Automatic Ranking by Extended Binary Classification

Hsuan-Tien Lin

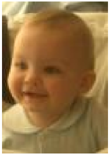
Joint work with Ling Li (*ALT '06, NIPS '06*)
Learning Systems Group, California Institute of Technology

Talk at Institute of Information Science, Academia Sinica
March 21, 2007

Introduction to Automatic Ranking

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

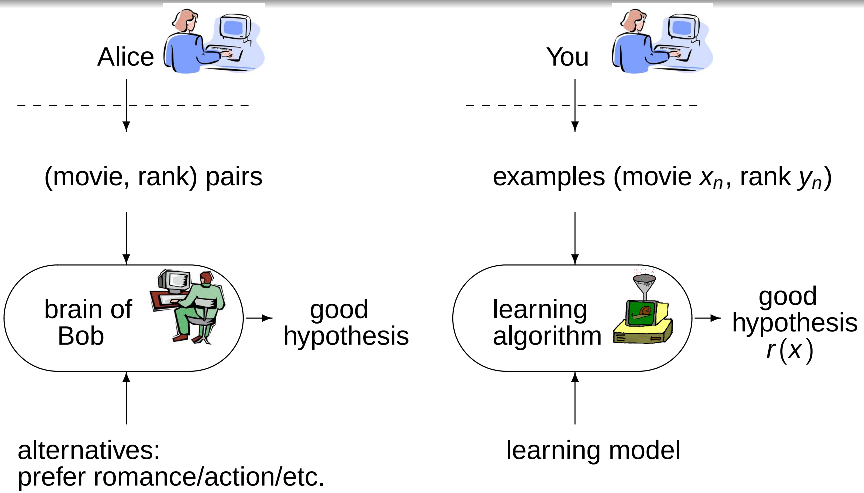
rank: natural representation of human preferences

How Much Did You Like These Movies?

http://www.netflix.com

goal: use "movies you've rated" to automatically predict your preferences (ranks) on "future movies"

Human Ranking v.s. Automatic Ranking



challenge: how to make the right-hand-side work?

Ranking (Ordinal Regression) Problem

- given: N examples (input x_n, rank y_n) $\in \mathcal{X} \times \mathcal{Y}$, e.g.
  `hotornot`: $\mathcal{X}$ = human pictures, $\mathcal{Y} = \{1, \cdots, 10\}$
  `netflix`: $\mathcal{X}$ = movies, $\mathcal{Y} = \{1, \cdots, 5\}$
- goal: a ranking function $r(x)$ that "closely predicts" the ranks y associated with some unseen inputs x

a hot research problem:

- relatively new for machine learning
- connecting classification and regression
- matching human preferences – many applications in social science and information retrieval

Ongoing Heat: Netflix Million Dollar Prize

Leaderboard

Display top 3 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
--	No Grand Prize candidates yet	--	--	--
Grand Prize - RMSE <= 0.8563				
1	Gravity	0.8872	6.75	2007-01-28 23:18:21
2	ICMLsubmission	0.8875	6.72	2007-03-16 19:30:34
3	ML@UToronto A	0.8883	6.63	2007-01-19 19:00:56

- a competition from 2006/10
- given: each user i ($480,000+$ users) rates N_i (from tens to hundreds) movies – a total of $\sum_i N_i \approx 100,000,000$ examples
- goal: personalized predictions $r_i(x)$ on $2,800,000+$ testing queries (i, x)
- a huge joint ranking problem

the first team being 10% better than existing Netflix system gets a million USD

Properties of Ranks $\mathcal{Y} = \{1, 2, \cdots, 5\}$

- representing **order**:
  ★★☆☆☆ $<$ ★★★★★
  – relabeling by $(3, 1, 2, 4, 5)$ erases information

  > general classification cannot
  > properly use ordering information

- **not** carrying numerical information:
  ★★★★★ not 2.5 times better than ★★☆☆☆
  – relabeling by $(2, 3, 5, 9, 16)$ shouldn't change results

  > general regression deteriorates
  > without correct numerical information

ranking resides uniquely between
classification and regression

Cost of Wrong Prediction

- ranks carry no numerical meaning: how to say “closely predict”?
- artificially quantify the **cost** of being wrong


infant (1)


child (2)


teen (3)

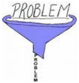

adult (4)

- small mistake – classify a child as a teen;
  big mistake – classify an infant as an adult
- $\mathcal{C}_{y,k}$: cost when rank y predicted as k, e.g. $\begin{pmatrix} 0 & 1 & 4 & 5 \\ 1 & 0 & 1 & 3 \\ 3 & 1 & 0 & 2 \\ 5 & 4 & 1 & 0 \end{pmatrix}$

  – will first focus on $\mathcal{C}_{y,k} = |y - k|$ (absolute cost)

closely predict: small testing cost

Our Accomplishments



a new framework that ...

