Machine Learning for Information Discovery

Thorsten Joachims

Cornell University
Department of Computer Science
## (Supervised) Machine Learning

### GENERAL:

**Input:**
- training examples
- design space

**Training:**
- automatically find the solution in design space that works well on the training data

**Prediction:**
- predict well on new examples

### EXAMPLE: Text Retrieval

**Input:**
- queries with relevance judgments
- parameters of retrieval function

**Training:**
- find parameters so that many relevant documents are ranked highly

**Prediction:**
- rank relevant documents high also for new queries
Common Machine Learning Tasks in ID

- **Text Retrieval**
  - provide good rankings for a query
  - use machine learning on relevance judgments to optimize ranking function

- **Text Classification**
  - classify documents by their semantic content
  - use machine learning and classified documents to learn classification rules

- **Information Extraction**
  - learn to extract particular attributes from a document
  - use machine learning to identify where in the text the information is located

- **Topic Detection and Tracking**
  - find and track new topics in a stream of documents
Text Retrieval

Query:
- "Support Vector Machine"

Goal:
- "rank the documents I want high in the list"
E.D. And F. MAN TO BUY INTO HONG KONG FIRM

The U.K. Based commodity house E.D. And F. Man Ltd and Singapore’s Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeos 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

About a corporate acquisition?

YES

NO
Information Extraction

FlipDog.com - Microsoft Internet Explorer

Step 1:
Location: Where do you want to work?

Step 2:
Category: What type of work?

Step 3:
Employer: Which employer?

Show recruiter & staffing agency listings

Keywords:

Location: United States
Category: All Categories
Employer: All Employers

263,540 job(s) found

Find Jobs by Country
41,782 jobs outside of the United States
Why Use Machine Learning?

Approach 1: Just do everything manually!
- pretty mind numbing
- too expensive (e.g. Reuters 11,000 stories per day, 90 indexers)
- does not scale

Approach 2: Construct automatic rules manually!
- humans are not really good at it (e.g. constructing classification rules)
- no expert is available (e.g. rules for filtering my email)
- its just too expensive to do by hand (e.g. ArXiv classification, personal retrieval functions)

Approach 3: Construct automatic rules via machine learning!
- training data is cheap and plenty (e.g. clickthrough)
- can be done on an (pretty much) arbitrary level of granularity
- works well without expert interventions
E.D. And F. MAN TO BUY INTO HONG KONG FIRM

The U.K. Based commodity house E.D. And F. Man Ltd and Singapores Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeos 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

About a corporate acquisition?

YES  NO
## Tasks and Applications

<table>
<thead>
<tr>
<th>Text-Classification Task</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Routing</td>
<td>Help-Desk Support:</td>
</tr>
<tr>
<td></td>
<td>Who is an appropriate expert for a particular problem?</td>
</tr>
<tr>
<td>Information Filtering</td>
<td>Information Agents:</td>
</tr>
<tr>
<td></td>
<td>Which news articles are interesting to a particular person?</td>
</tr>
<tr>
<td>Relevance Feedback</td>
<td>Information Retrieval:</td>
</tr>
<tr>
<td></td>
<td>What are other documents relevant for a particular query?</td>
</tr>
<tr>
<td>Text Categorization</td>
<td>Knowledge Management:</td>
</tr>
<tr>
<td></td>
<td>Organizing a document database by semantic categories.</td>
</tr>
</tbody>
</table>

Hand-coding text classifiers is costly or even impractical!
Learning Text Classifiers

Goal:
- Learner uses training set to find classifier with low prediction error.
Representing Text as Attribute Vectors

**Attributes:** Words (Word-Stems)

**Values:** Occurrence-Frequencies

---

From: xxx@sciences.sdsu.edu
Newsgroups: comp.graphics
Subject: Need specs on Apple QT

I need to get the specs, or at least a very verbose interpretation of the specs, for QuickTime. Technical articles from magazines and references to books would be nice, too.

I also need the specs in a format usable on a Unix or MS-Dos system. I can't do much with the QuickTime stuff they have on ...

