Large-Margin Thresholded Ensembles for Ordinal Regression

Hsuan-Tien Lin
(accepted by ALT ’06, joint work with Ling Li)

Learning Systems Group, Caltech

Workshop Talk in MLSS 2006, Taipei, Taiwan, 07/25/2006
Ordinal Regression Problem

Reduction Method

**Algorithmic**
1. Identify the type of learning problem $B$
2. Find premade reduction $R$ and oracle learning algorithm $A$
3. Build a $B$ predictor using $R^A +$ data

**Theoretical**
1. Identify the type of learning problem (ordinal regression)
2. Find premade reduction (thresholded ensemble) and known generalization bounds (large-margin ensembles)
3. Derive new bound (large-margin thresholded ensembles) using the reduction + known bound
Ordinal Regression Problem

Reduction Method

### Algorithmic
1. Identify the type of learning problem (ordinal regression)
2. Find premade reduction (thresholded ensemble) and oracle learning algorithm (AdaBoost)
3. Build a ordinal regression rule using (ORBoost) + data

### Theoretical
1. Identify the type of learning problem (ordinal regression)
2. Find premade reduction (thresholded ensemble) and known generalization bounds (large-margin ensembles)
3. Derive new bound (large-margin thresholded ensembles) using the reduction + known bound

H.-T. Lin and L. Li (Learning Systems Group) 07/25/2006 2 / 12
Algorithmic

1. Identify the type of learning problem (ordinal regression).
2. Find premade reduction (thresholded ensemble) and oracle learning algorithm (AdaBoost).
3. Build a ordinal regression rule using (ORBoost) + data.

Theoretical

1. Identify the type of learning problem (ordinal regression).
2. Find premade reduction (thresholded ensemble) and known generalization bounds (large-margin ensembles).
3. Derive new bound (large-margin thresholded ensembles) using the reduction + known bound.
## Reduction Method

### Algorithmic

1. identify the type of learning problem (ordinal regression)
2. find premade reduction (thresholded ensemble) and oracle learning algorithm (AdaBoost)
3. build a ordinal regression rule using (ORBoost) + data

### Theoretical

1. identify the type of learning problem (ordinal regression)
2. find premade reduction (thresholded ensemble) and known generalization bounds (large-margin ensembles)
3. derive new bound (large-margin thresholded ensembles) using the reduction + known bound

---

this work: a concrete instance of reductions
Ordinal Regression

what is the age-group of the person in the picture?

- rank: a finite ordered set of labels $\mathcal{Y} = \{1, 2, \ldots, K\}$
- ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^{N}$, find a decision function $g$ that predicts the ranks of unseen examples well
- e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

• what is the age-group of the person in the picture?

  ? infant

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

• ordinal regression:
  given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well

• e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

- what is the age-group of the person in the picture?

  infant

  child

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

- ordinal regression:
  given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well

- e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

- what is the age-group of the person in the picture?
  - infant
  - child
  - teenager

- rank: a finite ordered set of labels $\mathcal{Y} = \{1, 2, \cdots, K\}$
- ordinal regression:
  - given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well
- e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

what is the age-group of the person in the picture?

- Infant
- Child
- Teenager
- Adult

- Rank: a finite ordered set of labels $\mathcal{Y} = \{1, 2, \ldots, K\}$
- Ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well
- E.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

- what is the age-group of the person in the picture?

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

- ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well

- e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

what is the age-group of the person in the picture?

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

ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well

e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

- what is the age-group of the person in the picture?

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

- ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well

- e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

what is the age-group of the person in the picture?

- rank: a finite ordered set of labels \( Y = \{1, 2, \ldots, K\} \)
- ordinal regression: given training set \( \{(x_n, y_n)\}_{n=1}^{N} \), find a decision function \( g \) that predicts the ranks of unseen examples well
  - e.g. ranking movies, ranking by document relevance, etc.
Ordinal Regression

- what is the age-group of the person in the picture?

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

- ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well

- e.g. ranking movies, ranking by document relevance, etc.
**Ordinal Regression**

- what is the age-group of the person in the picture?

- rank: a finite ordered set of labels $\mathcal{Y} = \{1, 2, \cdots, K\}$
- ordinal regression: given training set $\{(x_n, y_n)\}_{n=1}^N$, find a decision function $g$ that predicts the ranks of unseen examples well
- e.g. ranking movies, ranking by document relevance, etc.

