Transduction - co-training

- Proposed by Blum and Mitchell (1998)

- Powerful idea rooted in cognitive science/pedagogy:

- Train two learning algorithms with a labeled sample using **two independent and sufficient views** of features (“two students”).

- Use each learning result to predict unlabeled examples and expand training set for the other (“students learn from each other”).

Two views in co-training

- Co-training assumption: two independent and sufficient views

- Example: the task of classification for CS faculty web pages

  - View 1: Page text. Words in home pages, such as research interests, teaching courses, etc.
  
  - View 2: Hyperlink text. Words in hyperlinks that point to that page.
Related work

- (Balcan et al. 2005) relaxed the assumption of two independent and sufficient views to an expansion of underlying data distribution.
- (Nigam & Ghani, 2000) showed dependent views may lead to successful co-training, but inferior to independent views.
- (Wang & Zhou, 2007) showed two arbitrary classifiers with large difference can be successful in co-training.

Transduction: co-train and co-test

Transcription:

(a) Distribution of all unlabeled instances classified by two view classifiers
(b) Random sampling
(c) Co-testing sampling method
(d) Our new sampling method
Illustration of co-training process

Co-training algorithm

Given:
- $L$: Set of labeled training examples
- $U$: set of unlabeled examples
- $V_1, V_2$: two views on data
- $h1, h2$: two classifiers trained on two views of $L$
- $n$: the number of unlabeled examples initially sampled into $U'$ from $U$
- $p$: the number of positive examples labeled and selected by $h1$ or $h2$ from $U'$
- $n$: the number of negative examples labeled and selected by $h1$ or $h2$ from $U'$

Create a pool $U'$ of examples by choosing $n$ examples at random from $U$

while $iteration\, number < k$ do

Use $L$ to train two classifiers $h1$ and $h2$ from $V1$ and $V2$

Use $h1$ to label all instances in $U'$ and select $p$ positive and $n$ negative most confidently predicted instances from $U'$

Use $h2$ to label all instances in $U'$ and select $p$ positive and $n$ negative most confidently predicted instances from $U'$

$U' = U' - \{2p + 2n \text{ examples selected by } h1 \text{ and } h2\}$

$L = L + \{2p + 2n \text{ instances selected by } h1 \text{ and } h2\}$

Randomly choose $2p + 2n$ examples from $U$ to replenish $U'$

endwhile
Sampling in co-training

- Random sampling is used in co-training to select unlabeled instances from $U$ to $U'$

- Since random sampling was shown not perform well in active learning, it may not lead to optimal performance for co-training

Using confidence-based sampling to improve co-training

- Using multiple $U'$ instead of a single $U'$
- Train a selector classifier on labeled set $L$ to classify and rank all unlabeled instances in $U$
- Sampling unlabeled instances from $U$ to multiple $U$'s according to ranking in $U$
- Each $U'$ can expand one labeled set $L$
- Selecting the $U'$ that may potentially lead to the best expanded $L$
Illustration of new co-training process

- $U$ is the unlabeled set.
- $L$ is the labeled set.
- Selector selects samples from $U$.
- $U'(1), U'(2), \ldots, U'(m)$ are subsets of $U$.
- Co-training is used to expand $L$.
- $L(1), L(2), \ldots, L(m)$ are the labeled sets.
- Find the best $L(t)$.

Empirical study: comparing original and new co-training methods

- Experiment setting:
  - Using web page classification task in (Blum & Mitchell, 1998) and six UCI binary datasets.
  - Using same learning algorithm for two view classifiers $h_1$ and $h_2$, and selector classifier $h_3$.
  - Using SVM and random forest learning algorithms.
  - Using a modeling classifier $h_4$ trained on expanded labeled set to evaluate co-training performance.
Views used in experiments

- For each dataset, all attributes are split into two sets of attributes as a pair of views.
- The number of pairs of views is exponential to the number of attributes.
- Since it is difficult to find good views, for each dataset 200 pairs of views are randomly chosen.
- The co-training results on 200 pairs of views are averaged.

Experiment:

- Using the same learning algorithm for modeling classifier and view classifiers.

