Decision-theoretic Machine Learning

Krzysztof Dembczyński and Wojciech Kotłowski

Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland

Poznań University of Technology, Summer 2019

Agenda

1. Introduction to Machine Learning
2. Binary Classification
3. **Bipartite Ranking**
4. Multi-Label Classification

Outline

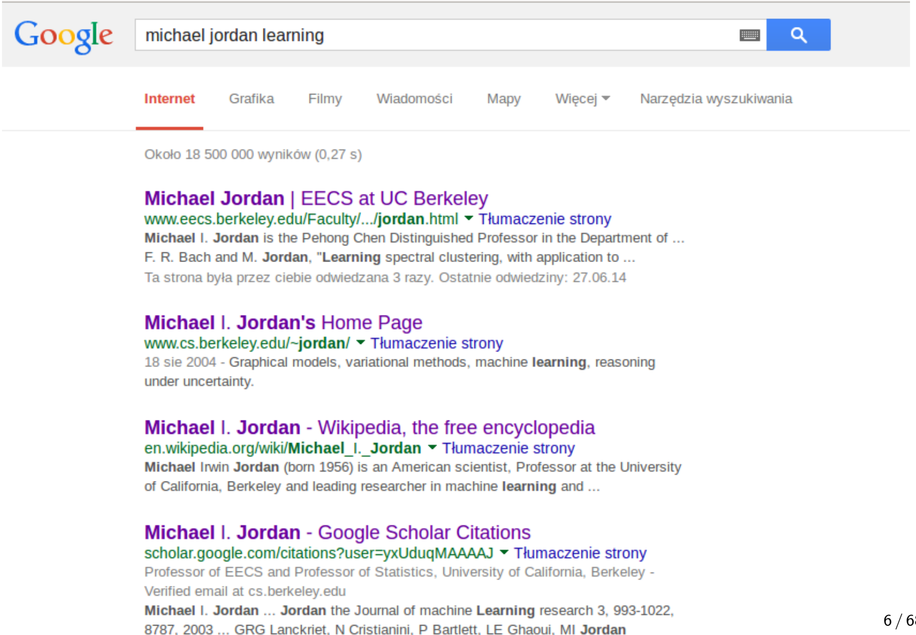
Outline

Ranking problem

Order a set of **objects** $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ according to the **preferences** of a **subject**.

Example – book recommendations

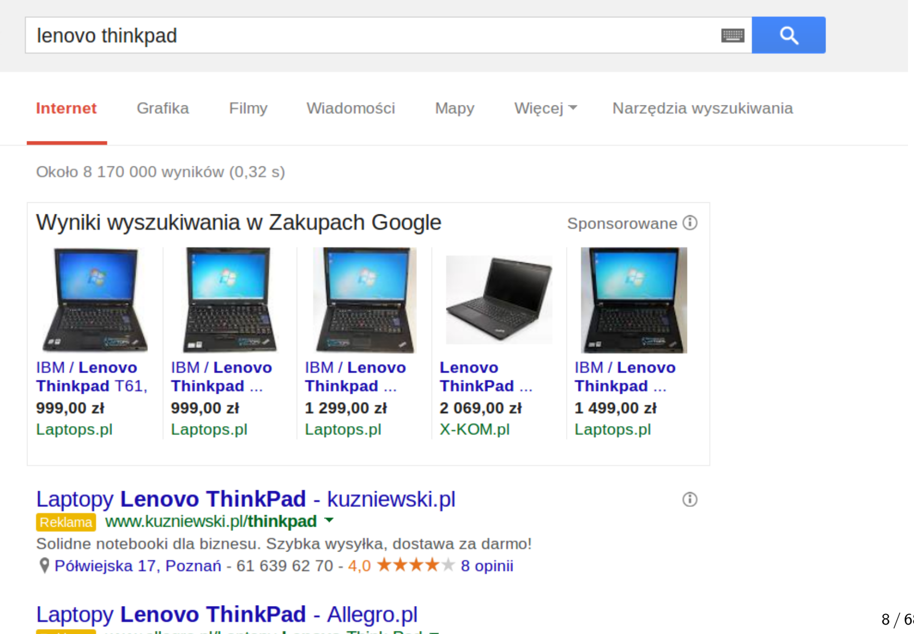
Example – information retrieval



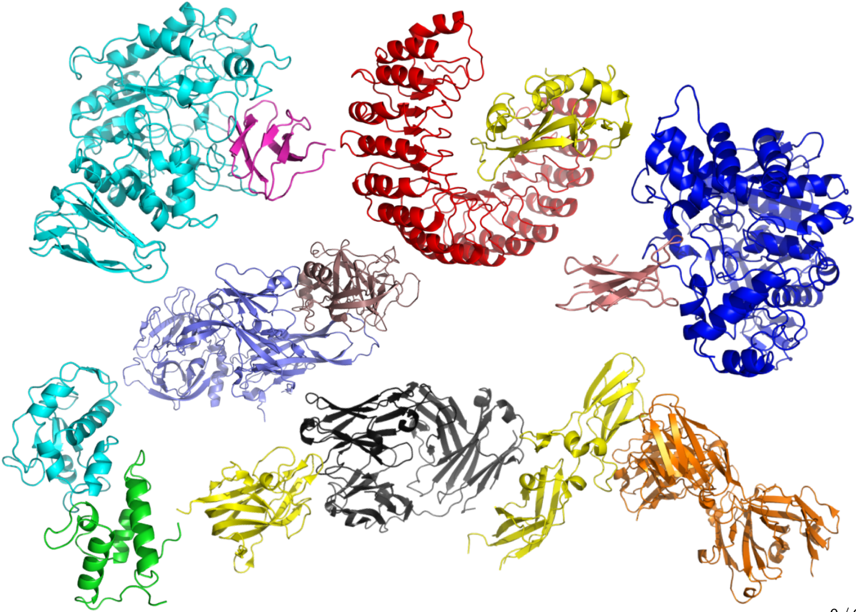
Example – rank aggregation problem

	Jack Corbett	Mark Ginnebaug	Geoff Hiten	Steve Jones	Allen Kinsel	Douglas McDowe	Andy Warren
Leadership - How much leadership experience has the candidate demonstrated?	2.00	3.57	3.00	2.33	3.14	3.86	3.71
Educational Experience - Do they have the requisite Education to be able to contribute on the Board?	2.40	3.57	2.83	3.00	2.86	4.00	3.29
Professional Background - Do they have the skills and experience (managerial, financial, and fiduciary) on offer to serve PASS?	2.00	3.71	2.67	3.00	2.86	3.86	3.71
Vision - Do they have a compelling vision for how they can contribute to the growth/expansion of PASS?	1.80	2.86	1.83	2.33	2.71	3.71	3.71
Volunteer Experience outside PASS - Do they have a compelling history of volunteerism?	2.00	3.14	2.17	2.17	2.86	3.43	2.86
Volunteer Contribution inside PASS – Do they show a history of dedication and involvement towards helping PASS achieve its mission and goals?	3.00	3.29	2.67	2.00	4.00	4.00	4.00
Reputation (inside PASS) - Do they have a good reputation for their contributions (volunteer or otherwise) to PASS in the community?	2.60	3.00	2.83	2.67	3.86	3.86	3.71
References (all) - Do they have strong references? Does the Board/PASS community support their bid for a Board seat?	3.00	3.29	3.00	2.50	3.43	3.57	3.57
Fit - How do their skills, experience, and strengths fit/complement the profile of the sitting Board?	1.80	3.29	2.33	2.17	3.00	4.00	3.86
Accountability - Do they do what they say they will?	2.60	3.57	3.33	3.00	3.71	3.86	3.71
Bias to action - Are they driven to deliver results?	2.60	3.86	3.33	2.83	3.57	3.86	3.86
Performance - Do they deliver on their commitments, and do they make a significant contribution?	2.60	3.57	3.33	2.67	3.71	3.86	3.71

Example – computational advertising



Example – protein structure prediction



Bipartite ranking

- Feedback information: **binary labels**.

x_1	-1
x_2	$+1$
x_3	$+1$
x_4	$+1$
x_5	-1

$$x_2 \succ x_1, x_3 \succ x_1,$$
$$x_4 \succ x_1, x_2 \succ x_5,$$
$$x_3 \succ x_5, x_4 \succ x_5.$$

Labels express preference, relevance, interest, etc.

Bipartite ranking

- Feedback information: **binary labels**.

x_1	-1
x_2	$+1$
x_3	$+1$
x_4	$+1$
x_5	-1

$$x_2 \succ x_1, x_3 \succ x_1,$$
$$x_4 \succ x_1, x_2 \succ x_5,$$
$$x_3 \succ x_5, x_4 \succ x_5.$$

Labels express preference, relevance, interest, etc.

Arguably the **simplest** problem of learning to rank.

- The feedback easy to acquire, sometimes implicitly.
- Good testbed for ranking algorithms and theoretical analysis.

Bipartite ranking

- Feedback information: **binary labels**.

x_1	-1
x_2	$+1$
x_3	$+1$
x_4	$+1$
x_5	-1

$$x_2 \succ x_1, x_3 \succ x_1,$$
$$x_4 \succ x_1, x_2 \succ x_5,$$
$$x_3 \succ x_5, x_4 \succ x_5.$$

Labels express preference, relevance, interest, etc.

Arguably the **simplest** problem of learning to rank.

- The feedback easy to acquire, sometimes implicitly.
- Good testbed for ranking algorithms and theoretical analysis.

Example

- Implicit feedback from search engine results.

Bipartite ranking

- Training data: $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\} \quad y_i \in \{-1, +1\}$.

	X_1	X_2	X_3	Y
$\boldsymbol{x}_1$	0.5	5	1	$+1$
$\boldsymbol{x}_2$	2.1	7	0	$+1$
$\boldsymbol{x}_3$	0.7	2	1	-1
$\boldsymbol{x}_4$	1.8	5	0	-1
$\boldsymbol{x}_5$	5.4	0	1	-1
...	...	...	...	...

Bipartite ranking

- Training data: $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\} \quad y_i \in \{-1, +1\}$.

	X_1	X_2	X_3	Y
$\boldsymbol{x}_1$	0.5	5	1	+1
$\boldsymbol{x}_2$	2.1	7	0	+1
$\boldsymbol{x}_3$	0.7	2	1	−1
$\boldsymbol{x}_4$	1.8	5	0	−1
$\boldsymbol{x}_5$	5.4	0	1	−1
...	...	...	...	...

- **Sort** objects, so that objects with $y_i = +1$ are ranked **higher** than objects with $y_i = -1$.

