From Ordinal Ranking to Binary Classification

林 軒 田

Hsuan-Tien Lin

Learning Systems Group, California Institute of Technology

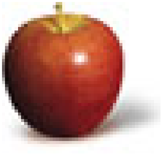
Talk at CS Department, National Tsing-Hua University
March 27, 2008

Benefited from joint work with Dr. Ling Li (ALT'06, NIPS'06)
& discussions with Prof. Yaser Abu-Mostafa and Dr. Amrit Pratap

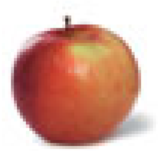
Outline

1. **Introduction to Machine Learning**
2. The Ordinal Ranking Setup
3. Reduction from Ordinal Ranking to Binary Classification
   - Algorithmic Usefulness of Reduction
   - Theoretical Usefulness of Reduction
   - Experimental Performance of Reduction
4. Conclusion

Apple, Orange, or Strawberry?



?



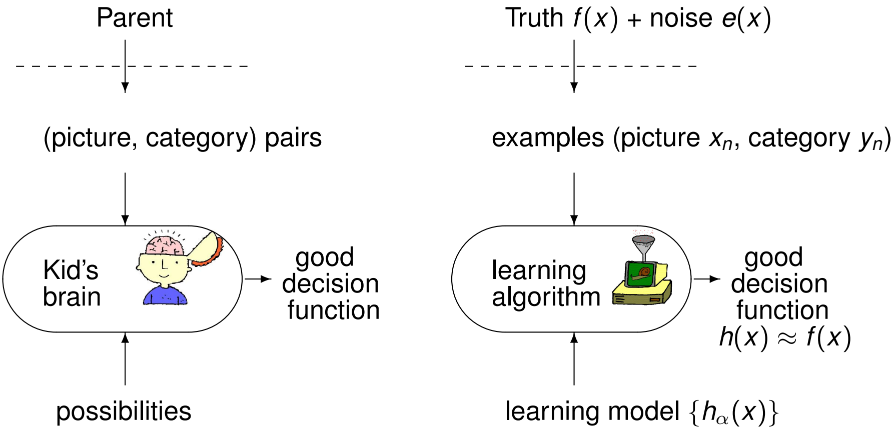
apple

orange

strawberry

how can machine learn to classify?

Supervised Machine Learning



challenge:

see only $\{(x_n, y_n)\}$ without knowing $f(x)$ or $e(x)$

$\stackrel{?}{\Longrightarrow}$ **generalize** to unseen (x, y) w.r.t. $f(x)$

Machine Learning Research

- What can the machines learn? (application)
  - concrete:
    computer vision, architecture optimization, information retrieval, bio-informatics, computational finance, $\cdots$
  - abstract setups:
    classification, regression, $\cdots$
- How can the machines learn? (algorithm)
  - faster
  - better **generalization**
- Why can the machines learn? (theory)
  - paradigms:
    statistical learning, reinforcement learning, $\cdots$
  - generalization guarantees

new opportunities keep coming
from new applications/setups

Outline

1. Introduction to Machine Learning
2. **The Ordinal Ranking Setup**
3. Reduction from Ordinal Ranking to Binary Classification
   - Algorithmic Usefulness of Reduction
   - Theoretical Usefulness of Reduction
   - Experimental Performance of Reduction
4. Conclusion

Which Age-Group?

2

infant (1)

child (2)

teen (3)

adult (4)

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

Properties of Ordinal Ranking (1/2)

ranks represent **order** information

infant (1) $<$

child (2) $<$

teen (3) $<$

adult (4)

general multiclass classification cannot properly use order information

Hot or Not?

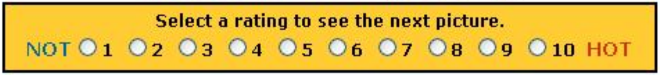
http://www.hotornot.com

rank: natural representation of human preferences

Properties of Ordinal Ranking (2/2)

ranks do **not** carry numerical information

- rating 9 not 2.25 times "hotter" than rating 4



- actual metric hidden

			
infant (ages 1–3)	child (ages 4–12)	teen (ages 13–19)	adult (ages 20–)

general metric regression deteriorates without correct numerical information

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

Ordinal Ranking Setup

Given

N examples (input x_n, rank y_n) $\in \mathcal{X} \times \mathcal{Y}$

- `age-group`: $\mathcal{X}$ = encoding(human pictures), $\mathcal{Y} = \{1, \cdots, 4\}$
- `hotornot`: $\mathcal{X}$ = encoding(human pictures), $\mathcal{Y} = \{1, \cdots, 10\}$
- `netflix`: $\mathcal{X}$ = encoding(movies), $\mathcal{Y} = \{1, \cdots, 5\}$