- connects ranking and binary classification **systematically**
- unifies and **clearly explains** many existing ranking algorithms
- makes the design of new ranking algorithms **much easier**
- allows **simple and intuitive** proof for new ranking theorems
- leads to **promising experimental results**

next: start with a concrete and specific case;
then: introduce the general framework

Automatic Ranking using Ensembles

Intuition behind Ensemble Learning

Ensemble Regression

- "the stock price tomorrow?"
- expert t suggests $h_t(x)$
- the ensemble (committee) reports weighted average of experts

$$\sum_t w_t h_t(x)$$

- **stable**: errors of a few experts diluted by weighted average

Ensemble Classification

- "shall we watch movie x?"
- member t: $h_t(x) \in \pm 1$
- the ensemble (committee) reports weighted vote of members

$$\text{sign}\left(\sum_t w_t h_t(x) \right)$$

- **powerful**: complicated decisions approximated by weighted votes

ensemble: useful and successful in modeling regression and classification problems

Our Contributions

- new model for ranking: thresholded ensemble model
  – a ranking extension of ensemble learning
- new generalization bounds for thresholded ensembles
  – theoretical guarantee of testing performance
- new algorithms for constructing thresholded ensembles
  – simple and efficient

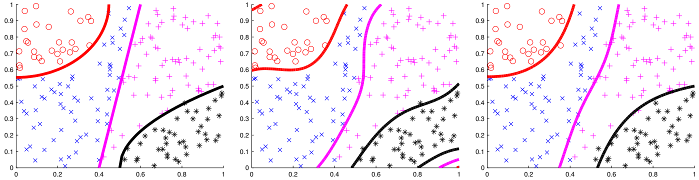
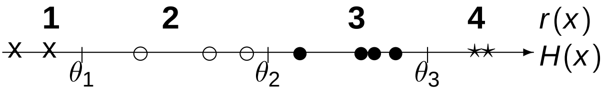


Figure: target; general regression; our algorithm

promising experimental results

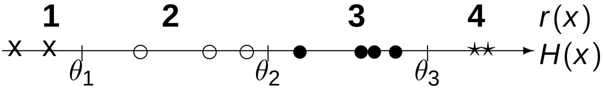
Thresholded Model

- commonly used in previous ranking work:
  - thresholded perceptrons (PRank, Crammer02)
  - thresholded hyperplanes (SVOR, Chu05)
- prediction procedure:
  1. compute a potential function $H(x)$
  2. quantize $H(x)$ by some **ordered** θ to get $r(x)$



thresholded model: $r(x) \equiv r_{H,\theta}(x) = \min \{k : H(x) < \theta_k\}$

Thresholded Ensemble Model



- the potential function $H(x)$ is an ensemble $H(x) \equiv H_T(x) = \sum_{t=1}^{T} w_t h_t(x)$
- intuition: if many people, h_t, say a movie x is "good", the potential of the movie $H(x)$ should be high
- ensemble classification: a special case when $K = 2$ and $\theta_1 = 0$

classification	ranking
$\text{sign}(H_T(x))$	$\min\{k : H_T(x) < \theta_k\}$

good theoretical and algorithmic properties inherited from ensemble classification

Recall: Goal and Cost

- goal: a ranking function $r(x)$ that closely predicts the ranks y associated with some unseen inputs x

  e.g. predicts your preference on future movies
- $\mathcal{C}_{y,k}$: cost when rank y predicted as rank k
  absolute cost $\mathcal{C}_{y,k} = |y - k|$

  e.g. loss of customer royalty when the system says ★★★★★ but you feel ★★☆☆☆

closely predict $\Longleftrightarrow$ small testing cost
how to formalize?

