

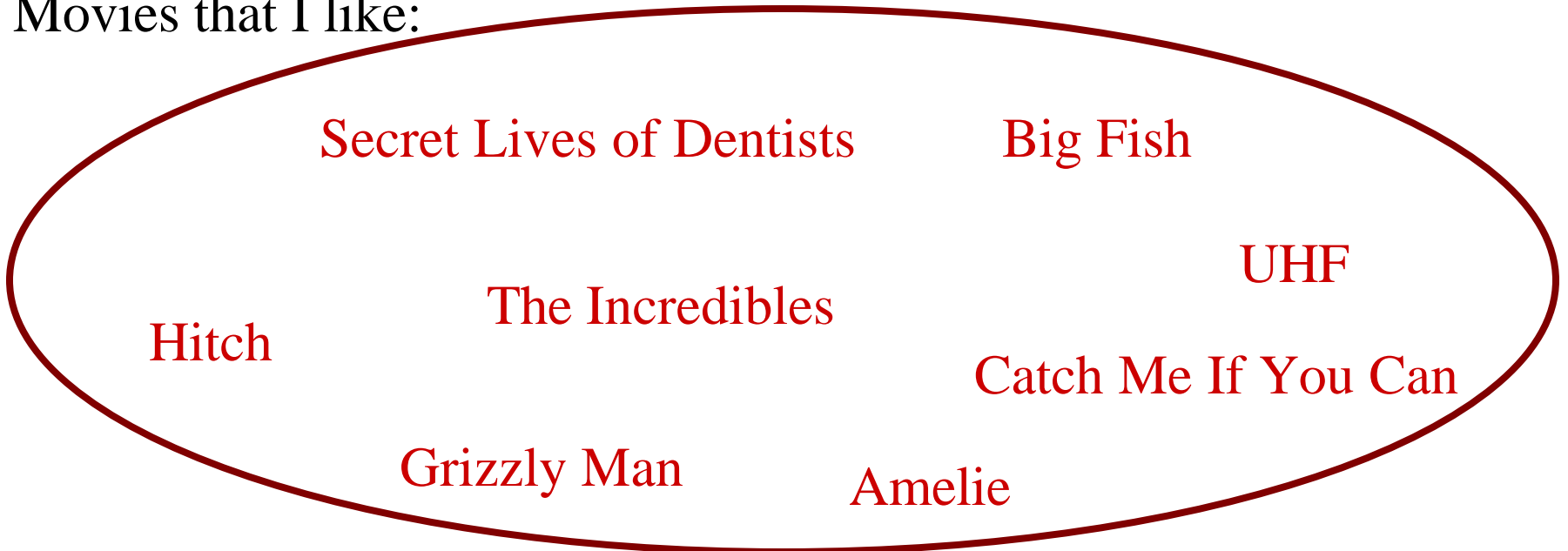
Ranking with a P-Norm Push

Cynthia Rudin

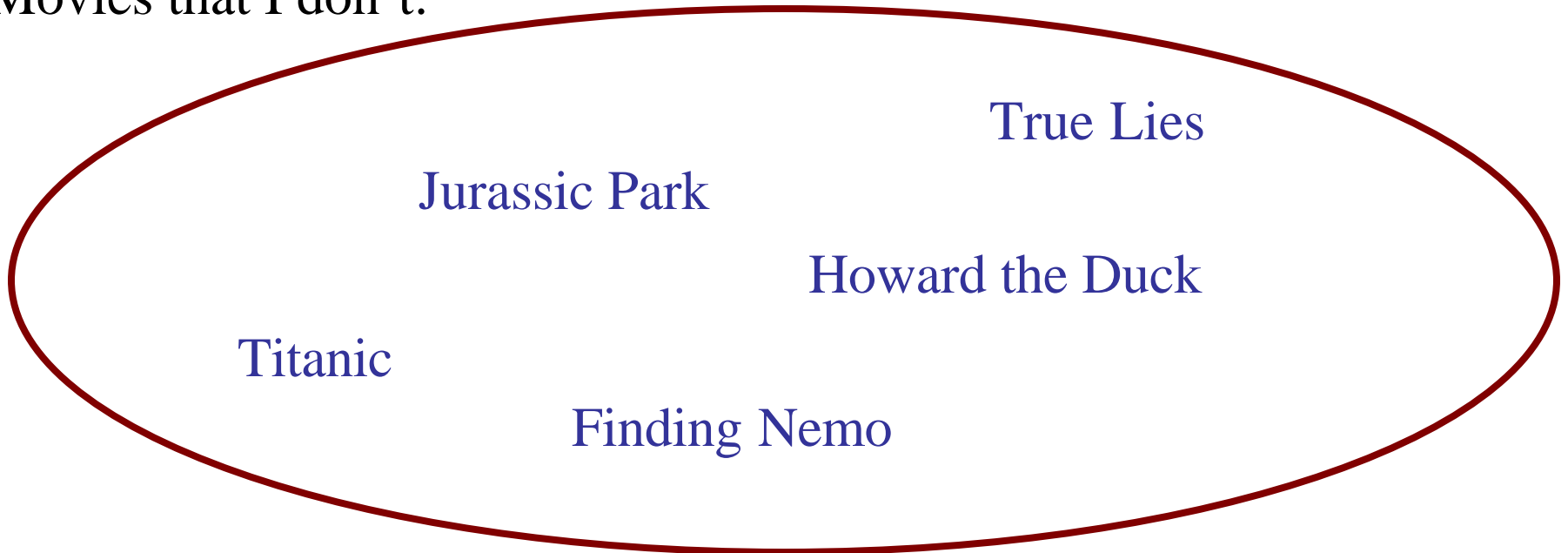
NSF Postdoctoral Research Fellow
NYU Center for Neural Science and Courant Institute

Impromptu session, Snowbird 2006

Movies that I like:



Movies that I don't:



Consider the following real-world problem:

Cynthia wants to go to the movies on Friday night...
... and she would like to see a good movie.

Where can she find a good movie recommendation
given her taste in movies?

IMDB - All Time Worldwide Box Office

| Rank | Title | Worldwide Box Office |
|------|--|----------------------|
| 1. | Titanic (1997) | \$1,835,300,000 |
| 2. | The Lord of the Rings: The Return of the King (2003) | \$1,129,219,252 |
| 3. | Harry Potter and the Sorcerer's Stone (2001) | \$968,600,000 |
| 4. | Star Wars: Episode I - The Phantom Menace (1999) | \$922,379,000 |
| 5. | The Lord of the Rings: The Two Towers (2002) | \$921,600,000 |
| 6. | Jurassic Park (1993) | \$919,700,000 |
| 7. | Shrek 2 (2004) | \$880,871,036 |
| 8. | Harry Potter and the Chamber of Secrets (2002) | \$866,300,000 |
| 9. | Finding Nemo (2003) | \$865,000,000 |
| 10. | The Lord of the Rings: The Fellowship of the Ring (2001) | \$860,700,000 |