Evaluation metrics — pairwise disagreement

- Counts the number of **reversed preferences** over all pairs of objects.

object	rank	feedback
$\boldsymbol{x}_1$	1	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	3	$+1$
$\boldsymbol{x}_4$	4	$+1$
$\boldsymbol{x}_5$	5	-1
$\boldsymbol{x}_6$	6	$+1$
$\boldsymbol{x}_7$	7	-1
$\boldsymbol{x}_8$	8	-1

Evaluation metrics — pairwise disagreement

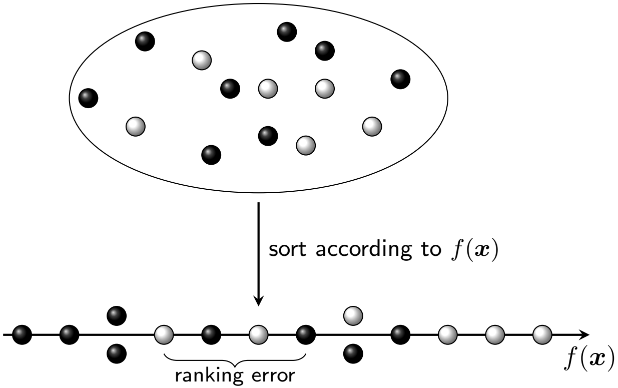
- Counts the number of **reversed preferences** over all pairs of objects.

object	rank	feedback
$\boldsymbol{x}_1$	1	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	3	$+1$
$\boldsymbol{x}_4$	4	$+1$
$\boldsymbol{x}_5$	5	-1
$\boldsymbol{x}_6$	6	$+1$
$\boldsymbol{x}_7$	7	-1
$\boldsymbol{x}_8$	8	-1

Number of reversed preferences: **4**.

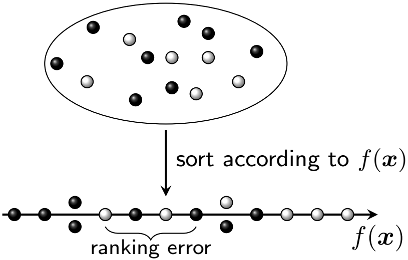
Ranking by scoring

- Learn a **scoring function** $f\colon \mathcal{X} \to \mathbb{R}$, which sorts objects according to the preferences.
- Error rate of $f \propto$ **number of reversed pairwise preferences**.



Ranking by scoring

- Learn a **scoring function** $f: X \to \mathbb{R}$, which sorts objects according to the preferences.
- Error rate of $f \propto$ **number of reversed pairwise preferences**.



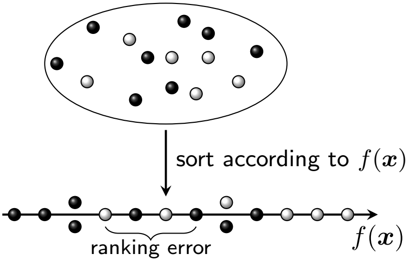
- **Empirical ranking risk**:

$$\widehat{L}_{\mathrm{rnk}}(f) = \frac{1}{n_+ n_-} \sum_{i:\, y_i=+1} \sum_{j:\, y_j=-1} \left([\![f(\boldsymbol{x}_i) < f(\boldsymbol{x}_j)]\!] + \frac{1}{2} [\![f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)]\!] \right),$$

where $n_+ = |\{i : y_i = +1\}|, n_- = |\{i : y_i = -1\}|$.

Ranking by scoring

- Learn a **scoring function** $f: X \to \mathbb{R}$, which sorts objects according to the preferences.
- Error rate of $f \propto$ **number of reversed pairwise preferences**.



- **Empirical ranking risk**:

$$\widehat{L}_{\mathrm{rnk}}(f) = \frac{1}{n_+ n_-} \sum_{i:\, y_i=+1} \sum_{j:\, y_j=-1} \left([\![f(\boldsymbol{x}_i) < f(\boldsymbol{x}_j)]\!] + \frac{1}{2} [\![f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)]\!] \right),$$

where $n_+ = |\{i : y_i = +1\}|, n_- = |\{i : y_i = -1\}|$.

- **(Empirical) Area under ROC Curve**: AUC$(f) = 1 - \widehat{L}_{\mathrm{rnk}}(f)$.

Area under ROC curve (AUC)

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

$$n_+ = 4, \qquad n_- = 4, \qquad \widehat{L}_{\text{rnk}}(f) = \frac{4}{4 \cdot 4} = 0.25 \qquad \text{AUC}(f) = 0.75$$

Area under ROC curve (AUC) in binary classification

- Real-valued scoring function $f\colon \mathcal{X} \to \mathbb{R}$.
- Objects with binary labels $y_i \in \{-1, +1\}$.

Area under ROC curve (AUC) in binary classification

- Real-valued scoring function $f\colon \mathcal{X} \to \mathbb{R}$.
- Objects with binary labels $y_i \in \{-1, +1\}$.
- Label prediction by **thresholding** f at some point θ:

$$\hat{y}(\boldsymbol{x}) = \begin{cases} +1 & \text{if } f(\boldsymbol{x}) > \theta, \\ -1 & \text{if } f(\boldsymbol{x}) \leq \theta. \end{cases}$$

Area under ROC curve (AUC) in binary classification

- Real-valued scoring function $f\colon \mathcal{X} \to \mathbb{R}$.
- Objects with binary labels $y_i \in \{-1, +1\}$.
- Label prediction by **thresholding** f at some point θ:

$$\hat{y}(\boldsymbol{x}) = \begin{cases} +1 & \text{if } f(\boldsymbol{x}) > \theta, \\ -1 & \text{if } f(\boldsymbol{x}) \leq \theta. \end{cases}$$

- **Vary** the threshold θ from $-\infty$ to ∞ and count the number of **true positives** and **false positives**:

$$\mathsf{TP} = \left|\{\boldsymbol{x}_i \colon \hat{y}(\boldsymbol{x}_i) = 1 \ \wedge \ y_i = 1\}\right|$$
$$\mathsf{FP} = \left|\{\boldsymbol{x}_i \colon \hat{y}(\boldsymbol{x}_i) = 1 \ \wedge \ y_i = -1\}\right|$$

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$		
$[2, 3.5)$		
$[1.2, 2.3)$		
$[0.6, 1.2)$		
$[0.1, 0.6)$		
$[-0.5, 0.1)$		
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$		
$[1.2, 2.3)$		
$[0.6, 1.2)$		
$[0.1, 0.6)$		
$[-0.5, 0.1)$		
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$		
$[0.6, 1.2)$		
$[0.1, 0.6)$		
$[-0.5, 0.1)$		
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$		
$[0.1, 0.6)$		
$[-0.5, 0.1)$		
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$	2	1
$[0.1, 0.6)$		
$[-0.5, 0.1)$		
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$	2	1
$[0.1, 0.6)$	3	1
$[-0.5, 0.1)$		
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$	2	1
$[0.1, 0.6)$	3	1
$[-0.5, 0.1)$	3	2
$[-1.2, -0.5)$		
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$	2	1
$[0.1, 0.6)$	3	1
$[-0.5, 0.1)$	3	2
$[-1.2, -0.5)$	4	2
$[-2.2, -1.2)$		
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

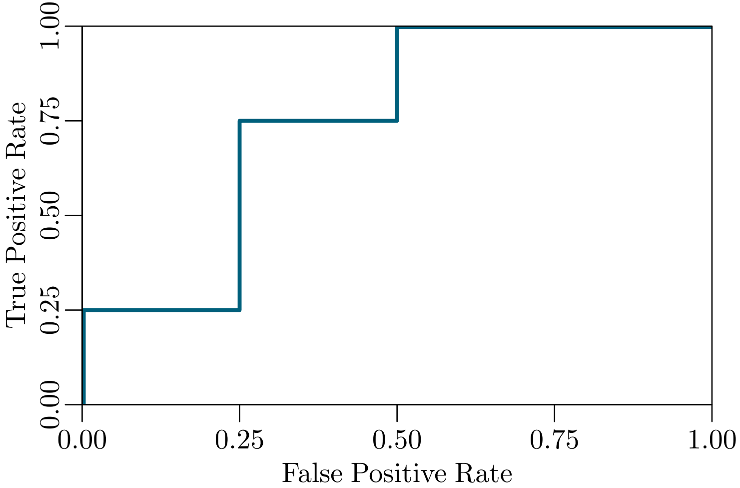
threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$	2	1
$[0.1, 0.6)$	3	1
$[-0.5, 0.1)$	3	2
$[-1.2, -0.5)$	4	2
$[-2.2, -1.2)$	4	3
$(-\infty, -2.2)$		

Area under ROC curve (AUC) in binary classification

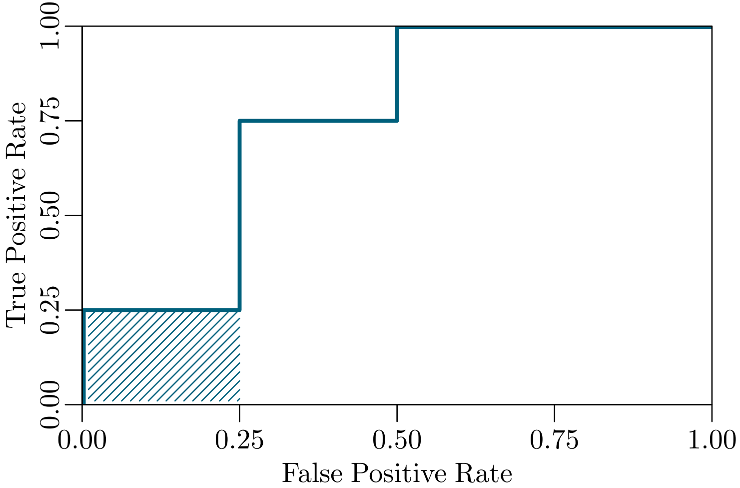
object	score $f(\boldsymbol{x})$	label
$\boldsymbol{x}_1$	3.5	$+1$
$\boldsymbol{x}_2$	2	-1
$\boldsymbol{x}_3$	1.2	$+1$
$\boldsymbol{x}_4$	0.6	$+1$
$\boldsymbol{x}_5$	0.1	-1
$\boldsymbol{x}_6$	-0.5	$+1$
$\boldsymbol{x}_7$	-1.2	-1
$\boldsymbol{x}_8$	-2.2	-1

threshold	TP	FP
$[3.5, \infty)$	0	0
$[2, 3.5)$	1	0
$[1.2, 2.3)$	1	1
$[0.6, 1.2)$	2	1
$[0.1, 0.6)$	3	1
$[-0.5, 0.1)$	3	2
$[-1.2, -0.5)$	4	2
$[-2.2, -1.2)$	4	3
$(-\infty, -2.2)$	4	4

