Pranking with Ranking

Koby Crammer and Yoram Singer

Presented by : Soham Dan

Content and some figures borrowed from [Crammer, Koby, and Yoram Singer. Pranking with ranking.NIPS. 2002] and talk slides.

Introduction

- Problem
  - Input : Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$
  - Output : A model(essentially a rank prediction rule) which assigns to each instance a rank.
  - Goal: To have the predicted rank as close as possible to the true rank.
  - Note : The ranks need not be unique!

Introduction

- Problem
  - Input : Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$
  - Output : A model(essentially a rank prediction rule) which assigns to each instance a rank.
  - Goal: To have the predicted rank as close as possible to the true rank.
  - Note : The ranks need not be unique!
- Similarity with
  - Classification Problems : Assign one of k possible labels to a new instance.
  - Regression Problems : Set of k labels is structured as there is a total order relation between labels.

Introduction

- Problem
  - Input : Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$
  - Output : A model(essentially a rank prediction rule) which assigns to each instance a rank.
  - Goal: To have the predicted rank as close as possible to the true rank.
  - Note : The ranks need not be unique!
- Similarity with
  - Classification Problems : Assign one of k possible labels to a new instance.
  - Regression Problems : Set of k labels is structured as there is a total order relation between labels.

Natural Settings to rank / rate instances

Information Retrieval , Collaborative Filtering

Problem

Figure 1: Movie rating prediction (Example : Netflix challenge)

Possible Solutions

- Cast as a regression or classification problem

Possible Solutions

- Cast as a regression or classification problem
- Reduce a total order into a set of preference over pairs. Drawback : Sample size blowup from n to $\emptyset(n^2)$. Also, no easy adaptation for online settings.

Possible Solutions

- Cast as a regression or classification problem
- Reduce a total order into a set of preference over pairs. Drawback : Sample size blowup from n to $\emptyset(n^2)$. Also, no easy adaptation for online settings.
- PRank Algorithm : Directly maintains totally ordered set by projection of instances into reals, associating ranks with distinct sub-intervals of the reals and adapting the support of each subinterval while learning.

Problem Setup

- Input Stream: Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$ where each instance $x_t \in \mathbb{R}^n$. Corresponding rank $y^t \in \mathcal{Y}$ which is a finite set with a total order relation (structured) . W.l.o.g. $\mathcal{Y} = 1, 2, 3..., k$ with $>$ as the order relation. $1 \prec 2 \prec ... \prec k$

Problem Setup

- Input Stream: Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$ where each instance $x_t \in \mathbb{R}^n$. Corresponding rank $y^t \in \mathcal{Y}$ which is a finite set with a total order relation (structured) . W.l.o.g. $\mathcal{Y} = 1, 2, 3..., k$ with $>$ as the order relation. $1 \prec 2 \prec ... \prec k$
- Ranking Rule $(\mathcal{H})$: Mapping from instances to ranks, $\mathbb{R}^n \to \mathcal{Y}$. The family of ranking rules considered here : $\mathrm{w} \in \mathbb{R}^n$ and k thresholds : $b_1 \leq b_2 \leq ... \leq b_k = \infty$

