SVM: Multiclass and Structured Prediction

Bin Zhao
Part I: Multi-Class SVM
2-Class SVM

- Primal form

\[
\min_{w,b} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\
\text{s.t.} \quad y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \forall i \\
\xi_i \geq 0, \quad \forall i
\]

- Dual form

\[
\max_{\alpha} \quad J(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\
\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \\
\sum_{i=1}^{m} \alpha_i y_i = 0.
\]
Real world classification problems

**Digit recognition**

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

**Automated protein classification**

- SCOP
  - Class: All beta, All alpha, Alpha + beta
  - Fold: Globular, Alpha-coiled
  - Superfamily: 8-stranded beta
  - Family: Short chain, Long chain
  - Protein: Immunoglobulin, O-CIF

**Object recognition**

- Various images of objects

**Phoneme recognition**

- [Waibel, Hanzawa, Hinton, Shikano, Lang 1989]

- The number of classes is sometimes big
- The multi-class algorithm can be heavy

**References**

- [Website for recognition](http://www.glue.umd.edu/~zhelin/recog.html)
How can we solve multi-class problem?

- One-against-one
- One-against-rest
- Crammer & Singer’s formulation
- Error-correcting output coding
- Empirical comparisons
One-against-one
One-against-rest
Problems

One-against-one

One-against-rest
Crammer & Singer’s formulation

• A Naïve approach

\[
\begin{align*}
\min_{w_1, \ldots, w_K} & \quad \frac{1}{2} \| (w_1, \ldots, w_K) \|^2 + C \sum_{ik} \xi_{ik} \\
\text{s.t.} & \quad \forall (i, k), \quad w_{y_i}^T x_i - w_k^T x_i \geq 1 \{ k \neq y^i \} - \xi_{ik}
\end{align*}
\]
Crammer & Singer’s formulation

• A Naïve approach

\[
\min_{w_1,\ldots,w_K} \quad \frac{1}{2} \|(w_1,\ldots,w_K)\|^2 + C \sum_{ik} \xi_{ik} \\
\text{s.t.} \quad \forall (i, k), \quad w_{yi}^T x^i - w_k^T x^i \geq 1 \{k \neq y^i\} - \xi_{ik}
\]

• C & S’s formulation

\[
\min_{w_1,\ldots,w_k,\xi} \quad \frac{1}{2} \beta \sum_{p=1}^{k} \|w_p\|^2 + \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad \forall i = 1, \ldots, n, r = 1, \ldots, k \\
\quad \quad \quad \quad \quad w_{yi}^T x_i + \delta_{y_i,r} - w_r^T x_i \geq 1 - \xi_i
\]
Error-Correcting Output Code (ECOC)

Table 3: A 15-bit error-correcting output code for a ten-class problem.

<table>
<thead>
<tr>
<th>Class</th>
<th>Code Word</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
</tr>
</tbody>
</table>

Source: Dietterich and Bakiri (1995)
Discussions

D&B suggest that $M$ be constructed to have good error-correcting properties — if the minimum Hamming distance between rows of $M$ is $d$, then the resulting multiclass classification will be able to correct any $\lfloor \frac{d-1}{2} \rfloor$ errors.

In learning, good column separation is also important — two identical (or opposite) columns will make identical errors. This highlights the key difference between the multiclass machine learning framework and standard error-correcting code applications.
Special cases of ECOC

### One-against-one

<table>
<thead>
<tr>
<th></th>
<th>+1</th>
<th>+1</th>
<th>+1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All-pairs $\rightarrow k$ by $\binom{k}{2}$

### One-against-rest

<table>
<thead>
<tr>
<th></th>
<th>+1</th>
<th>−1</th>
<th>−1</th>
<th>−1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>−1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

One-against-all $\rightarrow k$ by $k$
Empirical Study

• In Defense of One-Vs-All Classification – *JMLR* 2004
  – The most important step in good multiclass classification is to use the best binary classifier available. Once this is done, it seems to make little difference what multiclass scheme is applied, and therefore a simple scheme such as OVA (or AVA) is preferable to a more complex error-correcting coding scheme or single-machine scheme.
Part II: Structured SVM

Slides Courtesy: Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, Yasemin Altun
Local Classification

Classify using local information
⇒ Ignores correlations!

