Support vector comparison machines

Toby Dylan Hocking
toby@sg.cs.titech.ac.jp
joint work with Supaporn Spanurattana and Masashi Sugiyama

November 29, 2013
Introduction and related work

Learning a max-margin comparison function

Results and conclusions
Motivating example: learning to compare sushi

salmon is better than anago

chu-toro is as good as kani-miso

If I give you another sushi pair, can you tell me which one is better, or if they are equally good?
Motivating example: learning to compare sushi

salmon is better than anago

chu-toro is as good as kani-miso

If I give you another sushi pair, can you tell me which one is better, or if they are equally good?
Learning a comparison function

We are given \( n \) training pairs \((x_i, x'_i, y_i)\)

- **Input**: a pair of feature vectors \( x_i, x'_i \in \mathbb{R}^p \)
e.g. sushi fattiness, taster birthplace.

- **Output**: a label \( y_i = \)
\[
\begin{cases} 
-1 & \text{if } x_i \text{ is better} \\
0 & \text{if } x_i \text{ is as good as } x'_i \\
1 & \text{if } x'_i \text{ is better.}
\end{cases}
\]

**Goal**: find a comparison function \( c : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \{-1, 0, 1\} \)

- **Good prediction with respect to the zero-one loss**: 
\[
\minimize_c \sum_{i \in \text{test}} I[ y_i \neq c(x_i, x'_i)]
\]

- **Symmetry**: \( c(x, x') = -c(x', x) \).
Geometric interpretation when $r(x) = \|x\|_2^2$

Original features $x \in \mathbb{R}^p$

Enlarged features $\Phi(x)$

- Input feature $x_{i,1}$
- Additional feature $x_{i,2}^2$
- Label $y_i$

- $r(x) = w^\top \Phi(x)$
Geometric interpretation when \( r(x) = \|x\|^2 \)

**Enlarged features** \( \Phi(x) \)

**Difference** \( \Phi(x') - \Phi(x) \)

\[
\Phi(x'_i) - \Phi(x_i) = 0 \\
\Phi(x'_i) - \Phi(x_i) = 1 \\
\Phi(x'_i) - \Phi(x_i) = -1 \\
\Phi(x'_i) - \Phi(x_i) = -1
\]
Related work: reject, rank, and rate

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Inputs</th>
<th>single items $\mathbf{x}$</th>
<th>pairs of items $\mathbf{x}, \mathbf{x}'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y \in {-1, 1}$</td>
<td>SVM</td>
<td></td>
<td>SVMrank</td>
</tr>
<tr>
<td>$y \in {-1, 0, 1}$</td>
<td>Reject option</td>
<td></td>
<td>this work</td>
</tr>
</tbody>
</table>


- T Joachims. Optimizing search engines using clickthrough data. KDD 2002. (SVMrank)

- K Zhou et al. Learning to rank with ties. SIGIR 2008. (boosting, ties are more effective with more output values)

- R Herbrich et al. TrueSkill: a Bayesian skill rating system. NIPS 2006. (generalization of Elo for chess)
SVMrank ignores equality $y_i = 0$ pairs

Linear $r(x) = w^T x$.

$$\begin{align*}
\text{minimize} & \quad w^T w \\
\text{subject to} & \quad w^T (x'_i - x_i) y_i \geq 1, \quad \forall i \text{ such that } y_i \in \{-1, 1\}.
\end{align*}$$
Introduction and related work

Learning a max-margin comparison function

Results and conclusions
We will learn a

- Ranking function $r : \mathbb{R}^p \rightarrow \mathbb{R}$. Bigger is better.
- Threshold $\tau \in \mathbb{R}^+$. A small difference $|r(x') - r(x)| \leq \tau$ is not significant.
- Comparison function $c_\tau(x, x') = \begin{cases} -1 & \text{if } r(x') - r(x) < -\tau \\ 0 & \text{if } |r(x') - r(x)| \leq \tau \\ 1 & \text{if } r(x') - r(x) > \tau. \end{cases}$

The problem becomes

$$\minimize_{r, \tau} \sum_{i=1}^{n} l \left[ y_i \neq c_\tau(x_i, x'_i) \right].$$
Max margin comparison is a linear program (LP)

$\max_{\mu \in \mathbb{R}, w \in \mathbb{R}^p} \mu$

subject to

$\mu \leq 1 - |w^T(x'_i - x_i)|, \ \forall i \text{ such that } y_i = 0,$

$\mu \leq -1 + w^T(x'_i - x_i)y_i, \ \forall i \text{ such that } y_i \in \{-1, 1\}.$