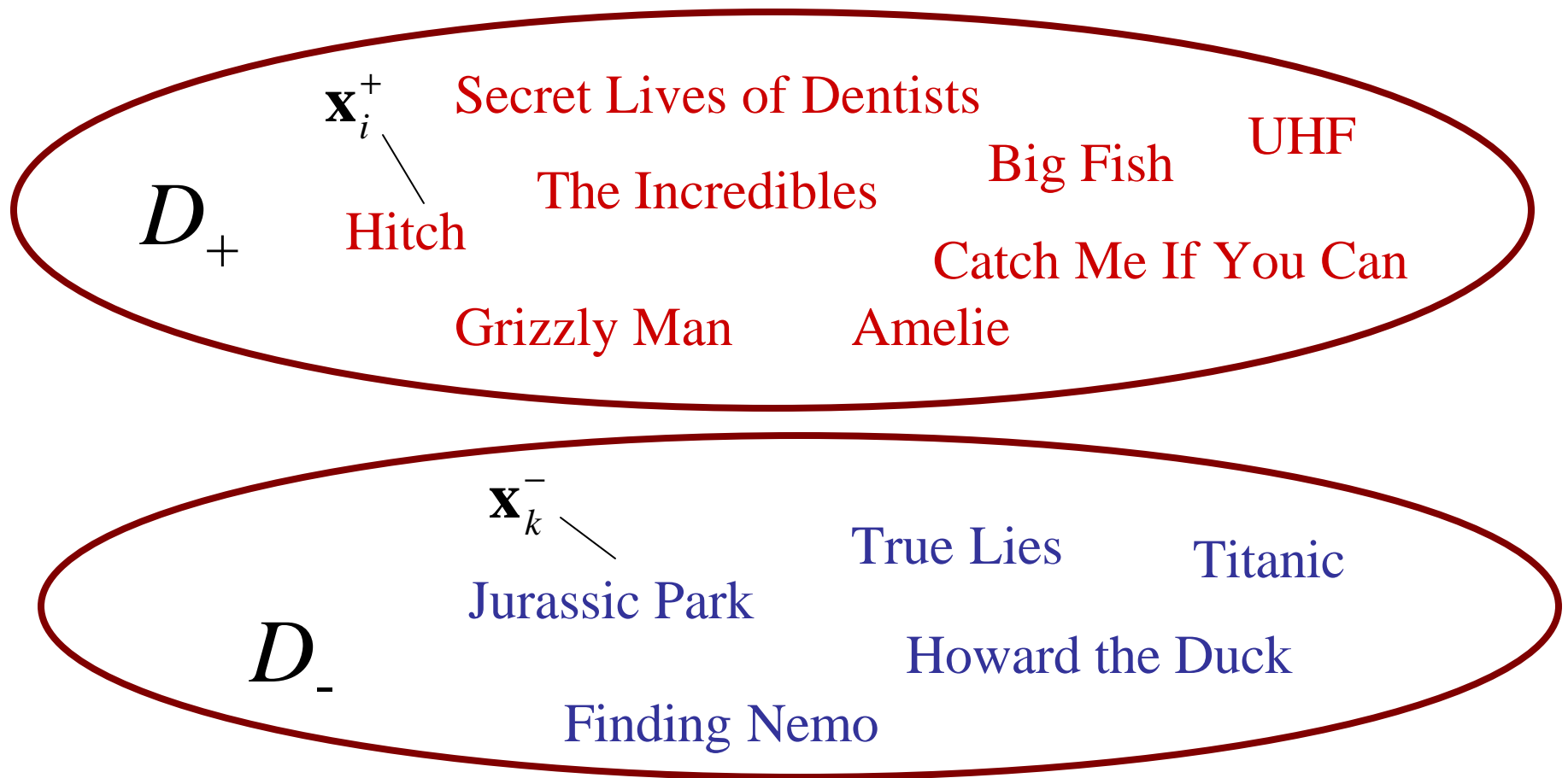
- I don't want a combined list that's supposed to work for everybody.
- I want personalized rankings, so it's a supervised learning problem.
- Remember, the best movies should be at the top!

(This is where the p -norms are going to come in handy!)

The Problem of Supervised Bipartite Ranking ∞

Given: *Training Data*

$\mathbf{x}_i^+ \in X, i=1..I$, chosen iid from unknown probability distribution D_+ .
Also $\mathbf{x}_k^- \in X, k=1..K$, chosen from D_- .



The Problem of Supervised Bipartite Ranking \mathfrak{R}

Given: *Training Data*

$\mathbf{x}_i^+ \in X, i=1..I$, chosen iid from unknown probability distribution D_+ .
Also $\mathbf{x}_k^- \in X, k=1..K$, chosen from D_- .