[thanks to Ben Taskar for slide!]
Structured Classification

- Use local information
- Exploit correlations

[thanks to Ben Taskar for slide!]
Structured Prediction

- **Given**: labeled training data \((x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}\)
- **Task**: learn mapping from inputs \(x \in \mathcal{X}\) to outputs \(y \in \mathcal{Y}\)
- **Special cases**
  - **Binary classification**: \(\mathcal{Y} = \{-1, 1\}\)
  - **Multiclass classification**: \(\mathcal{Y} = \{1, \ldots, k\}\)
- **Naive approach**: treat each possible output in \(\mathcal{Y}\) as discrete label
  - **Big problem**: \(\mathcal{Y}\) gets huge
Exploiting Structure

- Enumerating all members of $\mathcal{Y}$ is often intractable
- Hence, try to exploit structure and dependencies within the output space
- Approach: learn discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$
- Hypotheses have the form: $f(x; w) = \operatorname*{argmax}_{y \in \mathcal{Y}} F(x, y; w)$
- Parametrized by parameters $w$
- Assume $F$ linear in combined feature representation $\Psi(x, y)$

$$F(x, y; w) = \langle w, \Psi(x, y) \rangle$$
Optimization Problem

- First, assume separable data, i.e.,
  \[ \forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta \Psi_i(y) \rangle > 0, \quad \delta \Psi_i(y) \equiv \Psi(x_i, y_i) - \Psi(x_i, y) \]
- Requiring that margin \( \geq 1 \) gives the optimization problem
  \[
  \text{SVM}_0 : \min_w \frac{1}{2} \|w\|^2 \\
  \forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta \Psi_i(y) \rangle \geq 1.
  \]
- To handle non-separable data, add one slack variable per point
  \[
  \text{SVM}_1 : \min_{w, \xi} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i, \quad \text{s.t.} \forall i, \xi_i \geq 0 \\
  \forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta \Psi_i(y) \rangle \geq 1 - \xi_i.
  \]
Replacing the Zero-one Loss

\[ \text{SVM}_1 : \min_{\mathbf{w}, \xi} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i, \text{ s.t. } \forall i, \xi_i \geq 0 \]

\[ \forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \langle \mathbf{w}, \delta \Psi_i(y) \rangle \geq 1 - \xi_i. \]

- Problem: all margin violations are penalized identically
- A better way: assume existence of loss function \( \Delta(\mathbf{y}, \hat{\mathbf{y}}) \) to model closeness of predictions
- First idea: scale slack variables by loss incurred

\[ \text{SVM}^{\Delta_s}_1 : \min_{\mathbf{w}, \xi} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i, \text{ s.t. } \forall i, \xi_i \geq 0 \]

\[ \forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \langle \mathbf{w}, \delta \Psi_i(y) \rangle \geq 1 - \frac{\xi_i}{\Delta(\mathbf{y}_i, \mathbf{y})}. \]
Replacing the Zero-one Loss

\[ \text{SVM}_1: \min_{w, \xi} \frac{1}{2} ||w||^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i, \text{ s.t. } \forall i, \xi_i \geq 0 \]

\[ \forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta \Psi_i(y) \rangle \geq 1 - \xi_i. \]

- Problem: all margin violations are penalized identically
- A better way: assume existence of loss function \( \Delta(y, \hat{y}) \) to model closeness of predictions
- Another idea: scale the margin by the loss (e.g., Hamming loss in Max-Margin Markov Networks):

\[ \forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta \Psi_i(y) \rangle \geq \Delta(y_i, y) - \xi_i \]

- Also yields bound on empirical risk
Dual Optimization Problem

• Write dual problem using SVM-like Lagrangian techniques

• (take Lagrangian, find saddle point with KKT conditions, plug back into objective function)

• For $\text{SVM}_0$, we get
  \[
  \max_{\alpha} \sum_{i, y \neq y_i} \alpha_{iy} - \frac{1}{2} \sum_{i, y \neq y_i; j, y \neq y_j} \alpha_{iy} \alpha_{jy} \langle \delta \Psi_i(y), \delta \Psi_j(y) \rangle
  \]
  \[
  \text{s.t. } \forall i, \forall y \neq y_i \in Y \setminus y_i : \quad \alpha_{iy} \geq 0.
  \]

• As in SVMs, can replace inner product with kernel:
  \[
  K((x, y), (x', y'))
  \]

• Other optimization problems are also easily formulated in terms of dual QPs
Case Study: Max Margin Learning on Domain-Independent Web Information Extraction
Motivation

• “Understand” web page
  – Assign semantics to each component of a page
  – Understand functionality of each section
  – Distinguish main content from side contents

• Page layout reveals important cues
Motivation (Cont.)