**matching human preferences: applications in social science and info. retrieval**
Ordinal Regression Problem

Properties of Ordinal Regression

- regression without metric:
  - possibly metric underlying (age), but not encoded in \{1, 2, 3, 4\}
- classification with ordered categories:
  - small mistake – classify a teenager as a child;
  - big mistake – classify an infant as an adult
- common loss functions:
  - determine the category: classification error
    \[ L_C(g, x, y) = \begin{cases} \text{true} & \text{if } g(x) \neq y \\ \text{false} & \text{otherwise} \end{cases} \]
  - or at least have a close prediction: absolute error
    \[ L_A(g, x, y) = |g(x) - y| \]

will talk about \( L_A \) only;
similar for \( L_C \)
Ordinal Regression Problem

Properties of Ordinal Regression

- regression without metric:
  - possibly metric underlying (age), but not encoded in \{1, 2, 3, 4\}
- classification with ordered categories:
  - small mistake – classify a teenager as a child;
  - big mistake – classify an infant as an adult
- common loss functions:
  - determine the category: classification error
    \[ L_C(g, x, y) = [g(x) \neq y] \]
  - or at least have a close prediction: absolute error
    \[ L_A(g, x, y) = |g(x) - y| \]

will talk about \( L_A \) only; similar for \( L_C \)
Ordinal Regression Problem

Properties of Ordinal Regression

- regression without metric:
  - possibly metric underlying (age),
    but not encoded in \{1, 2, 3, 4\}

- classification with ordered categories:
  - small mistake – classify a teenager as a child;
    big mistake – classify an infant as an adult

- common loss functions:
  - determine the category: classification error
    \[ L_C(g, x, y) = \left[ g(x) \neq y \right] \]
  - or at least have a close prediction: absolute error
    \[ L_A(g, x, y) = |g(x) - y| \]

will talk about \( L_A \) only;
similar for \( L_C \)
Ordinal Regression Problem

Properties of Ordinal Regression

- regression without metric:
  - possibly metric underlying (age), but not encoded in \{1, 2, 3, 4\}

- classification with ordered categories:
  - small mistake – classify a teenager as a child;
  - big mistake – classify an infant as an adult

- common loss functions:
  - determine the category: classification error
    \[ L_C(g, x, y) = [g(x) \neq y] \]
  - or at least have a close prediction: absolute error
    \[ L_A(g, x, y) = |g(x) - y| \]
    will talk about \( L_A \) only;
    similar for \( L_C \)
Properties of Ordinal Regression

- regression without metric:
  - possibly metric underlying (age), but not encoded in \{1, 2, 3, 4\}

- classification with ordered categories:
  - small mistake – classify a teenager as a child;
  - big mistake – classify an infant as an adult

- common loss functions:
  - determine the category: classification error
    \[ L_C(g, x, y) = [g(x) \neq y] \]
  - or at least have a close prediction: absolute error
    \[ L_A(g, x, y) = |g(x) - y| \]

will talk about \( L_A \) only;
similar for \( L_C \)
Thresholded Ensemble Model

Thresholded Model for Ordinal Regression

- naive algorithm for ordinal regression:
  1. do general regression on \{ (x_n, y_n) \}, and get \( H(x) \)

- improved and generalized algorithm:
  1. estimate a potential function \( H(x) \)
  2. quantize \( H(x) \) by some ordered \( \theta \) to get \( g(x) \)

---

Thresholded Model for Ordinal Regression

- naive algorithm for ordinal regression:
  1. do general regression on \( \{(x_n, y_n)\} \), and get \( H(x) \)
  2. set \( g(x) = \text{clip}(\text{round}(H(x))) \)

- improved and generalized algorithm:
  1. estimate a potential function \( H(x) \)
  2. quantize \( H(x) \) by some ordered \( \theta \) to get \( g(x) \)

naive algorithm for ordinal regression:

1. do general regression on \( \{(x_n, y_n)\} \), and get \( H(x) \)
   - general regression performs badly without metric
2. set \( g(x) = \text{clip}(\text{round}(H(x))) \)

improved and generalized algorithm:

1. estimate a potential function \( H(x) \)
2. quantize \( H(x) \) by some ordered \( \theta \) to get \( g(x) \)
naive algorithm for ordinal regression:

1. do general regression on \((x_n, y_n)\), and get \(H(x)\)
   - general regression performs badly without metric
2. set \(g(x) = \text{clip} (\text{round}(H(x)))\)
   - roundoff operation (uniform quantization) cause large error

improved and generalized algorithm:

- estimate a potential function \(H(x)\)
- quantize \(H(x)\) by some ordered \(\theta\) to get \(g(x)\)
naive algorithm for ordinal regression:

1. do general regression on \( \{(x_n, y_n)\} \), and get \( H(x) \)
   - general regression performs badly without metric
2. set \( g(x) = \text{clip} (\text{round}(H(x))) \)
   - roundoff operation (uniform quantization) cause large error

improved and generalized algorithm:

1. estimate a potential function \( H(x) \)
2. quantize \( H(x) \) by some ordered \( \theta \) to get \( g(x) \)
naive algorithm for ordinal regression:

1. do general regression on \( \{(x_n, y_n)\} \), and get \( H(x) \)
   - general regression performs badly without metric
2. set \( g(x) = \text{clip}(\text{round}(H(x))) \)
   - roundoff operation (uniform quantization) cause large error

improved and generalized algorithm:

1. estimate a potential function \( H(x) \)
2. quantize \( H(x) \) by some ordered \( \theta \) to get \( g(x) \)
naive algorithm for ordinal regression:
1. do general regression on \((x_n, y_n)\), and get \(H(x)\)
   - general regression performs badly without metric
2. set \(g(x) = \text{clip} (\text{round} (H(x)))\)
   - roundoff operation (uniform quantization) cause large error

improved and generalized algorithm:
1. estimate a potential function \(H(x)\)
2. quantize \(H(x)\) by some ordered \(\theta\) to get \(g(x)\)
naive algorithm for ordinal regression:
1. do general regression on \( \{(x_n, y_n)\} \), and get \( H(x) \)
   - general regression performs badly without metric
2. set \( g(x) = \text{clip}(\text{round}(H(x))) \)
   - roundoff operation (uniform quantization) cause large error

improved and generalized algorithm:
1. estimate a potential function \( H(x) \)
2. quantize \( H(x) \) by some ordered \( \theta \) to get \( g(x) \)

thresholded model: \( g(x) \equiv g_{H,\theta}(x) = \min \{ k : H(x) < \theta_k \} \)
the potential function $H(x)$ is a weighted ensemble

$$H(x) \equiv H_T(x) = \sum_{t=1}^{T} w_t h_t(x)$$

intuition: combine preferences to estimate the overall confidence

e.g. if many people, $h_t$, say a movie $x$ is “good”,
the confidence of the movie $H(x)$ should be high

$h_t$ can be binary, multi-valued, or continuous

$w_t < 0$: allow reversing bad preferences

thresholded ensemble model:
ensemble learning for ordinal regression
the potential function $H(x)$ is a weighted ensemble

$$H(x) \equiv H_T(x) = \sum_{t=1}^{T} w_t h_t(x)$$

intuition: combine preferences to estimate the overall confidence

e.g. if many people, $h_t$, say a movie $x$ is “good”, the confidence of the movie $H(x)$ should be high

$h_t$ can be binary, multi-valued, or continuous

$w_t < 0$: allow reversing bad preferences

thresholded ensemble model: ensemble learning for ordinal regression
the potential function $H(x)$ is a weighted ensemble

$$H(x) \equiv H_T(x) = \sum_{t=1}^{T} w_t h_t(x)$$

intuition: combine preferences to estimate the overall confidence

e.g. if many people, $h_t$, say a movie $x$ is “good”,
the confidence of the movie $H(x)$ should be high

$h_t$ can be binary, multi-valued, or continuous

$w_t < 0$: allow reversing bad preferences
the potential function $H(x)$ is a weighted ensemble

$$H(x) \equiv H_T(x) = \sum_{t=1}^{T} w_t h_t(x)$$

intuition: combine preferences to estimate the overall confidence

e.g. if many people, $h_t$, say a movie $x$ is “good”,
the confidence of the movie $H(x)$ should be high

$h_t$ can be binary, multi-valued, or continuous

$w_t < 0$: allow reversing bad preferences
the potential function $H(x)$ is a weighted ensemble

$$H(x) \equiv H^T(x) = \sum_{t=1}^{T} w_t h_t(x)$$

intuition: combine preferences to estimate the overall confidence

e.g. if many people, $h_t$, say a movie $x$ is “good”,
the confidence of the movie $H(x)$ should be high