Generalization Error

- setup: training examples (x_n, y_n) and testing ones (x, y) generated i.i.d. from the same (unknown) distribution $\mathcal{D}$
- what can be said about the generalization error

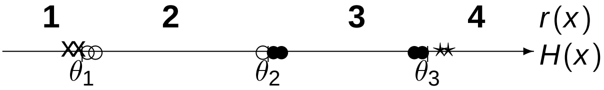
$$E(r) = \mathcal{E}_{(x,y)} \mathcal{C}_{y,r(x)}$$

  of the chosen $r(x)$?
- E_A: generalization error when using the absolute cost

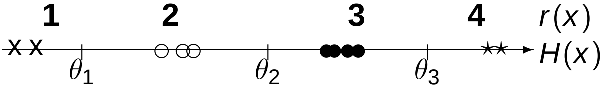
goal: some $r(x)$ with small generalization error

Good Thresholded Ensembles

- "bad" thresholded ensemble: predictions close to thresholds – small noise changes prediction

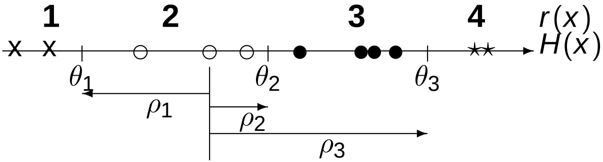


- "good" thresholded ensemble: clear separation using thresholds



next: good thresholded ensemble
$\Longrightarrow$ small generalization error

Margins of Thresholded Ensembles



- margin (confidence): safe distance from the threshold
- normalized margin for thresholded ensemble

$$\bar{\rho}(x, y, k) = \left\{ \begin{array}{l} H_T(x) - \theta_k, \text{ if } y > k \\ \theta_k - H_T(x), \text{ if } y \le k \end{array} \right\} \Bigg/ \left(\sum_{t=1}^{T} |w_t| + \sum_{k=1}^{K-1} |\theta_k| \right)$$

- negative margin implies wrong prediction:

$$\sum_{k=1}^{K-1} [\bar{\rho}(x, y, k) \le 0] = |y - r(x)|$$

good thresholded ensemble:

large and positive training margins

Large-Margin Bounds on Generalization Error

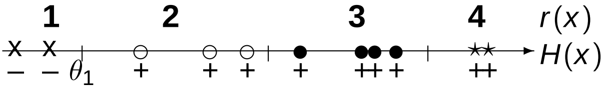
- core results:
  if (x_n, y_n) i.i.d. from $\mathcal{D}$, for all margin criteria $\Delta > 0$,
  with probability $> 1 - \delta$,

$$E_A \leq \underbrace{\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1}\left[\bar{\rho}(x_n, y_n, k) \leq \Delta\right]}_{\text{number of small margin training examples}} + \underbrace{O\left(K\sqrt{\frac{1}{N}\left(\frac{\log^2 N}{\Delta^2} + \log\frac{1}{\delta}\right)}\right)}_{\text{deviation that decreases with stronger criteria or more examples}}$$

- large-margin thresholded ensembles can generalize

key: connecting ranking to binary classification

Ranking to Binary Classification



- recall: ranking ensemble extended from classification ensemble
- $K - 1$ binary classification problems w.r.t. each θ_k
- let $\big((X)_k, (Y)_k\big)$ be binary examples
  - $(X)_k = (x, k)$: input w.r.t. k-th threshold
  - $(Y)_k = \text{sign}\big(y - k - \frac{1}{2}\big)$: binary label $+/-$
- key observation:

$$\begin{aligned} E_A &= \mathcal{E}_{(x,y)\sim\mathcal{D}} \big| y - r(x) \big| \\ &= \mathcal{E}_{(x,y)\sim\mathcal{D}} \sum\nolimits_{k=1}^{K-1} \big[\bar{\rho}(x,y,k) \le 0\big] \\ &= (K-1)\mathcal{E}_{(x,y)\sim\mathcal{D},k\sim\mathcal{K}} \big[\bar{\rho}(x,y,k) \le 0\big] \\ &= (K-1) \text{ gen. error in binary classification} \end{aligned}$$

ensemble ranking problem equivalent to one big joint ensemble classification problem

Parallel Between Ranking and Binary Classification

Bin. Classification (Schapire98)

$$\text{gen. error} \leq \frac{1}{N}\sum_{n=1}^{N}[\bar{\rho}(X_n, Y_n) \leq \Delta] + O\left(\sqrt{\frac{1}{N}\left(\frac{\log^2 N}{\Delta^2} + \log\frac{1}{\delta}\right)}\right)$$