Area under ROC curve (AUC) in binary classification

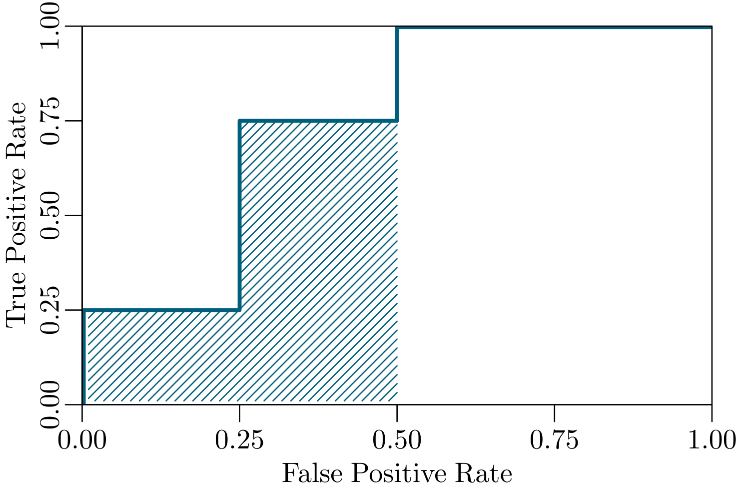


Area under ROC curve (AUC) in binary classification



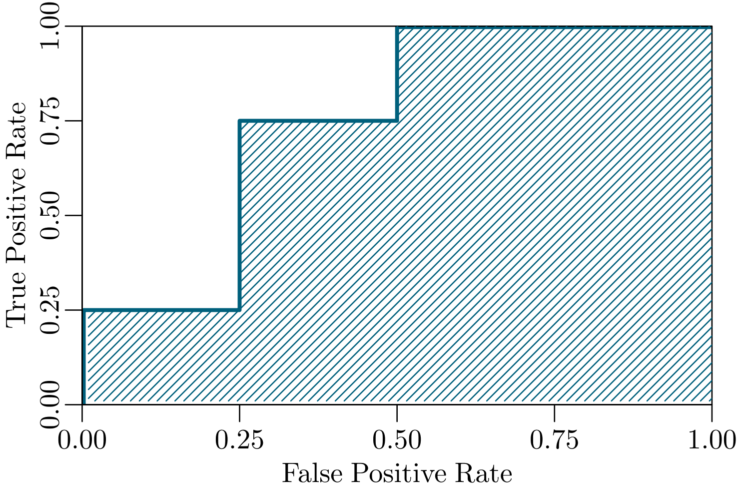
AUC $= 1/4 \cdot 1/4$

Area under ROC curve (AUC) in binary classification



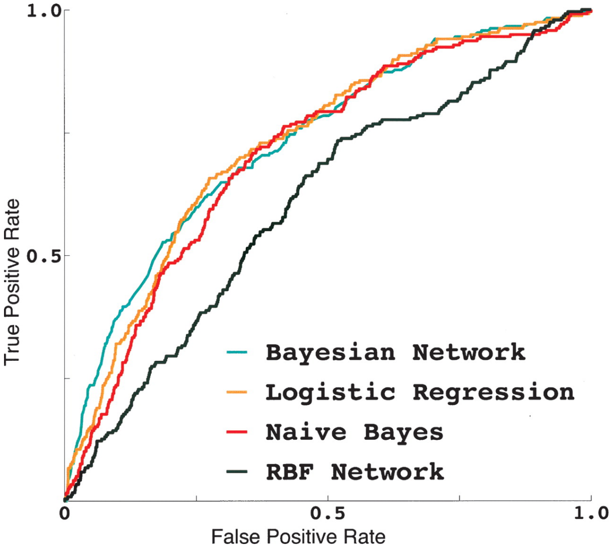
$\mathsf{AUC} = 1/4 \cdot 1/4 + 1/4 \cdot 3/4$

Area under ROC curve (AUC) in binary classification



$$\text{AUC} = 1/4 \cdot 1/4 + 1/4 \cdot 3/4 + 1/2 \cdot 1 = 0.75$$

Area under ROC curve (AUC) in binary classification



Area under ROC curve (AUC) in binary classification

- ROC curve measures the **performance** of binary classifier as threshold is **varied**.

Area under ROC curve (AUC) in binary classification

- ROC curve measures the **performance** of binary classifier as threshold is **varied**.
- ROC curve gives full characteristic of the classifier in terms of **sensitivity** (TP rate) vs. **specificity** ($1-$ FP rate).

Area under ROC curve (AUC) in binary classification

- ROC curve measures the **performance** of binary classifier as threshold is **varied**.
- ROC curve gives full characteristic of the classifier in terms of **sensitivity** (TP rate) vs. **specificity** ($1-$ FP rate).
- Allows to make optimal decision for any **misclassification costs**.

Area under ROC curve (AUC) in binary classification

- ROC curve measures the **performance** of binary classifier as threshold is **varied**.
- ROC curve gives full characteristic of the classifier in terms of **sensitivity** (TP rate) vs. **specificity** ($1-$ FP rate).
- Allows to make optimal decision for any **misclassification costs**.
- **Area under the ROC curve** will often be a **better** classifier's evaluation metric than **accuracy** (thresholding at 0), especially for:
  - **Imbalanced** data.
  - **Unknown** misclassification costs.

Area under ROC curve (AUC) in binary classification

- ROC curve measures the **performance** of binary classifier as threshold is **varied**.
- ROC curve gives full characteristic of the classifier in terms of **sensitivity** (TP rate) vs. **specificity** ($1-$ FP rate).
- Allows to make optimal decision for any **misclassification costs**.
- **Area under the ROC curve** will often be a **better** classifier's evaluation metric than **accuracy** (thresholding at 0), especially for:
  - **Imbalanced** data.
  - **Unknown** misclassification costs.
- Interest in **optimizing AUC** for binary classification **without reference to ranking**.

Outline

Standard approach to learning to rank

Reduction from bipartite ranking to pairwise binary classification:

Given:

- Data set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where each $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$.
- Class $\mathcal{F}$ of real-valued prediction functions $f \colon \mathcal{X} \to \mathbb{R}$,

Standard approach to learning to rank

Reduction from bipartite ranking to pairwise binary classification:

Given:

- Data set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where each $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$.
- Class $\mathcal{F}$ of real-valued prediction functions $f\colon \mathcal{X} \to \mathbb{R}$,

Define:

- A new dataset $\{\tilde{\boldsymbol{x}}_k, \tilde{y}_k\}_{k=1}^{K}$, $K = n_+ n_-$,
- A new class $\tilde{\mathcal{F}}$ of functions $f\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Standard approach to learning to rank

Data transformation:

- Take each pair $\{(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j)\}$ with $y_i = +1$ and $y_j = -1$, and make a **learning example** $(\tilde{\boldsymbol{x}}_k, \tilde{y}_k)$, such that:

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

Standard approach to learning to rank

Data transformation:

- Take each pair $\{(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j)\}$ with $y_i = +1$ and $y_j = -1$, and make a **learning example** $(\tilde{\boldsymbol{x}}_k, \tilde{y}_k)$, such that:

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

Function transformation:

- For any $f \in \mathcal{F}$, define $\tilde{f} \in \tilde{\mathcal{F}}$ by:

$$\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j), \qquad \text{for any } \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j).$$

Standard approach to learning to rank

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

$$\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j), \qquad \text{for any } \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j).$$

- Easy to see that for any f, the **empirical ranking risk of f** is equal to the **empirical $0/1$-risk of $\tilde{f}$**:

Standard approach to learning to rank

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

$$\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j), \qquad \text{for any } \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j).$$

- Easy to see that for any f, the **empirical ranking risk of f** is equal to the **empirical $0/1$-risk of $\tilde{f}$**:

$$\begin{aligned}\ell_{0/1}\left(\tilde{y}, \tilde{f}(\tilde{\boldsymbol{x}}_k)\right) &= [\![\tilde{y}\tilde{f}(\tilde{\boldsymbol{x}}_k) < 0]\!] + \frac{1}{2}[\![\tilde{y}\tilde{f}(\tilde{\boldsymbol{x}}_k) = 0]\!] \\ &= [\![f(\boldsymbol{x}_i) < f(\boldsymbol{x}_j)]\!] + \frac{1}{2}[\![f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)]\!].\end{aligned}$$

Standard approach to learning to rank

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

$$\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j), \qquad \text{for any } \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j).$$

- Easy to see that for any f, the **empirical ranking risk of f** is equal to the **empirical $0/1$-risk of $\tilde{f}$**:

$$\begin{aligned}\ell_{0/1}\left(\tilde{y}, \tilde{f}(\tilde{\boldsymbol{x}}_k)\right) &= [\![\tilde{y}\tilde{f}(\tilde{\boldsymbol{x}}_k) < 0]\!] + \frac{1}{2}[\![\tilde{y}\tilde{f}(\tilde{\boldsymbol{x}}_k) = 0]\!] \\ &= [\![f(\boldsymbol{x}_i) < f(\boldsymbol{x}_j)]\!] + \frac{1}{2}[\![f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)]\!].\end{aligned}$$

  Summing over pairs of positive and negative examples gives ranking risk.

Standard approach to learning to rank

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

$$\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j), \qquad \text{for any } \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j).$$

- Easy to see that for any f, the **empirical ranking risk of f** is equal to the **empirical $0/1$-risk of $\tilde{f}$**:

$$\begin{aligned}\ell_{0/1}\left(\tilde{y}, \tilde{f}(\tilde{\boldsymbol{x}}_k)\right) &= [\![\tilde{y}\tilde{f}(\tilde{\boldsymbol{x}}_k) < 0]\!] + \frac{1}{2}[\![\tilde{y}\tilde{f}(\tilde{\boldsymbol{x}}_k) = 0]\!] \\ &= [\![f(\boldsymbol{x}_i) < f(\boldsymbol{x}_j)]\!] + \frac{1}{2}[\![f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j)]\!].\end{aligned}$$

  Summing over pairs of positive and negative examples gives ranking risk.

- Take your favourite **surrogate loss** for binary classification $\ell(y, f(\boldsymbol{x}))$, and use it for $\tilde{y}$ and $\tilde{f}(\tilde{\boldsymbol{x}})$. **Problem solved**.

Standard approach to learning to rank

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = +1.$$

$$\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j), \qquad \text{for any } \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j).$$

Questions

- Why not include as well negative examples in the reduction:

$$\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \tilde{y}_k = \operatorname{sgn}(y_i - y_j)$$

- Does $\tilde{f}$ need to have a structure: $\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)$?

Standard approach to learning to rank

Examples:

- **SVM-OR**[1]: hinge loss.
- **RankBoost**[2]: exponential loss.
- A vast number of other pairwise approaches.

[1] R. Herbrich, T. Graepel, and K. Obermayer. Regression models for ordinal data: A machine learning approach. Technical report TR-99/03, Technical University of Berlin, 1999

[2] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003

Standard approach to learning to rank

Pros:

- Reduction to classification: we can **reuse** known concepts and methods.
- This reduction can solve much more **general** ranking problem, not necessarily bipartite.

Cons:

- Scales **quadratically** with sample size (tricks to reduce complexity on some special cases).
- **Cannot** reuse standard classification algorithms **without modification** due to structure on $\tilde{f}$, i.e. $\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)$.

Standard approach to learning to rank

Pros:

- Reduction to classification: we can **reuse** known concepts and methods.
- This reduction can solve much more **general** ranking problem, not necessarily bipartite.

Cons:

- Scales **quadratically** with sample size (tricks to reduce complexity on some special cases).
- **Cannot** reuse standard classification algorithms **without modification** due to structure on $\tilde{f}$, i.e. $\tilde{f}(\tilde{\boldsymbol{x}}_k) = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)$.

$O(n^2)$ is often **unacceptable**! How about **training a real-valued classifier** (works in $O(n)$) and **use it as a ranker**?