$h_t$ can be binary, multi-valued, or continuous

$w_t < 0$: allow reversing bad preferences
the potential function $H(x)$ is a weighted ensemble

\[ H(x) \equiv H_T(x) = \sum_{t=1}^{T} w_t h_t(x) \]

intuition: combine preferences to estimate the overall confidence
e.g. if many people, $h_t$, say a movie $x$ is “good”, the confidence of the movie $H(x)$ should be high

$h_t$ can be binary, multi-valued, or continuous

$w_t < 0$: allow reversing bad preferences
Bounds for Large-Margin Thresholded Ensembles

Margins of Thresholded Ensembles

- margin: safe from the boundary
- normalized margin for thresholded ensemble

\[ \bar{\rho}(x, y, k) = \begin{cases} 
H_T(x) - \theta_k, & \text{if } y > k \\
\theta_k - H_T(x), & \text{if } y \leq k 
\end{cases} \bigg/ \left( \sum_{t=1}^{T} |w_t| + \sum_{k=1}^{K-1} |\theta_k| \right) \]

- negative margin \iff \text{wrong prediction}

\[ \sum_{k=1}^{K-1} [\bar{\rho}(x, y, k) \leq 0] \iff |g(x) - y| \]
Margins of Thresholded Ensembles

- margin: safe from the boundary
- normalized margin for thresholded ensemble

$$\bar{\rho}(x, y, k) = \begin{cases} H_T(x) - \theta_k, & \text{if } y > k \\ \theta_k - H_T(x), & \text{if } y \leq k \end{cases} \div \left( \sum_{t=1}^{T} |w_t| + \sum_{k=1}^{K-1} |\theta_k| \right)$$

negative margin $\iff$ wrong prediction

$$\sum_{k=1}^{K-1} [\bar{\rho}(x, y, k) \leq 0] \iff |g(x) - y|$$
Margins of Thresholded Ensembles

- margin: safe from the boundary
- normalized margin for thresholded ensemble

\[ \bar{\rho}(x, y, k) = \begin{cases} H_T(x) - \theta_k, & \text{if } y > k \\ \theta_k - H_T(x), & \text{if } y \leq k \end{cases} \]

\[
\sum_{k=1}^{K-1} \left[ \bar{\rho}(x, y, k) \leq 0 \right] \iff |g(x) - y|
\]

negative margin \( \iff \) wrong prediction
Bounds for Large-Margin Thresholded Ensembles

Theoretical Reduction

- (K – 1) binary classification problems w.r.t. each \( \theta_k \):
  \[ ((X)_k, (Y)_k) = ((x, k), +/-) \]

  (Schapire et al., 1998) binary classification: with probability at least 1 – \( \delta \), for all \( \Delta > 0 \) and binary classifiers \( g_c \),

  \[
  \mathcal{E}_{(X,Y) \sim D'}[g_c(X) \neq Y] \leq \frac{1}{N} \sum_{n=1}^{N} \left[ \bar{\rho}(X_n, Y_n) \leq \Delta \right] + O\left( \frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}} \right)
  \]

- (Lin and Li, 2006) ordinal regression: with similar settings, for all thresholded ensembles \( g \),

  \[
  \mathcal{E}_{(x,y) \sim D} L_A(g, x, y) \leq \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K-1} \left[ \bar{\rho}(x_n, y_n, k) \leq \Delta \right] + O\left( K, \frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}} \right)
  \]

large-margin thresholded ensembles can generalize
**Theoretical Reduction**

1. \((K - 1)\) binary classification problems w.r.t. each \(\theta_k: ((X)_k, (Y)_k) = ((x, k), +/-)\)

2. (Schapire et al., 1998) binary classification: with probability at least \(1 - \delta\), for all \(\Delta > 0\) and binary classifiers \(g_c\),

\[
\mathcal{E}_{(X,Y) \sim D'}[g_c(X) \neq Y] \leq \frac{1}{N} \sum_{n=1}^{N} [\bar{\rho}(X_n, Y_n) \leq \Delta] + O\left(\frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}}\right)
\]

3. (Lin and Li, 2006) ordinal regression: with similar settings, for all thresholded ensembles \(g\),

\[
\mathcal{E}_{(x,y) \sim D} L_A(g, x, y) \leq \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K-1} [\bar{\rho}(x_n, y_n, k) \leq \Delta] + O(K, \frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}})
\]

large-margin thresholded ensembles can generalize
Bounds for Large-Margin Thresholded Ensembles

Theoretical Reduction

- $(K - 1)$ binary classification problems w.r.t. each $\theta_k$:
  $((X)_k, (Y)_k) = ((x, k), +/−)$