⇕

Adaptive Boosting (Freund96)

one of the most successful algorithms in bin. classification

Ranking

$$E_A \leq \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1}[\bar{\rho}(x_n, y_n, k) \leq \Delta] + O\left(K\sqrt{\frac{1}{N}\left(\frac{\log^2 N}{\Delta^2} + \log\frac{1}{\delta}\right)}\right)$$

⇕

Ordinal Reg. Boosting

new algorithm for ranking that connects to the bound above

other theoretical results derived;
same technique applied to algorithms

Intuition behind Boosting

- boosting: a popular family of algorithms for ensemble learning

AdaBoost for ensemble classification

for $t = 1, 2, \cdots, T$,

1. add an h_t that matches best with the current "view" of training examples
2. give a larger weight w_t to h_t if the match is stronger
3. update "view" by emphasizing training examples with small margins

output: $\text{sign}(H_T(x))$

- better h_t gets more weights (votes) in the ensemble
- each h_t improves small-margin examples

how to perform ensemble ranking with boosting?

ORBoost: Ordinal Regression Boosting

AdaBoost for classification

for $t = 1, 2, \cdots, T$,

1. add an h_t that matches best with the current "view" of training examples
2. give a larger weight w_t to h_t if the match is stronger
3. update "view" by emphasizing training examples with small margins

output: $\text{sign}(H_T(x))$

ORBoost for ranking

for $t = 1, 2, \cdots, T$,

1. for fixed θ, add an h_t that matches current "view" of the tuples (x_n, y_n, k) well
2. give a larger weight w_t to h_t if the match is stronger
3. **update θ_k based on the newly added (h_t, w_t)**
4. update "view" by emphasizing tuples with small margins

output: $r_{H_T,\theta}(x)$

ORBoost: closely connected to large-margin bounds

Connection to Large-Margin Bounds

Bin. Classification (Schapire98)

gen. error $\leq \frac{1}{N}\sum_{n=1}^{N}[\bar{\rho}(X_n, Y_n) \leq \Delta]$

$+ \quad O\left(\sqrt{\frac{1}{N}\left(\frac{\log^2 N}{\Delta^2} + \log\frac{1}{\delta}\right)}\right)$

Ranking

$E_A \leq \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1}[\bar{\rho}(x_n, y_n, k) \leq \Delta]$

$+ \quad O\left(K\sqrt{\frac{1}{N}\left(\frac{\log^2 N}{\Delta^2} + \log\frac{1}{\delta}\right)}\right)$

AdaBoost

implicitly minimizing

$$\sum_{n=1}^{N}[\bar{\rho}(X_n, Y_n) \leq \Delta]$$

ORBoost

implicitly minimizing:

$$\sum_{n=1}^{N}\sum_{k=1}^{K-1}[\bar{\rho}(x_n, y_n, k) \leq \Delta]$$

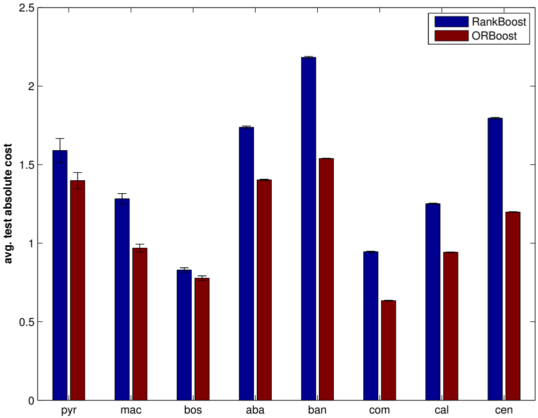
algorithmic reduction analogous to theoretical reduction

Advantages of ORBoost

- ensemble learning:
  combine simple preferences to approximate complex targets
- thresholding:
  adaptively estimating scales to predict ranks
- benefits inherited from AdaBoost
  - simple implementation
  - ranking function $r(x)$ improves when adding more h_t

ORBoost not very vulnerable to overfitting in practice

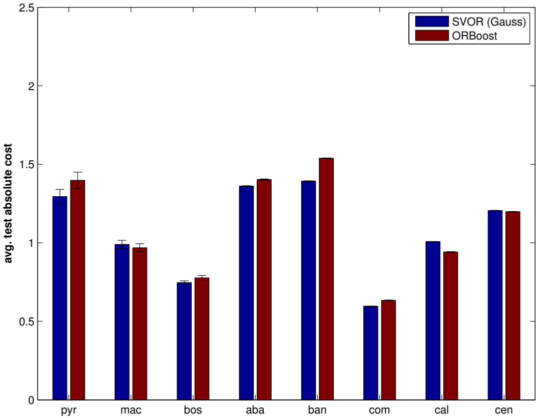
ORBoost v.s. RankBoost



- RankBoost (Freund03): best existing ensemble ranking algorithm
- ORBoost significantly better than RankBoost
- simpler to implement; faster to train