0  baseball
3  specs
0  graphics
1  references
0  hockey
0  car
0  clinton
.
.
1  unix
0  space
2  quicktime
0  computer

---

The ordering of words is ignored!
Support Vector Machines

Training Examples: \((x_1, y_1), \ldots, (x_n, y_n)\) \(x_i \in \mathbb{R}^N\) \(y_i \in \{1, -1\}\)

Hypothesis Space: \(h(x) = \text{sgn} [\vec{w} \cdot \vec{x} + b]\) with \(\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i\)

Training: Find hyperplane \(\langle \vec{w}, \vec{b} \rangle\) with minimal \(\frac{\delta^2}{2} + C \sum_{i=1}^{n} \xi_i\)

Hard Margin (separable)

Soft Margin (training error)
# Experimental Results

**Reuters Newswire**
- 90 categories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

**WebKB Collection**
- 4 categories
- 4183 training doc.
- 226 test doc.
- ~38000 features

**Ohsumed MeSH**
- 20 categories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

<table>
<thead>
<tr>
<th>microaveraged precision/recall breakeven-point [0..100]</th>
<th>Reuters</th>
<th>WebKB</th>
<th>Ohsumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>72.3</td>
<td>82.0</td>
<td>62.4</td>
</tr>
<tr>
<td>Rocchio Algorithm</td>
<td>79.9</td>
<td>74.1</td>
<td>61.5</td>
</tr>
<tr>
<td>C4.5 Decision Tree</td>
<td>79.4</td>
<td>79.1</td>
<td>56.7</td>
</tr>
<tr>
<td>k-Nearest Neighbors</td>
<td>82.6</td>
<td>80.5</td>
<td>63.4</td>
</tr>
<tr>
<td>SVM</td>
<td>87.5</td>
<td>90.3</td>
<td>71.6</td>
</tr>
</tbody>
</table>

Table from [Joachims, 2002]
Humans vs. Machine Learning

Task: Write query that retrieves all CS documents in ArXiv.org!

Data: 29,890 training examples / 32,487 test examples (relevant:=in_CS)
Humans vs. Machine Learning (Setting 2)

Task: Improve query using the training data!

Data: 29,890 training examples / 32,487 test examples (relevant:=in_CS)
What is a Good Retrieval Function?

Query:
- "Support Vector Machine"

Goal:
- "rank the documents I want high in the list"
Training Examples from Clickthrough

Assumption: If a user skips a link $a$ and clicks on a link $b$ ranked lower, then the user preference reflects $\text{rank}(b) < \text{rank}(a)$.

Example: $(3 < 2)$ and $(7 < 2)$, $(7 < 4)$, $(7 < 5)$, $(7 < 6)$

<table>
<thead>
<tr>
<th>Ranking Presented to User:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Kernel Machines</td>
</tr>
<tr>
<td><a href="http://svm.first.gmd.de/">http://svm.first.gmd.de/</a></td>
</tr>
<tr>
<td>2. Support Vector Machine</td>
</tr>
<tr>
<td><a href="http://jbolivar.freeservers.com/">http://jbolivar.freeservers.com/</a></td>
</tr>
<tr>
<td>3. SVM-Light Support Vector Machine</td>
</tr>
<tr>
<td><a href="http://ais.gmd.de/~thorsten/svm_light/">http://ais.gmd.de/~thorsten/svm_light/</a></td>
</tr>
<tr>
<td>4. An Introduction to Support Vector Machines</td>
</tr>
<tr>
<td><a href="http://www.support-vector.net/">http://www.support-vector.net/</a></td>
</tr>
<tr>
<td>5. Support Vector Machine and Kernel ... References</td>
</tr>
<tr>
<td><a href="http://svm.research.bell-labs.com/SVMrefs.html">http://svm.research.bell-labs.com/SVMrefs.html</a></td>
</tr>
<tr>
<td>6. Archives of SUPPORT-VECTOR-MACHINES ...</td>
</tr>
<tr>
<td><a href="http://www.jiscmail.ac.uk/lists/SUPPORT...">http://www.jiscmail.ac.uk/lists/SUPPORT...</a></td>
</tr>
<tr>
<td>7. Lucent Technologies: SVM demo applet</td>
</tr>
<tr>
<td><a href="http://svm.research.bell-labs.com/SVT/SVMsvt.html">http://svm.research.bell-labs.com/SVT/SVMsvt.html</a></td>
</tr>
<tr>
<td>8. Royal Holloway Support Vector Machine</td>
</tr>
<tr>
<td><a href="http://svm.dcs.rhbnc.ac.uk/">http://svm.dcs.rhbnc.ac.uk/</a></td>
</tr>
</tbody>
</table>
Training Examples from Clickthrough

Assumption: If a user skips a link $a$ and clicks on a link $b$ ranked lower, then the user preference reflects $\text{rank}(b) < \text{rank}(a)$.

