A Tale of Two Curves

Data vs. Time

- Left: Data vs. Time graph with an increasing line indicating growth.
- Right: A static horizontal line indicating a constant value.
Scaling Machine Learning to the Web

Examples: Given the World Wide Web...
1. which nanotech companies are hiring on the West Coast?
2. get me documents about Russian politics.

Learn function $f$ over strings/docs $x$

$$f(x) = \begin{cases} 1 & \text{if } x \text{ relevant,} \\ 0 & \text{otherwise} \end{cases}$$

from examples $\{(x, f(x))\}$

Challenges: new $f$'s arise all the time
billions of $x$'s
Classical Supervised Learning

Learn function from $\mathbf{x} = (x_1, \ldots, x_d)$ to $y \in \{0, 1\}$
given labeled examples $\{(\mathbf{x}, y)\}$
Semi-Supervised Learning (SSL)

Learn function from $\mathbf{x} = (x_1, ..., x_d)$ to $y \in \{0, 1\}$ given labeled examples $\{(\mathbf{x}, y)\}$ and unlabeled examples $\{\mathbf{x}\}$

But existing SSL techniques scan all unlabeled data for each new concept!
Outline

1) Intro

2) Scaling Semi-supervised Learning
   A. Hidden Markov Models for Word Representations
   B. Semi-supervised Naïve Bayes with Feature Marginals

3) Applications
   A. Wikitables
   B. Atlasify

4) Conclusion
Throw(person, x) ?

Weight(x) < 50lbs ^
Max_dim(x) < 20ft ^ ...
=> Throw(person, x)

Weight(baseball) = 5oz ^ .... =>
Throw(person, baseball)
"throwable objects such as"

About 5,050 results (0.19 seconds)

**Patent US5984812 - Grippable surface for throwable object - Google ...**
www.google.com/patents/US5984812
This invention relates to a grippable surface for **throwable objects such as** a football, baseball, etc. which enhances the ease with which the object may be ...

**[PDF] Name Juggle.pdf - GOAL Consulting**
Materials: Many soft **throwable objects such as** fleece balls, wadded up pieces of paper, Nerf™ balls. Level: Grades K and higher. Suggested Procedure. 1.
Cities such as X, Y, mayor of X

• The Web makes hard AI problems easier
Understanding Language is Hard

| Michael Bloomberg, mayor of New York City, was born in ... |
| Eric Johnson, CEO of Texas Instruments, mayor of Dallas from 1964-1971, and ... |
Redundancy enables Information Extraction

Michael Bloomberg, mayor of New York City, was born in...

Announced by Michael Bloomberg, the mayor of New York City...

Michael Bloomberg, mayor of New York City, stated that...

...+10,000s more

Eric Johnson, CEO of Texas Instruments, mayor of Dallas from 1964-1971, and...

(only once)

<table>
<thead>
<tr>
<th>Mayor</th>
<th>City</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Bloomberg</td>
<td>New York City</td>
<td>0.99</td>
</tr>
<tr>
<td>Texas Instruments</td>
<td>Dallas</td>
<td>0.05</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

[Downey et al., AIJ 2010]
Challenge: the “long tail”

A mixture of correct and incorrect
Our Information Extraction Task

• Given:
  – A handful of examples of a concept
    • (mayor/city pairs, throwable objects, etc.)
  – The Web

• Determine:
  – (all) other examples of the concept
Scaling Information Extraction

Cities such as X

Y, mayor of X

• The Web makes hard AI problems easier...but:
  – New concepts of interest arise all the time
  – Available data is enormous
  – Can’t use standard SSL

• Idea: **pre-process** the Web to make SSL fast
  – Search engine is a simple example
  – ...but not sufficient
The Distributional Hypothesis

- Terms with similar meanings tend to appear in similar contexts.

<table>
<thead>
<tr>
<th>Context</th>
<th>Hits with Chicago</th>
<th>Hits with Twisp</th>
</tr>
</thead>
<tbody>
<tr>
<td>“cities including __”</td>
<td>42,000</td>
<td>1</td>
</tr>
<tr>
<td>“__ and other cities”</td>
<td>37,900</td>
<td>0</td>
</tr>
<tr>
<td>“__ hotels”</td>
<td>2,000,000</td>
<td>1,670</td>
</tr>
<tr>
<td>“mayor of __”</td>
<td>657,000</td>
<td>82</td>
</tr>
</tbody>
</table>
Baseline: context vectors

... cities such as Chicago, Boston, Los Angeles and Chicago.
... But Chicago isn’t the best.

