Efficient AUC Optimization for Classification

Toon Calders[1] Szymon Jaroszewicz[2]

[1]Eindhoven University of Technology
Eindhoven, the Netherlands

[2]National Institute of Telecommunications
Warsaw, Poland

PKDD 2007

Overview

Overview

Assessing Classifier Quality

- Usually measured using accuracy
  - Most algorithms are designed to optimize accuracy

Assessing Classifier Quality

- Usually measured using accuracy
  - Most algorithms are designed to optimize accuracy
- Accuracy is less suitable to assess classification models when
  - Classes are skewed
  - Classifier is a function f giving a score
    - Need to fix a threshold (how?)
    - Accuracy only looks at a single threshold

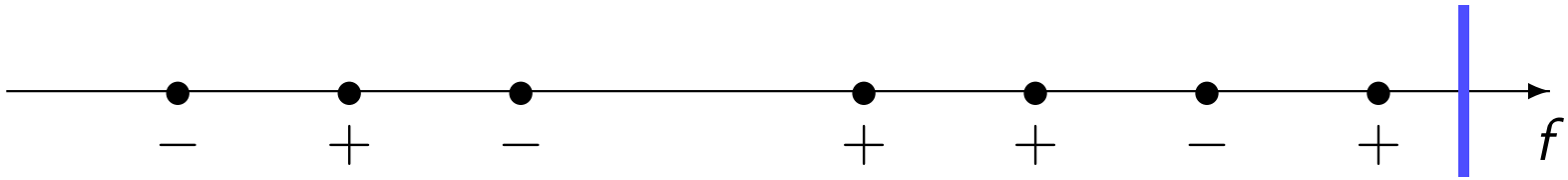
Assessing Classifier Quality

- Usually measured using accuracy
  - Most algorithms are designed to optimize accuracy
- Accuracy is less suitable to assess classification models when
  - Classes are skewed
  - Classifier is a function f giving a score
    - Need to fix a threshold (how?)
    - Accuracy only looks at a single threshold
- ROC-curves are better
  - Characterize precision-recall trade-off
  - Able to deal with skewed class distribution

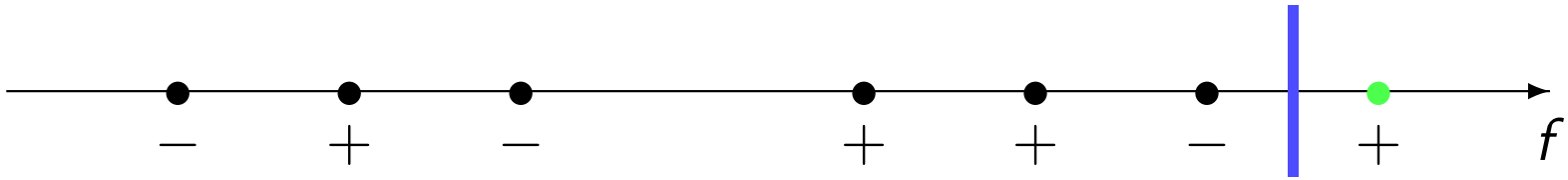
Assessing Classifier Quality

- Usually measured using accuracy
  - Most algorithms are designed to optimize accuracy
- Accuracy is less suitable to assess classification models when
  - Classes are skewed
  - Classifier is a function f giving a score
    - Need to fix a threshold (how?)
    - Accuracy only looks at a single threshold
- ROC-curves are better
  - Characterize precision-recall trade-off
  - Able to deal with skewed class distribution
- How to compare two ROC curves?
  - Compute Area under the ROC curve (AUC)

