boosting & active learning

Virgil Pavlu

Collaborative work with Jay Aslam and Robert Savell
**Problem: spam filter**

In an email inbox, there are several messages, one of which reads:

**From:** edwardth@aeroproducts.com  **To:** lilyhill@vermontel.com; denson@facil.umass.edu; 46279341@pager.icq.com; jennifersenior2000@yahoo.com; 46279397@pop.Popula.com

**Subject:** FIND OUT WHO IS IN LOVE WITH YOU

**Message:**

TOTALLY CRAZY ABOUT YOU!!
Someone [you know] has a secret crush.

We have been asked to send you this message.
Your admirer is [madly in love] with you and has said:

- You are Charming
- You are Attractive
- You Look Sweet
learning approaches

- PASSIVE
  - Get a collection of emails ALL LABELED
  - Feed the learning algorithm with them
learning approaches

- **PASSIVE**
  - Get a collection of emails ALL LABELED
  - Feed the learning algorithm with them

- **ACTIVE**
  - Get a collection of emails NOT LABELED
  - Classify few of them SPAM / NOT SPAM
  - Let Algorithm select emails to be labeled
this lecture

- active learning
- boosting algorithms
- active learning for boosting algorithms
supervised learning

TRAINING

Labeled data

LEARNING ALGORITHM

HYPOTHESIS

TRAINING ERROR
supervised learning

TRAINING
Labeled data
  → LEARNING ALGORITHM
  → HYPOTHESIS
  → TRAINING ERROR

WORKING
NOT Labeled data
  → DATA EVALUATION
  → LABEL PREDICTION
supervised learning

TRAINING
- Labeled data
- LEARNING ALGORITHM
- HYPOTHESIS
- TRAINING ERROR

WORKING
- NOT Labeled data
- DATA EVALUATION
- LABEL PREDICTION

TESTING
- Real Labels
- ERROR MEASUREMENT
- TESTING ERROR
supervised learning

Data model

\[ X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \]
\[ y_i = \text{label}(x_i); y_i \in \{-1, +1\} \]
Active learning

TRAINING

Labeled data

LEARNING ALGORITHM

HYPOTHESIS
Active learning

ACTIVE TRAINING

NOT labeled data (training set)

DATA SELECTOR → TRUTH MAKER SELLS LABELS → LEARNING ALGORITHM → HYPOTHESIS
trivial example

- High-low game
trivial example

- High-low game
- High-low game

- After getting P samples: Where is $W_0$?
  - $V_P = X_R - X_L$
active learning - summary

- $\varepsilon = \text{error allowed}$:
  - Active: needs $\Omega(\log\left(\frac{1}{\varepsilon}\right))$ instances of labeled data
  - Passive: needs $\Omega\left(\frac{1}{\varepsilon}\right)$ instances of labeled data

- Choose data where confidence is low

- What to look for:
  - Convergence rate of the classifier
  - High confidence

- Dependence of the classifier
dependence on the classifier

ACTIVE TRAINING

DATA SELECTOR

NOT labeled data (training set)

TRUTH MAKER SELLS LABELS

LEARNING ALGORITHM

HYPOTHESIS
boosting: introduction

- = Combine more classifiers in a master one

- Given
  - Labeled data set
  - Some weak predictors (error less than 50%)

- LOOP
  - Select weak predictor
  - Concentrate on the hard(wrong) instances
boosting example

- Start with uniform distribution on data
- Weak learners = halfplanes
Round 1

\[ h_1 \]

\[ \varepsilon_1 = 0.30 \quad \alpha_1 = \frac{1}{2} \log \left( \frac{1 - \varepsilon_1}{\varepsilon_1} \right) = 0.42 \]
round 2

\[ \varepsilon_2 = 0.21 \]

\[ \alpha_2 = 0.65 \]
final hypothesis

\[ H_{\text{final}} = \text{sign} \left( \begin{array}{c} 0.42 \\ + 0.65 \\ + 0.92 \end{array} \right) \]
Adaboost [Freund&Schapire ’95]

- Start with uniform distrib $D_1$: $D_1(i) = \frac{1}{m}$
- At every round $t=1$ to $T$
  - given $D_t$
    - find weak hypothesis $h_t : X \mapsto \{-1, 1\}$
    - with error
      $$\varepsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$$
    - compute “belief” in $h_t$
      $$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right) > 0$$
    - update distribution
      $$D_{t+1} = \frac{D_t}{Z_t} \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$
- final hypothesis:
  $$H_{\text{final}}(x) = \text{sgn} \left( \sum_t \alpha_t h_t(x) \right)$$
Theorem [Freund & Schapire ’97]:

write $\varepsilon_t$ as $\frac{1}{2} - \gamma_t$;
then

$$\text{training error}(H_{\text{final}}) \leq \exp\left(-2 \sum_t \gamma_t^2\right)$$

AdaBoost is adaptive:
- does not need to know $\gamma$ or $T$ a priori
- can exploit $\gamma_t >> \gamma$

GOOD: does not overfit
BAD: Susceptible to noise
the hypothesis points space

- Mapping from data space to hypothesis space
- The Hyper cube in hypothesis space
- Decision stumps

\[ U = \{-1,1\}^m \]
\[ h_i : X \rightarrow \{-1,1\} \quad i=1..m \]
\[ H: X \rightarrow U \]
\[ H(x) = (h_1(x), h_2(x), \ldots, h_m(x)) \]
**separation**

- **Separation hyperplane**
  \[ H_{\text{final}}(x) = \text{sgn} \left( \sum_i \alpha_i h_i(x) \right) \]

- **Geometric interpretation**

- **Optimal hyperplane**
  - Error measurements

- **Non-separable data**
  - SVMs
active learning - summary

- Recall high –low game

- choose data where **confidence is low**

- Which for AdaBoost means ....
active learning for AdaBoost
$T$ = training set; $X$ = initial labeled set

WHILE(condition)

\[
H_{FIN} = ADABOOST(X)
\]

Select $z = \arg \min |H_{FIN}(z)|$; label $z$

\[
X = X \cup \{(z, label(z))\}
\]
...meaning

- Neighborhoods
  - Wrongness expectation
    \[ W(u) = (P(1) \times nx^- + P(-1) \times nx^+) \times \text{neigh\_size} \]

- Inside and outside margin
  - Generalization of “min confidence” strategy
  - Resistance of the learner
    \[ R(u) = \sum_{i=\text{likely\_to\_change}} \text{confidence}(i) \]

- Time \( O(n^2) \)
experiments

- Real databases; gaussian distributions
- Cross validation
- Weak learners: decision stamps

DATA
- Discrete attributes
- Classes
results

database: /adaboost/krk

% testing error

% labeled data

- random
- virgil
- min confidence

ALL labels
4.4
ACTIVE: $\Omega\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ instances

PASSIVE: $\Omega\left(\frac{1}{\varepsilon}\right)$ instances

database: /adaboost/crx

% testing error

% labeled data
database: /adaboost/tic

% testing error

% labeled data

random
virgil
min confidence
database: /adaboost/monk

% testing error

% labeled data

random
virgil
min confidence

39.7 ALL labels
database: /adaboost/car

- random
- virgil
- min confidence

% testing error vs % labeled data

ALL labels 2.3

results
For typical data sets active learning is efficient
- Using only 20% of training data
- Gets 90% the performance of a full training

boosting simplicity make active learning controllable
A mathematical argument
  ◆ Comparison with random pooling data
  ◆ Information gain expectation

Better understanding of generalization ideas