Goal

an ordinal ranker (decision function) $r(x)$ that "closely predicts" the ranks y associated with some **unseen** inputs x

ordinal ranking: a hot and important research problem

Ongoing Heat: Netflix Million Dollar Prize (since 10/2006)

Given

each user u (480,189 users) rates N_u (from tens to thousands) movies x—a total of $\sum_u N_u = 100{,}480{,}507$ examples

Goal

personalized ordinal rankers $r_u(x)$ evaluated on 2,817,131 “unseen” queries (u, x)

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	When Gravity and Dinosaurs Unite	0.8686	8.70	2008-02-12 12:03:24
2	BellKor	0.8686	8.70	2008-02-26 23:26:28
3	Gravity	0.8708	8.47	2008-02-06 14:12:44

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

Cost of Wrong Prediction

- ranks carry no numerical information: how to say "better"?
- artificially quantify the **cost** of being wrong

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

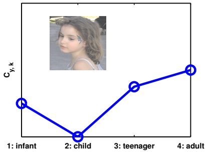
- cost vector **c** of example $(x, y, \mathbf{c})$:
  $\mathbf{c}[k]$ = cost when predicting (x, y) as rank k
  e.g. for (`Sweet Home Alabama` , ★★☆☆☆), a proper cost is $\mathbf{c} = (1, 0, 2, 10, 15)$

closely predict: small test cost

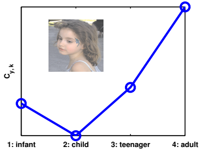
Ordinal Cost Vectors

For an ordinal example $(x, y, \mathbf{c})$, the cost vector $\mathbf{c}$ should

- follow the rank y: $\mathbf{c}[y] = 0$; $\mathbf{c}[k] \geq 0$
- respect the ordinal information: V-shaped (**ordinal**) or even convex (**strongly ordinal**)



V-shaped: pay more when predicting further away



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

$\mathbf{c}[k] = [\![y \neq k]\!]$	$\mathbf{c}[k] = \lvert y - k \rvert$	$\mathbf{c}[k] = (y - k)^2$
classification: ordinal	absolute: strongly ordinal	squared (Netflix): strongly ordinal
$(1, 0, 1, 1, 1)$	$(1, 0, 1, 2, 3)$	$(1, 0, 1, 4, 9)$

Our Contributions

a theoretical and algorithmic foundation of ordinal ranking, which ...

- provides a methodology for designing new ordinal ranking algorithms with **any** ordinal cost **effortlessly**
- takes many existing ordinal ranking algorithms as **special cases**
- introduces **new theoretical guarantee** on the generalization performance of ordinal rankers
- leads to **superior experimental results**

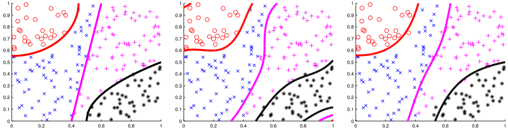
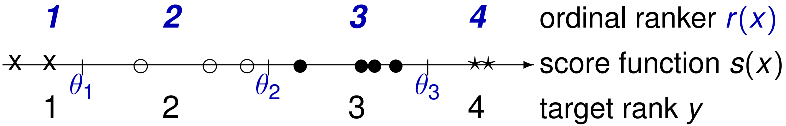


Figure: truth; traditional algorithm; our algorithm

Outline

Threshold Model

- If we can first get an ideal score $s(x)$ of a movie x, how can we construct the discrete $r(x)$ from an analog $s(x)$?



quantize $s(x)$ by some **ordered** threshold θ

- commonly used in previous work:
  - threshold perceptrons (PRank, Crammer and Singer, 2002)
  - threshold hyperplanes (SVOR, Chu and Keerthi, 2005)
  - threshold ensembles (ORBoost, Lin and Li, 2006)

threshold model: $r(x) = \min\{k\colon s(x) < \theta_k\}$

Key of Reduction: Associated Binary Queries

getting the rank using a threshold model

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

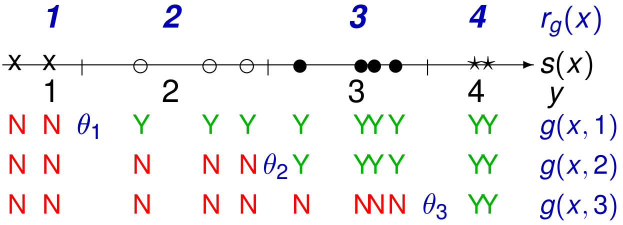
associated binary queries:

is movie x better than rank k?