Outline

Good classifier can be a bad ranker[3]

- $0/1$ loss of a classifier $f \colon X \to \mathbb{R}$:

$$\ell_{0/1}(y, f(\boldsymbol{x})) = [\![f(\boldsymbol{x}) y \leq 0]\!], \qquad \widehat{L}_{0/1}(f) = \frac{1}{n} \sum_i \ell_{0/1}(y_i, f(x_i))$$

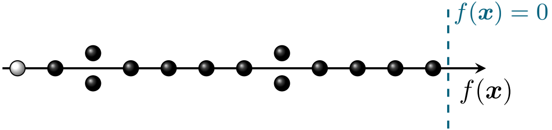
[3] W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Good classifier can be a bad ranker[3]

- $0/1$ loss of a classifier $f\colon X \to \mathbb{R}$:

$$\ell_{0/1}(y, f(\boldsymbol{x})) = [\![f(\boldsymbol{x})y \leq 0]\!], \qquad \widehat{L}_{0/1}(f) = \frac{1}{n}\sum_i \ell_{0/1}(y_i, f(x_i))$$

- Classifier with a fixed $0/1$-risk can have arbitrarily bad ranking risk



[3] W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

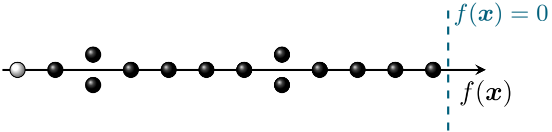
Good classifier can be a bad ranker[3]

- $0/1$ loss of a classifier $f\colon X \to \mathbb{R}$:

$$\ell_{0/1}(y, f(\boldsymbol{x})) = [\![f(\boldsymbol{x})y \leq 0]\!], \qquad \widehat{L}_{0/1}(f) = \frac{1}{n}\sum_i \ell_{0/1}(y_i, f(x_i))$$

- Classifier with a fixed $0/1$-risk can have arbitrarily bad ranking risk

$$\widehat{L}_{0/1}(f) = n_+/n,$$
$$\widehat{L}_{\mathrm{rnk}}(f) = 1.$$



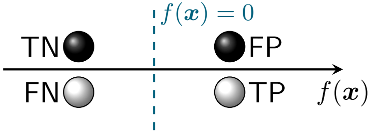
- This phenomenon is especially noticeable for unbalanced classes.

[3] W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Looking closer

- Assume for simplicity that $f(\boldsymbol{x}) \in \{-1, +1\}$.

		predicted $\hat{y} = f(\boldsymbol{x})$	
		-1	$+1$
true y	-1	TN	FP
	$+1$	FN	TP

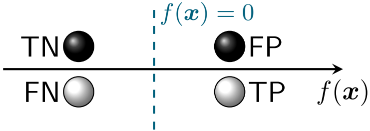


Looking closer

- Assume for simplicity that $f(\boldsymbol{x}) \in \{-1, +1\}$.

predicted $\hat{y} = f(\boldsymbol{x})$

		-1	$+1$
true y	-1	TN	FP
	$+1$	FN	TP

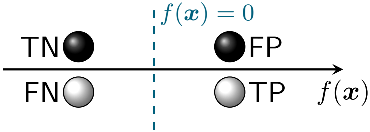


$$\widehat{L}_{\text{rnk}}(f) = \frac{FP \cdot FN + 0.5 \cdot TP \cdot FP + 0.5 \cdot FN \cdot TN}{n_+ n_-}$$

Looking closer

- Assume for simplicity that $f(\boldsymbol{x}) \in \{-1, +1\}$.

		predicted $\hat{y} = f(\boldsymbol{x})$	
		-1	$+1$
true y	-1	TN	FP
	$+1$	FN	TP



$$\widehat{L}_{\text{rnk}}(f) = \frac{FP \cdot FN + 0.5 \cdot TP \cdot FP + 0.5 \cdot FN \cdot TN}{n_+ n_-}$$

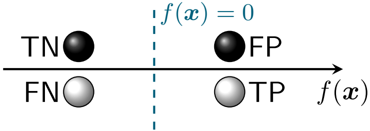
$$= \frac{FP(FN + TP) + FN(TN + FP)}{2 n_+ n_-}$$

Looking closer

- Assume for simplicity that $f(\boldsymbol{x}) \in \{-1, +1\}$.

predicted $\hat{y} = f(\boldsymbol{x})$

true y		-1	$+1$
	-1	TN	FP
	$+1$	FN	TP



$$\widehat{L}_{\mathrm{rnk}}(f) = \frac{FP \cdot FN + 0.5 \cdot TP \cdot FP + 0.5 \cdot FN \cdot TN}{n_+ n_-}$$

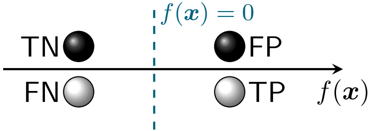
$$= \frac{FP(FN + TP) + FN(TN + FP)}{2n_+ n_-} = \frac{FP}{2n_-} + \frac{FN}{2n_+}$$

Looking closer

- Assume for simplicity that $f(\boldsymbol{x}) \in \{-1, +1\}$.

predicted $\hat{y} = f(\boldsymbol{x})$

		-1	$+1$
true y	-1	TN	FP
	$+1$	FN	TP



$$\widehat{L}_{\mathrm{rnk}}(f) = \frac{FP \cdot FN + 0.5 \cdot TP \cdot FP + 0.5 \cdot FN \cdot TN}{n_+ n_-}$$

$$= \frac{FP(FN + TP) + FN(TN + FP)}{2 n_+ n_-} = \frac{FP}{2n_-} + \frac{FN}{2n_+}$$

We can upperbound:

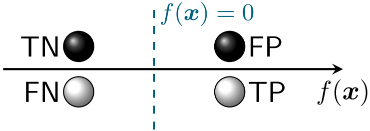
$$\widehat{L}_{\mathrm{rnk}}(f) \leq \frac{FP + FN}{2\min\{n_-, n_+\}} = \frac{n}{2\min\{n_-, n_+\}} \widehat{L}_{0/1}(f).$$

Looking closer

- Assume for simplicity that $f(\boldsymbol{x}) \in \{-1, +1\}$.

predicted $\hat{y} = f(\boldsymbol{x})$

		-1	$+1$
true y	-1	TN	FP
	$+1$	FN	TP



$$\widehat{L}_{\text{rnk}}(f) = \frac{FP \cdot FN + 0.5 \cdot TP \cdot FP + 0.5 \cdot FN \cdot TN}{n_+ n_-}$$

$$= \frac{FP(FN + TP) + FN(TN + FP)}{2n_+ n_-} = \frac{FP}{2n_-} + \frac{FN}{2n_+}$$

We can upperbound:

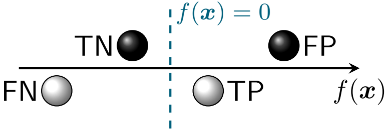
$$\widehat{L}_{\text{rnk}}(f) \leq \frac{FP + FN}{2\min\{n_-, n_+\}} = \frac{n}{2\min\{n_-, n_+\}} \widehat{L}_{0/1}(f).$$

Poor behavior of $0/1$ loss comes for **class imbalance**.

More general bound

- Assume now $f(\boldsymbol{x}) \in \mathbb{R}$.

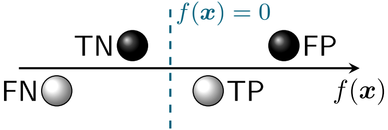
Given fixed TP, FN, FP, TP rate, what is the **worse-case** ranking risk?



More general bound

- Assume now $f(\boldsymbol{x}) \in \mathbb{R}$.

Given fixed TP, FN, FP, TP rate, what is the **worse-case** ranking risk?

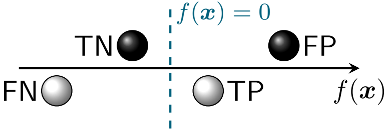


$$\widehat{L}_{\text{rnk}}(f) \quad = \quad \frac{FP \cdot FN + \cdot TP \cdot FP + \cdot FN \cdot TN}{n_+ n_-}$$

More general bound

- Assume now $f(\boldsymbol{x}) \in \mathbb{R}$.

Given fixed TP, FN, FP, TP rate, what is the **worse-case** ranking risk?

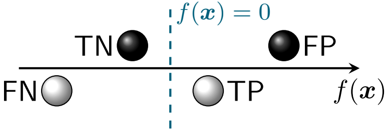


$$
\begin{aligned}
\widehat{L}_{\mathrm{rnk}}(f) &= \frac{FP \cdot FN + \cdot TP \cdot FP + \cdot FN \cdot TN}{n_+ n_-} \\
&= \frac{FP(FN + TP) + FN(TN + FP) - FNFP}{2n_+ n_-}
\end{aligned}
$$

More general bound

- Assume now $f(\boldsymbol{x}) \in \mathbb{R}$.

Given fixed TP, FN, FP, TP rate, what is the **worse-case** ranking risk?



$$
\begin{aligned}
\widehat{L}_{\mathrm{rnk}}(f) &= \frac{FP \cdot FN + \cdot TP \cdot FP + \cdot FN \cdot TN}{n_+ n_-} \\
&= \frac{FP(FN + TP) + FN(TN + FP) - FN FP}{2n_+ n_-} \\
&= \frac{FP}{n_-} + \frac{FN}{n_+} - \frac{FN}{n_-}\frac{FP}{n_+} \le \frac{FP}{n_-} + \frac{FN}{n_+}.
\end{aligned}
$$

Balanced $0/1$ Loss

$$\widehat{L}_{\text{rnk}}(f) \leq \frac{FP}{n_-} + \frac{FN}{n_+}$$

- $0/1$-risk $\widehat{L}_{0/1}(f) = \frac{FP+FN}{n}$ counts all mistakes with equal weights $\frac{1}{n}$.

Balanced $0/1$ **Loss**

$$\widehat{L}_{\mathrm{rnk}}(f) \leq \frac{FP}{n_-} + \frac{FN}{n_+}$$

- $0/1$-risk $\widehat{L}_{0/1}(f) = \frac{FP+FN}{n}$ counts all mistakes with equal weights $\frac{1}{n}$.
- **Balanced** $0/1$**-risk** $\widehat{L}_b(f) = \frac{FP}{2n_-} + \frac{FN}{2n_+}$ counts mistakes with weights proportional to the inverses of class cardinalities.

Balanced $0/1$ Loss

$$\widehat{L}_{\text{rnk}}(f) \leq \frac{FP}{n_-} + \frac{FN}{n_+}$$

- $0/1$-risk $\widehat{L}_{0/1}(f) = \frac{FP+FN}{n}$ counts all mistakes with equal weights $\frac{1}{n}$.
- **Balanced $0/1$-risk** $\widehat{L}_b(f) = \frac{FP}{2n_-} + \frac{FN}{2n_+}$ counts mistakes with weights proportional to the inverses of class cardinalities.
  - Proper normalization because: $\sum_{i:y_i=+1} \frac{1}{2n_+} + \sum_{i:y_i=-1} \frac{1}{2n_-} = \sum_i \frac{1}{n} = 1$.