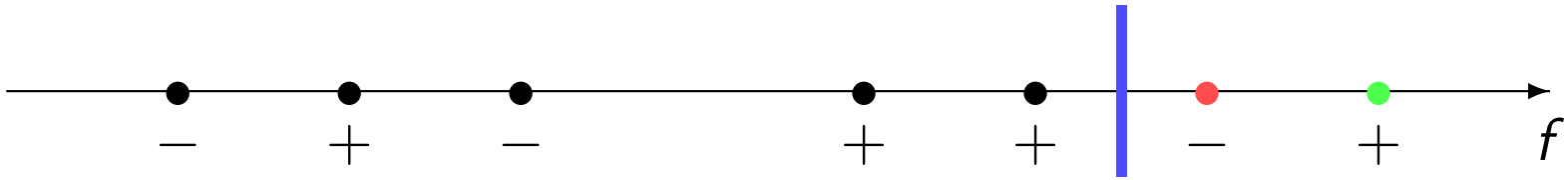
ROC Curve

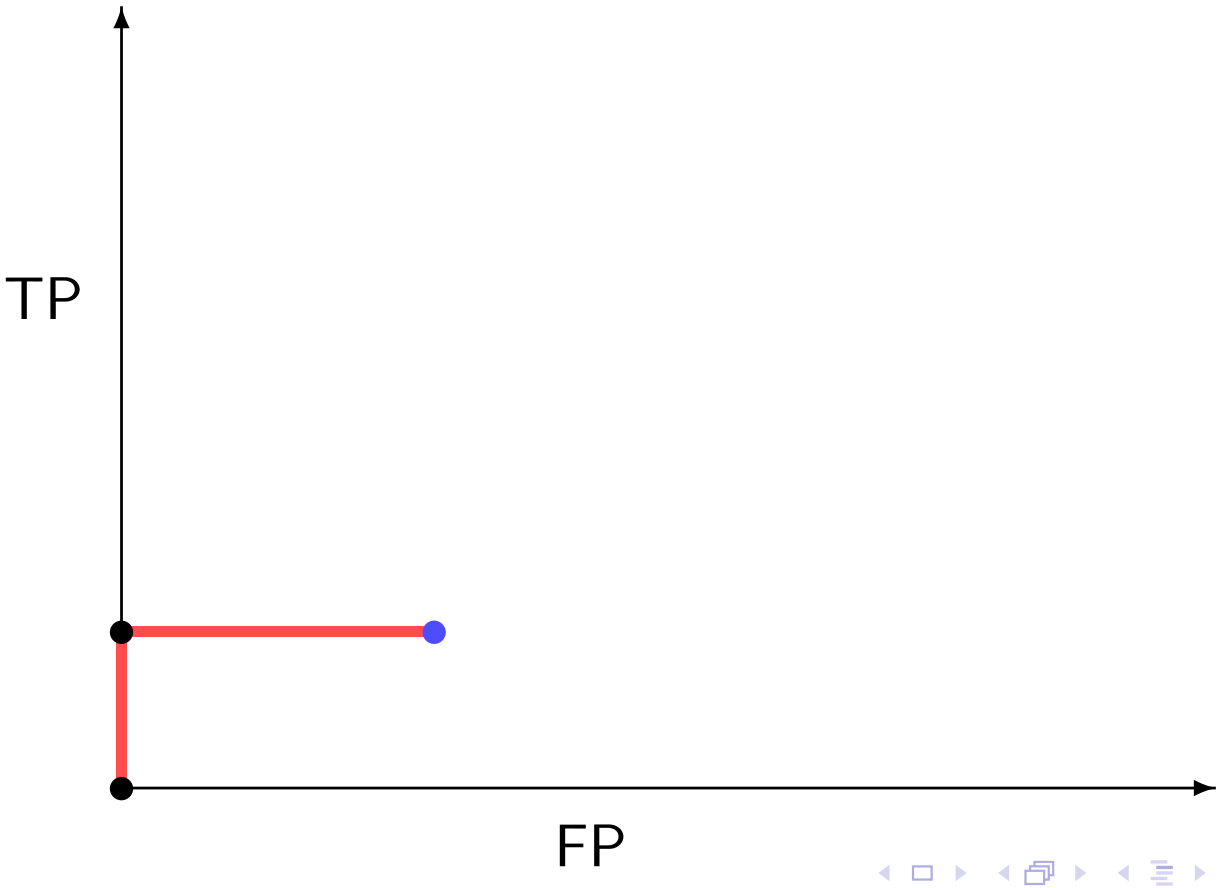


ROC Curve

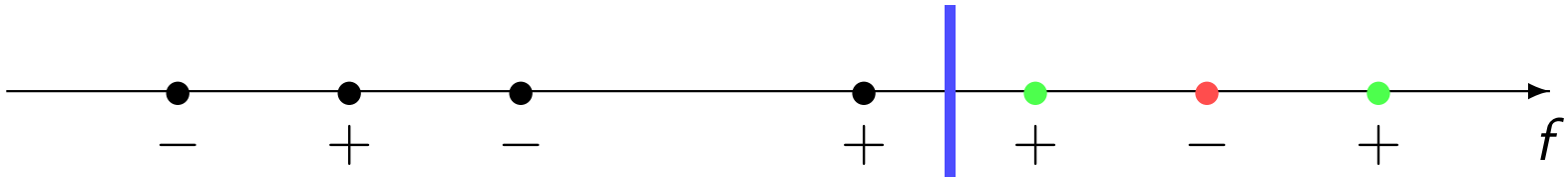


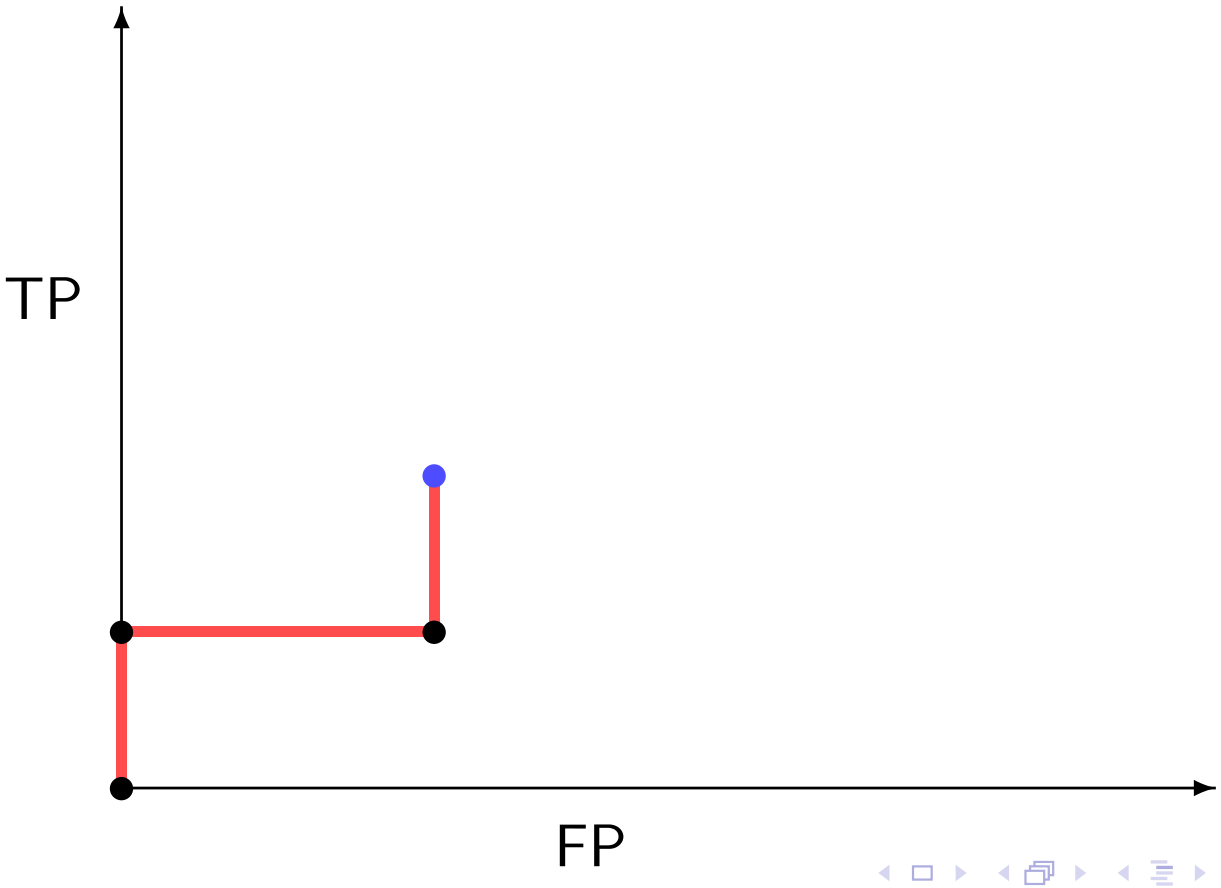
ROC Curve



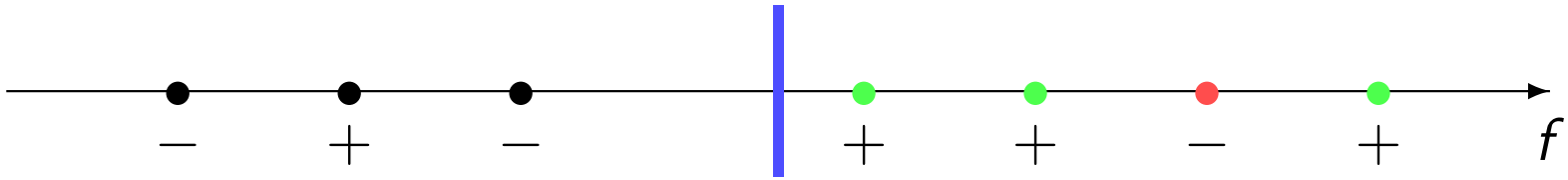


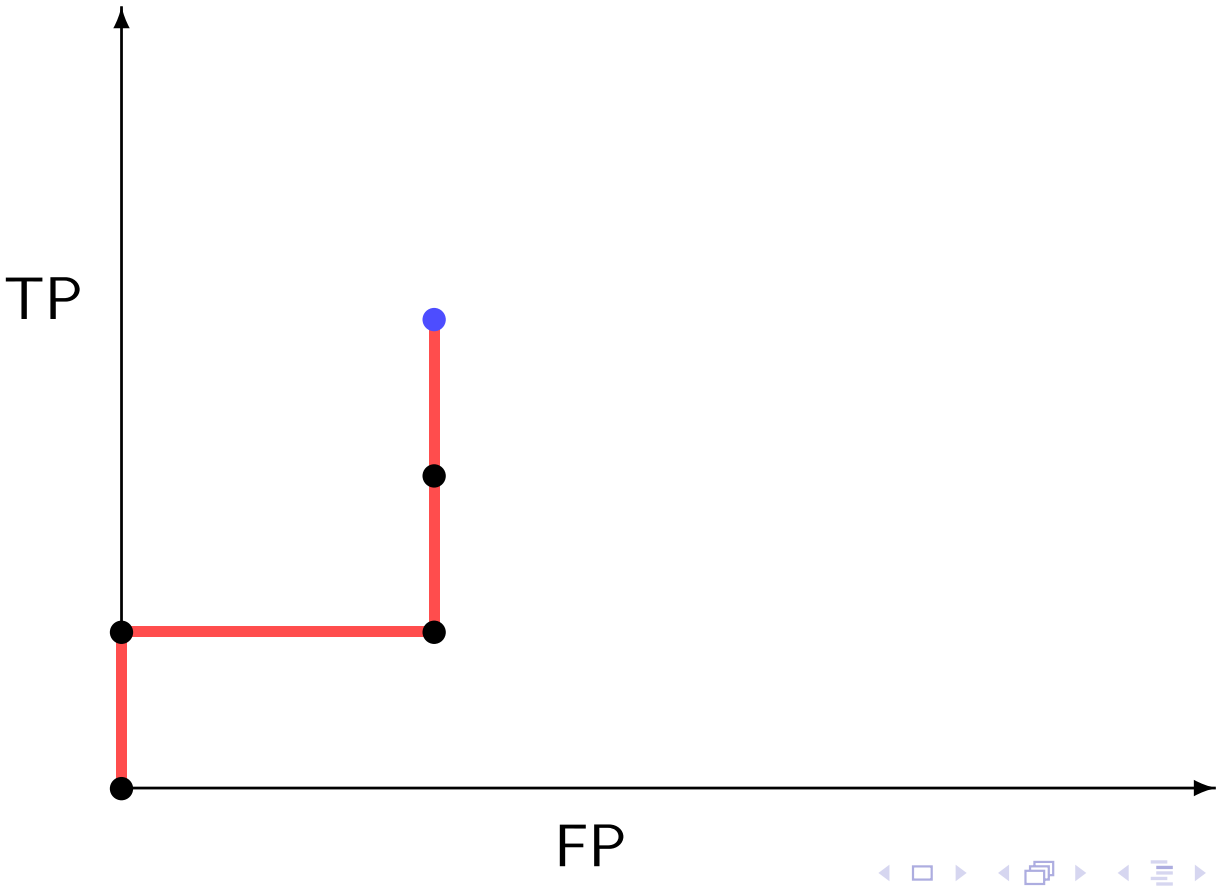
ROC Curve



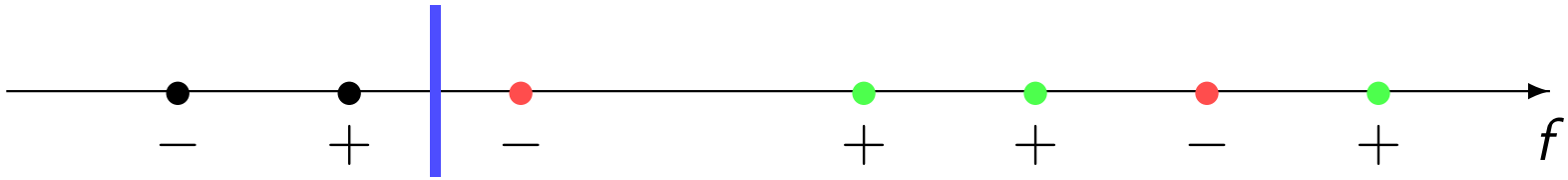


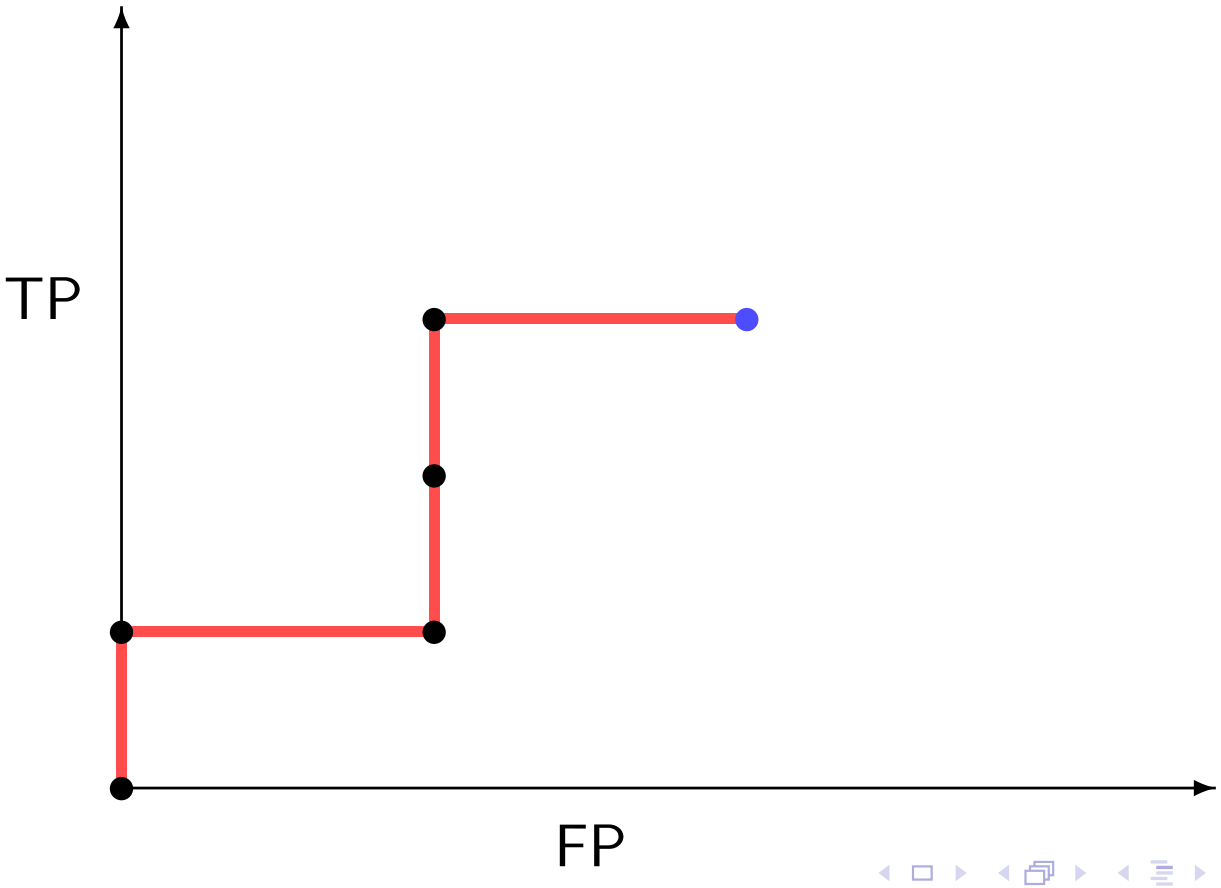
ROC Curve



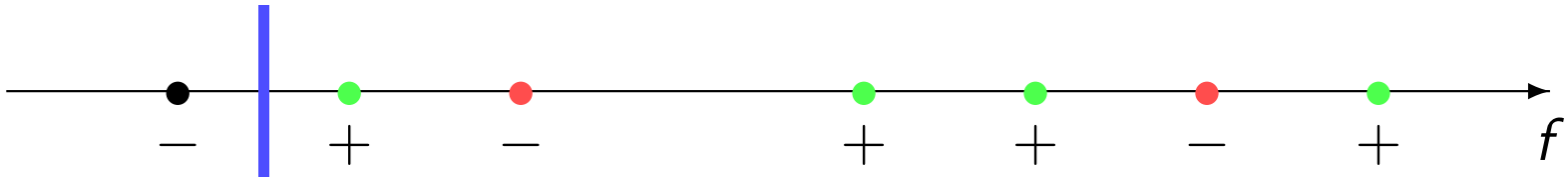


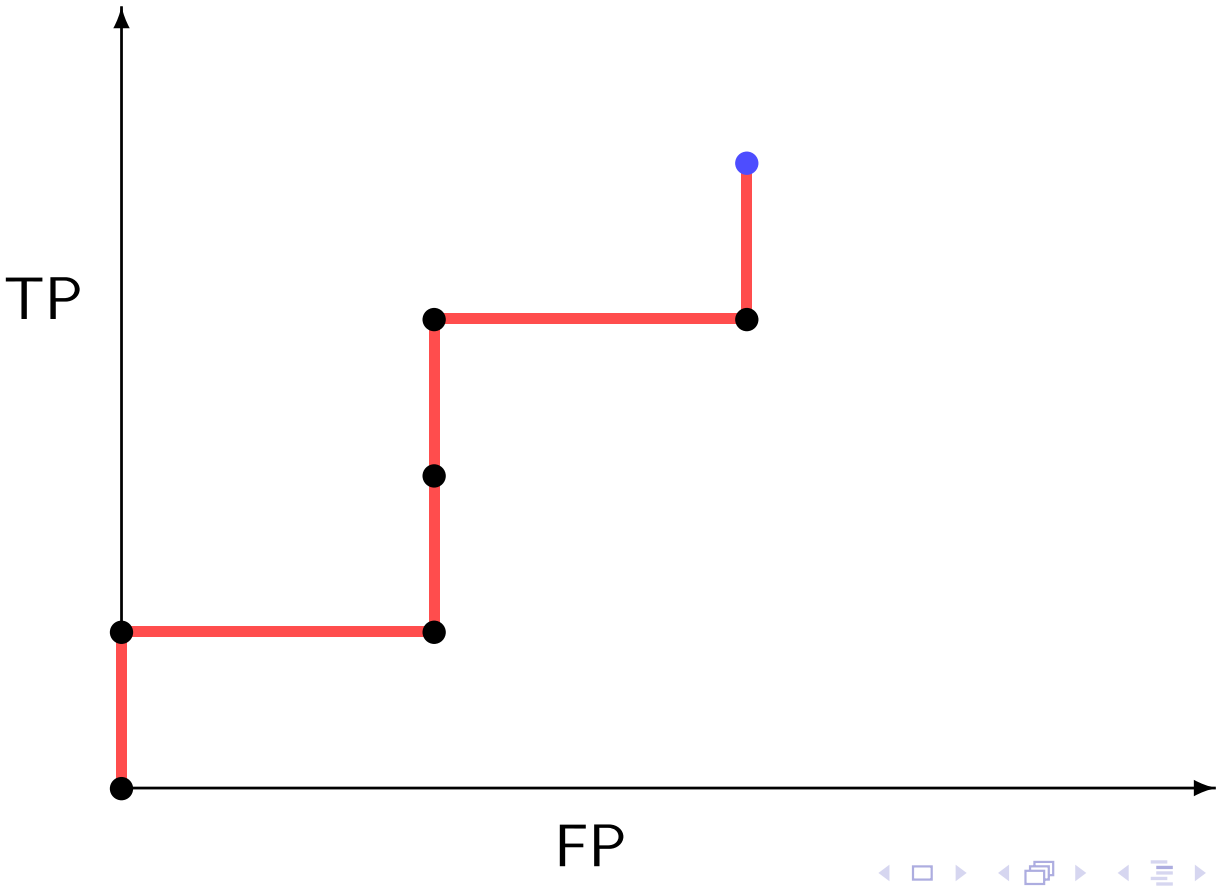
ROC Curve



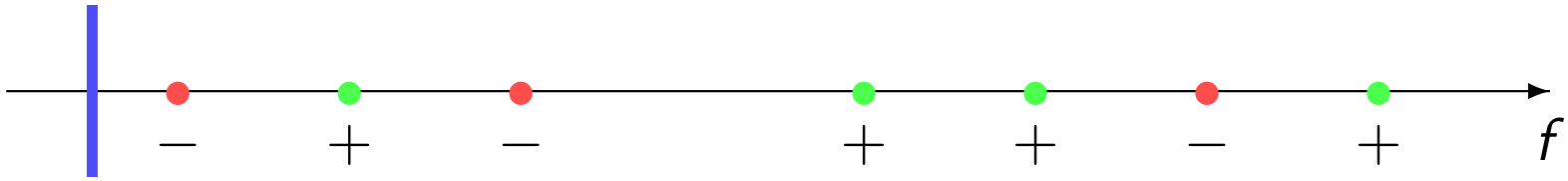


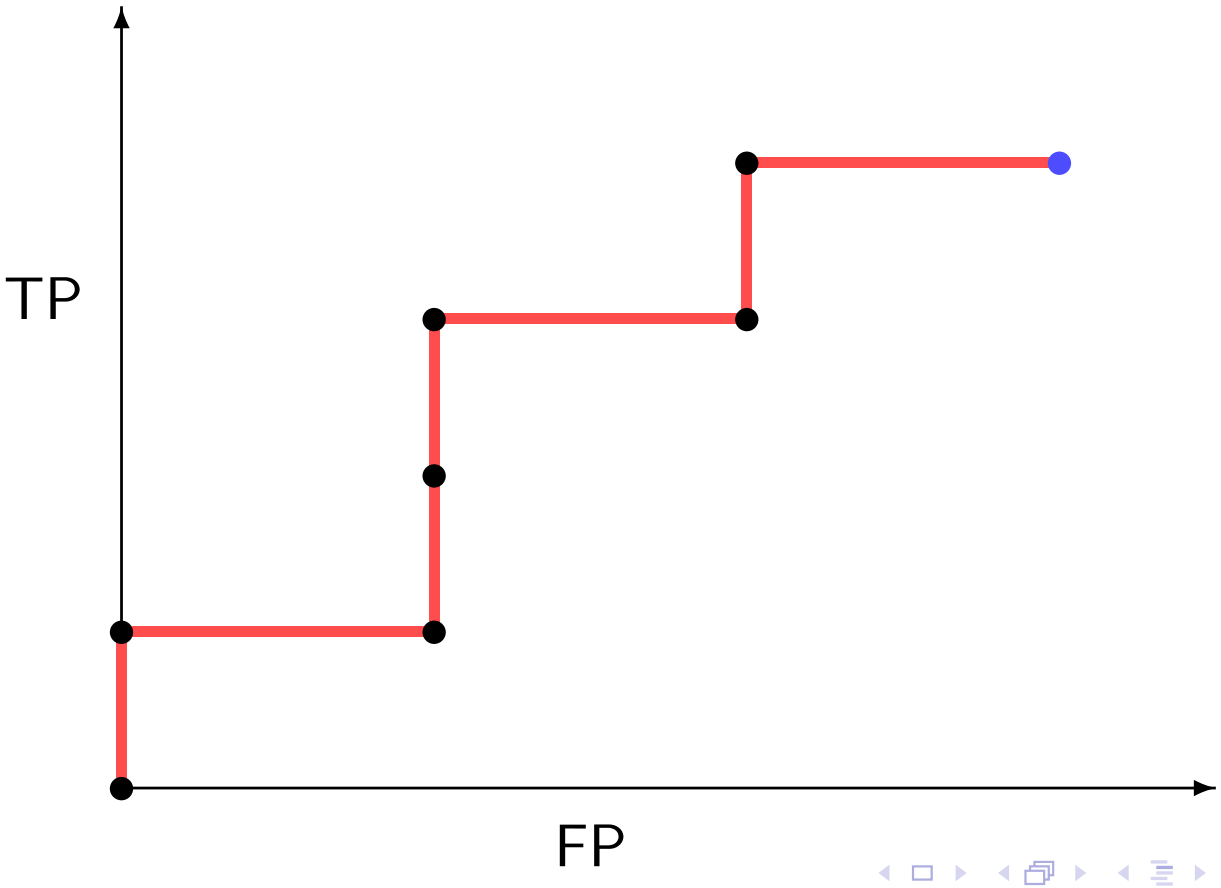
ROC Curve



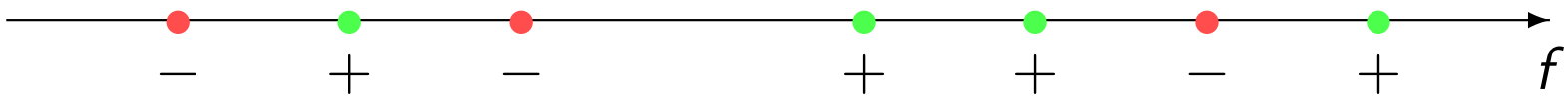


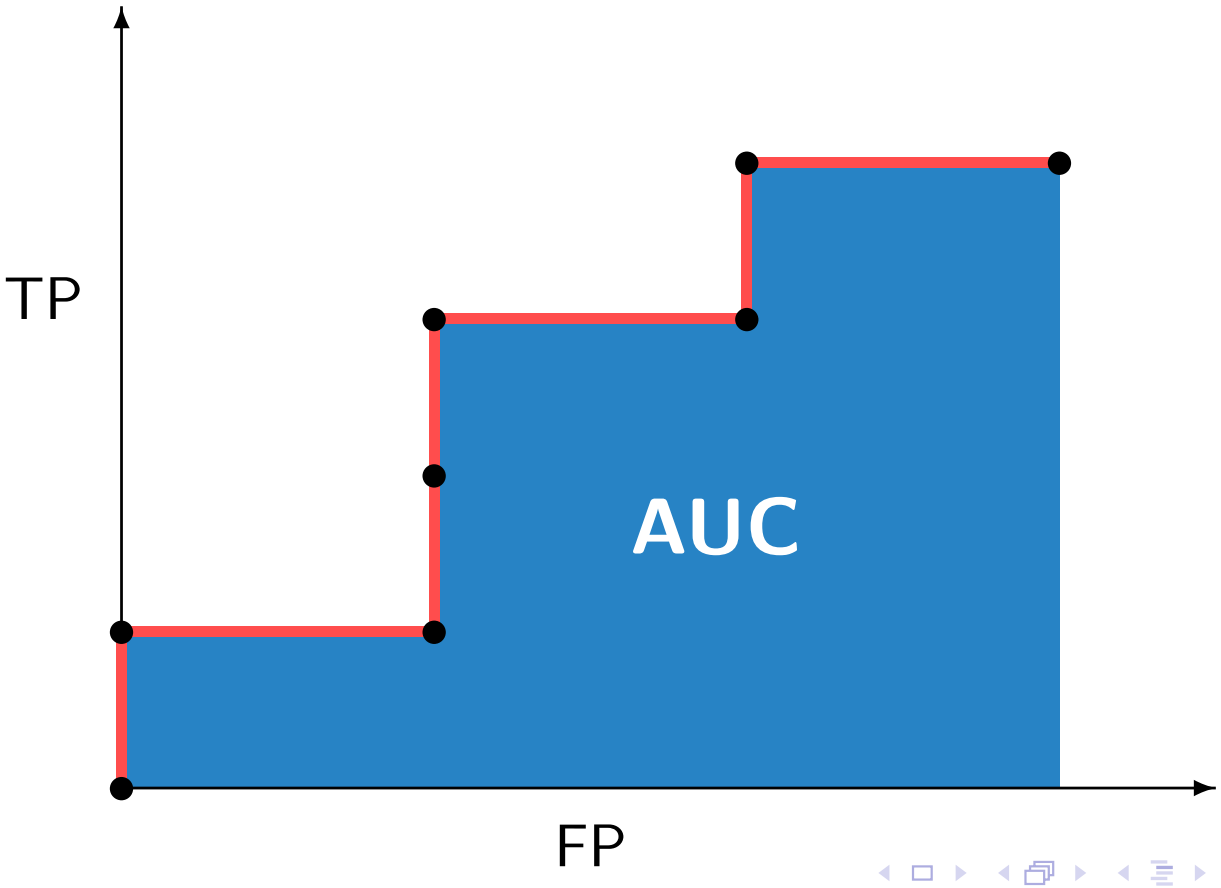
ROC Curve





ROC Curve





Area under ROC curve (AUC)

- Area under the curve has a statistical interpretation
  - $AUC = Pr\{f(x) < f(y) \mid x \in \text{class}_0, y \in \text{class}_1\}$

Definition: AUC of f on a dataset

$$AUC(f) = \frac{\sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \mathbf{1}[f(t_1) - f(t_0)]}{|\text{class}_0| \cdot |\text{class}_1|}$$

