Automatic Ranking by Extended Binary Classification

Hsuan-Tien Lin

Learning Systems Group
Joint work with Ling Li (NIPS 2006)

EE Pizza Meeting, November 17, 2006

What is the Age-Group?

2

1

2

3

4

rank: a finite ordered set of labels $\mathcal{Y} = \{1, 2, \cdots, K\}$

Hot or Not?

`http://www.hotornot.com`

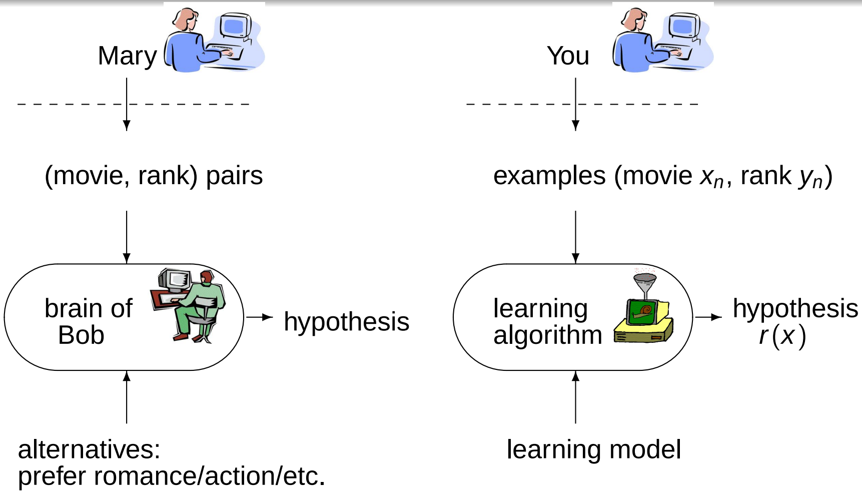
rank: natural representation of preferences in surveys

How Much Did You Like These Movies?

http://www.netflix.com

Can machines use **movies you've rated** to closely predict your preferences (i.e., ranks) on **future movies**?

How Machine Learns the Preference of You



machine learning:

an automatic route of system design

Poor Bob

Bob impresses Mary by memorizing every given (movie, rank);
but too nervous about a **new movie** and guesses randomly

- **memorize** $\neq$ **generalize**
- prefect **from Bob's view** $\neq$ good **for Mary**
- perfect **during training** $\neq$ good **when testing**

challenges:
algorithms and theories for doing well when testing

Ranking Problem

- input: N examples $(x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, e.g.
  `hotornot`: $\mathcal{X}$ = human pictures, $\mathcal{Y} = \{1, \cdots, 10\}$
  `netflix`: $\mathcal{X}$ = movies, $\mathcal{Y} = \{1, \cdots, 5\}$
- output: a ranking function $r(x)$ that ranks future unseen examples (x, y) "correctly"
- properties for the K elements in $\mathcal{Y}$:
  - **ordered**
    ★★☆☆☆ < ★★★★★
  - **not** carrying numerical information
    ★★★★★ not 2.5 times better than ★★☆☆☆

1. instance representation? some meaningful vectors
2. correctly? **cost of wrong prediction**

Cost of Wrong Prediction

- cannot quantify the numerical meaning of ranks;
  but can artificially quantify the **cost** of being wrong

infant (1)

child (2)

teen (3)

adult (4)

- small mistake – classify a child as a teenager;
  big mistake – classify an infant as an adult
- $C_{y,k}$: cost when rank y predicted as rank k
- V-shaped $C_{y,k}$ with $C_{y,y} = 0$,

  e.g. absolute cost $C_{y,k} = |y - k|$, $\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$

Even More Challenging: Netflix Million Dollar Prize

Leaderboard

Team Name	Best Score	% Improvement	Last Submit Time
No Grand Prize candidates yet	--	--	--
Grand Prize - RMSE <= 0.8563			
wxyzconsulting.com	0.9015	5.24	2006-11-15 06:05:32
ML@UToronto A	0.9021	5.18	2006-11-14 06:18:07
NIPS Reject	0.9034	5.05	2006-11-14 22:10:46

- input: N_i examples from each user i with $480,000+$ users and $\sum_i N_i \approx 100,000,000$
- output: personalized predictions $r(i, x)$ on $2,800,000+$ testing queries (i, x)
- cost: squared cost $C_{y,k} = (y - k)^2$
- a huge joint ranking problem

The first team that gets 10% better than existing Netflix system gets **a million USD**

Our Contributions

a new framework that ...

