsuperscript{1} Feng Jiao\textsuperscript{3}

\textsuperscript{1}Simon Fraser University, Canada
\textsuperscript{2}Wright State University, USA
\textsuperscript{3}Yahoo! Inc., USA

The 25th International Conference on Machine Learning
Helsinki, Finland, 2008
Supervised Classification

Given data set $D = \{x_i, y_i\}$, $x_i$ is the input vector, $y_i$ is the class label, learn a mapping function $F: \mathcal{X} \rightarrow \mathcal{Y}$

Classification with Incomplete Information

- Given two kinds of data sets $D_1 = \{x_i, y_i\}$, $D_2 = \{x_j, h_j, y_j\}$, learn a mapping function $F: \mathcal{X} \times \mathcal{H} \rightarrow \mathcal{Y}$
- This two data sets assumption is general and can be applied to many problems.
Motivation

Hassani, Wang, Wang, Mori, Jiao

Boosting with Incomplete Information
Motivation

Car

y

Car

y

h

x

x
Motivation

\[ x \]

\[ Y \]

\[ \text{You} \]
\[ \text{went} \]
\[ \text{home} \]
\[ \cdot \]

\[ \text{rafti} \]
\[ \text{khaane} \]
\[ \cdot \]
Motivation

\[ x \]
- You
- went
- home
- 

\[ y \]
- rafti
- khaane
- .

\[ x \ h \ y \]
- I
- will
- go
- to
- the
- university
- .

- man
- be
- daneshgah
- khaham
- raft
- .
Previous Work

- EM algorithm for generative models
- Max margin classification (Bi & Zhang, 2004; Chechik et al., 2007)
- Hidden conditional random fields (Koo & Collins, 2005; Quattoni et al., 2005)
- Second order cone programming (Shivaswamy et al., 2006)
Review of boosting

Basics

- Feature (weak learner, sufficient statistics): \( f_k : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \)
- Final classifier: \( y^* = \arg \max_y \left( \sum_k \lambda_k f_k(x, y) \right) \)

Learning parameters \( \lambda_k \)

- Unnormalized model
  - Minimize \( \sum_{x_i} \sum_y q_\lambda(y|x_i) \)
  - where \( q_\lambda(y|x) := \exp \sum_k \lambda_k \left[ f_k(x, y) - f_k(x, \tilde{y}_x) \right] \)

- Normalized model
  - Maximize \( \sum_{x_i} \log p_\lambda(\tilde{y}_{x_i} | x_i) \)
  - where \( p_\lambda(y|x) := q_\lambda(y|x) / Z_\lambda(x) \)

(Lebanon & Lafferty, 2002)
Primal/Dual Problem

Definition

- (extended) KL divergence:
  \[ D(p, q) := \sum_x \tilde{p}(x) \sum_y \left( p(y|x) \log \frac{p(y|x)}{q(y|x)} - p(y|x) + q(y|x) \right) \]

- feasible set:
  \[ \mathcal{F}(\tilde{p}, f) = \left\{ p \mid \sum_x \tilde{p}(x) \sum_y p(y|x)(f_j(x, y) - E_{\tilde{p}}[f_j|x]) = 0, \forall j \right\} \]

Primal problems

(P1) min. \( D(p, q_0) \)  
\[ \text{s.t.} \quad p \in \mathcal{F}(\tilde{p}, f) \]

(P2) min. \( D(p, q_0) \)  
\[ \text{s.t.} \quad p \in \mathcal{F}(\tilde{p}, f) \]
\[ \sum_y p(y|x) = 1 \quad \forall x \]

(Lebanon & Lafferty, 2002)
Problem Statement

- Data sets: $\mathcal{D}_1 = \{(x_i, y_i)\}$, $\mathcal{D}_2 = \{(x_j, h_j, y_j)\}$, $|\mathcal{D}_1| >>> |\mathcal{D}_2|$ in general
- Features:
  
  $\mathcal{F}_1 = \{f_k(x, y)\}$
  
  $\mathcal{F}_2 = \{f_k(x, h, y)\}$

- Goal: how to learn a classifier using $\mathcal{D}_1 \cup \mathcal{D}_2$ and $\mathcal{F}_1 \cup \mathcal{F}_2$?
Boosting with Hidden Variables

Normalized model

- Model: \( p_\lambda(y|x, h) \propto e^{\lambda^T_1 \cdot [f_1(x,y) - f_1(x,\tilde{y}_x)] + \lambda^T_2 \cdot [f_2(x,h,y) - f_2(x,h,\tilde{y}_x)]} \)

- Objective: maximize the log-likelihood

\[
\mathcal{L}(\lambda) := \sum_i \log p_\lambda(y_i|x_i) + \gamma \sum_j \log p_\lambda(y_j|x_j, h_j)
\]

Unnormalized model

- Model: \( q_\lambda(y|x, h) := e^{\lambda^T_1 \cdot [f_1(x,y) - f_1(x,\tilde{y}_x)] + \lambda^T_2 \cdot [f_2(x,h,y) - f_2(x,h,\tilde{y}_x)]} \)

- Objective: minimize the exponential loss

\[
\mathcal{E}(\lambda) := \sum_i \sum_h q_0(h|x) \sum_y q_\lambda(y|x_i, h) + \gamma \sum_j \sum_y q_\lambda(y|x_j, h_j)
\]
Primal/Dual Programs

Definitions

- extended KL-divergence

\[
KL(p||r) = 
\sum_{x,h} \tilde{p}(x)q_0(h|x) \sum_y p(y|h,x) \left[ \log \frac{p(y|x,h)}{r(x,h,y)} - 1 \right] + r(x, h, y)
\]