More on Associated Binary Queries

say, the machine uses $g(x, k)$ to answer the query
"*is movie x better than rank k?*"
e.g. threshold model $g(x, k) = \text{sign}(s(x) - \theta_k)$

- $K - 1$ binary classification problems w.r.t. each k



- let $((x, k), (z)_k)$ be binary examples
  - (x, k): extended input w.r.t. k-th query
  - $(z)_k$: desired binary answer Y/N

If $g(x, k) = (z)_k$ for all k,
we can compute $r_g(x)$ from $g(x, k)$ s.t. $r_g(x) = y$.

Computing Ranks from Associated Binary Queries

> when $g(x, k)$ answers "*is movie x better than rank k?*"

Consider $\big(g(x,1), g(x,2), \cdots, g(x,K-1)\big)$,

- consistent predictions: (Y, Y, N, N, N, N, N)
- extracting the rank:
  - minimum index searching: $r_g(x) = \min\{k : g(x,k) = \text{N}\}$
  - counting: $r_g(x) = 1 + \sum_k [\![g(x,k) = \text{Y}]\!]$
- two approaches equivalent for consistent predictions
- noisy/inconsistent predictions? e.g. (Y, N, Y, Y, N, N, Y)

> **counting: simpler to analyze and robust to noise**

The Counting Approach

Say $y = 5$, i.e., $\big((z)_1, (z)_2, \cdots, (z)_7\big) = $ (Y, Y, Y, Y, N, N, N)

- if $g_1(x, k)$ reports (Y, Y, N, N, N, N, N)
  - $g_1(x, k)$ made 2 errors
  - $r_{g_1}(x) = 3$; absolute cost $= 2$

absolute cost = # of binary classification errors

- if $g_2(x, k)$ reports (Y, N, Y, Y, N, N, Y)
  - $g_2(x, k)$ made 2 errors
  - $r_{g_2}(x) = 5$; absolute cost $= 0$

absolute cost $\leq$ # of binary classification errors

If $(z)_k$ = desired answer & r_g computed by counting,

$$\big|y - r_g(x)\big| \leq \sum_{k=1}^{K-1} [\![(z)_k \neq g(x, k)]\!] .$$

Binary Classification Error v.s. Ordinal Ranking Cost

Say $y = 5$, i.e., $\big((z)_1, (z)_2, \cdots, (z)_7\big) = $ (Y, Y, Y, Y, N, N, N)

- if $g_1(x, k)$ reports (Y, Y, N, N, N, N, N)
  - $g_1(x, k)$ made 2 errors
  - $r_{g_1}(x) = 3$; **squared** cost $= 4$
- if $g_3(x, k)$ reports consistent predictions (Y, N, N, N, N, N, N)
  - $g_3(x, k)$ made 3 errors
  - $r_{g_3}(x) = 2$; **squared** cost $= 9$

now 1 error can introduce up to 5 more in cost
—**how to take this into account?**

Importance of Associated Binary Queries

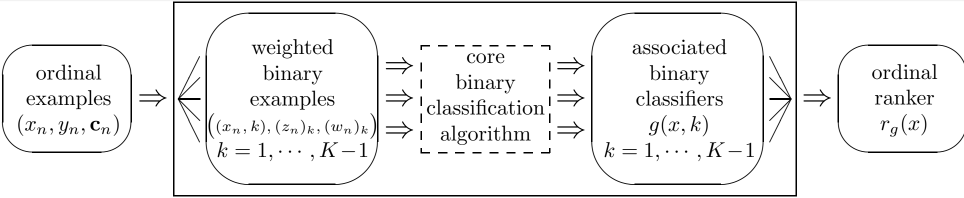
$(z)_k$	Y	Y	Y	Y	N	N	N	
$g_1(x,k)$	Y	Y	N	N	N	N	N	$\mathbf{c}\left[r_{g_1}(x)\right] = \mathbf{c}[3] = 4$
$g_3(x,k)$	Y	N	N	N	N	N	N	$\mathbf{c}\left[r_{g_3}(x)\right] = \mathbf{c}[2] = 9$
$(w)_k$	7	**5**	3	1	1	3	5	