where $\mathbf{1}[\,]$ is the step function:

$$\mathbf{1}[x] = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases}$$

Properties of AUC

- AUC can be computed in $\mathcal{O}(|\mathcal{D}| \log |\mathcal{D}|)$:

1: sort $\mathcal{D}$ descending on f
2: **for all** $t \in \mathcal{D}$ **do**
3: **if** $t \in \text{class}_1$ **then**
4: $TRUE\_POS := TRUE\_POS + 1$;
5: **if** $t \in \text{class}_0$ **then**
6: $AUC := AUC + TRUE\_POS$;

Properties of AUC

- AUC can be computed in $\mathcal{O}(|\mathcal{D}| \log |\mathcal{D}|)$:

1: sort $\mathcal{D}$ descending on f $\leftarrow$ SLOW FOR LARGE $\mathcal{D}$
2: **for all** $t \in \mathcal{D}$ **do**
3: **if** $t \in \text{class}_1$ **then**
4: $TRUE\_POS := TRUE\_POS + 1$;
5: **if** $t \in \text{class}_0$ **then**
6: $AUC := AUC + TRUE\_POS$;

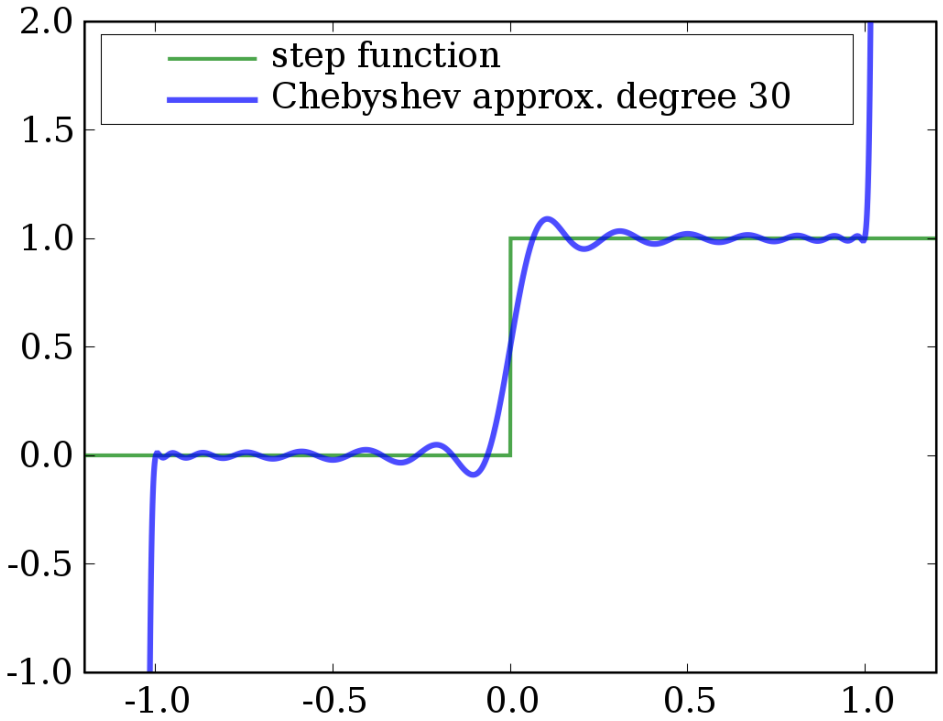
Overview

Approximating the AUC using polynomials

Approximate the step function using a polynomial

$$\mathbf{1}[x] \approx c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \ldots + c_d x^d,$$

and substitute into the definition of AUC:



Approximating the AUC using polynomials

$$AUC(f) = \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \mathbf{1}[f(t_1) - f(t_0)]$$

Approximating the AUC using polynomials

$$
\begin{aligned}
AUC(f) &= \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \mathbf{1}[f(t_1) - f(t_0)] \\
&\approx \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \sum_{i=0}^{d} c_i \left(f(t_1) - f(t_0)\right)^i
\end{aligned}
$$

Approximating the AUC using polynomials

$$
\begin{aligned}
AUC(f) &= \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \mathbf{1}[f(t_1) - f(t_0)] \\
&\approx \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \sum_{i=0}^{d} c_i \left(f(t_1) - f(t_0)\right)^i \\
&= \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \sum_{i=0}^{d} c_i \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} f^j(t_1) f^{i-j}(t_0)
\end{aligned}
$$