- Represent x’s as context vectors [cf. Ravichandran et al. 2005]
  - Problems: CV’s are large (inefficient) & sparse (inaccurate)
Hidden Markov Model (HMM)

Hidden States $t_i \in \{1, \ldots, N\}$  
(N fairly small)

Train on unlabeled text
- $P(t_i \mid w_i = w)$ is $N$-dim. representation of $w$
- Compare extractions using KL divergence
HMM Compresses Context Vectors

Twisp:  <

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>. &gt;</th>
</tr>
</thead>
</table>

\[ P(t \mid \text{Twisp}): \]

\[
\begin{array}{cccc}
  t=1 & 2 & N \\
  0.14 & 0.01 & \ldots & 0.06 \\
\end{array}
\]

Representation \( P(t \mid w) \)

- Compact (efficient – **10-50x** less data retrieved)
- Dense (accurate – **40%** accuracy improvement)
## Results

<table>
<thead>
<tr>
<th>Word Representation</th>
<th>Extraction Performance (AUC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 HMM ensemble</td>
<td>0.18</td>
</tr>
<tr>
<td>Single HMM</td>
<td>0.17</td>
</tr>
<tr>
<td><strong>Brown Clusters (1000)</strong></td>
<td><strong>0.18</strong></td>
</tr>
<tr>
<td>Brown Clusters (3200)</td>
<td>0.16</td>
</tr>
<tr>
<td>Brown Clusters (320)</td>
<td>0.15</td>
</tr>
<tr>
<td>Brown Clusters (100)</td>
<td>0.13</td>
</tr>
<tr>
<td>N-gram Context Vectors</td>
<td>0.10</td>
</tr>
<tr>
<td>Random Baseline</td>
<td>0.10</td>
</tr>
</tbody>
</table>

[Huang et al., Comp. Ling. 2013]
HMMs Handle Sparsity

<table>
<thead>
<tr>
<th>Word Representation</th>
<th>Extraction Performance (AUC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sparse</td>
</tr>
<tr>
<td>HMM</td>
<td>0.15</td>
</tr>
<tr>
<td>N-gram Context Vectors</td>
<td>0.06</td>
</tr>
</tbody>
</table>
IE performance correlation:  
- model ppl (-0.88)  
- corpus size (0.71)  
- # states $N$ (0.38)
How to Train Large HMMs

• Computational Clusters

• HMM training is “Embarrassingly Parallel”
  – Partition corpus across nodes
  – Memory Cost?
    • Corpus on disk, parameters in memory
    • $P(\text{word} \mid \text{state})$ has size $\#\text{Words} \times N$ (large!)
Idea: Partition Intelligently

• Typical approach: partition corpus randomly

• Our idea: Partition to minimize number of unique words on each node [Yang et al., NAACL-HLT 2013]
  – Minimize #params of \( P(\text{word} \mid \text{state}) \) on node
  – NP-hard optimization problem
    • \( \min(\max(\text{set of submodular functions})) \)
  – \textbf{BJAC:} Greedy approach. Iteratively:
    • choose doc approx. least similar to already placed docs
    • place doc on most similar node (Jaccard)
Intelligent partitioning with BJac reduces memory cost by 50% vs. random. Parallel w/ 100 nodes is 10x more space efficient than single node.
Outline

1) Intro

2) Scaling Semi-supervised Learning
   A. Hidden Markov Models for Word Representations
   B. Semi-supervised Naïve Bayes with Feature Marginals

3) Applications
   A. Wikitables
   B. Atlasify

4) Conclusion
Scaling Semi-supervised Naïve Bayes

• Key learning task: Estimate $P(x_i | y)$
  – Normally by counting, e.g.:

$$P(x_i = 1 | y = 0) = \frac{\#(\text{docs with } x_i = 1 \text{ and } y = 0)}{\#(\text{docs with } y = 0)}$$
Use unlabeled $P(x_i)$ as *constraint*

- $P(x_i) = P(x_i \mid y=1) P(y=1) + P(x_i \mid y=0) P(y=0)$

- E.g. $P(x_i=1) = \frac{10 \text{ million}}{1 \text{ billion}} \approx \frac{1}{500} \cdot \frac{4}{40} + \frac{5}{2000} \cdot \frac{36}{40}$

Labeled data says $x_i=1$ indicator that $y=0$
Unlabeled data reveals that opposite is true! (probably)
Results

- F1, 100 labeled documents on RCV1 doc classification [Lucas & Downey, ACL 2013]