- makes the design and implementation of ranking algorithms **almost effortless**
- makes the proof of ranking theories **much simpler**
- unifies many existing ranking algorithms and **helps understand their cons and pros**
- shows that ranking is theoretically **not much more complex than binary classification**
- leads to **promising experimental performance**

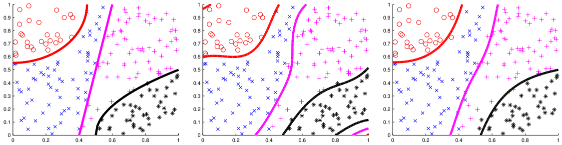


Figure: answer; traditional method; our method

Key Idea: Reduction

(iPod)

(adapter)

(cassette player)

complex ranking problems

(reduction)

simpler binary problems with well-known results on models, algorithms, proofs, etc.

many new results immediately come up;
many existing results unified

Intuition: Associated Binary Questions

- how we query the rank of a movie x?
  1. is movie x better than rank 1? Yes
  2. is movie x better than rank 2? No
  3. is movie x better than rank 3? No
  4. is movie x better than rank 4? No
  5. is movie x better than rank 5? No
- $g_b(x, k)$: is movie x better than rank k?
- consistent answers: $G(x) = (1, 1, 1, 0, \cdots, 0)$
- extract the rank from consistent answers:
  - searching: compare to a "middle" rank each time
  - voting: $r(x) = 1 + \sum_k g_b(x, k)$
- what if the answers are not consistent? e.g. $(1, 0, 1, 1, 0, 0, 1, 0)$
  – voting is simple enough to analyze, and still works

accurate binary answers $\Longrightarrow$ correct ranks

Reduction during Training

- input: N examples $(x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
- tool: your favorite binary classification algorithm
- output: a binary classifier $g_b(x, k)$ that can answer the associated questions correctly

- need to feed binary examples $(X_{n,k}, Y_{n,k})$ to the tool

$$X_{n,k} \equiv (x_n, k), Y_{n,k} \equiv [y_n > k]$$

- about NK extended binary examples extracted from given input
  – bigger, but not troublesome
- some approaches extract about N^2 binary examples using a different intuition
  – can be too big

Are extended binary examples of the same importance?

Importance of Extended Binary Examples

- for a given movie x_n with rank $y_n = 2$, and $C_{y,k} = (y-k)^2$

is x_n better than rank 1?	No	Yes	Yes	Yes
is x_n better than rank 2?	No	No	Yes	Yes
is x_n better than rank 3?	No	No	No	Yes
is x_n better than rank 4?	No	No	No	No
$r(x_n)$	1	2	3	4
cost	1	0	1	4

- 3 more for answering question 4 wrong;
  only 1 more for answering question 1 wrong
- $W_{n,k} \equiv \left| C_{n,k+1} - C_{n,k} \right|$: the importance of $(X_{n,k}, Y_{n,k})$
- most binary classification algorithm can handle $W_{n,k}$

analogy to economics:
additional cost (marginal) $\Longleftrightarrow$ importance

The Reduction Framework for Ranking

1. transform ranking examples (x_n, y_n) to extended binary examples $(X_{n,k}, Y_{n,k}, W_{n,k})$ based on $C_{y,k}$
2. use your favorite algorithm to learn from the extended binary examples, and get $g_b(x, k) \equiv g_b(X)$
3. for each new instance x, predict its rank using $r(x) = 1 + \sum_k g_b(x, k)$

- error equivalence: accurate binary answers $\Longrightarrow$ correct ranks
- simplicity: works with almost any $C_{y,k}$ and any algorithm
- up-to-date: new improvements in binary classification immediately propagates to ranking

If I have seen further it is by standing on ye shoulders of Giants – I. Newton

Unifying Existing Algorithms

- ranking with perceptrons
  – (PRank, Crammer and Singer, 2002)

  several long proof
  $\Rightarrow$ a few lines extended from binary perceptron results

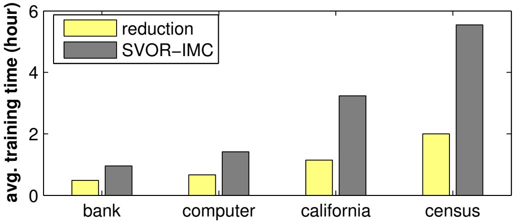
- large-margin (high confidence) formulations
  – (Rajaram et al., 2003), (SVORIM, Chu and Keerthi, 2005)

  results explained more directly; algorithm structure revealed

variants of existing algorithms can be designed quickly by tweaking reduction

Proposing New Algorithms

- ranking using ensemble (consensus) of classifiers – (ORBoost, Lin and Li, 2006), OR-AdaBoost
- ranking using decision trees – OR-C4.5
- ranking with large-margin classifiers – OR-SVM