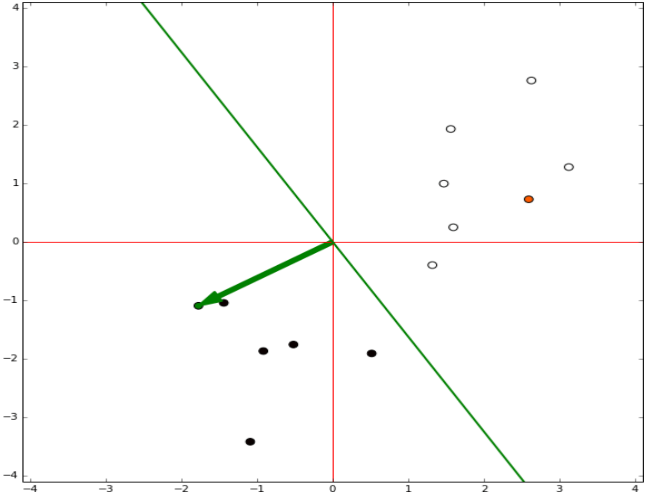
Problem Setup

- Input Stream: Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$ where each instance $x_t \in \mathbb{R}^n$. Corresponding rank $y^t \in \mathcal{Y}$ which is a finite set with a total order relation (structured) . W.l.o.g. $\mathcal{Y} = 1, 2, 3..., k$ with $>$ as the order relation. $1 \prec 2 \prec ... \prec k$
- Ranking Rule $(\mathcal{H})$: Mapping from instances to ranks, $\mathbb{R}^n \to \mathcal{Y}$. The family of ranking rules considered here : $\mathrm{w} \in \mathbb{R}^n$ and k thresholds : $b_1 \leq b_2 \leq ... \leq b_k = \infty$
- Given a ranking rule defined by w and b, the predicted rank $(\hat{y}^t)$ on a new instance x is $H(x) = \min_{r \in 1,2,..,k} \{r : w \cdot x - b_r < 0\}$

Problem Setup

- Input Stream: Sequence of instance-rank pairs $(x^1, y^1)...(x^t, y^t)$ where each instance $x_t \in \mathbb{R}^n$. Corresponding rank $y^t \in \mathcal{Y}$ which is a finite set with a total order relation (structured) . W.l.o.g. $\mathcal{Y} = 1, 2, 3..., k$ with $>$ as the order relation. $1 \prec 2 \prec ... \prec k$
- Ranking Rule $(\mathcal{H})$: Mapping from instances to ranks, $\mathbb{R}^n \to \mathcal{Y}$. The family of ranking rules considered here : $\mathrm{w} \in \mathbb{R}^n$ and k thresholds : $b_1 \leq b_2 \leq ... \leq b_k = \infty$
- Given a ranking rule defined by w and b, the predicted rank $(\hat{y}^t)$ on a new instance x is $H(x) = \min_{r \in 1,2,..,k} \{r : w \cdot x - b_r < 0\}$
- Algorithm makes a mistake on instance x^t if $\hat{y}^t \neq y^t$ and loss on that input is $|\hat{y}^t - y^t|$.
- Loss after T rounds is $\sum_{t=1}^{T} |\hat{y}^t - y^t|$

Perceptron Recap



Overview of Algorithm

- Online Algorithm

Overview of Algorithm

- Online Algorithm
- In each round the ranking algorithm
  - Gets an input instance
  - Outputs the rank as prediction
  - Receives the correct rank value
  - If there is an error
    - Computes loss
    - Updates the rank-prediction rule

Overview of Algorithm

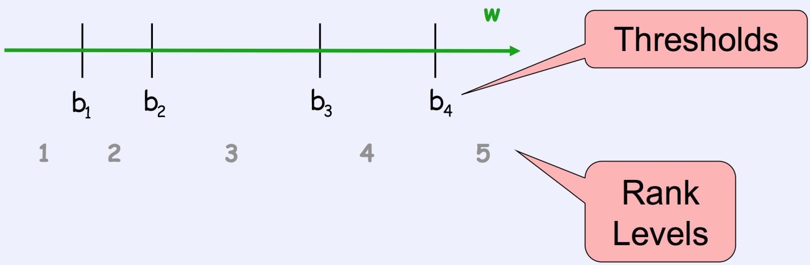
- Online Algorithm
- In each round the ranking algorithm
  - Gets an input instance
  - Outputs the rank as prediction
  - Receives the correct rank value
  - If there is an error
    - Computes loss
    - Updates the rank-prediction rule
- Conservative or Mistake driven algorithm :The algorithm updates its ranking rule only on rounds on which it made ranking mistakes.