<table>
<thead>
<tr>
<th>Dataset/Error rates</th>
<th>Original Co-training</th>
<th>Improved Co-training</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h1=h2=h3=SVM</td>
<td>h1=h2=h3=h4=SVM</td>
</tr>
<tr>
<td>web page</td>
<td>0.06±0.01</td>
<td>0.06±0.011</td>
</tr>
<tr>
<td>breast-c</td>
<td>0.36±0.021</td>
<td>0.34±0.014</td>
</tr>
<tr>
<td>credit-a</td>
<td>0.22±0.032</td>
<td><strong>0.20±0.022</strong></td>
</tr>
<tr>
<td>colic</td>
<td>0.35±0.037</td>
<td><strong>0.29±0.03</strong></td>
</tr>
<tr>
<td>diabetes</td>
<td>0.28±0.023</td>
<td>0.29±0.023</td>
</tr>
<tr>
<td>sick</td>
<td>0.08±0.009</td>
<td>0.08±0.008</td>
</tr>
<tr>
<td>vote</td>
<td>0.13±0.028</td>
<td><strong>0.11±0.025</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset/Error rates</th>
<th>Original Co-training</th>
<th>Improved Co-training</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h1=h2=h3=Random forest</td>
<td>h1=h2=h3=h4=Random forest</td>
</tr>
<tr>
<td>web page</td>
<td>0.07±0.01</td>
<td><strong>0.08±0.011</strong></td>
</tr>
<tr>
<td>breast-c</td>
<td>0.35±0.024</td>
<td>0.34±0.019</td>
</tr>
<tr>
<td>credit-a</td>
<td>0.21±0.017</td>
<td><strong>0.23±0.017</strong></td>
</tr>
<tr>
<td>colic</td>
<td>0.28±0.032</td>
<td><strong>0.23±0.03</strong></td>
</tr>
<tr>
<td>diabetes</td>
<td>0.27±0.018</td>
<td><strong>0.25±0.014</strong></td>
</tr>
<tr>
<td>sick</td>
<td>0.08±0.006</td>
<td><strong>0.001±0.006</strong></td>
</tr>
<tr>
<td>vote</td>
<td>0.13±0.024</td>
<td><strong>0.10±0.018</strong></td>
</tr>
</tbody>
</table>
Discussion

- For SVM, new method is significantly better in 2 of 7 datasets
- For random forest, new method is significantly better 3 of 7 datasets
- There is no case that new method is worse.

Second experiment:
Using decision tree algorithm for modeling
Discussion

- In 6 of 7 cases both the original and new method significantly improve the initial model performance.
- For SVM, new method is significantly better in 3 of 7 datasets.
- For random forest, new method is significantly better 2 of 7 datasets.
- There is no case that new method is worse.

Conclusion

- Co-training is effective in improving classifier performance for limited labeled dataset.
- Random sampling in original co-training with may not lead to optimal performance.
- Co-training with confidence based sampling can significantly improve original co-training method.
Active learning

- The transductive setting
- The learner **chooses** the best unlabeled instances to be labeled
- The best performance: Query By Committee:
  - A committee of classifiers
  - Classify unlabeled instances
  - Choose the instances on which there is most disagreement among the committee

Sampling techniques in active learning

- Some sampling approaches in active learning to improve over random sampling
  - (Tong & Koller, 2000) used a sampling strategy by minimizing the version space.
  - (Freund et al., 1997) sampled unlabeled instances that member classifiers of a committee disagree most: QBC
  - (Saar-Tsechansky & Provost, 2001) sampled unlabeled instances according to variances of probability estimated by multiple models.
Can active learning sampling techniques be applied to co-training?

- The answer is “No”.
  - Co-training is a passive learning process.
  - Active learning usually selects most unconfidently predicted instances, which are very likely to be misclassified in co-training.
- We have to design new sampling method for co-training.

Classifying medical abstracts

- Evidence-based medicine:
  “is the conscientious, explicit and judicious use of current best evidence in making decisions about the care of individual patients”
- Systematic Reviews
  the foundation of evidence-based medicine is the systematic review of evidence for particular treatments, mainly randomized controlled trials
Requirements

- 100% recall (!)
  - Is this realistic?
  - Does the human-manual system achieve this?
  - Perhaps recall at least as good as as good as human (90-95%)
- …and “decent” precision (at least as good as as good as human – 80-85%)
- Output a score, so that abstracts can be sorted by their relevance - ranking

Our initial solution

Initial database

SRS-Web interface

Relevant docs

Non-relevant docs

Compare and correct

Current result

SRS update on future data

Data Mining (ATC) subsystem

EMISAFR

user

SQL query

Medline
Performance evaluation

- Recall and precision are of interest, but the key measure is work savings.
- What are the objectives?
  - to minimize the number of relevant documents excluded by the classifier.
  - to reduce the reviewers' workload (i.e., to minimize the number of documents selected by the classifier to the next level).
- How is ML used:
  1. Train a high-recall classifier C
  2. Run C
  3. Comb the ‘included’ class for irrelevants

How to evaluate performance?