Balanced $0/1$ Loss

$$\widehat{L}_{\text{rnk}}(f) \leq \frac{FP}{n_-} + \frac{FN}{n_+}$$

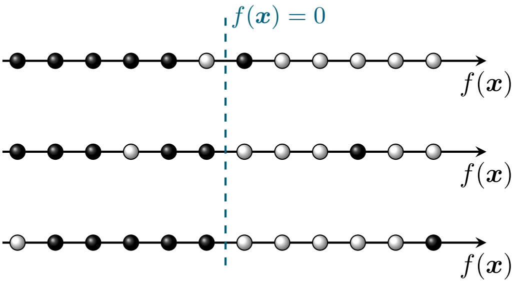
- $0/1$-risk $\widehat{L}_{0/1}(f) = \frac{FP+FN}{n}$ counts all mistakes with equal weights $\frac{1}{n}$.
- **Balanced $0/1$-risk** $\widehat{L}_b(f) = \frac{FP}{2n_-} + \frac{FN}{2n_+}$ counts mistakes with weights proportional to the inverses of class cardinalities.
  - Proper normalization because: $\sum_{i:y_i=+1} \frac{1}{2n_+} + \sum_{i:y_i=-1} \frac{1}{2n_-} = \sum_i \frac{1}{n} = 1$.
- Uneven misclassification costs **cancel out** class imbalance $\Longrightarrow$ balanced risk "sees" classes as being balanced.
- **Classifier which minimizes balanced risk also minimizes ranking risk!**

$$\widehat{L}_{\text{rnk}}(f) \leq 2\widehat{L}_b(f)$$

But...

- $0/1$ loss/risk (also balanced) is not sensitive to order.

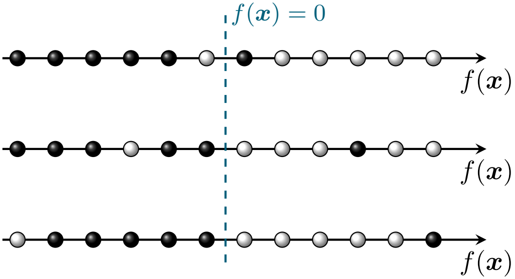
$\widehat{L}_{0/1}$	$\widehat{L}_{\mathrm{rnk}}$
1/6	1/36
1/6	6/36
1/6	11/36



But...

- $0/1$ loss/risk (also balanced) is not sensitive to order.

$\widehat{L}_{0/1}$	$\widehat{L}_{\text{rnk}}$
1/6	1/36
1/6	6/36
1/6	11/36



- Need to consider losses which penalize not only for classification mistake but also for the distance to 0.
  $\Longrightarrow$ **Margin-based losses.**

Outline

Setting

- Moving the theory from **empirical** level to the **population** level
  - **counting** $\rightarrow$ **distribution**.
- Accuracy measures used so far become expectations.
- Better measure of performance: **regret**.

Setting

- **Examples** $(\boldsymbol{x}, y)$ generated by a **distribution** $P(\boldsymbol{x}, y)$.
- A (real-valued) classifier $f\colon X \to \mathbb{R}$, with accuracy measured by the **risk**:

$$L_\ell(f) := \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\ell(y, f(\boldsymbol{x}))\right],$$

  where ℓ is a **pointwise** loss.
- The **regret** of a classifier f:

$$\mathrm{Reg}_\ell(f) = L_\ell(f) - L_\ell(f_\ell^*),$$

  where f_ℓ^* is the **Bayes classifier**, $f_\ell^* = \arg\min_f L_\ell(f)$.
- Regret measures how much worse we perform than the optimal classifier.

Setting

- A **ranker** $f\colon X \to \mathbb{R}$, with accuracy measured by **ranking risk**:

$$L_{\text{rnk}}(f) := P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|y > y') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|y > y'),$$

where $(\boldsymbol{x}, y)$, $(\boldsymbol{x}', y')$ are two **independent** random examples.

Setting

- A **ranker** $f\colon X \to \mathbb{R}$, with accuracy measured by **ranking risk**:

$$L_{\mathrm{rnk}}(f) := P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|y > y') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|y > y'),$$

  where $(\boldsymbol{x}, y)$, $(\boldsymbol{x}', y')$ are two **independent** random examples.

- Ranking risk is a probability that random **positive** example is ranked **lower** than random **negative** example.

Setting

- A **ranker** $f\colon X \to \mathbb{R}$, with accuracy measured by **ranking risk**:

$$L_{\mathrm{rnk}}(f) := P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|y > y') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|y > y'),$$

  where $(\boldsymbol{x}, y)$, $(\boldsymbol{x}', y')$ are two **independent** random examples.
- Ranking risk is a probability that random **positive** example is ranked **lower** than random **negative** example.
- The **ranking regret** is defined as:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) = L_{\mathrm{rnk}}(f) - L_{\mathrm{rnk}}(f_r^*),$$

  where $f_r^* = \arg\min_f L_{\mathrm{rnk}}(f)$ is the **Bayes ranker**.

Problem statement

- Each classifier f can be used as a ranker.

Problem statement

- Each classifier f can be used as a ranker.
- Given a classifier f with classification regret $\mathrm{Reg}_\ell(f)$ for some loss function ℓ, what is the maximum ranking regret of f, $\mathrm{Reg}_{\mathrm{rnk}}(f)$? **(regret bounds)**

Problem statement

- Each classifier f can be used as a ranker.
- Given a classifier f with classification regret $\mathrm{Reg}_\ell(f)$ for some loss function ℓ, what is the maximum ranking regret of f, $\mathrm{Reg}_{\mathrm{rnk}}(f)$? **(regret bounds)**
- In particular: if a classifier f is close to the optimal classifier f_ℓ^*, is its ranking risk close to to the ranking risk of the optimal ranker f_r^*? $\Longrightarrow$ **ranking calibration**.

The optimal ranker

$$L_{\mathrm{rnk}}(f) = P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|y > y') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|y > y')$$

The optimal ranker

$$L_{\mathrm{rnk}}(f) = P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|y > y') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|y > y')$$

Question

Define:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \eta(\boldsymbol{x})(1 - \eta(\boldsymbol{x}'))\Big([\![f(\boldsymbol{x}) < f(\boldsymbol{x}')]\!] + \frac{1}{2}[\![f(\boldsymbol{x}) = f(\boldsymbol{x}')]\!]\Big),$$

where $\eta(\boldsymbol{x}) = P(y = 1|\boldsymbol{x})$. Show that the ranking risk can be rewritten as:

$$\begin{aligned} L_{\mathrm{rnk}}(f) &= \frac{1}{p(1-p)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\left[K(\boldsymbol{x}, \boldsymbol{x}')\right] \\ &= \frac{1}{2p(1-p)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\left[K(\boldsymbol{x}, \boldsymbol{x}') + K(\boldsymbol{x}', \boldsymbol{x})\right]. \end{aligned}$$

where $p = P(y = 1)$ is the prior probability of positive class

The optimal ranker

$$L_{\mathrm{rnk}}(f) = P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|y > y') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|y > y')$$

Question

Based on the result of the previous question, argue that the Bayes ranker $f^*(\boldsymbol{x})$ minimizes $K(\boldsymbol{x}, \boldsymbol{x}') + K(\boldsymbol{x}', \boldsymbol{x})$ for every $(\boldsymbol{x}, \boldsymbol{x}')$. Show that this implies:

$$f^*(\boldsymbol{x}) > f^*(\boldsymbol{x}') \qquad \text{if and only if} \qquad \eta(\boldsymbol{x}) > \eta(\boldsymbol{x}'),$$

i.e., the Bayes ranker $f^*(\boldsymbol{x})$ is **any strictly monotone transformation** of $\eta(\boldsymbol{x})$. (examples should be **ordered** according to $\eta(\boldsymbol{x})$)

Surrogate losses and calibration

- Let $\ell(y, \boldsymbol{x})$ be a **pointwise surrogate** loss for ranking.

Surrogate losses and calibration

- Let $\ell(y, \boldsymbol{x})$ be a **pointwise surrogate** loss for ranking.
- We want to be **ranking calibrated**:

$$\mathrm{Reg}_{\ell}(f_n) \to 0 \Longrightarrow \mathrm{Reg}_{\mathrm{rnk}}(f_n) \to 0.$$

Surrogate losses and calibration

- Let $\ell(y, \boldsymbol{x})$ be a **pointwise surrogate** loss for ranking.
- We want to be **ranking calibrated**:

$$\mathrm{Reg}_\ell(f_n) \to 0 \Longrightarrow \mathrm{Reg}_{\mathrm{rnk}}(f_n) \to 0.$$

- This implies that the **Bayes classifier** f_ℓ^* must also be the **Bayes ranker**.

Surrogate losses and calibration

- Let $\ell(y, \boldsymbol{x})$ be a **pointwise surrogate** loss for ranking.
- We want to be **ranking calibrated**:

$$\mathrm{Reg}_{\ell}(f_n) \to 0 \Longrightarrow \mathrm{Reg}_{\mathrm{rnk}}(f_n) \to 0.$$

- This implies that the **Bayes classifier** f_ℓ^* must also be the **Bayes ranker**.
- Since the Bayes ranker is a strictly monotone transform of $\eta(\boldsymbol{x})$, **so must be** f_ℓ^*.

Surrogate losses and calibration

- Let $\ell(y, \boldsymbol{x})$ be a **pointwise surrogate** loss for ranking.
- We want to be **ranking calibrated**:

$$\mathrm{Reg}_\ell(f_n) \to 0 \Longrightarrow \mathrm{Reg}_{\mathrm{rnk}}(f_n) \to 0.$$

- This implies that the **Bayes classifier** f^*_ℓ must also be the **Bayes ranker**.
- Since the Bayes ranker is a strictly monotone transform of $\eta(\boldsymbol{x})$, **so must be** f^*_ℓ.
- The loss ℓ must **"estimate"** conditional probability function $\eta(\boldsymbol{x})$ or its strictly increasing transform!

Surrogate losses and calibration

- Let $\ell(y, \boldsymbol{x})$ be a **pointwise surrogate** loss for ranking.
- We want to be **ranking calibrated**:

$$\mathrm{Reg}_\ell(f_n) \to 0 \Longrightarrow \mathrm{Reg}_{\mathrm{rnk}}(f_n) \to 0.$$

- This implies that the **Bayes classifier** f_ℓ^* must also be the **Bayes ranker**.
- Since the Bayes ranker is a strictly monotone transform of $\eta(\boldsymbol{x})$, **so must be** f_ℓ^*.
- The loss ℓ must **"estimate"** conditional probability function $\eta(\boldsymbol{x})$ or its strictly increasing transform!
- $0/1$ loss **ruled out**: the Bayes classifier $f_{0/1}^*(\boldsymbol{x}) = \mathrm{sign}(\eta(\boldsymbol{x}) - 1/2)$ is **not** a strictly monotone transform of $\eta(\boldsymbol{x})$.

Outline

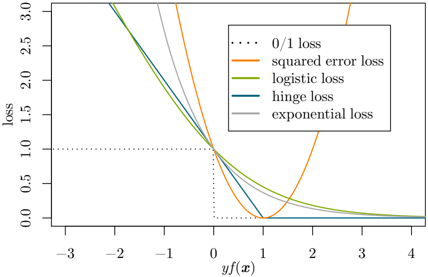
Margin-based losses

Motivation:

- **Empirical evidence** (from published papers, methods used in industry) suggests that simple scoring classifiers, notably those minimizing **margin-based loss functions**, perform quite **strongly** in terms of ranking loss (AUC).
- Can we **explain** this phenomenon on the **theoretical grounds**?