Our Goal: Construct a function $f : X \rightarrow \mathfrak{R}$ such that for $\mathbf{x}_+ \sim D_+$ and $\mathbf{x}_- \sim D_-$, we have $f(\mathbf{x}_+) > f(\mathbf{x}_-)$ with high probability.

(Notation: $\mathbf{x} \sim D$ means \mathbf{x} chosen randomly from D .)

But...

- Waterworld

+ Amelie

+ Big Fish

+

+

The top of the list is most important!

-

+

If this is our ranked list...

-

-

-

-

+

- True Lies

+ Hitch

- Titanic

By the way, this ranking problem isn't just for movies:

- meta search for search engines
- fraud detection (900 phone numbers, sponsored links)
- processes to be completed in a certain order
- natural language processing
- bioinformatics
- pharmaceuticals

HUGE number of applications

Outline of Talk ∞

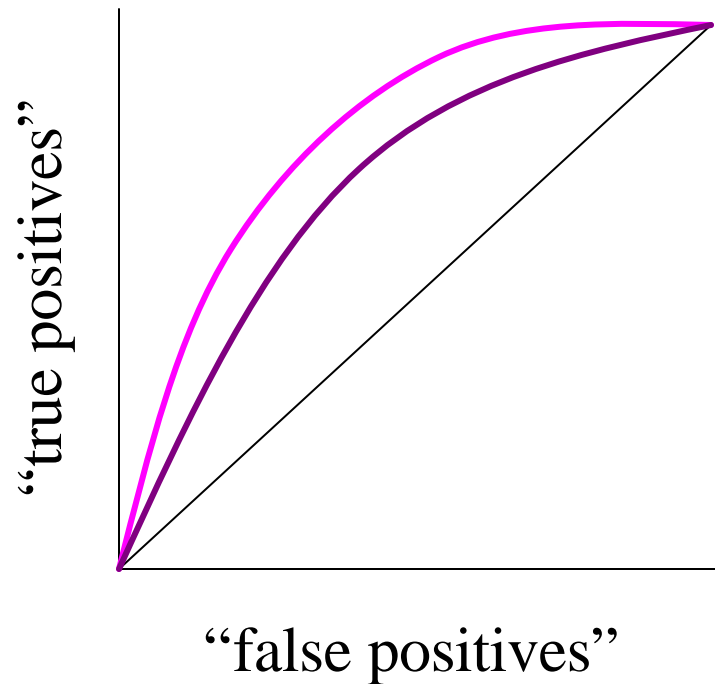
- Introduction to the bipartite supervised ranking problem
(Done)

Our Results :

- 1) Deriving an Objective Function
- 2) The “P-Norm Push” Algorithm
- 3) A Generalization Bound
- 4) Uniqueness

(Rudin, COLT 2006)

How to measure the goodness of a ranked list? use ROC curves!