- Work Saved Over Savings [Cohen et al. 06]
- the work saved by using the classifier is greater than the work needed in simple random sampling.
- run a classifier, check the excluded (N is the excluded, P is the included class)
- WSS = (TN + FN) / N - (1 - R)
First tack

- Use a standard BOW representation
  - Experience with MESH, pubtype
  - Apply a classifier likely to do well with imbalance, and tuned to obtaining a high recall: FCNB, but outputs a score

\[
I_{FCNB}(d) = \arg \max_c \left[ \log p(\theta) - F_c \sum_i f_{i,d} \log \frac{N_{c^i} + \alpha}{N^c + \alpha} \right]
\]

\[F_c\] factor applied for class \(C\)

\(N_{c^i}\) = the number of times feature \(i\) occurred in documents of classes other than \(c\),
\(N_c\) = the total number of feature occurrences in classes other than \(c\).

saving = \(\frac{TN}{TN+FP}\) Excluded (Negative) Class Recall

risk = \(\frac{FN}{TP+FN}\) False Negative Rate

![Saving and Risk Graph](image-url)

Sampling: 50% split: 30/70
Issues

- Unsatisfactory results
  - Can’t have both recall and precision at human level

Version 2

- Stronger linguistic representation:
  - Nounphrases (NPs) extracted (GENIA) from the included abstracts and ranked according to 5 ranking techniques USEFUL TO IMPROVE QUERY
  - TF-IDF: no advantage
  - BOW + frequency is the representation used in the next versions

-
Version 3

- More Classifiers (WEKA, slightly modified)
  - Factorized Discriminative Multinomial Naive Bayes (FDMNB).
  - Factorized Alternating Decision Trees
  - Regression Tree
  - Factorized AdaBoost (logistic regression and J48)

- Voting schemes (vote is a number of classifiers that predicted the article to pass, classifier-specific factor values applied consecutively; different voting schemes)

- Active Learning schemes

Classifiers used

- Factorized Discriminative Multinomial Naive Bayes (FDMNB).
  - recently developed [ICML 2008] modification of MNB
  - combines advantages of Generative and Discriminative approaches.
  - First real life data text categorization task implementation of this method. We have modified it by factorization
Classifiers used cont’d

- Factorized Alternating Decision Trees –
  - generalization of decision trees similar to boosting. We have modified it by factorization

- Regression Tree –
  - a variant of decision trees with continuous outcome

Co-occurrence based representation

Text → Correlation Matrix → Stop Word Removal
  Proper Segmentation
  Segment Vector
  Classification ← Attribute Selection ← Abstract Vector
**Word co-occurrence**

- **Association co-occurrence** – based on the co-occurrence of words inside a sentence (frequency).
- **Distance co-occurrence** – based on the co-occurrence of words considering the distance between words inside a sentence.

**Segmented based words Closeness**

- Different weights based on different positions of a pair of words! (distance; Is there a , or ” or ; or /n in between?)
- Undirected
Word–similarity measures across the whole abstract

\[
\frac{2\left(w_1 df_{1,xy} + w_2 df_{2,xy} + \ldots + w_m df_{m,xy}\right)}{df_x + df_y}
\]

\[
df_x = df_{1,x} + df_{2,x} + \ldots + df_{m,x}
\]

\(df_{1,xy}\) : a "configuration" of co-occurrence of x and y

Closeness Word Vector

\[
W_{\text{seventies}} = \begin{bmatrix}
\text{seventies} \\
W_1 \\
\ldots \\
\text{Cholesterol} & 0.4912 \\
\text{Diabetic} & 0.6836 \\
\text{Arthritis} & 0.5119 \\
\text{Cardio-} & 0.4263 \\
\text{Chronic} & 0.7823 \\
\ldots \\
W_{4618} & \\
\ldots
\end{bmatrix}
\]
**Sentence Characteristic Vector**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>It</th>
<th>Has</th>
<th>Been</th>
<th>Reported</th>
<th>In</th>
<th>Some</th>
<th>Serious</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_1 )</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( W_{4618} )</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

\[
S_1 = \frac{W_{\text{It}} + W_{\text{has}} + W_{\text{been}} + W_{\text{reported}} + \ldots + W_{\text{cases}}}{8}
\]

Number of words in sentence.