Note: if the optimal $\mu > 0$ then the data are separable.
Learned function is invariant to training direction

\[ r(x') - r(x) = 1 \]

\[ r(x') - r(x) = -1 \]

\[
\begin{align*}
\text{maximize} & \quad \mu \\
\text{subject to} & \quad \mu \leq 1 - |w^T(x'_i - x_i)|, \; \forall \; i \text{ such that } y_i = 0, \\
& \quad \mu \leq -1 + w^T(x'_i - x_i)y_i, \; \forall \; i \text{ such that } y_i \in \{-1, 1\}. 
\end{align*}
\]

Note: if the optimal \( \mu > 0 \) then the data are separable.
We can use an SVM solver to learn a ranking function!
Introduction and related work

Learning a max-margin comparison function

Results and conclusions
Simulation: true patterns $r$ and noisy training pairs

Validation and test data have the same number of pairs $n$ and the same proportion of equality pairs $\rho$. 
Details of simulation setup

- Inputs $x_i, x'_i \in [-3, 3]^2$.
- True ranking function $r(x) = ||x||_j^2$ for $j \in \{1, 2, \infty\}$.
- Noisy labels $y_i = t_1[r(x'_i) - r(x_i)] + \epsilon_i$.
- Threshold function $t_1(x) = \begin{cases} 
-1 & \text{if } x < -1, \\
0 & \text{if } |x| \leq 1, \\
1 & \text{if } x > 1. 
\end{cases}$
- Noise $\epsilon_i \sim N(0, \sigma)$ with standard deviation $\sigma = 1/4$.
- Train, validation, and test sets with
  - same number of training pairs $n$, and
  - same proportion of equality pairs $\rho$.
- Fit a $10 \times 10$ grid of models to the training set:
  - Cost parameter $C = 10^{-3}, \ldots, 10^3$,
  - Gaussian kernel width $2^{-7}, \ldots, 2^4$.
- Select the model with minimal zero-one loss on the validation set.
We ran 3 different algorithms on each data set

<table>
<thead>
<tr>
<th>Input:</th>
<th>equality pairs</th>
<th>inequality pairs</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank</td>
<td></td>
<td>$</td>
<td>I_1</td>
</tr>
<tr>
<td>rank2</td>
<td>$2</td>
<td>I_0</td>
<td>$</td>
</tr>
<tr>
<td>compare</td>
<td>$2</td>
<td>I_0</td>
<td>$</td>
</tr>
</tbody>
</table>

Equality $y_i = 0$ pairs are shown as – segments.

Inequality $y_i \in \{-1, 1\}$ pairs are shown as $\rightarrow$ arrows.

- **rank** ignores each input equality pair.
- **rank2** converts each input equality pair to two contradictory inequality pairs.
- **compare** directly models the equality pairs.
Test error lowest for proposed SVM compare model

\[ r(x) = \|x\|^2 \]

Number of labeled pairs, half equality and half inequality

Percent incorrectly predicted test pairs

Function:
- rank
- rank2
- compare
- truth

Number of labeled pairs, half equality and half inequality
Test error lowest for proposed SVM compare model

\[ r(x) = \|x\|_1^2 \]

\[ r(x) = \|x\|_2^2 \]

\[ r(x) = \|x\|_\infty^2 \]
No difference for few equality pairs, rank worse when there are many equality pairs

\[ r(x) = \|x\|_\infty^2 \]

\[ \rho = \text{proportion of equality } y_i = 0 \text{ pairs} \]
No difference for few equality pairs, rank worse when there are many equality pairs.
Sushi data of Kamishima et al.

- [http://www.kamishima.net/sushi/](http://www.kamishima.net/sushi/)
- 100 different sushis rated by 5000 different people.
- Each person rated 10 sushis on a 5 point scale.
- Convert 10 ratings to 5 preference pairs.
- 17,832 equality \( y_i = 0 \) pairs and
- 7,168 inequality \( y_i \in \{-1, 1\} \) pairs.
- Feature pairs \( x_i, x'_i \in \mathbb{R}^{14} \).
- 7 sushi features: style, major, minor, oily, eating frequency, price, and selling frequency.
- 7 taster/person features: Sushi gender, age, time, birthplace and current home (we converted Japanese prefecture codes to latitude/longitude coordinates).
Sushi data are harder, but SVMcompare still has lowest test error.

![Graph showing Test error and Test AUC with n = number of labeled pairs and ρ = proportion of equality yᵢ = 0 pairs.](image)
Conclusions and future work

- Learned a nonlinear ranking function $r(x) \in \mathbb{R}$, and
- a comparison function $c(x, x') \in \{-1, 0, 1\}$.
- Results: $\text{rank} < \text{rank2} < \text{compare}$.
- Directly learning from $y_i = 0$ equality pairs is important, when they are present!
- https://github.com/tdhock/rankSVMcompare
- Scaling to large data? Stochastic gradient methods?
Thank you!

Supplementary slides appear after this one.