Example: $(3 < 2)$ and $(7 < 2)$, $(7 < 4)$, $(7 < 5)$, $(7 < 6)$

<table>
<thead>
<tr>
<th>Ranking Presented to User:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Kernel Machines</td>
</tr>
<tr>
<td><a href="http://svm.first.gmd.de/">http://svm.first.gmd.de/</a></td>
</tr>
<tr>
<td>2. Support Vector Machine</td>
</tr>
<tr>
<td><a href="http://jbolivar.freeservers.com/">http://jbolivar.freeservers.com/</a></td>
</tr>
<tr>
<td>3. SVM-Light Support Vector Machine</td>
</tr>
<tr>
<td><a href="http://ais.gmd.de/~thorsten/svm.light/">http://ais.gmd.de/~thorsten/svm.light/</a></td>
</tr>
<tr>
<td>4. An Introduction to Support Vector Machines</td>
</tr>
<tr>
<td><a href="http://www.support-vector.net/">http://www.support-vector.net/</a></td>
</tr>
<tr>
<td>5. Support Vector Machine and Kernel ... References</td>
</tr>
<tr>
<td><a href="http://svm.research.bell-labs.com/SVMrefs.html">http://svm.research.bell-labs.com/SVMrefs.html</a></td>
</tr>
<tr>
<td>6. Archives of SUPPORT-VECTOR-MACHINES ...</td>
</tr>
<tr>
<td><a href="http://www.jiscmail.ac.uk/lists/SUPPORT">http://www.jiscmail.ac.uk/lists/SUPPORT</a>...</td>
</tr>
<tr>
<td>7. Lucent Technologies: SVM demo applet</td>
</tr>
<tr>
<td><a href="http://svm.research.bell-labs.com/SVT/SVMsvt.html">http://svm.research.bell-labs.com/SVT/SVMsvt.html</a></td>
</tr>
<tr>
<td>8. Royal Holloway Support Vector Machine</td>
</tr>
<tr>
<td><a href="http://svm.dcs.rhbnc.ac.uk/">http://svm.dcs.rhbnc.ac.uk/</a></td>
</tr>
</tbody>
</table>
Learning to Rank

Assume:
- distribution of queries $P(Q)$
- distribution of target rankings for query $P(R \mid Q)$

Given:
- collection $D$ of $m$ documents
- i.i.d. training sample $(q_1, r_1), \ldots, (q_n, r_n)$

Design:
- set of ranking functions $F$, with elements $f: Q \rightarrow P^D \times D$ (weak ordering)
- loss function $l(r_a, r_b)$
- learning algorithm

Goal:
- find $f^* \in F$ with minimal
  \[
  R_P(f) = \int l(f(q), r) dP(q, r)
  \]
A Loss Function for Rankings

For two orderings $r_a$ and $r_b$, a pair $d_i \neq d_j$ is

- **concordant**, if $r_a$ and $r_b$ agree in their ordering
  \[ P = \text{number of concordant pairs} \]

- **discordant**, if $r_a$ and $r_b$ disagree in their ordering
  \[ Q = \text{number of discordant pairs} \]

**Loss function:** [Kemeny & Snell, 62], [Wong et al, 88], [Cohen et al, 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

\[
l(r_a, r_b) = Q
\]

**Example:**

\[
r_a = (a, c, d, b, e, f, g, h)
\]

\[
r_b = (a, b, c, d, e, f, g, h)
\]

=> discordant pairs $(c,b), (d,b)$ => \[ l(r_a, r_b) = 2 \]
A Loss Function for Rankings

For two orderings $r_a$ and $r_b$, a pair $d_i \neq d_j$ is

- **concordant**, if $r_a$ and $r_b$ agree in their ordering
  
  $P = \text{number of concordant pairs}$

- **discordant**, if $r_a$ and $r_b$ disagree in their ordering
  
  $Q = \text{number of discordant pairs}$

**Loss function**: [Kemeny & Snell, 62], [Wong et al, 88], [Cohen et al, 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

$$l(r_a, r_b) = Q$$

**Example:**

$$r_a = (a, c, d, b, e, f, g, h)$$

$$r_b = (a, b, c, d, e, f, g, h)$$

=> discordant pairs $(c, b), (d, b)$ => $l(r_a, r_b) = 2$
A Loss Function for Rankings

For two orderings $r_a$ and $r_b$, a pair $d_i \neq d_j$ is

- **concordant**, if $r_a$ and $r_b$ agree in their ordering
  
  $P = \text{number of concordant pairs}$

- **discordant**, if $r_a$ and $r_b$ disagree in their ordering
  
  $Q = \text{number of discordant pairs}$

**Loss function:** [Kemeny & Snell, 62], [Wong et al, 88], [Cohen et al, 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

$$l(r_a, r_b) = Q$$

**Example:**

$$r_a = (a, c, d, b, e, f, g, h)$$
$$r_b = (a, b, c, d, e, f, g, h)$$

=>$\text{discordant pairs } (c, b), (d, b) =>$  

$$l(r_a, r_b) = 2$$
What does the Retrieval Function Look Like?

