Introduction to Active Learning

Alexey Voropaev

go.mail.ru
Mail.Ru have own Search Engine

– It is not «patched Google»

– Completely out product

– 8.3% of market

- The same engine for:
  - web
  - image
  - video
  - news
  - real time
  - e-mail search
  - etc.
We love Machine Learning :)

Simplified Search Engine Architecture:

- **Web**
  - Crawler
    - ML
    - StaticRank: crawling priority
    - Antispam
    - Antiporn
    - Object extraction
    - User behaviour
    - Ranking
    - Lingustic
    - Query processing
  - Indexer
  - Searchers
  - Frontend
  - Antirobot
  - Decision when to show vertical search
What is Supervised Machine Learning?

«Machine learning is the hot new thing»

John Hennessy, President, Stanford
There is unknown function \( f(\vec{x}) = y \)

**Given:**
Set of samples \( T = \{ \vec{x}_1, y_1 \} \ldots \{ \vec{x}_n, y_n \} \)

**Goal:**
Build approximation \( f' \) of \( f \) using \( T \)
Take good features

A — impossible

B — possible

Linear model
Proper model selection

Importance of train set construction

Usually:
- random sampling is not the best strategy
- labeling is very expensive
Modern books about ML:

- Pattern Recognition and Machine Learning — 0
- Elements of Statistical Learning — 0
- An Introduction to Information Retrieval — 1 paragraph

Book about TS construction:

Burr Settles. *Active learning.*
- Publication Date: 2 July 2012
Idea of Active Learning

Main assumption: obtaining an unlabeled instance is free
Sampling scenarios

membership query synthesis

model generates a query de novo

stream-based selective sampling

sample an instance

model decides to query or discard

pool-based sampling

sample a large pool of instances

model selects the best query

query is labeled by the oracle

instance space or input distribution

Uncertainty Sampling

Take instances about which it is least certain how to label.

Least confident:

\[ x_{LC}^* = \operatorname{argmax}_x 1 - P_\theta(\hat{y} | x) \]

Margin sampling

\[ x_M^* = \operatorname{argmin}_x P_\theta(\hat{y}_1 | x) - P_\theta(\hat{y}_2 | x) \]

Entropy

\[ x_H^* = \operatorname{argmax}_x - \sum_i P_\theta(y_i | x) \log P_\theta(y_i | x) \]
Query-By-Committee
Query-By-Committee
Query-By-Committee
Measure of committee disagreement:

**Vote entropy**

\[
x^*_\text{VE} = \arg \max_x \left( \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C} \right)
\]

**Kullback-Leibler (KL) divergence:**

\[
x^*_\text{KL} = \arg \max_x \frac{1}{C} \sum_{c=1}^C D(P_{\theta(c)} || P_C),
\]

\[
D(P_{\theta(c)} || P_C) = \sum_i P_{\theta(c)}(y_i|x) \log \frac{P_{\theta(c)}(y_i|x)}{P_C(y_i|x)}
\]
Query-By-Committee (QBag)

Input: $T$ – labeled train set
$C$ – size of the committee
$A$ – learning algorithm
$U$ – set of unlabeled objects

1. Uniformly resample $T$, obtain $T_1$...$T_C$, where $|T_i| < |T|$
2. For each $T_i$ build model $H_i$ using $A$
3. Select $x^* = \min_{x \in U} |H_i(x) = 1| - |H_i(x) = 0|$
4. Pass $x^*$ to oracle and update $T$
5. Repeat from 1 until convergence
QBag: Quality

(b) Error rate

K.Dwyer, R.Holte, *Decision Tree Instability and Active Learning*, 2007
QBag: Stability

(a) Structural FinalStab ($\epsilon = 0$)
Idea: Inhabit dense/sparse regions of the input space

\[ x_{ID}^* = \arg\max_x \phi_A(x) \times \left( \frac{1}{U} \sum_{u=1}^{U} \text{sim}(x, x^{(u)}) \right)^\beta \]
Other methods

Expected Model Change

Expected Error Reduction

Variance Reduction

Many «model dependent» methods
I recommend Mitchell (1997) or Duda et al. (2001). I have strived...

**Idea:**

1. Make set of simple regexp-based rules like
   «big letter at the right» or
   «digits at the right and left»
   (40 rules)

2. Build classifier:
   Gradient Boosted Decision Trees

Example 1: Recognition of sentence boundaries

## Example 1: Recognition of sentence boundaries

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenNLP / Sentence Boundary Detector / Random sampling</td>
<td>41.2 %</td>
</tr>
<tr>
<td>AOT project</td>
<td>30.4 %</td>
</tr>
<tr>
<td>Mail.Ru detector / Random sampling</td>
<td>8.2 %</td>
</tr>
<tr>
<td>Mail.Ru detector / least confident</td>
<td>0.8 %</td>
</tr>
</tbody>
</table>

Train set: 9820 examples
Validation: 500 examples
**Example 2: Ranking formula**

**Idea:**

1. Query-Document presented by $x$, where:
   - $x_1$ – tf-idf rank
   - $x_2$ – query geographical
   - $x_3$ – geo-region of user is equal to document's region
   - (600 other factors)

2. Build train set:
   - 5 – vital
   - 4 – exact answer
   - 3 – usefull
   - 2 – slightly usefull
   - 1 – out of topic
   - 0 – can't be labeled (unknown language, etc.)

3. Train ranking formula using modified LambdaRank
Example 2: Self-organizing map

Idea: mapping from N-dimentional to 2-dimentional
Example 2: Map of train set for ranking
Example 2: Density of QDocuments on the map

High density
Low density
Example 2: Result of “SOM miss” selection

Old documents
New documents
Example 2: Qbag + SOM heuristic

1. Transform regression to binary classification problem:
   - We should order pair of QDocuments
   - Apply QBag and select pairs of QDocuments

2. Apply SOM heuristic
   - Construct initial trainset $T$
   - Filter selected pairs

A) Don't take pair
B) May be
C) Take pair
Example 2: Qbag + SOM heuristic: results
Example 3.4: Antispam, Antiporn

**Antispam**
- Neuron net
- SOM balancing

**Antiporn**
- Decision trees
- Uncertainty sampling
Thank you!

Reference:

- http://active-learning.net/