Approximating the AUC using polynomials

$$
\begin{aligned}
AUC(f) &= \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \mathbf{1}[f(t_1) - f(t_0)] \\
&\approx \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \sum_{i=0}^{d} c_i \left(f(t_1) - f(t_0)\right)^i \\
&= \sum_{t_0 \in \text{class}_0} \sum_{t_1 \in \text{class}_1} \sum_{i=0}^{d} c_i \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} f^j(t_1) f^{i-j}(t_0) \\
&= \sum_{i=0}^{d} \sum_{j=0}^{i} \alpha_{ij} \left(\sum_{t_0 \in \text{class}_0} f^{i-j}(t_0) \right) \left(\sum_{t_1 \in \text{class}_1} f^i(t_1) \right)
\end{aligned}
$$

Approximating the AUC using polynomials

$$AUC(f) \approx \sum_{i=0}^{d} \sum_{j=0}^{i} \alpha_{ij} \underbrace{\left(\sum_{t_0 \in \text{class}_0} f(t_0)^{i-j} \right)}_{\mathcal{O}(|\mathcal{D}|)} \underbrace{\left(\sum_{t_1 \in \text{class}_1} f(t_1)^{j} \right)}_{\mathcal{O}(|\mathcal{D}|)}$$

Total cost: $\mathcal{O}(d \cdot |\mathcal{D}| + d^2) = \mathcal{O}(d \cdot |\mathcal{D}|)$

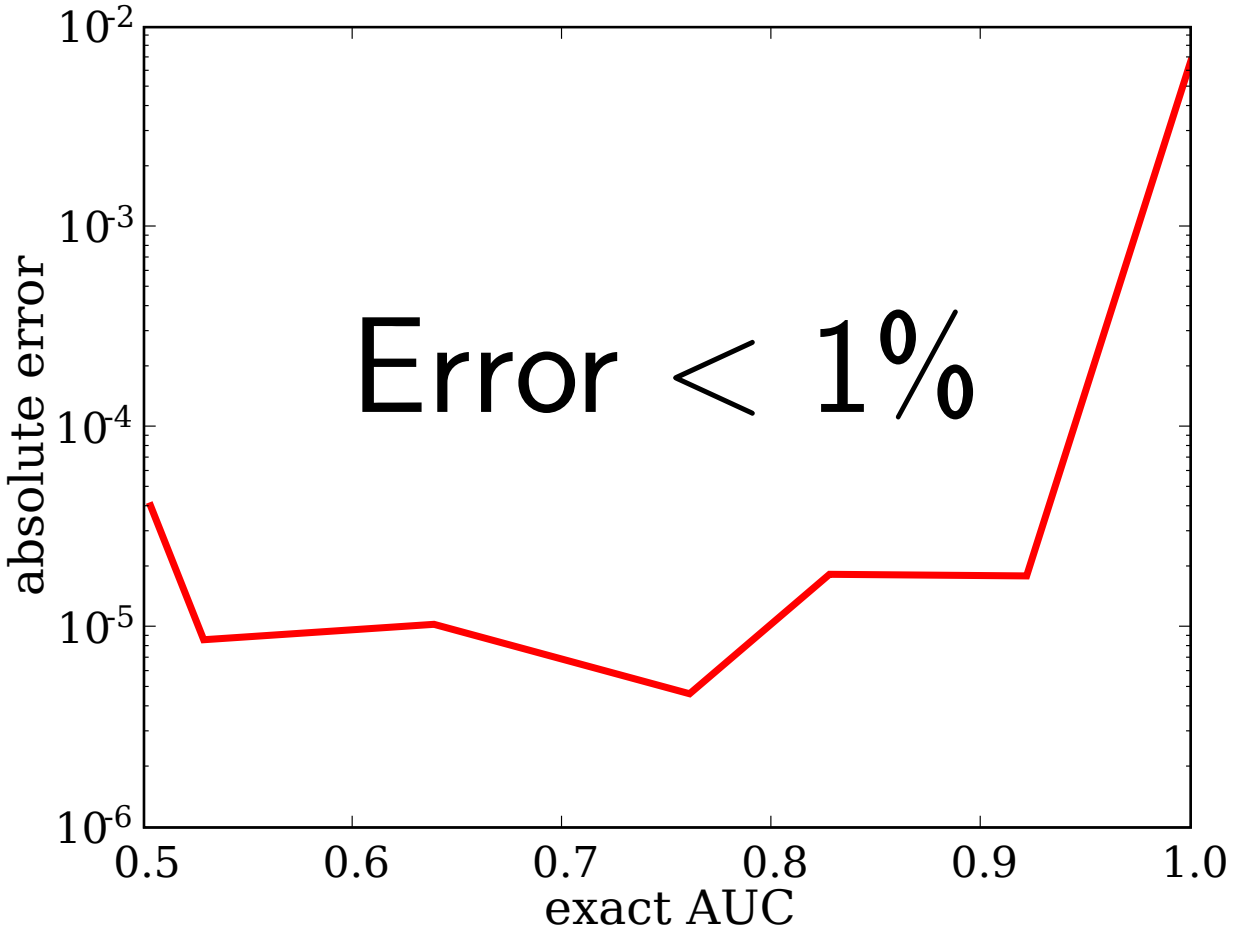
- Linear in database size
- Only one data scan (can be used for data streams)

Accuracy of approximation—first experiment

- $f \sim Normal(0, 1)$ for class 0
- $f \sim Normal(m, 1)$ for class 1

Accuracy of approximation—first experiment

- $f \sim Normal(0, 1)$ for class 0
- $f \sim Normal(m, 1)$ for class 1



Approximating the AUC using polynomials

Take-away message

- Exact AUC requires sorting the database ($\mathcal{O}(|\mathcal{D}| \log(|\mathcal{D}|))$)
- Our polynomial approximation of AUC is:

  efficient: only 1 database scan (no sorting required)

  accurate: usually less than 1% error; often 4-5 digits

Overview

Application - maximizing AUC for classification

- Build a classifier maximizing AUC directly
  - Linear classifier $f_{\mathbf{w}} = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m$

Application - maximizing AUC for classification

- Build a classifier maximizing AUC directly
  - Linear classifier $f_{\mathbf{w}} = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m$

Optimization Problem

Find weights $\mathbf{w} = (w_1, \ldots, w_m)$ such that $AUC(f_{\mathbf{w}})$ is maximized.

Application - maximizing AUC for classification

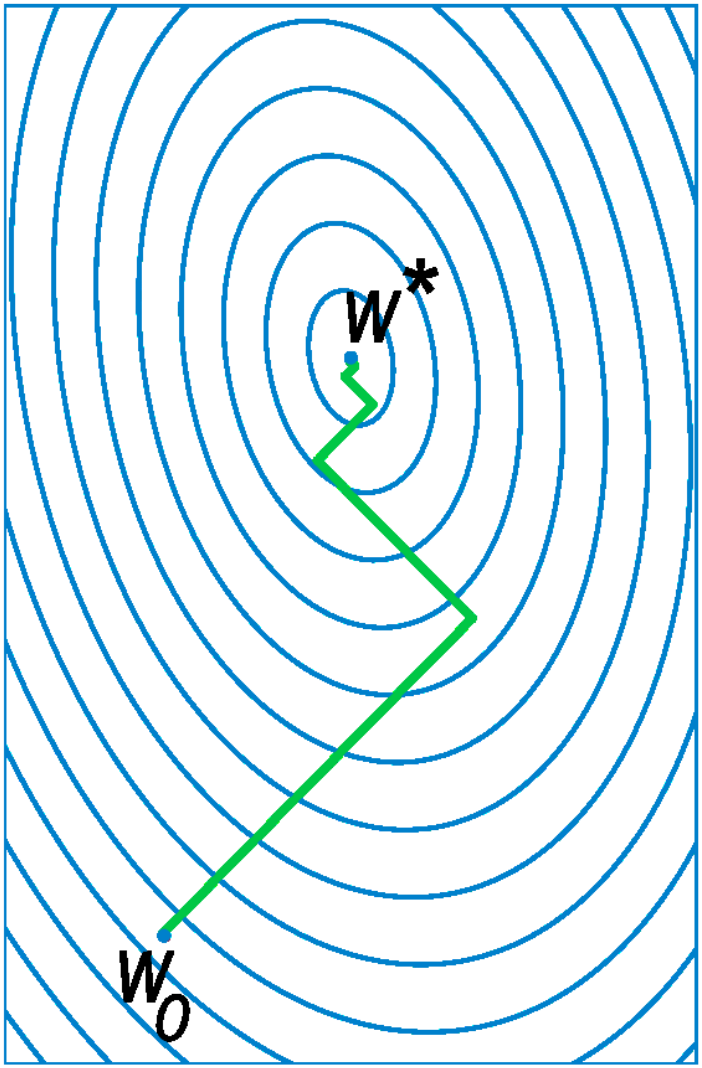
- Build a classifier maximizing AUC directly
  - Linear classifier $f_{\mathbf{w}} = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m$

Optimization Problem

Find weights $\mathbf{w} = (w_1, \ldots, w_m)$ such that $AUC(f_{\mathbf{w}})$ is maximized.