Overview of Algorithm

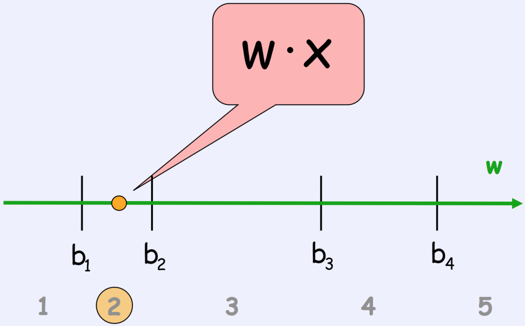
- Online Algorithm
- In each round the ranking algorithm
  - Gets an input instance
  - Outputs the rank as prediction
  - Receives the correct rank value
  - If there is an error
    - Computes loss
    - Updates the rank-prediction rule
- Conservative or Mistake driven algorithm :The algorithm updates its ranking rule only on rounds on which it made ranking mistakes.
- No statistical assumptions over data.The algorithm should do well irrespectively of specific sequence of inputs and target labels

Algorithm Illustration

- Direction **w**, Thresholds $b_{k-1}, \ldots, b_1$

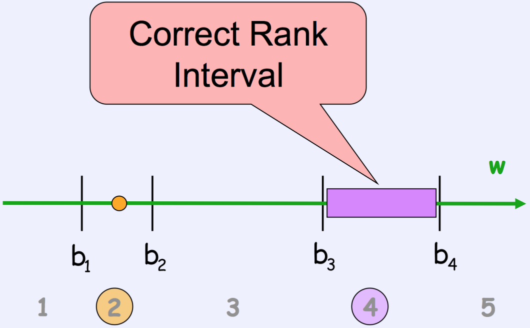


Algorithm Illustration



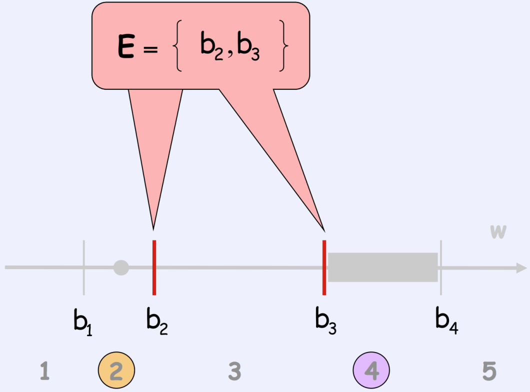
- Direction **w**,
  Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance **x**

Algorithm Illustration



- Direction **w**, Thresholds $b_{k-1},\ldots,b_1$
- Rank a new instance **x**
- Get the correct rank **y**

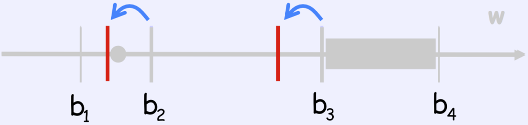
Algorithm Illustration



- Direction **w**, Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance **x**
- Get the correct rank **y**
- Compute Error-Set **E**

Algorithm Illustration

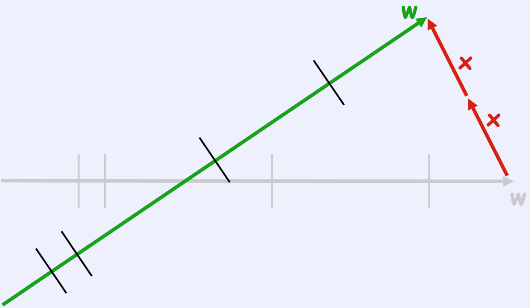
$$\mathbf{E} = \{ b_2, b_3 \}$$



- Direction **w**, Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance $\mathbf{x}$
- Get the correct rank $\mathbf{y}$
- Compute Error-Set $\mathbf{E}$
- Update :
  - $b_r \leftarrow b_r - 1 \quad r \in \mathbf{E}$

Algorithm Illustration

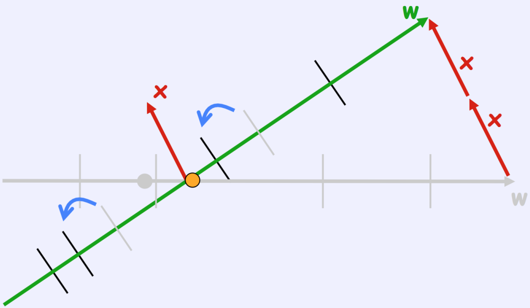
$$\mathbf{E} = \{\ b_2, b_3\ \}$$



- Direction **w**, Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance **x**
- Get the correct rank **y**
- Compute Error-Set **E**
- Update :
  - $b_r \leftarrow b_r - 1 \quad r \in \mathbf{E}$
  - $\mathbf{w} \leftarrow \mathbf{w} + |\mathbf{E}|\mathbf{x}$