Sort documents $d_i$ by their "retrieval status value" $rsv(q,d_i)$ with query $q$ [Fuhr, 89]:

$$rsv(q,d_i) = w_1 \cdot \#(\text{of query words in title of } d_i)$$

$$+ w_2 \cdot \#(\text{of query words in H1 headlines of } d_i)$$

$$\ldots$$

$$+ w_N \cdot \text{PageRank}(d_i)$$

$$= \vec{w} \cdot \Phi(q,d_i).$$

Select $F$ as:

$$d_i > d_j$$

$$\iff$$

$$(d_i, d_j) \in f_{\vec{w}}(q)$$

$$\iff$$

$$\vec{w} \Phi(q,d_i) > \vec{w} \Phi(q,d_j)$$
Experiment

Experiment Setup:
- meta-search engine (Google, MSNSearch, Altavista, Hotbot, Excite)
- approx. 20 users
- machine learning students and researchers from University of Dortmund AI Unit (Prof. Morik)
- asked to use system as any other search engine
- display title and URL of document

October 31\textsuperscript{st}  

collected training data  

$\Rightarrow$ 260 training queries  

(with at least one click)

November 20\textsuperscript{th}

trained Ranking SVM

December 2\textsuperscript{nd}

test ranking function  

$\Rightarrow$ 139 queries
Query/Document Match Features $\Phi(q,d)$

Rank in other search engine:
- Google, MSNSearch, Altavista, Hotbot, Excite

Query/Content Match:
- cosine between URL-words and query
- cosine between title-words and query
- query contains domain-name

Popularity-Attributes:
- length of URL in characters
- country code of URL
- domain of URL
- word "home" appears in title
- URL contains "tilde"
- URL as an atom
## Experiment: Learning vs. Google/MSNSearch

<table>
<thead>
<tr>
<th>Ranking A</th>
<th>Ranking B</th>
<th>A better</th>
<th>B better</th>
<th>Tie</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned</td>
<td>Google</td>
<td>29</td>
<td>13</td>
<td>27</td>
<td>69</td>
</tr>
<tr>
<td>Learned</td>
<td>MSNSearch</td>
<td>18</td>
<td>4</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>Learned</td>
<td>Toprank</td>
<td>21</td>
<td>9</td>
<td>11</td>
<td>41</td>
</tr>
</tbody>
</table>

~20 users, as of 2nd of December

**Toprank:** rank by increasing mimium rank over all 5 search engines

**Result:** Learned > Google  
Learned > MSNSearch  
Learned > Toprank
## Learned Weights

<table>
<thead>
<tr>
<th>weight</th>
<th>feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60</td>
<td>cosine between query and abstract</td>
</tr>
<tr>
<td>0.48</td>
<td>ranked in top 10 from Google</td>
</tr>
<tr>
<td>0.24</td>
<td>cosine between query and the words in the URL</td>
</tr>
<tr>
<td>0.24</td>
<td>document was ranked at rank 1 by exactly one of the 5 search engines</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>0.17</td>
<td>country code of URL is &quot;\text{.de}&quot;</td>
</tr>
<tr>
<td>0.16</td>
<td>ranked top 1 by HotBot</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>-0.15</td>
<td>country code of URL is &quot;\text{.fi}&quot;</td>
</tr>
<tr>
<td>-0.17</td>
<td>length of URL in characters</td>
</tr>
<tr>
<td>-0.32</td>
<td>not ranked in top 10 by any of the 5 search engines</td>
</tr>
<tr>
<td>-0.38</td>
<td>not ranked top 1 by any of the 5 search engines</td>
</tr>
</tbody>
</table>
Summary

Why and when is it good to use ML?
- humans are not really good at it (e.g. constructing classification rules)
- training data is cheap and plenty (e.g. clickthrough)
- no expert is available (e.g. rules for filtering my email)
- its just too expensive to do by hand (e.g. ArXiv classification, personal retrieval functions)

Further Info:
- Demo retrieval system for Cornell
  => Striver: http://www.cs.cornell.edu/~tj/striver
- CS478: Introduction to Machine Learning (Spring 03)
- CS678: Advanced Topics in Machine Learning (Spring 03)
- CS574: Language Technologies (currently)