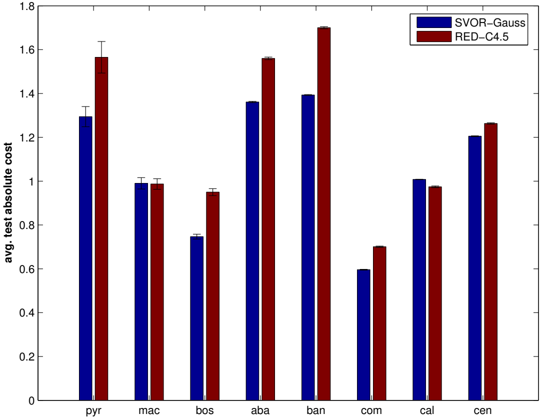
advantages of underlying binary algorithm inherited in the new ranking one

Proving New Theorems

- simpler cost bound for PRank
- new guarantee of ranking performance using ensemble of classifiers (Lin and Li, 2006)
- new guarantee of ranking performance using large-margin classifiers, e.g.,

$$\underbrace{\mathcal{E}_{(x,y)} C_{y,r(x)}}_{\text{expected cost during testing}} \leq \frac{1}{N} \sum_{n} \sum_{k} \Big[\, \underbrace{\rho(X_{n,k}, Y_{n,k}) \leq \Delta}_{\text{low confidence extended examples}} \,\Big] + K \cdot \underbrace{h_\delta(N, \Delta)}_{\text{deviation func. that decreases with more data or confidence}}$$

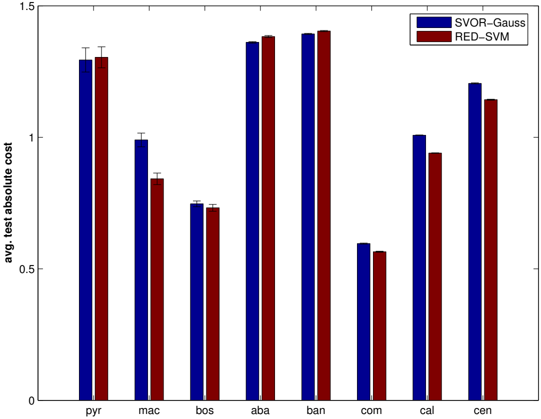
Reduction-C4.5 vs. SVORIM



- C4.5: decision tree, a intuitive, but often too simple, binary classifier
- SVORIM: state-of-the-art ranking algorithm

even reduction to simple C4.5 beats SVORIM some time

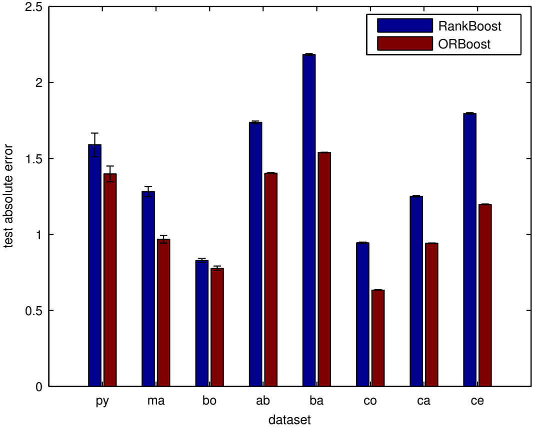
Reduction-SVM vs. SVORIM



- SVM: one of the most powerful binary classifier
- SVORIM: state-of-the-art ranking algorithm extended from a modified SVM

reducing to SVM without modification often better than SVORIM

Reduction-Boost vs. RankBoost



- Boost: a popular ensemble algorithm
- RankBoost: state-of-the-art ensemble ranking algorithm

our reduction to boosting approaches results in significantly better ensemble ranking algorithm

Conclusion

- reduction framework: simple, intuitive, and useful for ranking
- algorithmic reduction:
  - unifying existing ranking algorithms
  - proposing new ranking algorithms
- theoretic reduction:
  - new guarantee on ranking performance
- promising experimental results:
  - some for better performance
  - some for faster training time
- next level: the Netflix challenge?
  - handling huge datasets
  - finding useful representations (features)
  - using collaborative information from other users

Thank you. Questions?