- $(w)_k \equiv \left|\mathbf{c}[k+1] - \mathbf{c}[k]\right|$: the importance of $\left((x,k),(z)_k\right)$
- per-example cost bound (Li and Lin, 2007; Lin, 2008):
  for **consistent predictions** or **strongly ordinal costs**

$$\mathbf{c}\left[r_g(x)\right] \le \sum_{k=1}^{K-1} (w)_k \left[\!\left[(z)_k \ne g(x,k)\right]\!\right]$$

accurate binary predictions $\Longrightarrow$ correct ranks

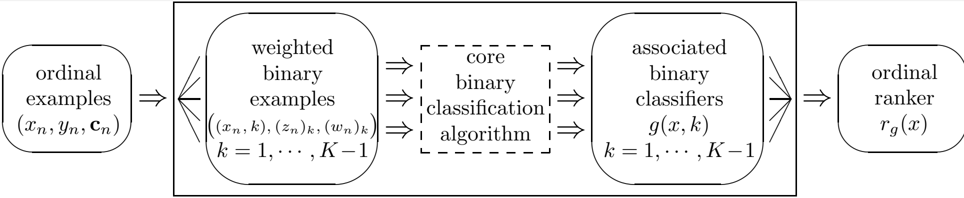
The Reduction Framework (1/2)



1. transform ordinal examples $(x_n, y_n, \mathbf{c}_n)$ to weighted binary examples $\big((x_n, k), (z_n)_k, (w_n)_k\big)$
2. apply your favorite algorithm and get one big joint binary classifier $g(x, k)$
3. for each new input x, predict its rank using $r_g(x) = 1 + \sum_k [\![g(x, k) = \text{Y}]\!]$

the reduction framework:
systematic & easy to implement

The Reduction Framework (2/2)



- **performance guarantee**:
  accurate binary predictions $\Longrightarrow$ correct ranks
- **wide applicability**:
  works with any ordinal **c** & any binary classification algorithm
- **simplicity**:
  mild computation overheads with $O(NK)$ binary examples
- **up-to-date**:
  allows new improvements in binary classification to be immediately inherited by ordinal ranking

Theoretical Guarantees of Reduction (1/3)

- is reduction a practical approach? **YES!**

error transformation theorem (Li and Lin, 2007)

For **consistent predictions** or **strongly ordinal costs**,
if g makes test error Δ in the induced binary problem,
then r_g pays test cost at most Δ in ordinal ranking.

- a one-step extension of the per-example cost bound
- conditions: general and minor
- performance guarantee in the absolute sense:

accuracy in binary classification $\Longrightarrow$ correctness in ordinal ranking

Is reduction really **optimal**?
—what if the induced binary problem is “too hard”?

Theoretical Guarantees of Reduction (2/3)

- is reduction an optimal approach? **YES!**

regret transformation theorem (Lin, 2008)

For a general class of **ordinal costs**,
if g is ϵ-close to the optimal binary classifier g_*,
then r_g is ϵ-close to the optimal ordinal ranker r_*.

- error guarantee in the relative setting:

  regardless of the absolute hardness of the induced binary prob.,
  optimality in binary classification $\Longrightarrow$ optimality in ordinal ranking

- reduction does not introduce additional hardness

"reduction to binary" sufficient, but necessary?
i.e., is reduction a **principled** approach?

Theoretical Guarantees of Reduction (3/3)

- is reduction a principled approach? **YES!**

equivalence theorem (Lin, 2008)

For a general class of **ordinal costs**,
ordinal ranking is learnable by a learning model
if and only if binary classification is learnable by the associated learning model.

- a surprising equivalence:

ordinal ranking is **as easy as** binary classification

reduction to binary classification:
practical, optimal, and principled

Outline

Unifying Existing Algorithms

ordinal ranking = reduction + cost + binary classification

ordinal ranking	cost	binary classification algorithm
PRank (Crammer and Singer, 2002)	absolute	modified perceptron rule
kernel ranking (Rajaram et al., 2003)	classification	modified hard-margin SVM
SVOR-EXP SVOR-IMC (Chu and Keerthi, 2005)	classification absolute	modified soft-margin SVM
ORBoost-LR ORBoost-All (Lin and Li, 2006)	classification absolute	modified AdaBoost

- correctness proof significantly simplified (PRank)
- algorithmic structure revealed (SVOR, ORBoost)

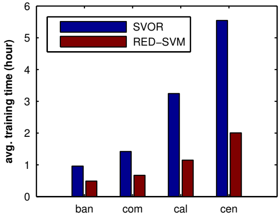
variants of existing algorithms can be designed quickly by tweaking reduction