ORBoost: promising ensemble ranking algorithm

ORBoost v.s. SVOR



- SVOR: state-of-the-art ranking algorithm using thresholded hyperplane
- ORBoost: comparable performance
- much faster training (1 hour v.s. 2 days on 6000 examples)

ORBoost: especially useful for large-scale tasks

Summary for Ensemble Ranking

- thresholded ensemble model: useful for ranking
  - theoretical reduction: new large-margin bounds
  - algorithmic reduction: new learning algorithms
- ORBoost:
  - simplicity and better performance over existing ensemble algorithm
  - comparable performance to state-of-the-art algorithms
  - fast training and not very vulnerable to overfitting

next: apply the steps more generally

Reduction from Ranking to Extended Binary Classification

Ranking v.s. Binary Classification

- parallel between ranking and binary classification

result	ensemble ranking	ensemble classification
model	thresholded ensemble	signed ensemble
theorem	large-margin bounds	large-margin bounds
algorithm	ORBoost	AdaBoost

- many more in literature

result	ranking	classification
model	thresholded perceptron	perceptron
algorithm	PRank	perceptron rule
model	thresholded hyperplane	hyperplane
algorithm	SVOR	SVM

next: systematically reducing ranking to binary classification

Intuition of Reduction: Associated Binary Questions

getting the rank with a thresholded ensemble

1. is $H_T(x) > \theta_1$? Yes
2. is $H_T(x) > \theta_2$? No
3. is $H_T(x) > \theta_3$? No
4. is $H_T(x) > \theta_4$? No

generally, how do 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

associated binary questions $g_b(x, k) = g_b((X)_k)$:

is movie x better than rank k?

Predicting from Associated Binary Questions

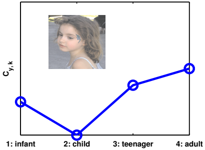
$g_b(x, k)$: is movie x better than rank k?
e.g. thresholded model $g_b(x, k) = \text{sign}(H(x) - \theta_k)$

- consistent answers: $(+, +, +, -, \cdots, -)$
- extract the rank from consistent answers:
  - minimum index searching: $r(x) = \min\{k : g_b(x, k) < 0\}$
  - counting: $r(x) = 1 + \sum_k [g_b(x, k) > 0]$
- two approaches equivalent for consistent answers
- inconsistent answers? e.g. $(+, -, +, +, -, -, -, +)$: counting is simple enough to analyze, and still works

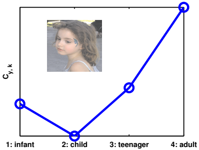
are all binary questions of the same importance?

Cost Revisited: Reasonable Cost Functions

- $\mathcal{C}_{y,k}$: cost when rank y predicted as k
- cost function that respects ranking properties



V-shaped: pay more when predicting further away



convex: pay **increasingly** more when further away

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

classification: V-shaped only

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

absolute: convex

$$\begin{pmatrix} 0 & 1 & 4 & 9 \\ 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 \\ 9 & 4 & 1 & 0 \end{pmatrix}$$

squared: convex

Importance of Extended Binary Examples

- given movie x_n with rank $y_n = 2$, and $\mathcal{C}_{y,k} = (y-k)^2$

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 3 wrong; only 1 more for answering question 1 wrong
- $W_{y,k} \equiv \left|\mathcal{C}_{y,k+1} - \mathcal{C}_{y,k}\right|$: the importance of $((X)_k, (Y)_k)$
- error reduction theorem: for **consistent answers** or **convex costs**

$$\mathcal{C}_{y,k} \le \sum_{k=1}^{K-1} W_{y,k} \left[(Y)_k \ne g_b((X)_k)\right]$$

accurate binary answers $\Longrightarrow$ correct ranks

The Reduction Framework

1. transform ranking examples (x_n, y_n) to extended binary examples $((X_n)_k, (Y_n)_k, W_{y_n,k})$ based on $\mathcal{C}_{y,k}$
2. use your favorite algorithm to learn from the extended binary examples, and get $g_b(x,k) \equiv g_b\big((X)_k\big)$
3. for each new instance x, predict its rank using $r(x) = 1 + \sum_k \big[g_b(x,k) > 0\big]$