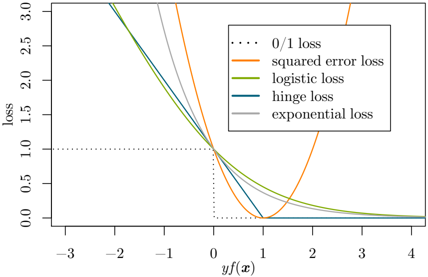
Margin-based losses

- Loss functions of the form $\ell(y, f(\boldsymbol{x})) = \ell(yf(\boldsymbol{x}))$.



Margin-based losses

- Loss functions of the form $\ell(y, f(\boldsymbol{x})) = \ell(yf(\boldsymbol{x}))$.

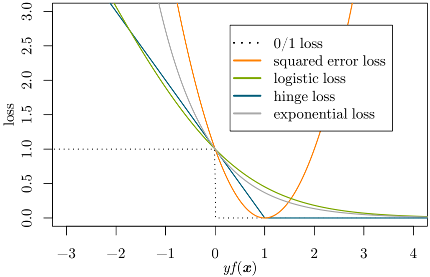


- Bayes classifiers:

loss	$f^*(\eta)$	$\frac{\mathrm{d}f^*(\eta)}{\mathrm{d}\eta}$
squared error	$2\eta - 1$	$2 > 0$
logistic	$\log \frac{\eta}{1-\eta}$	$\frac{1}{\eta(1-\eta)} > 0$
exponential	$\frac{1}{2} \log \frac{\eta}{1-\eta}$	$\frac{1}{2\eta(1-\eta)} > 0$
hinge	$\operatorname{sgn}(\eta - 1/2)$	0

Margin-based losses

- Loss functions of the form $\ell(y, f(\boldsymbol{x})) = \ell(yf(\boldsymbol{x}))$.



- Bayes classifiers:

loss	$f^*(\eta)$	$\frac{\mathrm{d}f^*(\eta)}{\mathrm{d}\eta}$
squared error	$2\eta - 1$	$2 > 0$
logistic	$\log \frac{\eta}{1-\eta}$	$\frac{1}{\eta(1-\eta)} > 0$
exponential	$\frac{1}{2} \log \frac{\eta}{1-\eta}$	$\frac{1}{2\eta(1-\eta)} > 0$
hinge	$\mathrm{sgn}\,(\eta - 1/2)$	0

- **Hinge loss ruled out!**

Regret bounds for exponential and logistic surrogate losses

Theorem[4]:
The following regret bounds hold for the exponential loss and the logistic loss, respectively:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{2p(1-p)}\sqrt{\frac{3}{2}}\sqrt{\mathrm{Reg}_{\exp}(f)},$$
$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{2p(1-p)}\sqrt{2}\sqrt{\mathrm{Reg}_{\log}(f)},$$

where $\mathrm{Reg}_{\exp}$ and $\mathrm{Reg}_{\log}$ are the regrets for exponential and logistic loss, respectively, and $p = P(y = 1)$.

[4] K. Dembczyński, W. Kotłowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *International Conference on Machine Learning*, 2012
W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Regret bounds for exponential and logistic surrogate losses

Theorem[4]:
The following regret bounds hold for the exponential loss and the logistic loss, respectively:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \le \frac{1}{2p(1-p)}\sqrt{\frac{3}{2}}\sqrt{\mathrm{Reg}_{\exp}(f)},$$
$$\mathrm{Reg}_{\mathrm{rnk}}(f) \le \frac{1}{2p(1-p)}\sqrt{2}\sqrt{\mathrm{Reg}_{\log}(f)},$$

where $\mathrm{Reg}_{\exp}$ and $\mathrm{Reg}_{\log}$ are the regrets for exponential and logistic loss, respectively, and $p = P(y = 1)$.

Can we **get rid of** the ugly constant $1/(2p(1-p))$?

[4] K. Dembczyński, W. Kotłowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *International Conference on Machine Learning*, 2012
W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Regret bounds for exponential and logistic surrogate losses

Theorem[4]:
The following regret bounds hold for the exponential loss and the logistic loss, respectively:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{2p(1-p)}\sqrt{\frac{3}{2}}\sqrt{\mathrm{Reg}_{\exp}(f)},$$
$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{2p(1-p)}\sqrt{2}\sqrt{\mathrm{Reg}_{\log}(f)},$$

where $\mathrm{Reg}_{\exp}$ and $\mathrm{Reg}_{\log}$ are the regrets for exponential and logistic loss, respectively, and $p = P(y=1)$.

Can we **get rid of** the ugly constant $1/(2p(1-p))$? **Not** with the current loss functions!

[4] K. Dembczyński, W. Kotłowski, and E. Hüllermeier. Consistent multilabel ranking through univariate losses. In *International Conference on Machine Learning*, 2012
W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Sensitivity to class priors

- Ranking risk is **insensitive** to any change of the class prior $P(y)$.
  - Changing $P(y)$ while keeping $P(\boldsymbol{x}|y)$ fixed does not change the ranking risk.

$$L_{\mathrm{rnk}}(f) := P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|\boldsymbol{y} > \boldsymbol{y}') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|\boldsymbol{y} > \boldsymbol{y}')$$

(depends **only** on $P(\boldsymbol{x}|y)$, not on $p = P(y = 1)$)

Sensitivity to class priors

- Ranking risk is **insensitive** to any change of the class prior $P(y)$.
  - Changing $P(y)$ while keeping $P(\boldsymbol{x}|y)$ fixed does not change the ranking risk.

$$L_{\mathrm{rnk}}(f) := P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|\boldsymbol{y} > \boldsymbol{y}') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|\boldsymbol{y} > \boldsymbol{y}')$$

(depends **only** on $P(\boldsymbol{x}|y)$, not on $p = P(y = 1)$)

- Surrogate losses are **sensitive** to class priors.
  - This is the origin of the term $1/(2p(1-p))$.

Sensitivity to class priors

- Ranking risk is **insensitive** to any change of the class prior $P(y)$.
  - Changing $P(y)$ while keeping $P(\boldsymbol{x}|y)$ fixed does not change the ranking risk.

$$L_{\mathrm{rnk}}(f) := P(f(\boldsymbol{x}) < f(\boldsymbol{x}')|\boldsymbol{y} > \boldsymbol{y}') + \frac{1}{2}P(f(\boldsymbol{x}) = f(\boldsymbol{x}')|\boldsymbol{y} > \boldsymbol{y}')$$

  (depends **only** on $P(\boldsymbol{x}|y)$, not on $p = P(y = 1)$)

- Surrogate losses are **sensitive** to class priors.
  - This is the origin of the term $1/(2p(1-p))$.
- Can we make the surrogate loss **insensitive to the priors**?

Balancing

- Given a loss function $\ell(y, \hat{y})$, define its **weighted** version as:

$$\ell_{\mathrm{w}}(y, \hat{y}) = w(y)\ell(y, \hat{y}).$$

Balancing

- Given a loss function $\ell(y, \hat{y})$, define its **weighted** version as:

$$\ell_{\mathrm{w}}(y, \hat{y}) = w(y)\ell(y, \hat{y}).$$

- Require weights to satisfy the **normalization** constraint:

$$\mathbb{E}_y[w(y)] = 1,$$

  i.e., weighting only **redistributes** the loss without changing its scale.

Balancing

- Given a loss function $\ell(y, \hat{y})$, define its **weighted** version as:

$$\ell_{\mathrm{w}}(y, \hat{y}) = w(y)\ell(y, \hat{y}).$$

- Require weights to satisfy the **normalization** constraint:

$$\mathbb{E}_y[w(y)] = 1,$$

  i.e., weighting only **redistributes** the loss without changing its scale.
- Given a loss function $\ell(y, \hat{y})$, define its **balanced** version as:

$$\ell_{\mathrm{b}}(y, \hat{y}) = \frac{1}{2P(y)}\ell(y, \hat{y}), \qquad \text{i.e., } w(y) = \frac{1}{2P(y)}.$$

Balancing

- Given a loss function $\ell(y, \hat{y})$, define its **weighted** version as:

$$\ell_{\mathrm{w}}(y, \hat{y}) = w(y)\ell(y, \hat{y}).$$

- Require weights to satisfy the **normalization** constraint:

$$\mathbb{E}_y[w(y)] = 1,$$

i.e., weighting only **redistributes** the loss without changing its scale.

- Given a loss function $\ell(y, \hat{y})$, define its **balanced** version as:

$$\ell_{\mathrm{b}}(y, \hat{y}) = \frac{1}{2P(y)}\ell(y, \hat{y}), \qquad \text{i.e., } w(y) = \frac{1}{2P(y)}.$$

- Properly normalized:

$$\mathbb{E}_y\left[\frac{1}{2P(y)}\right] = \frac{P(y=1)}{2P(y=1)} + \frac{P(y=-1)}{2P(y=-1)} = 1.$$

Balancing

- Given a loss function $\ell(y, \hat{y})$, define its **weighted** version as:

$$\ell_{\mathrm{w}}(y, \hat{y}) = w(y)\ell(y, \hat{y}).$$

- Require weights to satisfy the **normalization** constraint:

$$\mathbb{E}_y[w(y)] = 1,$$

  i.e., weighting only **redistributes** the loss without changing its scale.
- Given a loss function $\ell(y, \hat{y})$, define its **balanced** version as:

$$\ell_{\mathrm{b}}(y, \hat{y}) = \frac{1}{2P(y)}\ell(y, \hat{y}), \qquad \text{i.e., } w(y) = \frac{1}{2P(y)}.$$

- Properly normalized:

$$\mathbb{E}_y\left[\frac{1}{2P(y)}\right] = \frac{P(y=1)}{2P(y=1)} + \frac{P(y=-1)}{2P(y=-1)} = 1.$$

- Requires knowing the class priors $P(y)$, but these can be easily **estimated** from the training data.

Balancing counteracts the uneven priors.

The expected balanced loss $\ell_{\mathrm{b}}(y, f(\boldsymbol{x}))$ with respect to a distribution $P(\boldsymbol{x}, y)$ with class prior p, is the same as the expected original loss $\ell(y, f(\boldsymbol{x}))$ with respect to a distribution $\tilde{P}(\boldsymbol{x}, y)$, such that:

$$\tilde{P}(\boldsymbol{x}|y) = P(\boldsymbol{x}|y), y \in \{-1, 1\}, \qquad \tilde{P}(y = 1) = \tilde{P}(y = -1) = 1/2.$$

Balancing counteracts the uneven priors.