Algorithm Illustration

$$E = \{ b_2, b_3 \}$$



- Direction **w**,
  Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance **x**
- Get the correct rank **y**
- Compute Error-Set **E**
- Update :
  - $b_r \leftarrow b_r - 1 \quad r \in E$
  - $\mathbf{w} \leftarrow \mathbf{w} + |E| \mathbf{x}$

Algorithm

Initialize: Set $\mathbf{w}^1 = 0$, $b_1^1, \ldots, b_{k-1}^1 = 0, b_k^1 = \infty$.
Loop: For $t = 1, 2, \ldots, T$

- Get a new rank-value $\mathbf{x}^t \in \mathbb{R}^n$.
- Predict $\hat{y}^t = \min_{r \in \{1,\ldots,k\}} \{r : \mathbf{w}^t \cdot \mathbf{x}^t - b_r^t < 0\}$.
- Get a new label y^t.
- If $\hat{y}^t \neq y^t$ update $\mathbf{w}^t$ (otherwise set $\mathbf{w}^{t+1} = \mathbf{w}^t$, $\forall r : b_r^{t+1} = b_r^t$) :
  1. For $r = 1, \ldots, k-1$: If $y^t \leq r$ Then $y_r^t = -1$
     Else $y_r^t = 1$.
  2. For $r = 1, \ldots, k-1$: If $(\mathbf{w}^t \cdot \mathbf{x}^t - b_r^t) y_r^t \leq 0$ Then $\tau_r^t = y_r^t$
     Else $\tau_r^t = 0$.
  3. Update $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + (\sum_r \tau_r^t)\mathbf{x}^t$.
     For $r = 1, \ldots, k-1$ update: $b_r^{t+1} \leftarrow b_r^t - \tau_r^t$

Output : $H(\mathbf{x}) = \min_{r \in \{1,\ldots,k\}} \{r : \mathbf{w}^{T+1} \cdot \mathbf{x} - b_r^{T+1} < 0\}$.

Figure 2: The PRank Algorithm

Algorithm

Initialize: Set $\mathbf{w}^1 = 0$, $b_1^1, \ldots, b_{k-1}^1 = 0, b_k^1 = \infty$.
Loop: For $t = 1, 2, \ldots, T$

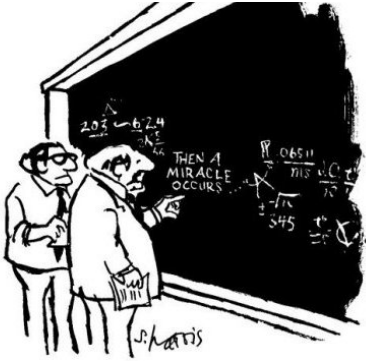
- Get a new rank-value $\mathbf{x}^t \in \mathbb{R}^n$.
- Predict $\hat{y}^t = \min_{r \in \{1,\ldots,k\}}\{r : \mathbf{w}^t \cdot \mathbf{x}^t - b_r^t < 0\}$.
- Get a new label y^t.
- If $\hat{y}^t \neq y^t$ update $\mathbf{w}^t$ (otherwise set $\mathbf{w}^{t+1} = \mathbf{w}^t$, $\forall r : b_r^{t+1} = b_r^t$) :
  1. For $r = 1, \ldots, k-1$: If $y^t \leq r$ Then $y_r^t = -1$
     Else $y_r^t = 1$.
  2. For $r = 1, \ldots, k-1$: If $(\mathbf{w}^t \cdot \mathbf{x}^t - b_r^t)y_r^t \leq 0$ Then $\tau_r^t = y_r^t$
     Else $\tau_r^t = 0$.
  3. Update $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + (\sum_r \tau_r^t)\mathbf{x}^t$.
     For $r = 1, \ldots, k-1$ update: $b_r^{t+1} \leftarrow b_r^t - \tau_r^t$

Output : $H(\mathbf{x}) = \min_{r \in \{1,\ldots,k\}}\{r : \mathbf{w}^{T+1} \cdot \mathbf{x} - b_r^{T+1} < 0\}$.