• Human can “understand” a page written in foreign language
  – Position
  – Size
  – Font
  – Color
  – Boldness
  – Relative position to other sections

• Can a computer fulfill similar task?
  – Domain-independent web information extraction
The Proposed Approach

• Page Segmentation: vision tree
  – Built based on DOM tree
  – With visual information
  – Correspond to how each node is displayed

• Structured Segmentation
  – Assign label for each node in the vision tree
  – Information extraction based on node classification
Structured Classification

• Label space for **leaf** nodes
  – attribute name, attribute value, non attribute, image, nav bar, main title, page tail, image caption
  – non-attribute (anything else)

• Label space for **non-leaf** nodes
  – structure block, data record, nav bar block, non attribute block, value block, page tail block, name block, image block, image caption block, main title block
Max Margin Learning

• Structured output space (tree)
  – If treated as conventional classification: exponential number of classes → unable to train
  – Augmented loss: misclassify a tree by one node should receive much less penalty than misclassifying the entire tree

• Input-output feature mapping: $\Phi(x_l, y_l) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}^d$

• Linear discriminant function: $F(x, y) = \langle w, \Phi(x, y) \rangle$

• Response to input $x$:

$$h(x) = \arg \max_{y \in \mathcal{Y}} F(x, y)$$
Max Margin Learning (Cont.)

• Augmented loss: $\Delta(y, y')$
  – # of disagreements between $y$ and $y'$
• Learning: use SVM to find optimal $w$

$$
\min_w \quad \frac{1}{2}\|w\|^2 + C \sum_{l=1}^{m} \xi_l \\
\text{s.t.} \quad w^T \Phi(x_l, y_l) - w^T \Phi(x_l, y') \geq \Delta(y_l, y') - \xi_l, \forall y', \forall l \\
\xi_l \geq 0, \forall l
$$

• Inference: dynamical programming

$$
y^* = \arg \max_{y \in Y} F(x, y) = \arg \max_{y \in Y} \langle w, \Phi(x, y) \rangle
$$
Input-Output Feature Mapping

• Two types of cliques in the hierarchical model
  – Cliques $C^b$ covering observation-state node pairs
  – Cliques $C^p$ covering state-state node pairs

• Correspondingly, two types of features
  – Features describing cliques $C^b$
    \[ \phi^b_{l}(x, y) = \sum_{(x_i, y_i) \in C^b} \mathbb{I}[y_i = l] \phi(x_i) \]
  – Features describing cliques $C^p$
    \[ \phi^p_{l}(x, y) = \sum_{(y_i, y_j) \in C^p} \mathbb{I}[y_i = l] \mathbb{I}[y_j = l] \]
Input-Output Feature Mapping: Type I

• Spatial features
  – Position of block center, block height, block weight, block area

• Features for all nodes
  – Link number, number of various HTML tags, number of child nodes

• Features for leaf nodes only
  – Text length, font, bold, italic, word count, number of images, image size, link text length
Input-Output Feature Mapping: Type II

• Parent-child relationship
  – Label co-occurrence pattern
    \[ \forall l, l': \; \phi^{P}_{st}(l, l') = \mathbb{I}_{[y_s=l]} \mathbb{I}_{[y_t=l']} \]

• Connect spatially adjacent blocks
  – Link a node with its k nearest neighbors
  – Define label co-occurrence pattern for connected node pair \((i,j)\)
    \[ \forall l, l': \; \phi^{S}_{ij}(l, l') = \mathbb{I}_{[y_i=l]} \mathbb{I}_{[y_j=l]} \]
Learning and Inference

• Learning
  – Quadratic programming with exponential number of constraints $\rightarrow$ cutting plane algorithm (a.k.a., constraint generation, bundle method)

• Inference
  – Without edges between spatially adjacent blocks $\rightarrow$ dynamical programming (a.k.a. Viterbi decoding)
  – With edges between spatially adjacent blocks $\rightarrow$ loopy belief propagation
Empirical Study

- Block level prediction results

<table>
<thead>
<tr>
<th></th>
<th>Multiclass SVM</th>
<th>SVM-Tree</th>
<th>SVM-Cyclic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>52.11</td>
<td>71.93</td>
<td>74.96</td>
</tr>
<tr>
<td>pName</td>
<td>65.55</td>
<td>69.08</td>
<td>72.62</td>
</tr>
<tr>
<td>rName</td>
<td>12.13</td>
<td>69.83</td>
<td>73.41</td>
</tr>
<tr>
<td>pValue</td>
<td>52.55</td>
<td>70.69</td>
<td>74.00</td>
</tr>
<tr>
<td>rValue</td>
<td>35.78</td>
<td>75.41</td>
<td>78.55</td>
</tr>
</tbody>
</table>

- Attribute name-value pair extraction
  - 1000 web pages
  - Precision: 56.91%
  - Recall: 59.37%
Thank you