  - Use gradient descent method

Gradient Descent



1: **for** *iter* := 1 . . . *maxiter* **do**
2: Approximate gradient $\mathbf{g}$
3: Find λ that maximizes $AUC(\mathbf{w} + \lambda\mathbf{g})$ via a linear optimization
4: $\mathbf{w} \leftarrow \mathbf{w} + \lambda\mathbf{g}$
5: **return w**

Computing the gradient

- Gradient is $\left[\frac{\partial AUC(f)}{\partial w_1}, \ldots, \frac{\partial AUC(f)}{\partial w_m}\right]$

Computing the gradient

- Gradient is $\left[\frac{\partial AUC(f)}{\partial w_1}, \ldots, \frac{\partial AUC(f)}{\partial w_m}\right]$
- $\frac{\partial AUC(f)}{\partial w_k}$ comes down to deriving a polynomial

Computing the gradient

- Gradient is $\left[\frac{\partial AUC(f)}{\partial w_1}, \ldots, \frac{\partial AUC(f)}{\partial w_m}\right]$
- $\frac{\partial AUC(f)}{\partial w_k}$ comes down to deriving a polynomial
- Final result: $\frac{\partial AUC(f)}{\partial w_k} \approx$

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left[j\cdot\left(\sum_{t_1\in \text{class}_1} x_k f^{j-1}(t_1)\right)\cdot\left(\sum_{t_0\in \text{class}_0} f^{i-j}(t_0)\right)\right.$$
$$\left.+\,(i-j)\cdot\left(\sum_{t_1\in \text{class}_1} f^{j}(t_1)\right)\cdot\left(\sum_{t_0\in \text{class}_0} x_k f^{i-j-1}(t_0)\right)\right]$$

Computing the gradient

- Gradient is $\left[\frac{\partial AUC(f)}{\partial w_1}, \ldots, \frac{\partial AUC(f)}{\partial w_m}\right]$
- $\frac{\partial AUC(f)}{\partial w_k}$ comes down to deriving a polynomial
- Final result: $\frac{\partial AUC(f)}{\partial w_k} \approx$

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left[j\cdot\left(\sum_{t_1\in\text{class}_1}x_k f^{j-1}(t_1)\right)\cdot\left(\sum_{t_0\in\text{class}_0}f^{i-j}(t_0)\right)\right.$$
$$\left.+\,(i-j)\cdot\left(\sum_{t_1\in\text{class}_1}f^{j}(t_1)\right)\cdot\left(\sum_{t_0\in\text{class}_0}x_k f^{i-j-1}(t_0)\right)\right]$$

Computing the gradient is easy; *requires only one database scan.*

Computing values of $AUC(f)$ along the gradient

- Suppose gradient direction $\mathbf{g}$ is found
- Update weights as follows:

$$\mathbf{w} \leftarrow \mathbf{w} + \lambda \mathbf{g}$$

- What is the optimal λ?

Computing values of $AUC(f)$ along the gradient

- Suppose gradient direction $\mathbf{g}$ is found
- Update weights as follows:

$$\mathbf{w} \leftarrow \mathbf{w} + \lambda \mathbf{g}$$

- What is the optimal λ?

$$AUC(f_{\mathbf{w}+\lambda\mathbf{g}}) \approx \sum_{i=0}^{d} \sum_{j=0}^{i} \alpha_{ij} \left(\sum_{t_0 \in \text{class}_0} f^{i-j}_{\mathbf{w}+\lambda\mathbf{g}}(t_0) \right) \left(\sum_{t_1 \in \text{class}_1} f^{j}_{\mathbf{w}+\lambda\mathbf{g}}(t_1) \right)$$

Computing values of $AUC(f)$ along the gradient

$$\sum_{i=0}^{d} \sum_{j=0}^{i} \alpha_{ij} \left(\sum_{t_0 \in \text{class}_0} f_{\mathbf{w}+\lambda\mathbf{g}}^{i-j}(t_0) \right) \left(\sum_{t_1 \in \text{class}_1} f_{\mathbf{w}+\lambda\mathbf{g}}^{j}(t_1) \right)$$

Computing values of $AUC(f)$ along the gradient

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left(\sum_{t_0\in\text{class}_0} f^{i-j}_{\mathbf{w}+\lambda\mathbf{g}}(t_0)\right)\left(\sum_{t_1\in\text{class}_1} f^{j}_{\mathbf{w}+\lambda\mathbf{g}}(t_1)\right)$$

Computing values of $AUC(f)$ along the gradient

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left(\sum_{t_0\in\text{class}_0} f^{i-j}_{\mathbf{w}+\lambda\mathbf{g}}(t_0)\right)\left(\sum_{t_1\in\text{class}_1} f^{j}_{\mathbf{w}+\lambda\mathbf{g}}(t_1)\right)$$

Since f is linear: $f_{\mathbf{w}+\lambda\mathbf{g}} = f_{\mathbf{w}} + \lambda f_{\mathbf{g}}$

Computing values of $AUC(f)$ along the gradient

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left(\sum_{t_0\in\text{class}_0} f_{\mathbf{w}+\lambda\mathbf{g}}^{i-j}(t_0)\right)\left(\sum_{t_1\in\text{class}_1} f_{\mathbf{w}+\lambda\mathbf{g}}^{j}(t_1)\right)$$

Since f is linear: $f_{\mathbf{w}+\lambda\mathbf{g}} = f_{\mathbf{w}} + \lambda f_{\mathbf{g}}$

$$\sum_{t_1\in\text{class}_1} \left(f_{\mathbf{w}}(t_1) + \lambda f_{\mathbf{g}}(t_1)\right)^j$$

Computing values of $AUC(f)$ along the gradient

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left(\sum_{t_0\in\text{class}_0} f_{\mathbf{w}+\lambda\mathbf{g}}^{i-j}(t_0)\right)\left(\sum_{t_1\in\text{class}_1} f_{\mathbf{w}+\lambda\mathbf{g}}^{j}(t_1)\right)$$

Since f is linear: $f_{\mathbf{w}+\lambda\mathbf{g}} = f_{\mathbf{w}} + \lambda f_{\mathbf{g}}$

$$\sum_{t_1\in\text{class}_1}\left(f_{\mathbf{w}}(t_1)+\lambda f_{\mathbf{g}}(t_1)\right)^j$$

$$= \sum_{t_1\in\text{class}_1}\sum_{k=0}^{j}\lambda^{j-k}\binom{j}{k} f_{\mathbf{w}}^{k}(t_1) f_{\mathbf{g}}^{j-k}(t_1)$$

Computing values of $AUC(f)$ along the gradient

$$\sum_{i=0}^{d}\sum_{j=0}^{i}\alpha_{ij}\left(\sum_{t_0\in\text{class}_0} f_{\mathbf{w}+\lambda\mathbf{g}}^{i-j}(t_0)\right)\left(\sum_{t_1\in\text{class}_1} f_{\mathbf{w}+\lambda\mathbf{g}}^{j}(t_1)\right)$$

Since f is linear: $f_{\mathbf{w}+\lambda\mathbf{g}} = f_{\mathbf{w}} + \lambda f_{\mathbf{g}}$

$$\sum_{t_1\in\text{class}_1} \left(f_{\mathbf{w}}(t_1) + \lambda f_{\mathbf{g}}(t_1)\right)^j$$

$$= \sum_{t_1\in\text{class}_1}\sum_{k=0}^{j}\lambda^{j-k}\binom{j}{k} f_{\mathbf{w}}^{k}(t_1) f_{\mathbf{g}}^{j-k}(t_1)$$

$$= \sum_{k=0}^{j}\beta_{jk}\left(\sum_{t_1\in\text{class}_1} f_{\mathbf{w}}^{k}(t_1) f_{\mathbf{g}}^{j-k}(t_1)\right)$$

Computing values of $AUC(f)$ along the gradient

Take-away message

1. Collect $2(d+1)^2$ values in one scan over the database:

$$\sum_{t \in \text{class}_c} f_{\mathbf{w}}^r(t) f_{\mathbf{g}}^s(t) \quad c \in \{0,1\}, r, s \in \{0 \ldots d\}$$

2. Maximizing along the gradient is done without accessing $\mathcal{D}$ at all.

Putting it all together

1: **for** *iter* $:= 1 \ldots$ *maxiter* **do**
2: Approximate gradient $\mathbf{g}$
3: Find λ that maximizes $AUC(\mathbf{w} + \lambda\mathbf{g})$ via a linear optimization
4: $\mathbf{w} \leftarrow \mathbf{w} + \lambda\mathbf{g}$
5: **return w**

Putting it all together

1: **for** *iter* $:= 1 \ldots$ *maxiter* **do**
2: Approximate gradient $\mathbf{g} \leftarrow$ 1 scan $(\mathcal{O}(d|\mathcal{D}|))$
3: Find λ that maximizes $AUC(\mathbf{w} + \lambda\mathbf{g})$ via a linear optimization
4: $\mathbf{w} \leftarrow \mathbf{w} + \lambda\mathbf{g}$
5: **return w**

Putting it all together

1: **for** *iter* $:= 1 \ldots$ *maxiter* **do**
2: Approximate gradient $\mathbf{g} \leftarrow$ 1 scan $(\mathcal{O}(d|\mathcal{D}|))$
3: Find λ that maximizes $AUC(\mathbf{w} + \lambda\mathbf{g})$ via a linear optimization $\leftarrow$ 1 scan: $\mathcal{O}(d^2|\mathcal{D}|)$
4: $\mathbf{w} \leftarrow \mathbf{w} + \lambda\mathbf{g}$
5: **return w**