Figure 2: The PRank Algorithm

- Rank y is expanded into $k-1$ virtual variables $y_1, .., y_{k-1}$, where $y_r = +1$ if $w \cdot x > b_r$ and $y_r = -1$ otherwise.
- On mistakes, b and $w \cdot x$ are moved towards each other.

Analysis



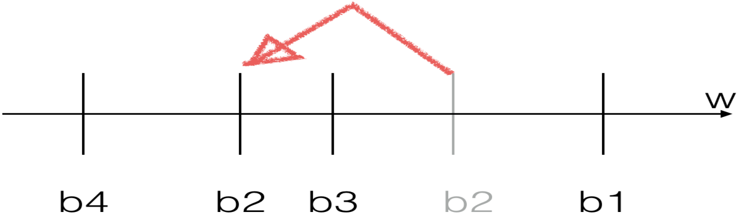
"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

1. Lemma : Order Preservation

2. Theorem : Mistake Bound

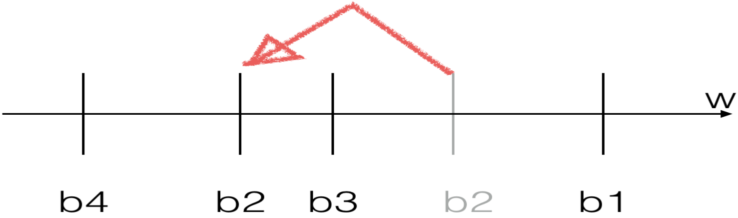
Lemma : Order Preservation

Can this happen ?



Lemma : Order Preservation

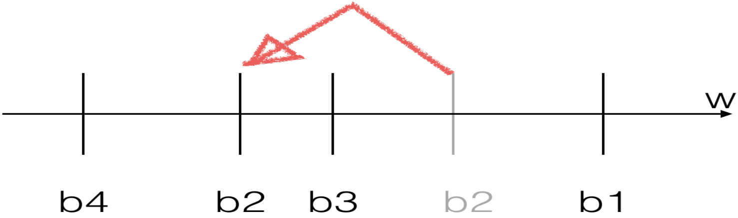
Can this happen ?



NO

Lemma : Order Preservation

Can this happen ?



NO

Let w_t and b_t be the current ranking rule, where $b_1^t \leq \ldots \leq b_{k-1}^t$ and let (x_t, y_t) be an instance-rank pair fed to PRank on round t. Denote by w_{t+1} and b_{t+1} the resulting ranking rule after the update of PRank, then $b_1^{t+1} \leq \ldots \leq b_{k-1}^{t+1}$

Lemma : Order Preservation

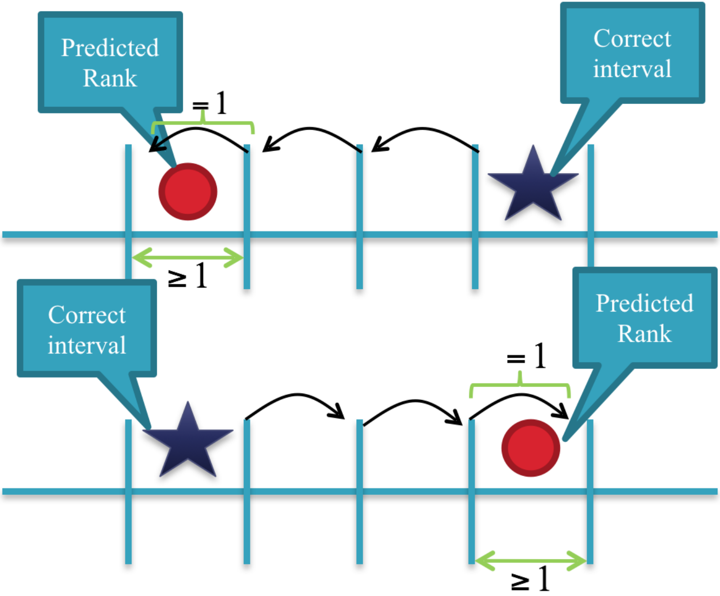
Let w_t and b_t be the current ranking rule, where $b_1^t \leq ... \leq b_{k-1}^t$ and let (x_t, y_t) be an instance-rank pair fed to PRank on round t. Denote by w_{t+1} and b_{t+1} the resulting ranking rule after the update of PRank, then $b_1^{t+1} \leq ... \leq b_{k-1}^{t+1}$