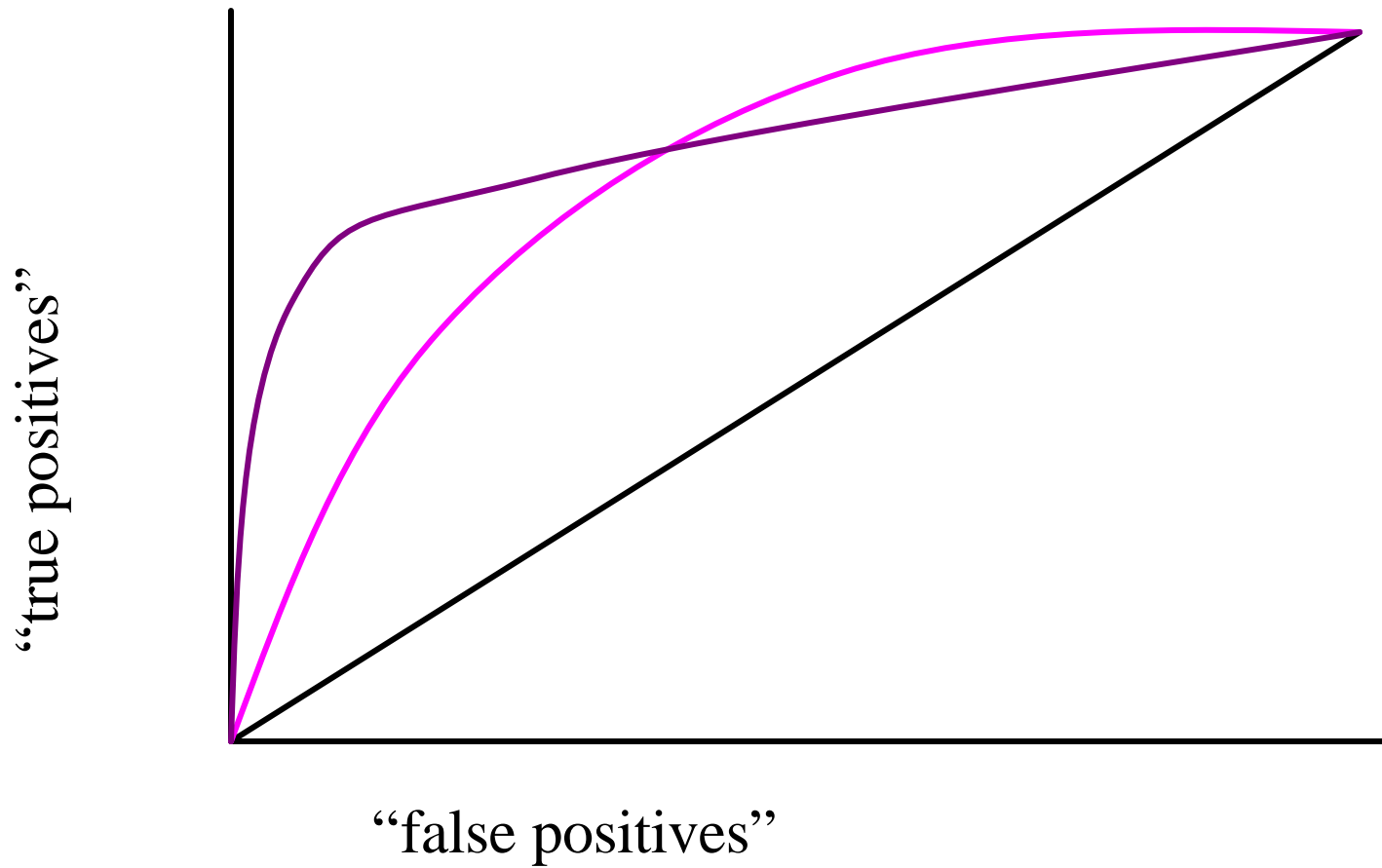


- Algorithms for ranking often evaluate the AUC (Area Under the ROC)

AUC Optimization

- Algorithms for ranking often evaluate the AUC (Area Under the ROC)
 - Boosting AUC Maximizers:
 - RankBoost (Freund et al.)
 - AdaBoost (Freund&Schapire 97) (see Rudin et al. 05)
 - SVM AUC Maximizers: (Yan et al. 03), (Rakotomamonjy 04)
 - Other AUC Maximizers: (Fawcett 01), (Bostrom 05), (Liu and Wu), (Zhang et al 02), (Ferri et al. 03), ...
 - Tutorials on AUC:
 - (Flach 04) At ICML 2004
 - J. A. Hanley and B. J. McNeil. *The meaning and use of the area under the receiver operating characteristic (roc) curve*. Radiology, 143:29-36, 1982

How to measure the goodness of a ranked list? use ROC curves!