  (Schapire et al., 1998) binary classification: with probability at least $1 - \delta$, for all $\Delta > 0$ and binary classifiers $g_c$,

  $$
  \mathcal{E}_{(X,Y)\sim D'}[g_c(X) \neq Y] \leq \frac{1}{N} \sum_{n=1}^{N} [\bar{\rho}(X_n, Y_n) \leq \Delta] + O(\frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}})
  $$

- (Lin and Li, 2006) ordinal regression: with similar settings, for all thresholded ensembles $g$,

  $$
  \mathcal{E}_{(x,y)\sim D} L_A(g, x, y) \leq \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K-1} [\bar{\rho}(x_n, y_n, k) \leq \Delta] + O(K, \frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}})
  $$

large-margin thresholded ensembles can generalize
Bounds for Large-Margin Thresholded Ensembles

Theoretical Reduction

$(K - 1)$ binary classification problems w.r.t. each $\theta_k$:

$$((X)_k, (Y)_k) = ((x, k), +/−)$$

(Schapire et al., 1998) binary classification: with probability at least $1 − \delta$, for all $\Delta > 0$ and binary classifiers $g_c$,

$$\mathcal{E}_{(X, Y) \sim \mathcal{D}'} [g_c(X) \neq Y] \leq \frac{1}{N} \sum_{n=1}^{N} \bar{\rho}(X_n, Y_n) \leq \Delta + O\left(\frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}}\right)$$

(Lin and Li, 2006) ordinal regression: with similar settings, for all thresholded ensembles $g$,

$$\mathcal{E}_{(x, y) \sim \mathcal{D}} L_A(g, x, y) \leq \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K-1} \bar{\rho}(x_n, y_n, k) \leq \Delta + O(K, \frac{\log N}{\sqrt{N}}, \frac{1}{\Delta}, \sqrt{\log \frac{1}{\delta}})$$

large-margin thresholded ensembles can generalize
(Freund and Schapire, 1996) AdaBoost: binary classification by operationally optimizing

$$\min_{n=1}^{N} \sum_{n} \exp(-\rho(x_n, y_n)) \approx \max \text{softmin}_n \tilde{\rho}(x_n, y_n)$$

(Lin and Li, 2006)

**ORBoost-LR (left-right):**

$$\min \sum_{n=1}^{N} \sum_{k=y_n-1}^{y_n} \exp(-\rho(x_n, y_n, k))$$

**ORBoost-All:**

$$\min \sum_{n=1}^{N} \sum_{k=1}^{K-1} \exp(-\rho(x_n, y_n, k))$$

algorithmic reduction to AdaBoost
Algorithmic Reduction

(Freund and Schapire, 1996) AdaBoost: binary classification by operationally optimizing

$$\min_{n=1}^{N} \sum \exp(-\rho(x_n, y_n)) \approx \max \text{softmin}_n \tilde{\rho}(x_n, y_n)$$

(Lin and Li, 2006)

**ORBoost-LR (left-right):**

$$\min \sum_{n=1}^{N} \sum_{k=y_n-1}^{y_n} \exp(-\rho(x_n, y_n, k))$$

**ORBoost-All:**

$$\min \sum_{n=1}^{N} \sum_{k=1}^{K-1} \exp(-\rho(x_n, y_n, k))$$

algorithmic reduction to AdaBoost
Algorithmic Reduction

(Freund and Schapire, 1996) AdaBoost: binary classification by operationally optimizing

\[
\min \sum_{n=1}^{N} \exp(-\rho(x_n, y_n)) \approx \max \text{softmin}_n \bar{\rho}(x_n, y_n)
\]

(Lin and Li, 2006)

ORBoost-LR (left-right):

\[
\min \sum_{n=1}^{N} \sum_{k=y_n-1}^{y_n} \exp(-\rho(x_n, y_n, k))
\]

ORBoost-All:

\[
\min \sum_{n=1}^{N} \sum_{k=1}^{K-1} \exp(-\rho(x_n, y_n, k))
\]

algorithmic reduction to AdaBoost
Advantages of ORBoost

- ensemble learning: combine simple preferences to approximate complex targets
- threshold: adaptively estimated scales to perform ordinal regression
- inherit from AdaBoost:
  - simple implementation
  - guarantee on minimizing $\sum_{n,k} \bar{\rho}(x_n, y_n, k) \leq \Delta$ fast
  - practically less vulnerable to overfitting
Advantages of ORBoost