<table>
<thead>
<tr>
<th>Class</th>
<th>MNB-FM</th>
<th>SFE</th>
<th>MNB</th>
<th>NBEM</th>
<th>Logist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCAT</td>
<td>0.797</td>
<td>0.793</td>
<td>0.624</td>
<td>0.713</td>
<td>0.754</td>
</tr>
<tr>
<td>GCAT</td>
<td>0.849</td>
<td>0.848</td>
<td>0.731</td>
<td>0.837</td>
<td>0.831</td>
</tr>
<tr>
<td>MCAT</td>
<td>0.776</td>
<td>0.737</td>
<td>0.313</td>
<td>0.516</td>
<td>0.689</td>
</tr>
<tr>
<td>ECAT</td>
<td>0.463</td>
<td>0.317</td>
<td>0.017</td>
<td>0.193</td>
<td>0.203</td>
</tr>
<tr>
<td>GPOL</td>
<td>0.499</td>
<td>0.370</td>
<td>0.002</td>
<td>0.089</td>
<td>0.114</td>
</tr>
<tr>
<td>Average</td>
<td>0.677</td>
<td>0.613</td>
<td>0.337</td>
<td>0.470</td>
<td>0.518</td>
</tr>
</tbody>
</table>
Outline

1) Intro

2) Scaling Semi-supervised Learning
   A. Hidden Markov Models for Word Representations
   B. Semi-supervised Naïve Bayes with Feature Marginals

3) Applications
   A. Wikitables
   B. Atlasify

4) Conclusion
Table Search Results

- WikiTable’s learned ranker outperforms Google Fusion Tables (\texttt{TABLE}):  

<table>
<thead>
<tr>
<th>Method</th>
<th>All Ratings</th>
<th>Ratings by Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>(a)</td>
</tr>
<tr>
<td>\texttt{WikiTables}</td>
<td>500</td>
<td>54</td>
</tr>
<tr>
<td>\texttt{TABLE}</td>
<td>175</td>
<td>69</td>
</tr>
<tr>
<td>\texttt{DOCUMENT}</td>
<td>399</td>
<td>24</td>
</tr>
<tr>
<td>\texttt{GOOG}</td>
<td>493</td>
<td>63</td>
</tr>
<tr>
<td>\texttt{GOOGR}</td>
<td>156</td>
<td>43</td>
</tr>
</tbody>
</table>
Semantic Relatedness

• Given two Wikipedia concepts, how “related” are they?
  – SR(Napa Valley, Wine) = High
  – SP(Napa Valley, Monster Trucks) = Low

• Can be estimated well from Wikipedia links, text, categories, ...  [Hecht et al., SIGIR 2012]
Adding Relevant Columns

<table>
<thead>
<tr>
<th>Model</th>
<th>columns added</th>
<th>accuracy</th>
<th>DCG@4</th>
<th>nDCG'@4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>423</td>
<td>0.29</td>
<td>11.38</td>
<td>0.43</td>
</tr>
<tr>
<td>WikiTables-SR</td>
<td>414</td>
<td>0.43</td>
<td>13.71</td>
<td>0.48</td>
</tr>
<tr>
<td>WikiTables</td>
<td>410</td>
<td>0.62</td>
<td>30.71</td>
<td>0.76</td>
</tr>
</tbody>
</table>

WikiTables doubles accuracy over baselines
Semantic Relatedness (SR) provides 58% of the increase
Outline

1) Intro

2) Scaling Semi-supervised Learning
   A. Hidden Markov Models for Word Representations
   B. Semi-supervised Naïve Bayes with Feature Marginals

3) Applications
   A. Wikitables
   B. Atlasify

4) Conclusion
Median Family Income
by U.S. state

Data source:
Department of Housing and Urban Development
Nuclear Weapon
Thematic Maps | Make Reference System

Concept: Nuclear_weapon  Language: English  Atlas of: The Earth

Execute!  Options

Map of Africa with various locations labeled, including South_Africa and weapons_of_mass_destruction.
South Africa links to Nuclear weapon

SOUTH AFRICA: 20TH CENTURY:

In the late 1970s, South Africa began a programme of nuclear weapons development.

South Africa links to Nuclear weapon

SOUTH AFRICA: FOREIGN RELATIONS AND MILITARY:

South Africa undertook a nuclear weapons programme in the 1970s and may have conducted a nuclear test over the Atlantic in
Nuclear Weapon
Cobalt bomb discusses both Nuclear weapon and Cobalt

COBALT BOMB:

A cobalt bomb is a theoretical type of "salted bomb": a nuclear weapon intended to contaminate an area by radioactive material, with relatively little blast.

-------------

Cobalt bomb discusses both Nuclear weapon and Cobalt

COBALT BOMB:
Ad-hoc Reference Systems

Country music

Rock music

Hip hop music
Ad-hoc Reference Systems
Ad-hoc Reference Systems

Tim McGraw
Ad-hoc Reference Systems

Tim McGraw

Country music

Jay-Z

Rock music

Hip hop music
“There is a huge mistake here” – Bryan Pardo (musician)
Conclusions

• The Web makes AI problems easier, but:
  – It’s huge
  – Concepts of interest are not known in advance

• Key task: pre-process Web to make SSL fast
  – New SSL approaches
    • HMMs for Word Representations
    • MNB with Feature Marginals
  – Applications
    • Wikitables
    • Atlasify