Proof Sketch :

- b_r^t are integers for all r and t since for all r we initialize $b_r^1 = 0$, and $b_r^{t+1} - b_r^t \in \{-1, 0, +1\}$.
- Proof by Induction :
  Showing $b_{r+1}^{t+1} \geq b_r^{t+1}$ is equivalent to proving

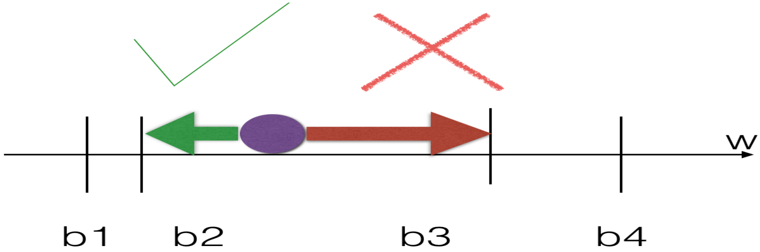
$$b_{r+1}^t - b_r^t \geq y_{r+1}^t[(w_t \cdot x_t - b_{r+1}^t) y_{r+1}^t \leq 0] - y_r^t[(w_t \cdot x_t - b_r^t) y_r^t \leq 0]$$

Lemma : Order Preservation



Theorem : Mistake Bound

Let $(x_I, y_1), ..., (x_T, y_T)$ be an input sequence for PRank where $x_t \in \mathbb{R}^n$ and $y_t \in I, ..., k$. Denote by $R^2 = \max_t ||x_t||^2$. Assume that there is a ranking rule $v^* = (w^*, b^*)$ with $b_1^* \leq ... \leq b_{k-1}^*$ of a unit norm that classifies the entire sequence correctly with margin $\gamma = \min_{r,t}(w^* \cdot x_t - b_r^*)y_r^t > 0$. Then, the rank loss of the algorithm $\sum_{t=1}^T |\hat{y}^t - y^t|$, is at most $\frac{(k-1)(R^2+1)}{\gamma^2}$.



Proof of Theorem

- $w_{t+1} = w_t + (\sum_r \tau_r^t) x_t$ and $b_r^{t+1} = b_r^t - \tau_r^t$
- Let $n_t = |\hat{y}^t - y^t|$ be difference between the true rank and the predicted rank. Clearly, $n^t = \sum_r |\tau_r^t|$
- To prove the theorem we bound $\sum_t n^t$ from above by bounding $||v^t||^2$ from above and below.
- $v^* \cdot v^{t+1} = v^* \cdot v^t + \sum_{r=1}^{k-1} \tau_r^t (w^* x^t - b_r^*)$
- $\sum_{r=1}^{k-1} \tau_r^t (w^* x^t - b_r^*) \geq n^t \gamma \implies v^* v^{T+1} \geq \gamma \sum_t n^t \implies ||v^{T+1}||^2 \geq \gamma^2 (\sum_t n^t)^2$

- To bound the norm of v from above :
- $||v^{t+1}||^2 = ||w^t||^2 + ||b^t||^2 + 2\sum_r \tau_r^t(w^t \cdot x^t - b_r^t) + (\sum_r \tau_r^t)^2||x^t||^2 + \sum_r(\tau_r^t)^2$
- Since, $(\sum_r \tau_r^t)^2 \le (n^t)^2$ and $\sum_r(\tau_r^t)^2 = n^t$
- $||v^{t+1}||^2 = ||v^t||^2 + 2\sum_r \tau_r^t(w^t \cdot x^t - b_r^t) + (n^t)^2||x^t||^2 + n^t$
- $\sum_r \tau_r^t(w^t \cdot x^t - b_r^t) = \sum_r[(w^t \cdot x^t - b_r^t) \le 0](w^t \cdot x^t - b_r^t)y_r \le 0$
- Since, $||x^t||^2 \le R^2 \implies ||v^{t+1}||^2 = ||v^t||^2 + (n^t)^2 R^2 + n^t$
- Using the lower bound, we get, $\sum_t n^t \le \frac{R^2[\sum_t (n^t)^2]/[\sum_t n^t]+1}{\gamma^2}$
- $n^t \le k-1 \implies \sum_t (n^t)^2 \le (k-1)\sum_t n^t \implies \sum_t n^t \le \frac{(k-1)(R^2+1)}{\gamma^2}$