Designing New Algorithms Effortlessly

ordinal ranking = reduction + cost + binary classification

ordinal ranking	cost	binary classification algorithm
Reduction-C4.5	absolute	standard C4.5 decision tree
Reduction-SVM	absolute	standard soft-margin SVM

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

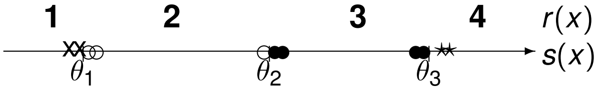


advantages of core binary classification algorithm inherited in the new ordinal ranking one

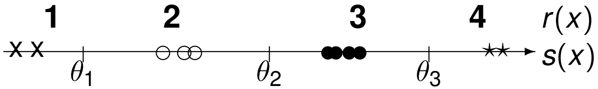
Outline

Recall: Threshold Model

- “bad” ordinal ranker: predictions close to thresholds
  —small noise changes prediction



- “good” ordinal ranker: clear separation using thresholds



next: good ordinal ranker $\Longrightarrow$ small expected test cost

Proving New Generalization Theorems

Ordinal Ranking (Li and Lin, 2007)

For SVOR or Reduction-SVM,
with probability $> 1 - \delta$,

$$\text{expected test abs. cost of } r \le \underbrace{\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1}\left[\!\left[\bar{\rho}(r(x_n), y_n, k) \le \Phi\right]\!\right]}_{\text{“goodness” in training}} + \underbrace{O\left(\text{poly}\left(K, \frac{\log N}{\sqrt{N}}, \frac{1}{\Phi}, \sqrt{\log\frac{1}{\delta}}\right)\right)}_{\text{deviation that decreases with more examples}}$$

Bi. Class. (Bartlett and Shawe-Taylor, 1998)

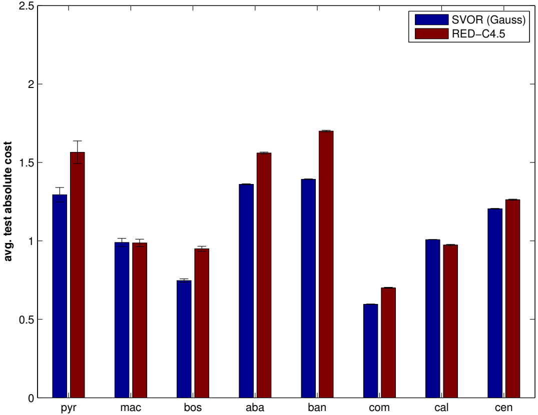
For SVM,
with probability $> 1 - \delta$,

$$\text{expected test err. of } g \le \underbrace{\frac{1}{N}\sum_{n=1}^{N}\left[\!\left[\bar{\rho}(g(x_n), y_n) \le \Phi\right]\!\right]}_{\text{“goodness” in training}} + \underbrace{O\left(\text{poly}\left(\frac{\log N}{\sqrt{N}}, \frac{1}{\Phi}, \sqrt{\log\frac{1}{\delta}}\right)\right)}_{\text{deviation that decreases with more examples}}$$

new ordinal ranking theorem
= reduction + any cost + bin. thm. + math derivation

Outline

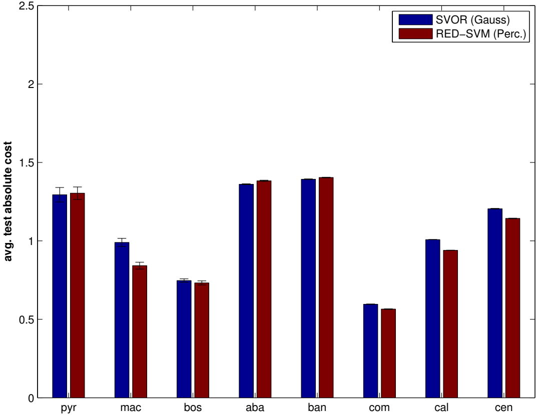
Reduction-C4.5 v.s. SVOR



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

even simple Reduction-C4.5 sometimes beats SVOR

Reduction-SVM v.s. SVOR



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

Reduction-SVM without modification often better than SVOR and faster

Outline

Conclusion

- reduction framework:
  practical, optimal, and principled
- algorithmic reduction:
  - take existing ones as **special cases**
  - design new and better ones **easily**
- theoretic reduction:
  - **new generalization guarantee** of ordinal rankers
- **superior** experimental results:
  better performance and faster training time

reduction keeps ordinal ranking up-to-date