The expected balanced loss $\ell_{\mathrm{b}}(y, f(\boldsymbol{x}))$ with respect to a distribution $P(\boldsymbol{x}, y)$ with class prior p, is the same as the expected original loss $\ell(y, f(\boldsymbol{x}))$ with respect to a distribution $\tilde{P}(\boldsymbol{x}, y)$, such that:

$$\tilde{P}(\boldsymbol{x}|y) = P(\boldsymbol{x}|y), y \in \{-1, 1\}, \qquad \tilde{P}(y=1) = \tilde{P}(y=-1) = 1/2.$$

Proof:

$$\begin{aligned} L_{\ell_{\mathrm{b}}}(f) &= \int \ell_{\mathrm{b}}(y, f(\boldsymbol{x})) P(\boldsymbol{x}, y) \mathrm{d}\boldsymbol{x} \mathrm{d}y = \int \frac{1}{2P(y)} \ell(y, f(\boldsymbol{x})) P(\boldsymbol{x}|y) P(y) \mathrm{d}\boldsymbol{x} \mathrm{d}y \\ &= \int \ell(y, f(\boldsymbol{x})) P(\boldsymbol{x}|y) \frac{1}{2} \mathrm{d}\boldsymbol{x} \mathrm{d}y = \int \ell(y, f(\boldsymbol{x})) \tilde{P}(\boldsymbol{x}, y) \mathrm{d}\boldsymbol{x} \mathrm{d}y = \tilde{L}_{\ell}(f). \end{aligned}$$

Regret bounds for balanced exponential and logistic surrogate losses

Theorem[5]:

The following regret bounds hold for the **balanced exponential** loss and **balanced logistic** loss, respectively:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq 2\sqrt{\frac{3}{2}}\sqrt{\mathrm{Reg}_{\mathrm{b.exp}}(f)},$$

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq 2\sqrt{2}\sqrt{\mathrm{Reg}_{\mathrm{b.log}}(f)},$$

where $\mathrm{Reg}_{\mathrm{b.exp}}$ and $\mathrm{Reg}_{\mathrm{b.log}}$ are the regrets for balanced exponential and balanced logistic losses, respectively.

the term $1/(2p(1-p))$ has been replaced by 2.

[5] W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Regret bounds for balanced exponential and logistic surrogate losses

Theorem[5]:
The following regret bounds hold for the **balanced exponential** loss and **balanced logistic** loss, respectively:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \le 2\sqrt{\frac{3}{2}}\sqrt{\mathrm{Reg}_{\mathrm{b.exp}}(f)},$$

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \le 2\sqrt{2}\sqrt{\mathrm{Reg}_{\mathrm{b.log}}(f)},$$

where $\mathrm{Reg}_{\mathrm{b.exp}}$ and $\mathrm{Reg}_{\mathrm{b.log}}$ are the regrets for balanced exponential and balanced logistic losses, respectively.

the term $1/(2p(1-p))$ has been replaced by 2.

Proof: The **expected balanced loss** is **equal** to the **expected original loss** w.r.t $\tilde{P}(\boldsymbol{x}, y)$ with priors equal to $1/2$. Apply **previous theorem** for $\tilde{P}(\boldsymbol{x}, y)$ and note that ranking regret is **invariant** to changing the priors.

[5] W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning*, pages 1113–1120, 2011

Does balancing matters?

- The Bayes classifiers for balanced losses

$$f^*_{\mathrm{b.\,exp}}(\boldsymbol{x}) = \frac{1}{2}\log\frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})} - \frac{1}{2}\log\frac{p}{1-p} = f^*_{\exp}(\boldsymbol{x}) + f_0,$$

$$f^*_{\mathrm{b.\,log}}(\boldsymbol{x}) = \log\frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})} - \log\frac{p}{1-p} = f^*_{\log}(\boldsymbol{x}) + f_1,$$

are **shifted** versions of the unbalanced counterparts.
$\Rightarrow$ **constant shift does not influence ranking!**

Does balancing matters?

- The Bayes classifiers for balanced losses

$$f^*_{\text{b. exp}}(\boldsymbol{x}) = \frac{1}{2}\log\frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})} - \frac{1}{2}\log\frac{p}{1-p} = f^*_{\exp}(\boldsymbol{x}) + f_0,$$

$$f^*_{\text{b. log}}(\boldsymbol{x}) = \log\frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})} - \log\frac{p}{1-p} = f^*_{\log}(\boldsymbol{x}) + f_1,$$

  are **shifted** versions of the unbalanced counterparts.
  $\Rightarrow$ **constant shift does not influence ranking!**
- For exponential loss, the above can be shown not only for Bayes classifier, but also for classifiers trained by minimizing the empirical risk.

Outline

Overview

- Artificial and real data.
- We train standard linear classifiers based on:
  - logistic loss (**logistic regression**),
  - exponential loss (**AdaBoost**).
- We check how they perform compared to a specialized "state-of-the-art" linear algorithm for bipartite ranking (SVM-OR).

Overview

- Artificial and real data.
- We train standard linear classifiers based on:
  - logistic loss (**logistic regression**),
  - exponential loss (**AdaBoost**).
- We check how they perform compared to a specialized "state-of-the-art" linear algorithm for bipartite ranking (SVM-OR).
- No significant difference in ranking accuracy...

Overview

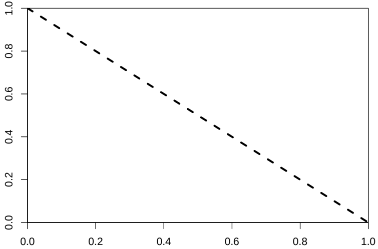
- Artificial and real data.
- We train standard linear classifiers based on:
  - logistic loss (**logistic regression**),
  - exponential loss (**AdaBoost**).
- We check how they perform compared to a specialized "state-of-the-art" linear algorithm for bipartite ranking (SVM-OR).
- No significant difference in ranking accuracy...
- ...but that's what we want, as our algorithms are **simple, fast and widely accessible** in software packages.

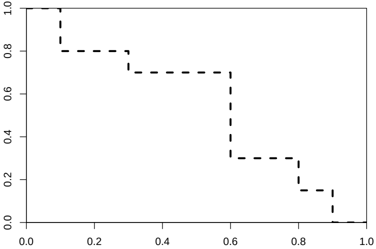
Experiment – Artificial Data

- Input $\boldsymbol{x} = (x_1, \ldots, x_{50}) \in [0,1]^{50}$ drawn uniformly.
- Output y is generated by thresholding a function $f(\boldsymbol{x})$, i.e., $y = \operatorname{sgn}(f(\boldsymbol{x})) +$ random noise (Bayes rank risk 0.1).

Experiment – Artificial Data

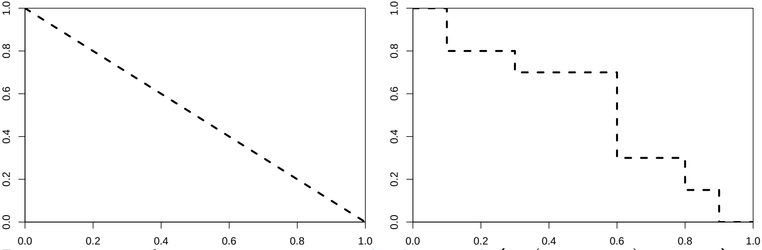
- Input $\boldsymbol{x} = (x_1, \ldots, x_{50}) \in [0,1]^{50}$ drawn uniformly.
- Output y is generated by thresholding a function $f(\boldsymbol{x})$, i.e., $y = \operatorname{sgn}(f(\boldsymbol{x})) +$ random noise (Bayes rank risk 0.1).
- Two models for $f(\boldsymbol{x})$: **linear** and **nonlinear**.





Experiment – Artificial Data

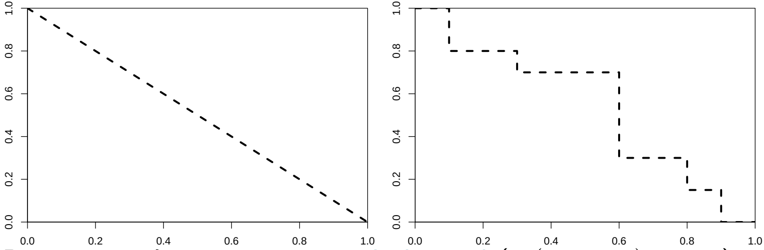
- Input $\boldsymbol{x} = (x_1, \ldots, x_{50}) \in [0,1]^{50}$ drawn uniformly.
- Output y is generated by thresholding a function $f(\boldsymbol{x})$, i.e., $y = \operatorname{sgn}(f(\boldsymbol{x})) +$ random noise (Bayes rank risk 0.1).
- Two models for $f(\boldsymbol{x})$: **linear** and **nonlinear**.



- By varying class priors we get **balanced** ($P(y = +1) = 0.5$) and **imbalanced** ($P(y = +1) = 0.9$) models.

Experiment – Artificial Data

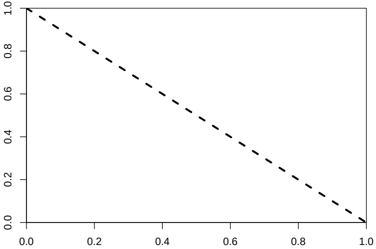
- Input $\boldsymbol{x} = (x_1, \ldots, x_{50}) \in [0,1]^{50}$ drawn uniformly.
- Output y is generated by thresholding a function $f(\boldsymbol{x})$, i.e., $y = \operatorname{sgn}(f(\boldsymbol{x})) +$ random noise (Bayes rank risk 0.1).
- Two models for $f(\boldsymbol{x})$: **linear** and **nonlinear**.

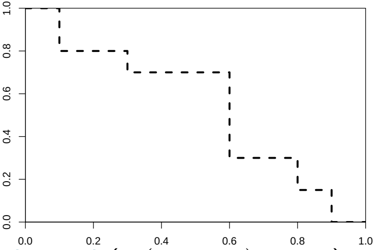


- By varying class priors we get **balanced** ($P(y=+1) = 0.5$) and **imbalanced** ($P(y=+1) = 0.9$) models.
- 30 random models, 30 training sets (of size 1000) per model, test set of size 10000.

Experiment – Artificial Data

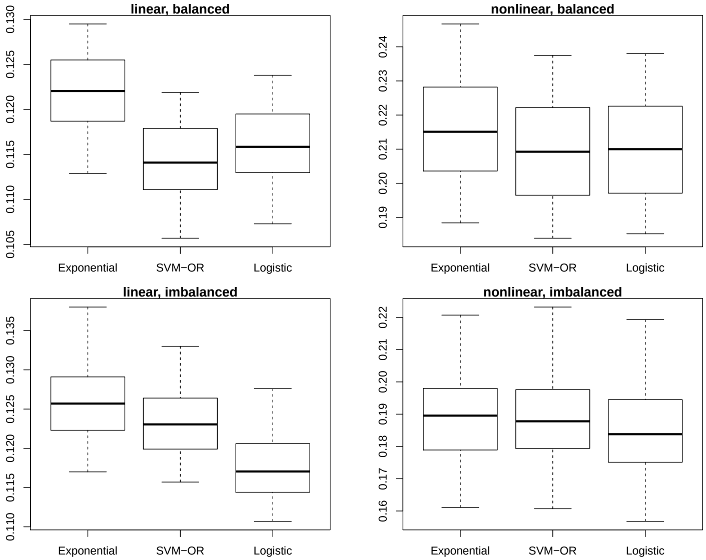
- Input $\boldsymbol{x} = (x_1, \ldots, x_{50}) \in [0,1]^{50}$ drawn uniformly.
- Output y is generated by thresholding a function $f(\boldsymbol{x})$, i.e., $y = \operatorname{sgn}(f(\boldsymbol{x})) +$ random noise (Bayes rank risk 0.1).
- Two models for $f(\boldsymbol{x})$: **linear** and **nonlinear**.