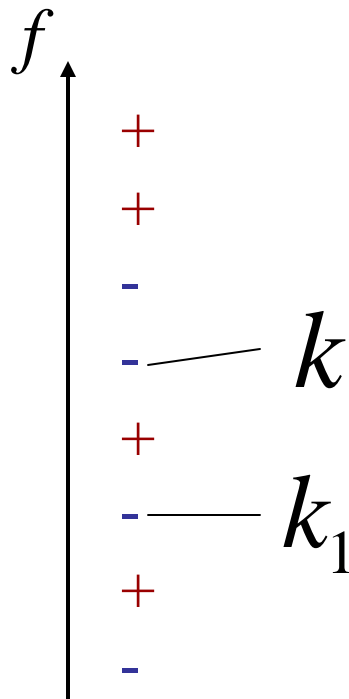


How to measure the goodness of a ranked list? use ROC curves!

- Algorithms for ranking often evaluate the AUC (Area Under the ROC)
 - The AUC concentrates *uniformly* along the ranked list
 - Our problem is slightly different!
 - We care mostly about the *leftmost portion* of the ROC curve
 - It's ok to sacrifice the rightmost portion a bit.
 - Only a small amount of literature on this problem:
(Mozer et al. 02, Yan et al. 03)
- Other related literature: “Log-Linear Models for Label Ranking” (Dekel et al. 03), “Permutation Groups” (Lebanon Lafferty 02), “Partial AUC” (Dodd&Pepe 03)

Deriving A Convex Objective ∞

$$\text{Height_of}(k) = \sum_{i=1}^I \mathbf{1}_{f(\mathbf{x}_i^+) \leq f(\mathbf{x}_k^-)} = \text{Number of positive examples ranked below negative example } k$$



If k is ranked above k_1 then:

$$\text{Height_of}(k) = \text{Height_of}(k_1)$$

We want to concentrate harder on k than on k_1 .

- Put a price on each negative example:

Price for example k is: $g\left(\sum_{i=1}^I 1_{f(\mathbf{x}_i^+) \leq f(\mathbf{x}_k^-)}\right)$

where $g : \mathfrak{R}_+ \rightarrow \mathfrak{R}_+$ is convex, monotonically increasing

Note: When $g(z)=z$, then $R_{g,1} = \text{const}*(1-\text{AUC})$.

- For example,

$$g(z) = z, \quad g(z) = \exp(z), \quad g(z) = z^p \text{ for } p \text{ large}$$

- Objective function

$$R_{g,1}(f) := \sum_{k=1}^K g\left(\sum_{i=1}^I 1_{f(\mathbf{x}_i^+) \leq f(\mathbf{x}_k^-)}\right)$$

f ↑

Let $g(z) = z^4$

+
+
-
-
+
-
+
-

Original Total Price : $2^4 + 2^4 + 1^4 = 33$

Swap lower pair : $2^4 + 2^4 + 1^4 + 1^4 = 34$

Swap higher pair : $3^4 + 2^4 + 1^4 = 98$

$$R_{g,1}(f) := \sum_{k=1}^K g \left(\sum_{i=1}^I \mathbf{1}_{f(\mathbf{x}_i^+) \leq f(\mathbf{x}_k^-)} \right)$$

0-1 Objective:

$$R_{g,1}(f) := \sum_{k=1}^K g \left(\sum_{i=1}^I \mathbf{1}_{f(\mathbf{x}_i^+) \leq f(\mathbf{x}_k^-)} \right)$$

\leq

Related convex objective:

$$R_{g,\ell}(f) := \sum_{k=1}^K g \left(\sum_{i=1}^I \ell(f(\mathbf{x}_i^+) - f(\mathbf{x}_k^-)) \right)$$

where $\ell : \mathfrak{R} \rightarrow \mathfrak{R}_+$ is a convex upper bound on $\mathbf{1}_{z \leq 0}$.