Putting it all together

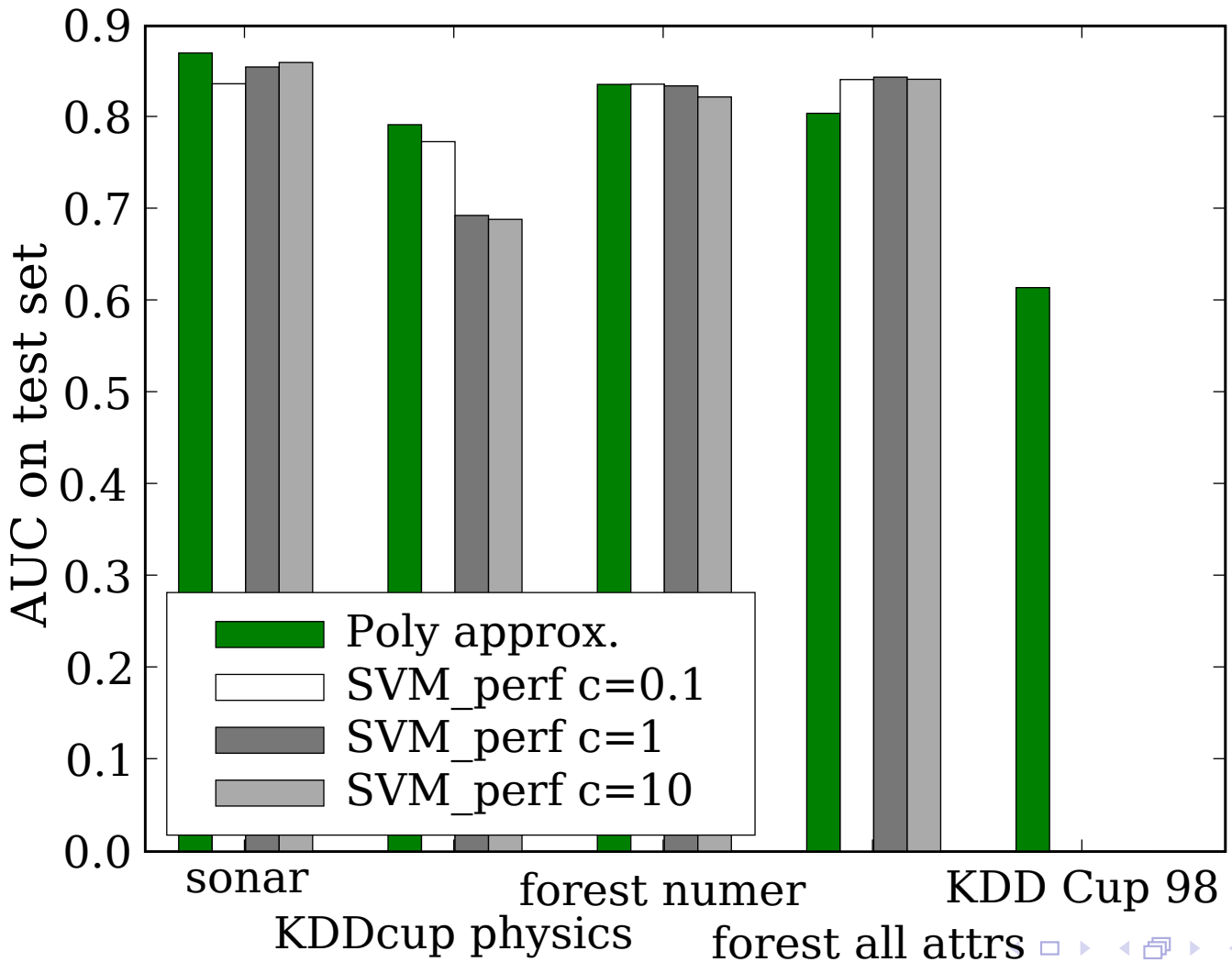
1: **for** *iter* $:= 1 \ldots$ *maxiter* **do**
2: Approximate gradient $\mathbf{g} \leftarrow$ 1 scan $(\mathcal{O}(d|\mathcal{D}|))$
3: Find λ that maximizes $AUC(\mathbf{w} + \lambda\mathbf{g})$ via a linear optimization $\leftarrow$ 1 scan: $\mathcal{O}(d^2|\mathcal{D}|)$
4: $\mathbf{w} \leftarrow \mathbf{w} + \lambda\mathbf{g}$
5: **return w**

Total time

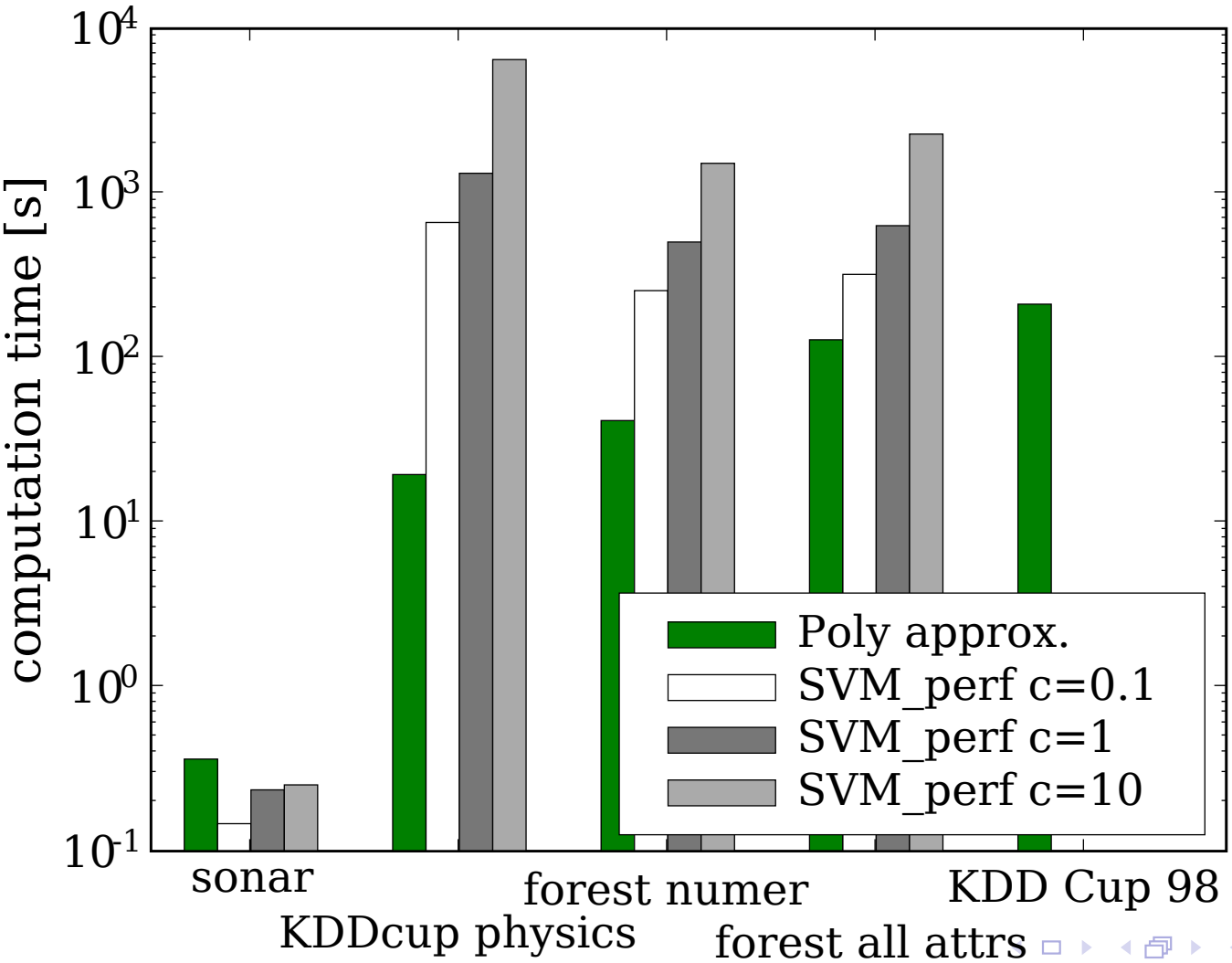
- $2 \cdot$ *maxiter* database scans
- Total complexity: $\mathcal{O}(\textit{maxiter} \cdot d^2|\mathcal{D}|)$

Overview

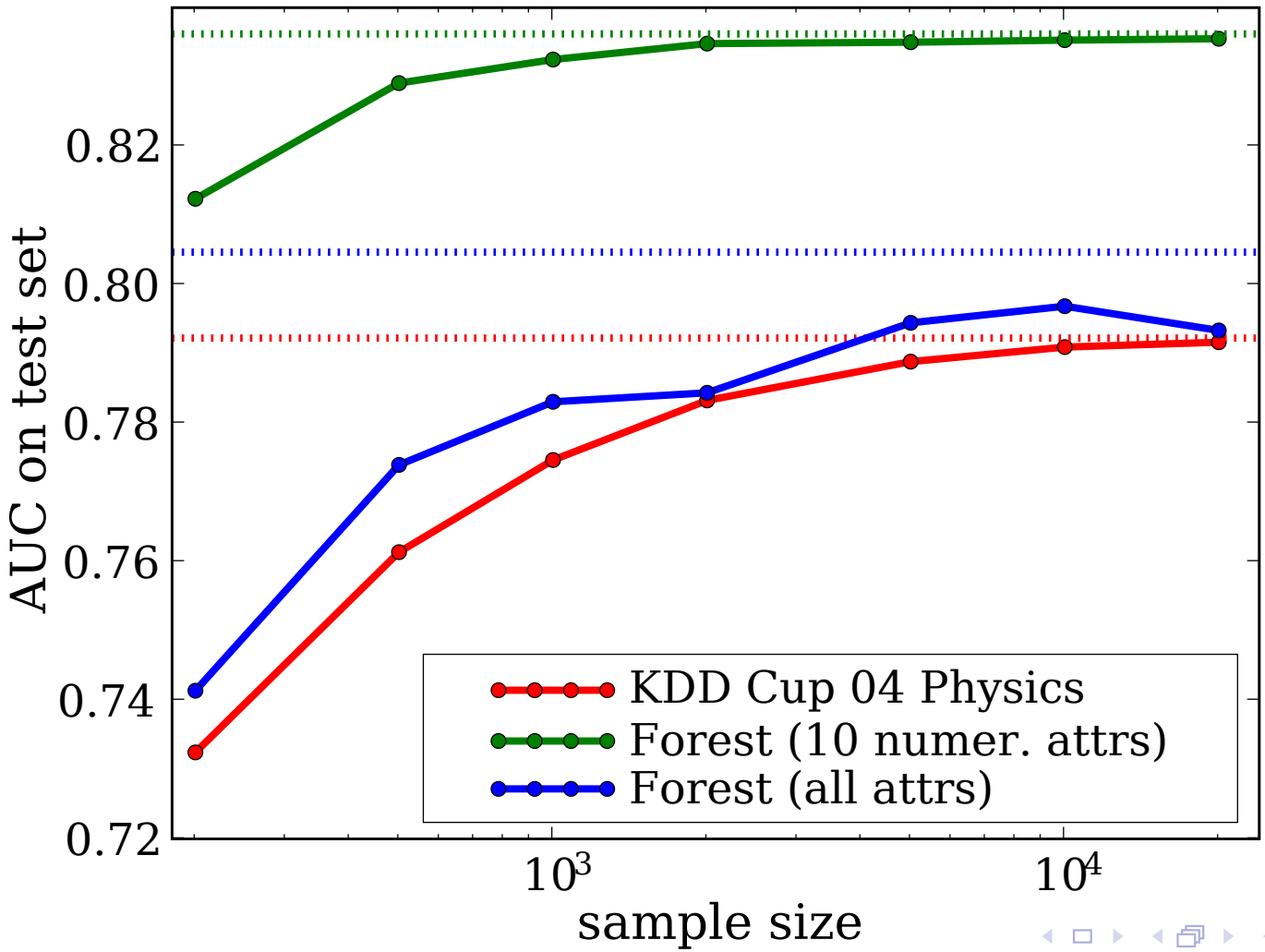
Empirical tests—AUC
Approximation vs SVM_Perf