- ensemble learning: combine simple preferences to approximate complex targets
- threshold: adaptively estimated scales to perform ordinal regression
- inherit from AdaBoost:
  - simple implementation
  - guarantee on minimizing $\sum_{n,k} [\tilde{\rho}(x_n, y_n, k) \leq \Delta]$ fast
  - practically less vulnerable to overfitting

useful properties inherited with reduction
Advantages of ORBoost

- ensemble learning: combine simple preferences to approximate complex targets
- threshold: adaptively estimated scales to perform ordinal regression
- inherit from AdaBoost:
  - simple implementation
  - guarantee on minimizing $\sum_{n,k} [\bar{\rho}(x_n, y_n, k) \leq \Delta]$ fast
  - practically less vulnerable to overfitting

useful properties inherited with reduction
Advantages of ORBoost

- ensemble learning: combine simple preferences to approximate complex targets
- threshold: adaptively estimated scales to perform ordinal regression
- inherit from AdaBoost:
  - simple implementation
  - guarantee on minimizing $\sum_{n,k} \bar{\rho}(x_n, y_n, k) \leq \Delta$ fast
  - practically less vulnerable to overfitting
Advantages of ORBoost

- ensemble learning: combine simple preferences to approximate complex targets
- threshold: adaptively estimated scales to perform ordinal regression
- inherit from AdaBoost:
  - simple implementation
  - guarantee on minimizing $\sum_{n,k} [\bar{\rho}(x_n, y_n, k) \leq \Delta]$ fast
  - practically less vulnerable to overfitting
Advantages of ORBoost

- ensemble learning: combine simple preferences to approximate complex targets
- threshold: adaptively estimated scales to perform ordinal regression
- inherit from AdaBoost:
  - simple implementation
  - guarantee on minimizing $\sum_{n,k} \bar{\rho}(x_n, y_n, k) \leq \Delta$ fast
  - practically less vulnerable to overfitting
ORBoost Experiments

Results (ORBoost-All)
- ORBoost-All simpler, and much better than RankBoost (Freund et al., 2003)
- ORBoost-All much faster, and comparable to SVM (Chu and Keerthi, 2005)
- similar for ORBoost-LR
Results (ORBoost-All)

- ORBoost-All simpler, and much better than RankBoost (Freund et al., 2003)
- ORBoost-All much faster, and comparable to SVM (Chu and Keerthi, 2005)

similar for ORBoost-LR
Results (ORBoost-All)

- ORBoost-All simpler, and much better than RankBoost (Freund et al., 2003)
- ORBoost-All much faster, and comparable to SVM (Chu and Keerthi, 2005)
- similar for ORBoost-LR
Conclusion

- thresholded ensemble model: useful for ordinal regression
  - theoretical reduction: new large-margin bounds
  - algorithmic reduction: new training algorithms – ORBoost

ORBoost:
- simplicity over existing boosting algorithms
- comparable performance to state-of-the-art algorithms
- fast training and less vulnerable to overfitting

on-going work: similar reduction technique for other theoretical and algorithmic results with more general loss functions (Li and Lin, 2006)
thresholded ensemble model: useful for ordinal regression
  - theoretical reduction: new large-margin bounds
  - algorithmic reduction: new training algorithms – ORBoost

ORBoost:
  - simplicity over existing boosting algorithms
  - comparable performance to state-of-the-art algorithms
  - fast training and less vulnerable to overfitting

on-going work: similar reduction technique for other theoretical and algorithmic results with more general loss functions (Li and Lin, 2006)
thresholded ensemble model: useful for ordinal regression
- theoretical reduction: new large-margin bounds
- algorithmic reduction: new training algorithms – ORBoost

ORBoost:
- simplicity over existing boosting algorithms
- comparable performance to state-of-the-art algorithms
- fast training and less vulnerable to overfitting

on-going work: similar reduction technique for other theoretical and algorithmic results with more general loss functions (Li and Lin, 2006)
Conclusion

- thresholded ensemble model: useful for ordinal regression
- theoretical reduction: new large-margin bounds
- algorithmic reduction: new training algorithms – ORBoost

ORBoost:
- simplicity over existing boosting algorithms
- comparable performance to state-of-the-art algorithms
- fast training and less vulnerable to overfitting

on-going work: similar reduction technique for other theoretical and algorithmic results with more general loss functions (Li and Lin, 2006)