Experiments

Experiments

Experiments

- Models
  - Multi-class Generalization of Perceptron (MCP) : kn parameters : under-constrained
  - Widrow Hoff Algorithm for Online Regression (WH): n parameters : over-constrained
  - PRank : $n + k - 1$ parameters : accurately constrained

Experiments

- Models
  - Multi-class Generalization of Perceptron (MCP) : kn parameters : under-constrained
  - Widrow Hoff Algorithm for Online Regression (WH): n parameters : over-constrained
  - PRank : $n + k - 1$ parameters : accurately constrained
- Datasets
  - Synthetic dataset
  - EachMovie dataset-used for collaborative filtering tasks
  - Evaluation in batch setting- outperforms multi-class SVM, SVR

Experiments

- Models
  - Multi-class Generalization of Perceptron (MCP) : kn parameters : under-constrained
  - Widrow Hoff Algorithm for Online Regression (WH): n parameters : over-constrained
  - PRank : $n + k - 1$ parameters : accurately constrained
- Datasets
  - Synthetic dataset
  - EachMovie dataset-used for collaborative filtering tasks
  - Evaluation in batch setting- outperforms multi-class SVM, SVR

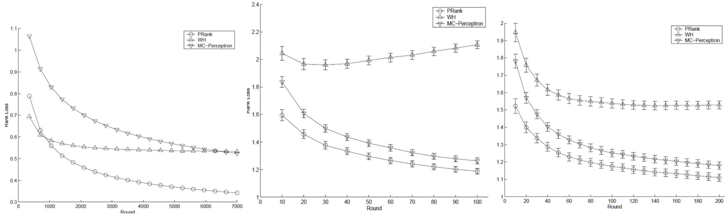


Figure 4: Time-averaged ranking-loss comparison of MCP,WH,PRank on the synthetic dataset, EachMovie-100 and 200 datasets respectively

Key takeaways

Key takeaways

1. The ranking problem is a structured prediction task because of the total order between the different ratings.

Key takeaways

1. The ranking problem is a structured prediction task because of the total order between the different ratings.
2. Online algorithm for ranking problem via projections and conservative update of the projection's direction and the threshold values.

Key takeaways

1. The ranking problem is a structured prediction task because of the total order between the different ratings.
2. Online algorithm for ranking problem via projections and conservative update of the projection's direction and the threshold values.
3. Experiments indicate this algorithm performs better than regression and classification models for ranking tasks.

Further Reading

Types of Ranking Algorithms:

- Point-wise Approaches - PRanking
- Pair-wise Approaches - RankSVM, RankNet, Rankboost
- List-wise Approaches - SVM^{map}, AdaRank, SoftRank

Further Reading

Types of Ranking Algorithms:

- Point-wise Approaches - PRanking
- Pair-wise Approaches - RankSVM, RankNet, Rankboost
- List-wise Approaches - SVM^{map}, AdaRank, SoftRank

References:

- Liu, Tie-Yan. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval 3.3 (2009): 225-331.

Further Reading

Types of Ranking Algorithms:

- Point-wise Approaches - PRanking
- Pair-wise Approaches - RankSVM, RankNet, Rankboost
- List-wise Approaches - SVM^{map}, AdaRank, SoftRank

References:

- Liu, Tie-Yan. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval 3.3 (2009): 225-331.
- Agarwal, Shivani, and Partha Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. Journal of Machine Learning Research 10.Feb (2009): 441-474.