- By varying class priors we get **balanced** ($P(y = +1) = 0.5$) and **imbalanced** ($P(y = +1) = 0.9$) models.
- 30 random models, 30 training sets (of size 1000) per model, test set of size 10000.
- Linear classifier trained by minimizing (1) **exponential**, (2) **logistic**, and (3) **pairwise hinge loss** (SVM-OR)

Artificial Data – Results



Real Data – Results

Dataset	Exponential	SVM-OR	Logistic
Breast-w	0.0051	0.0049	0.0054
Breast-c	0.3077	0.2955	0.3005
Colic	0.1251	0.1352	0.1179
Diabetes	0.1724	0.1702	0.1804
Haberman	0.3684	0.3153	0.3820
Heart-h	0.0887	0.1005	0.0929
Hepatitis	0.1289	0.1321	0.1230
Ionosphere	0.0811	0.0773	0.0884
Vote	0.0098	0.0103	0.0096
Covtype	0.1635	0.1604	0.1623
KDD04	0.2114	0.2083	0.2143

Outline

Proper composite loss[6]

- Given a pointwise margin loss $\ell(f)$, define its **conditional risk**:

$$C_\eta(f) = \eta\ell(f) + (1-\eta)\ell(-f).$$

[6] Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014

Proper composite loss[6]

- Given a pointwise margin loss $\ell(f)$, define its **conditional risk**:

$$C_\eta(f) = \eta\ell(f) + (1-\eta)\ell(-f).$$

- We call $\ell(f)$ **proper composite** if there exists a **strictly increasing** (and therefore invertible) **link function** $\psi\colon [0,1] \to \mathbb{R}$, such that:

$$\psi(\eta) \in \arg\min_f C_\eta(f) \qquad \text{for any } \eta.$$

[6] Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014

Proper composite loss[6]

- Given a pointwise margin loss $\ell(f)$, define its **conditional risk**:

$$C_\eta(f) = \eta\ell(f) + (1-\eta)\ell(-f).$$

- We call $\ell(f)$ **proper composite** if there exists a **strictly increasing** (and therefore invertible) **link function** $\psi\colon [0,1] \to \mathbb{R}$, such that:

$$\psi(\eta) \in \arg\min_f C_\eta(f) \qquad \text{for any } \eta.$$

- Bayes classsifier is an **invertible** function of conditional probability η. (inverting the relation we get probability estimate from f)

[6] Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014

Proper composite loss[6]

- Given a pointwise margin loss $\ell(f)$, define its **conditional risk**:

$$C_\eta(f) = \eta\ell(f) + (1-\eta)\ell(-f).$$

- We call $\ell(f)$ **proper composite** if there exists a **strictly increasing** (and therefore invertible) **link function** $\psi\colon [0,1] \to \mathbb{R}$, such that:

$$\psi(\eta) \in \arg\min_f C_\eta(f) \qquad \text{for any } \eta.$$

- Bayes classsifier is an **invertible** function of conditional probability η. (inverting the relation we get probability estimate from f)
- Holds for most of considered margin-based losses:

loss	$f^*(\eta) = \psi(\eta)$	$\eta(f^*) = \psi^{-1}(f^*)$
squared error	$2\eta - 1$	$\frac{1+f^*}{2}$
logistic	$\log\frac{\eta}{1-\eta}$	$\frac{1}{1+e^{-f^*}}$
exponential	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$\frac{1}{1+e^{-2f^*}}$

[6] Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014

Strongly proper composite loss[7]

- We call $\ell(f)$ **λ-strongly proper composite** if

$$C_\eta(f) - H(\eta) \geq \frac{\lambda}{2}\left(\eta - \psi^{-1}(f)\right)^2, \qquad H(\eta) = \min_f C_\eta(f),$$

i.e. conditional regret is **lowerbounded** by squared difference between the true conditional probability η and estimated conditional probability $\psi^{-1}(f)$.

[7] Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014

Main result[8]

Theorem: Let $\ell(y, f(\boldsymbol{x}))$ be λ-strongly proper composite margin loss. Then:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \sqrt{\frac{2}{\lambda}} \sqrt{\mathrm{Reg}_{\ell}(f)}.$$

[8] Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014

Proof

Proof

Use strong properness:

$$C_\eta(f(\boldsymbol{x})) - H(\eta(\boldsymbol{x})) \geq \frac{\lambda}{2}\left(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\right)^2$$

to bound:

$$\left(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\right)^2 \leq \frac{2}{\lambda}\left(C_\eta(f(\boldsymbol{x})) - H(\eta(\boldsymbol{x}))\right)$$

Proof

Use strong properness:

$$C_\eta(f(\boldsymbol{x})) - H(\eta(\boldsymbol{x})) \geq \frac{\lambda}{2}\left(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\right)^2$$

to bound:

$$\left(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\right)^2 \leq \frac{2}{\lambda}\left(C_\eta(f(\boldsymbol{x})) - H(\eta(\boldsymbol{x}))\right)$$

Take expectation on both sides:

$$\mathbb{E}_{\boldsymbol{x}}\left[\left(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\right)^2\right] \leq \frac{2}{\lambda}\left(L_\ell(f) - L_\ell^*\right) = \frac{2}{\lambda}\mathrm{Reg}_\ell(f).$$

Proof — cont.

We now need a lemma, which will not be proved here:

Lemma: For any f', such that $f'(\boldsymbol{x}) \in [0, 1]$ for all $\boldsymbol{x}$:

$$\mathrm{Reg}_{\mathrm{rnk}}(f') \leq \frac{1}{p(1-p)} \mathbb{E}_{\boldsymbol{x}} \left[|\eta(\boldsymbol{x}) - f'(\boldsymbol{x})| \right]$$

Proof — cont.

We now need a lemma, which will not be proved here:

Lemma: For any f', such that $f'(\boldsymbol{x}) \in [0,1]$ for all $\boldsymbol{x}$:

$$\mathrm{Reg}_{\mathrm{rnk}}(f') \leq \frac{1}{p(1-p)} \mathbb{E}_{\boldsymbol{x}} \left[|\eta(\boldsymbol{x}) - f'(\boldsymbol{x})| \right]$$

We take $f'(\boldsymbol{x}) := \psi^{-1}(f(\boldsymbol{x}))$ and get from the lemma and Jensen's inequality:

$$\begin{aligned}\mathrm{Reg}_{\mathrm{rnk}}(f') &\leq \frac{1}{p(1-p)} \mathbb{E}_{\boldsymbol{x}} \left[|\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))| \right] \\ &\leq \frac{1}{p(1-p)} \sqrt{\mathbb{E}_{\boldsymbol{x}} \left[(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x})))^2 \right]}\end{aligned}$$

Jensen: if f convex, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

Proof — cont.

We now need a lemma, which will not be proved here:

Lemma: For any f', such that $f'(\boldsymbol{x}) \in [0,1]$ for all $\boldsymbol{x}$:

$$\mathrm{Reg}_{\mathrm{rnk}}(f') \leq \frac{1}{p(1-p)} \mathbb{E}_{\boldsymbol{x}} \left[|\eta(\boldsymbol{x}) - f'(\boldsymbol{x})| \right]$$

We take $f'(\boldsymbol{x}) := \psi^{-1}(f(\boldsymbol{x}))$ and get from the lemma and Jensen's inequality:

$$\begin{aligned} \mathrm{Reg}_{\mathrm{rnk}}(f') &\leq \frac{1}{p(1-p)} \mathbb{E}_{\boldsymbol{x}} \left[|\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))| \right] \\ &\leq \frac{1}{p(1-p)} \sqrt{\mathbb{E}_{\boldsymbol{x}} \left[(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x})))^2 \right]} \end{aligned}$$

Jensen: if f convex, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

Finally, since f' and f are strictly monotonically related,

$$\mathrm{Reg}_{\mathrm{rnk}}(f') = \mathrm{Reg}_{\mathrm{rnk}}(f).$$

Proof – cont.

Taking it all together:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \sqrt{\mathbb{E}_{\boldsymbol{x}} \left[(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x})))^2 \right]},$$

$$\mathbb{E}_{\boldsymbol{x}} \left[\big(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\big)^2 \right] \leq \frac{2}{\lambda} \mathrm{Reg}_{\ell}(f),$$

Proof – cont.

Taking it all together:

$$\begin{aligned}\mathrm{Reg}_{\mathrm{rnk}}(f) &\leq \frac{1}{p(1-p)}\sqrt{\mathbb{E}_{\boldsymbol{x}}\left[(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x})))^2\right]}, \\ \mathbb{E}_{\boldsymbol{x}}\left[\left(\eta(\boldsymbol{x}) - \psi^{-1}(f(\boldsymbol{x}))\right)^2\right] &\leq \frac{2}{\lambda}\mathrm{Reg}_{\ell}(f),\end{aligned}$$

we get the desired bound:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)}\sqrt{\frac{2}{\lambda}}\sqrt{\mathrm{Reg}_{\ell}(f)}.$$

How to calculate λ?

Fact: if $H(\eta)$ is twice differentiable, and $-\frac{\mathrm{d}H^2(\eta)}{\mathrm{d}\eta^2} > \lambda$ for any η, then ℓ is λ-strongly proper.

loss	$H(\eta)$	$-\frac{\mathrm{d}H^2(\eta)}{\mathrm{d}\eta^2}$	λ
squared error	$4\eta(1-\eta)$	8	8
logistic	$-\eta\log\eta - (1-\eta)\log(1-\eta)$	$\frac{1}{\eta(1-\eta)}$	4
exponential	$2\sqrt{\eta(1-\eta)}$	$\frac{1}{2(\eta(1-\eta))^{3/2}}$	4

Regret bounds

Corrolary:

- For **squared error loss**:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \frac{1}{2} \sqrt{\mathrm{Reg}_{\mathrm{sq}}(f)}.$$

- For **logistic loss**:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \frac{1}{\sqrt{2}} \sqrt{\mathrm{Reg}_{\log}(f)}.$$

- For **exponential loss**:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \frac{1}{\sqrt{2}} \sqrt{\mathrm{Reg}_{\exp}(f)}.$$

Regret bounds

Corrolary:

- For **squared error loss**:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \frac{1}{2} \sqrt{\mathrm{Reg}_{\mathrm{sq}}(f)}.$$

- For **logistic loss**:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \frac{1}{\sqrt{2}} \sqrt{\mathrm{Reg}_{\mathrm{log}}(f)}.$$

- For **exponential loss**:

$$\mathrm{Reg}_{\mathrm{rnk}}(f) \leq \frac{1}{p(1-p)} \frac{1}{\sqrt{2}} \sqrt{\mathrm{Reg}_{\mathrm{exp}}(f)}.$$

The term $\frac{1}{p(1-p)}$ can be removed by **balancing** the loss, as before.

Conclusions

- Theoretical results suggesting that minimizing margin-based pointwise loss functions is **sufficient** to achieve low rank regret.
- Also confirmed by experimental results, both for synthetic and benchmark data.
- The results are intuitively plausible (and hence not very surprising), yet they provide a **sound theoretical explanation** of previous observations and give some new insights.

Thank you for your attention!