- feasible set \( S(\tilde{p}, q_0, F) = \{ p \in \mathcal{M} \mid \sum_x \tilde{p}(x) \mathbb{E}_{q_0(h|x)p(y|x,h)} \left[ f - \mathbb{E}_{\tilde{p}(y|x)}[f] \right] = 0, \forall f \in F \} \)

Primal problems

(P1) min. \( KL(p||r) \) \[\text{s.t.} \quad p \in S\]

(P2) min. \( KL(p||r) \) \[\text{s.t.} \quad p \in S, \quad \sum_y p(y|x,h) = 1 \quad \forall x, h\]
Learning and Inference

### Learning
- Construct auxiliary function to bound the change of $\mathcal{E}(\lambda + \Delta \lambda) - \mathcal{E}(\lambda)$ or $\mathcal{L}(\lambda) - \mathcal{L}(\lambda + \Delta \lambda)$
- Both parallel and sequential update rules can be derived

### Inference
- If $h$ is observed on test data, $y^* = \arg\max p(y|h, x)$
- If $h$ is unobserved on test data, $y^* = \arg\max p(y|x)$. This requires summing over $h$. 
Experiments: Visual Object Recognition

airplane

car

face

motorbike
Experiments: Visual Object Recognition

- 1000 training/testing images, 4 categories
- 30% fully observed training images
- Baselines algorithms

BL1

\[ \mathcal{D}_1 \begin{bmatrix} x_i, & y_i \end{bmatrix} \]

\[ \mathcal{D}_2 \begin{bmatrix} x_j, & h_j, & y_j \end{bmatrix} \]
Experiments: Visual Object Recognition

- 1000 training/testing images, 4 categories
- 30% fully observed training images
- Baselines algorithms

\[ D_1 \] \[ x_i, y_i \] \[ D_2 \] \[ x_j, h_j, y_j \]

\[ D_1 \] \[ x_i, y_i \]

\[ f_1 \] \[ f_2 \]
Experiments: Visual Object Recognition

- 1000 training/testing images, 4 categories
- 30% fully observed training images
- Baselines algorithms

\[
\begin{align*}
\mathcal{D}_1 & \quad x_i, \ y_i \\
\mathcal{D}_2 & \quad x_j, h_j, y_j
\end{align*}
\]

Baselines algorithms:

- BL1
- BL2
- BL3
Experiments: Visual Object Recognition
Experiments: Visual Object Recognition
## Experiments: Visual Object Recognition

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Log-Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>97.22%</td>
<td>-0.0916</td>
</tr>
<tr>
<td>BL1</td>
<td>89.26%</td>
<td>-1.1417</td>
</tr>
<tr>
<td>BL2</td>
<td>88.01%</td>
<td>-0.5698</td>
</tr>
<tr>
<td>BL3</td>
<td>90.43%</td>
<td>-0.4375</td>
</tr>
</tbody>
</table>

### Normalized Model

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Log of Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>94.83%</td>
<td>-0.7412</td>
</tr>
<tr>
<td>BL1</td>
<td>82.57%</td>
<td>-1.1231</td>
</tr>
<tr>
<td>BL2</td>
<td>89.86%</td>
<td>-0.7977</td>
</tr>
<tr>
<td>BL3</td>
<td>87.64%</td>
<td>-0.8068</td>
</tr>
</tbody>
</table>

### Unnormalized Model
Experiments: Named Entity Recognition

- CoNLL03 shared task: 5000 fully observed, 6000 partially observed, 1000 testing
- Features:
  - Lexical: word forms and their positions in the window
  - Syntactic: part-of-speech tags (if available)
  - Orthographic: capitalized, include digits, ...
  - Affixes: suffixes and prefixes
  - Left predict: predicted labels for the two previous words
**Experiments: Named Entity Recognition**

$h$ is unobserved on test data

<table>
<thead>
<tr>
<th></th>
<th>f-measure</th>
<th>log-likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>49.45%</td>
<td>-0.5784</td>
</tr>
<tr>
<td>BL1</td>
<td>46.63%</td>
<td>-0.5932</td>
</tr>
<tr>
<td>BL2</td>
<td>48.10%</td>
<td>-0.5803</td>
</tr>
<tr>
<td>BL3</td>
<td>47.80%</td>
<td>-0.5880</td>
</tr>
</tbody>
</table>

**normalized model**

<table>
<thead>
<tr>
<th></th>
<th>f-measure</th>
<th>log of loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>49.04%</td>
<td>-2.6337</td>
</tr>
<tr>
<td>BL1</td>
<td>46.24%</td>
<td>-2.6458</td>
</tr>
<tr>
<td>BL2</td>
<td>47.58%</td>
<td>-2.6378</td>
</tr>
<tr>
<td>BL3</td>
<td>46.39%</td>
<td>-2.6434</td>
</tr>
</tbody>
</table>

**unnormalized model**
H is observed on test data

<table>
<thead>
<tr>
<th></th>
<th>f-measure</th>
<th>log-likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>59.60%</td>
<td>-0.5759</td>
</tr>
<tr>
<td>BL1</td>
<td>56.51%</td>
<td>-0.5916</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>f-measure</th>
<th>log of loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>60.17%</td>
<td>-0.2586</td>
</tr>
<tr>
<td>BL1</td>
<td>55.46%</td>
<td>-0.2655</td>
</tr>
</tbody>
</table>

normalized model

unnormalized model
Summary

Conclusion
A boosting approach that extends the traditional boosting framework by incorporating hidden variables, and achieves better results than baseline approaches.

Future work
- Extension to more complex dependent hidden variables (e.g., trees, graphs), variational methods (e.g., loopy BP) may be used
- Connection with confidence-rated AdaBoost (Schapire & Singer, 1999)