**Abstract Characteristic Vector**

<table>
<thead>
<tr>
<th>Abstract</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>It has been reported in some serious cases.</td>
<td></td>
</tr>
<tr>
<td>It needs to be followed.</td>
<td>S2</td>
</tr>
<tr>
<td>Drowsiness could be one of the side affects.</td>
<td>S3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Specially in elderly people dosage of the drug should be monitored.</td>
<td>Sn</td>
</tr>
</tbody>
</table>

\[
A_i = \frac{S_1 + S_2 + S_3 + \ldots + S_n}{n}
\]

Number of sentences in the abstract.
Word/Sentence Correlation (Dimensions Range)

A sentence vector consisting of its words vectors

An abstract vector consisting of its sentences vectors

Illustrative abstract vector (Highest Smoothing Parameter)

The white area is depicting the calculated abstract vectors (Application in Sentiment/Concept Analysis)
Illustrative abstract vector
(Lowest Smoothing Parameter)

The white area is depicting the calculated abstract vectors
(Application in Scientific articles/papers)

Axis X: Corpus feature space (Words)
Axis Y: Relevancy

Sentences:
S1, S2, S3, S4, S5, S6

Issues with Version 3

- Results still unsatisfactory
- On a dataset with 9.6% imbalance, recall close to 100%
  - Precision 12% in Version 1
  - Precision 15% in Version 3
- Settings: factors, voting scheme – how to choose?
- Is it possible to have high recall AND precision just for part of the data?
Version 4

- Focus the classifier [committee] on zones of high confidence in the prediction
- Factors are dropped

Version 4

- Three areas of interest:
  - Positive (top ranked)
  - Negative (bottom ranked)
  - Grey zone (not sure, need to be re-categorized, automatically or manually)
Grey Zone – Prediction Zone Trade Off

WHOLE DATA 100%

Test Set 70%

Training Set 30% Labeled by Human

"Grey Zone" 35%

Prediction Zone

- Negative Prediction 33%
- Positive Prediction 2%

Ranking Algorithm. Local Ranks

Local Ranking:

\[ R_j(W_{ij}^+, W_{ij}^-) \rightarrow S_{ij} \]

\( R_j \): Ranking Rule for classifier \( j \)

\( W_{ij}^+ \): Decision weight, assigned to instance \( i \) by classifier \( j \) with respect to positive class

\( W_{ij}^- \): Decision weight, assigned to instance \( i \) by classifier \( j \) with respect to negative class

\( S_{ij} \): Local Ranking Score for instance \( i \) and classifier \( j \) wrt to a class

\[ S_{ij} \rightarrow L_{ij} \]

\( L_{ij} \): Local Rank (number in ordered list) for instance \( i \) and classifier \( j \)

\[ L_{ij} \in \{1, 2, \ldots, N\} \], \( N \) – number of instances to be classified
# Ranking Algorithm. Global Ranks

Global Ranking:

\[ Gi = \sum_{j} L_{ij} \]

**Global Score for instance** \( i \)

\[ Gi \rightarrow Xi \]

**Global Rank for instance** \( i \), \( Xi \in \{1, 2, \ldots, N\} \)

---

# Global Classification Rule

**Global Classification:** \( K'(T, B, Xi) \rightarrow C'i \)

<table>
<thead>
<tr>
<th>( K' )</th>
<th>Global Classification Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C'i )</td>
<td>Global Class Prediction</td>
</tr>
<tr>
<td>for instance ( i )</td>
<td></td>
</tr>
<tr>
<td>( B )</td>
<td>Bottom Threshold, ( B &lt; N )</td>
</tr>
<tr>
<td>( T )</td>
<td>Top threshold, ( 0 &lt; T &lt; B )</td>
</tr>
</tbody>
</table>

- \( (Xi \geq T) \Rightarrow i \in Z^+, \quad C'i = 1 \)
- \( (Xi \leq B) \Rightarrow i \in Z^-, \quad C'i = 0 \)
- \( (B < Xi < T) \Rightarrow i \in Z^N \)
What Should be the Prediction Zone Thresholds

$T'$ and $B'$ - thresholds established on New Prediction Zone with respect to required level of confidence
Treat the training set as validation set for this

$M = \text{"new training set size"}/\text{"original training set size"}$

$f(k_1, M, B), f(k_2, M, T)$. Simple case: $f(k_1M,B)=B' = k_1*M*B$
Determine params with a validation set

---

Experiment settings:

Number of instances:

Whole Data Set: 23324, split: 30% / 70%
  - Training set: 7000
    - Positive class: 626
    - Negative class: 6374
  - Test set: 16334
    - Positive class: 1461
    - Negative class: 14873

Classifiers used in committees: (2*5)
  - Complement Naive Bayes (CNB)
  - Discriminative Multinomial Naive Bayes (DMNB).
  - Alternating Decision Trees (AT)
  - AdaBoost logistic regression
  - AdaBoost J48

Data Representation methods:
  - BOW and Co-occurrence based method (Co-oc) both are in use
  - Each classifier run twice: first run on BOW, second run on Co-oc
### Results

**Prediction Zone:** 8700 (37.3% over whole data)

<table>
<thead>
<tr>
<th></th>
<th>Aria</th>
<th>Threshold</th>
<th>Correctly classified</th>
<th>Incorrectly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>700</td>
<td>590 (TP)</td>
<td></td>
<td>110 (FP)</td>
</tr>
<tr>
<td>Bottom</td>
<td>8000</td>
<td>7951 (TN)</td>
<td></td>
<td>49 (FN)</td>
</tr>
</tbody>
</table>

**Table 2. ML performance vs. Human Performance**

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>ML results on the Prediction Zone</th>
<th>Average Human Reviewer results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall on positive class:</td>
<td>92.3%</td>
<td>90-95%</td>
</tr>
<tr>
<td>Precision on positive class:</td>
<td>84.3%</td>
<td>80-85%</td>
</tr>
</tbody>
</table>

**Trade-Off Between Prediction Zone Size and Prediction Confidence**

**Reduce Prediction Zone:**
- Prediction Zone: 5000 (21.4% over whole data)

<table>
<thead>
<tr>
<th>zone</th>
<th>Threshold</th>
<th>Correctly classified</th>
<th>Incorrectly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>500</td>
<td>441 (TP)</td>
<td>59 (FP)</td>
</tr>
<tr>
<td>Bottom</td>
<td>4500</td>
<td>4388 (TN)</td>
<td>12 (FN)</td>
</tr>
</tbody>
</table>

**Increase Prediction Confidence:**
- Recall on positive class: 97.3%
- Precision on positive class: 88.3%
**Impact of each classifier on committee results**

(cont.)

Experiment settings:
8 members committees (4 on BOW + 4 on Co-oc) - excluding one classifier from the previous setup
Prediction Zone: 5500

Table 4. Ranking performance with respect to the classifier to be excluded

<table>
<thead>
<tr>
<th>Classifier to be excluded</th>
<th>FN</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Classifier to be excluded</td>
<td>17</td>
<td>49</td>
</tr>
<tr>
<td>Ada Boost regression</td>
<td>17</td>
<td>55</td>
</tr>
<tr>
<td>CNB</td>
<td>20</td>
<td>62</td>
</tr>
<tr>
<td>DMNB</td>
<td>21</td>
<td>55</td>
</tr>
<tr>
<td>AT</td>
<td>22</td>
<td>60</td>
</tr>
<tr>
<td>Ada Boost J48</td>
<td>26</td>
<td>72</td>
</tr>
</tbody>
</table>

5 classifiers team work better than any 4-classifier team

---

**Synergetic Effect**

Data representation methods work better together than alone

Experiment settings: all classifiers are included,
Prediction Zone: 8700 (Top:700, Bottom: 8000)

- **BOW + Co-oc**
  - **FP**: 110
  - **FN**: 49

- **BOW**
  - **FP**: 157
  - **FN**: 95

- **Co-oc**
  - **FP**: 141
  - **FN**: 84
Lessons learned

- Not really a standard [topic discrimination] ATC problem
- Imbalanced data
- Noise in the data
- Reviews are often heterogeneous
- Classifier committee does well
- Some advantage of a “global” representation
Conclusion

- Classifiers committee and Global Ranking approach give acceptable performance at the tails of the distribution.

- Product prototype has been deployed on the industry partner side and demonstrates possibility to reduce workload more than 30%.

Future Work

- Incremental Training set and Grey Zone Reduction (multi-iterations approach to reduce Grey Zone; active learning)

- Increasing Performance by selection optimal classifiers committee
Roadmap:
- Machine Learning (ML) in image processing (IP): bird’s eyes view
- Past and current challenges
- ML 2.0 in IP: some dimensions
- A success story and lessons learned
- When there is insufficient labelled data
- Beyond the horizon: neuroscience...
- Questions...

ML in CV/PR/IA: bird’s eyes view

- TASK
- Sensed signal
- Internal representation
- Knowledge needed for the task
- ML
Historically...