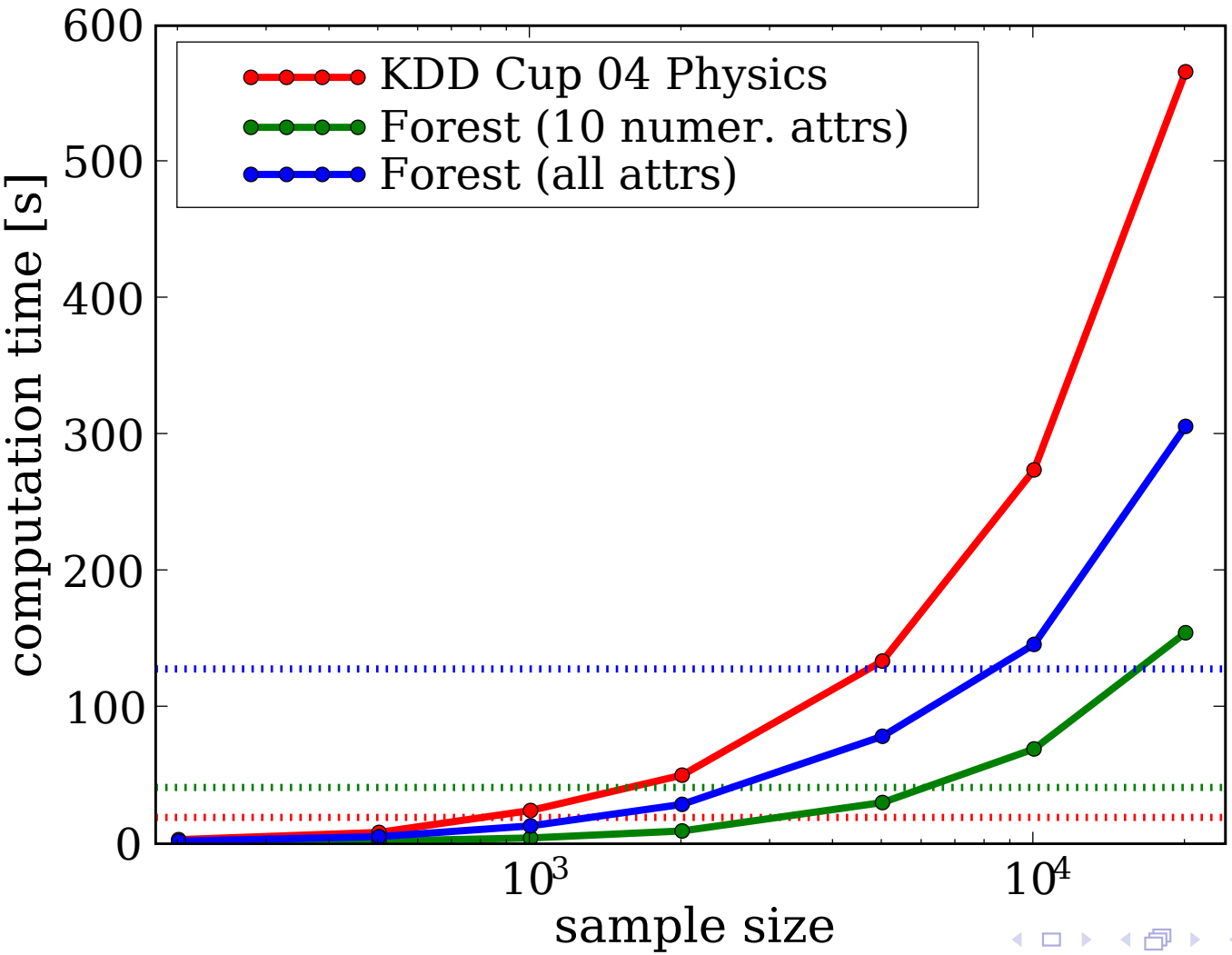
Empirical tests—Time
Approximation vs SVM_Perf



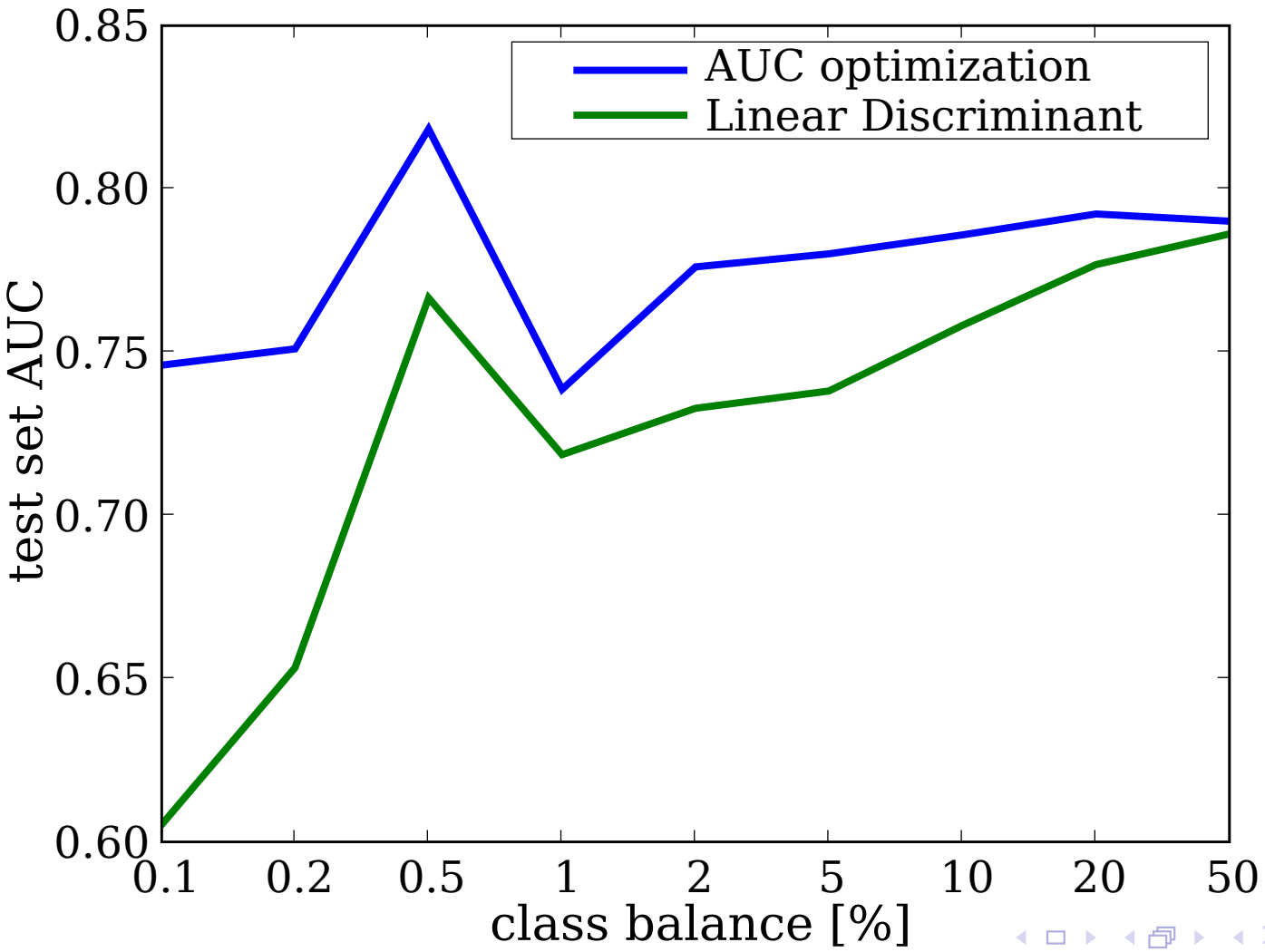
Empirical tests—AUC
Approximation vs Sampling



Empirical tests—Time
Approximation vs Sampling



Empirical tests—Skewed class distribution
KDD cup physics dataset



Overview

Conclusions and Future Work

- Efficient approximation for AUC.
- Allows for adapting gradient descent method to directly optimize linear classifier.
- Empirical results are promising

Future work—Open problems

- Extend to other measures
  - In the paper: soft-AUC
  - ...
- Extending the approach to nonlinear classifiers
- Other applications
  - In SDM'07: Efficient curve fitting
  - Stream mining
  - ...

Questions?