- error reduction: accurate binary answers $\Longrightarrow$ correct ranks
- simplicity: works with any reasonable $\mathcal{C}_{y,k}$ 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 the shoulders of Giants – I. Newton

Unifying Existing Algorithms with the Framework

ranking	cost	binary algorithm
PRank (Crammer02)	absolute	modified perceptron rule
kernel ranking (Rajaram03)	classification	modified hard-margin SVM
SVOR-EXP SVOR-IMC (Chu05)	classification absolute	modified soft-margin SVM modified soft-margin SVM
ORBoost-LR ORBoost-All	classification absolute	modified AdaBoost modified AdaBoost

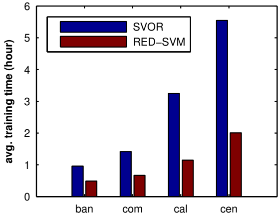
- development and implementation time saved
- correctness proof significantly simplified (PRank)
- algorithmic structure revealed (SVOR, ORBoost)

variants of existing algorithms can be designed quickly by tweaking reduction

Proposing New Algorithms with the Framework

ranking	cost	binary algorithm
Red.-C4.5	absolute	standard C4.5 decision tree
Red.-AdaBoost	absolute	standard AdaBoost
Red.-SVM	absolute	standard soft-margin SVM

SVOR (modified SVM) v.s. Red.-SVM (standard SVM):



advantages of underlying binary algorithm inherited in the new ranking one

Proving New Theorems with the Framework

- showed: new bounds of generalization error using large-margin ensembles
- similarly, new bounds of generalization error using large-margin hyperplanes

Binary Classification (Bartlett98)

gen. error

$$\leq \frac{1}{N}\sum_{n=1}^{N}[\bar{\rho}(X_n, Y_n) \leq \Delta]$$

$$+ \quad O\left(\frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}}\right)$$

Ranking

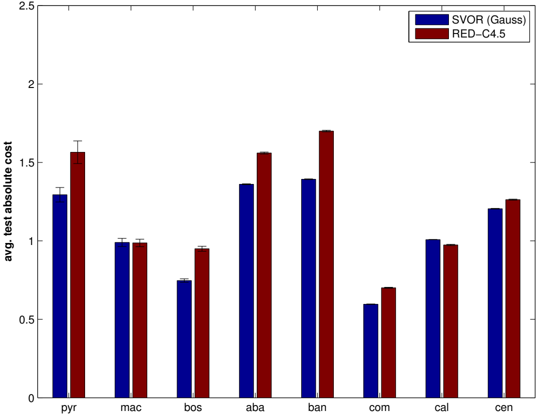
$$\mathcal{E}_{(x,y)}\mathcal{C}_{y,r(x)}$$

$$\leq \frac{\beta}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1} W_{y_n,k}\left[\bar{\rho}(x_n, y_n, k) \leq \Delta\right]$$

$$+ \quad O\left(\frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}}\right)$$

new large-margin bounds for any reasonable $\mathcal{C}_{y,k}$

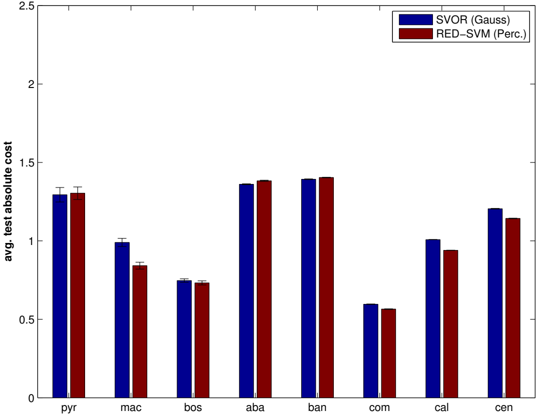
Red.-C4.5 v.s. SVOR



- C4.5: a (too) simple binary classifier – decision trees
- SVOR: state-of-the-art ranking algorithm

even simple Red.-C4.5 sometimes beats SVOR

Red.-SVM v.s. SVOR



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

Red.-SVM without modification often better than SVOR* and faster

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

reduction keeps ranking up-to-date with binary classification

Acknowledgments

- Prof. Yaser S. Abu-Mostafa, and Amrit Pratap for many helpful discussions
- Dr. John Langford, reviewers, and previous audience for useful comments
- Dr. Tyng-Luh Liu for talk invitation

Thank you. Questions?