- ML has provided **useful tools** for IP; it usually involves:
  - Feature selection
  - Segmentation (classification?)
  - Registration (classification?)

Challenges

- Representation (e.g. features good for registration might not work well for the task)
- Data challenges:
  - Stream data (video)
  - Noisy data
  - Sheer volume of data
  - Insufficient labeled data
  - ...

More challenges...

- Need for layered representations, taking into account
  - Relationships between parts of images
  - Knowledge about these parts

ML and ML 2.0

- "standard" ML:
  - Data fully labeled
  - Data reasonably balanced
  - Prior feature selection
  - Attribute-vector representation
  - Evaluated on accuracy/R/P
- "Deep" learning
- Transductive learning
- From search to optimization
- Relational learning
- Ensemble learning without feature selection
- Imbalanced data
- Interpretability of results
ML 2.0 goes some way towards meeting the challenges
More work needed, especially at the „symbolic” and „relational” level
Some success stories:
- Features are layered and learned
- Model is interpretable

Some successes and ML 2.0
- Viola, Jones 2001: very fast face detection
- „smart” feature engineering: discriminating rectangles
For 384 x 288 images and 24 x 24 detector, there are 180,000 possible rectangular features

No feature selection:
- integral image \( \rightarrow \) rectangular features in constant time;
- useful for image normalization

Adaboost on [single feature classifiers + threshold] acts as
- feature selection AND
- as a preliminary classifier with high recall and low precision

200 selected features give 95% recall with very high precision; features are interpretable

Recall not sufficient
To improve recall classifiers are cascaded
Viola, Jones cont’d

- Result wrt state of the art
  - 15 times faster
  - recall rates equal or better, precision 90%
- No feature selection; starts with a large number of simple features
  - See also [Agrawal Triggs 2006]
- Sophisticated feature engineering
  (generalizable, e.g. to learning from text?)
- Ensemble learning, tuned to the imbalanced data set

Viola, Jones lessons learned

- Informed feature engineering - No feature selection
- Method tuned to basic characteristics of the data (imbalance)
- Has “cognitive plausability”
- Search → optimization
  - # of stages
  - # of features in stage
  - threshold
Some successes cont’d

- similar ideas have been proposed in ML as well:
  - [Fereira & Gama 2007] show how association rules can “learn” what are the good attributes, and then combine simple classifiers in a cascade
  - [Kramer & DeRaedt 2001] use relational queries with frequency constraints. Solutions to the query → version space → attributes for ‘standard’ classifiers

Labaled data - availability

- Joachim’s Transductive Induction [1999] for text data.
- Expectation Maximization [EM]: assumes data is a mix of two Gaussians;
  - soft-assign each instance to each model [E],
  - maximize the log-likelihood fit [M]
Where’s the future in IP?

- Inspiration from Cognitive Neuroscience?
- Treisman: only elementary features processed in parallel
- Vision process focused by Boolean maps

Recent “breakthrough” in face recognition [Yi Ma, UIUC, CACM 2009]

- training data = large dictionary for representing test images. A given image is represented as a linear combination of images in a database:
- the algorithm uses the fewest images to interpret a test image. For each test image, the algorithm seeks the sparsest representation from the main dictionary
- Handles occlusions and noise very well
- Similar to CBR with boosting
- Performance beats human
Achieving visual attention [Huang, Pasher Psychology Review 2007]

- Boolean map: a spatial representation that focuses on a single feature: is it present or absent? If both objects are selected, they become undistinguishable.
- Only one feature can be associated with a map.

Only One Feature Value Can Be Accessed at One Instant

Which display is easier to detect differences between left and right?

Colour asymmetries are detected faster in ABCD because they involve 4 colours (2 in ABBA).
Important questions

- What are the lessons learned from successes (and failures) of ML in IP?
- Why is it harder for ML’ers to learn IP than the other way round? Is there a chance for a two-way street?
- Not much symbolic/knowledge ML in IP – why?
- Will Neuroscience shed new light on vision in humans that will inspire IP?

Important questions

- Difficult models (e.g. For face detection, face recognition, etc.) can be learned
- Learning can be used for both signal → representation mapping AND representation → knowledge mapping
Interpretable, related levels of abstraction and **mutual dependency relationships between them** could perhaps be learned through deep learning [Hinton].

Like multi-level perceptrons, or multi-level Bayesian Belief Networks, trained in both generative (Gibbs sampling) and discriminative manner