The “P-Norm Push” Algorithm ∞

Related convex objective:

$$R_{g,\ell}(f) := \sum_{k=1}^K g \left(\sum_{i=1}^I \ell(f(\mathbf{x}_i^+) - f(\mathbf{x}_k^-)) \right)$$

Choose $\ell(z) := \exp(-z)$

$$g(z) := z^p \text{ for } p \text{ large}$$

$$F_p(f) := \sum_{k=1}^K \left(\sum_{i=1}^I \exp(-f(\mathbf{x}_i^+) + f(\mathbf{x}_k^-)) \right)^p$$

Choose a form for f (i.e., choose an appropriate hypothesis space)

Boosting-type approach:

$$f(\mathbf{x}) = \sum_{j=1}^n \mathbf{1}_j h_j(\mathbf{x})$$

Where $h_j : X \rightarrow [0,1]$ are “weak rankers”.

The “P-Norm Push” Algorithm \propto

Related convex objective:

$$R_{g,\ell}(f) := \sum_{k=1}^K g \left(\sum_{i=1}^I \ell(f(\mathbf{x}_i^+) - f(\mathbf{x}_k^-)) \right)$$

Choose $\ell(z) := \exp(-z)$

$g(z) := z^p$ for p large

Boosting algorithms combine weak learning rules to create a strong learning rule. (Schapire ‘89)

h_1 = Movies where most of the characters survive to the end are good

$$h_1(\text{Amelie})=1, h_1(\text{Spiderman})=1,$$

$$h_1(\text{The Matrix})=0, h_1(\text{Titanic})=0, h_1(\text{Boogeyman})=0$$

h_n = Classic superhero movies are good

$$h_n(\text{Spiderman})=1,$$

$$h_n(\text{The Matrix})=0, h_n(\text{Titanic})=0, h_n(\text{Boogeyman})=0, h_n(\text{Amelie})=0$$

Where $h_j : X \rightarrow [0,1]$ are “weak rankers”.

The “P-Norm Push” Algorithm ∞

$$\min_{\mathbf{I} \in \mathcal{R}^n} F_p(\mathbf{I}) := \sum_{k=1}^K \left(\sum_{i=1}^I \exp(-f(\mathbf{x}_i^+) + f(\mathbf{x}_k^-)) \right)^p$$

where $f(\mathbf{x}) = \sum_{j=1}^n I_j h_j(\mathbf{x})$ and $h_j : X \rightarrow [0,1]$ $j = 1, \dots, n$

loss function

boosting-type

price

Height_of

- minimization is convex!
- use coordinate descent to optimize
- algorithm is pretty simple
- generalizes RankBoost (take $p=1$).

The “P-Norm Push” Algorithm ∞

Input : $\{\mathbf{x}_i^+\}_{i=1,\dots,I}$ positive examples, $\{\mathbf{x}_k^-\}_{k=1,\dots,K}$ negative examples,
 $\{h_j\}_{j=1,\dots,n}$ weak rankers, t_{\max} number of iterations, p power

Initialize : $\mathbf{l}_{1,j} = 0$ for $j = 1, \dots, n$, $d_{1,ik} = 1/IK$ for $i = 1, \dots, I$, $k = 1, \dots, K$,

$$M_{ikj} = h_j(\mathbf{x}_i^+) - h_j(\mathbf{x}_k^-) \text{ for all } i, k, j$$

Loop for $t = 1, \dots, t_{\max}$

$$(a) \quad j_t \in \arg \max_j \left[\sum_{k=1}^K \left[\left(\sum_{i=1}^I d_{t,ik} \right)^{p-1} \sum_{i=1}^I d_{t,ik} M_{ikj} \right] \right]$$

(b) Perform a linesearch for \mathbf{a}_t (details omitted)

(c) $\mathbf{l}_{t+1} = \mathbf{l}_t + \mathbf{a}_t \mathbf{e}_{j_t}$ where \mathbf{e}_{j_t} is 1 in position j_t and 0 elsewhere.

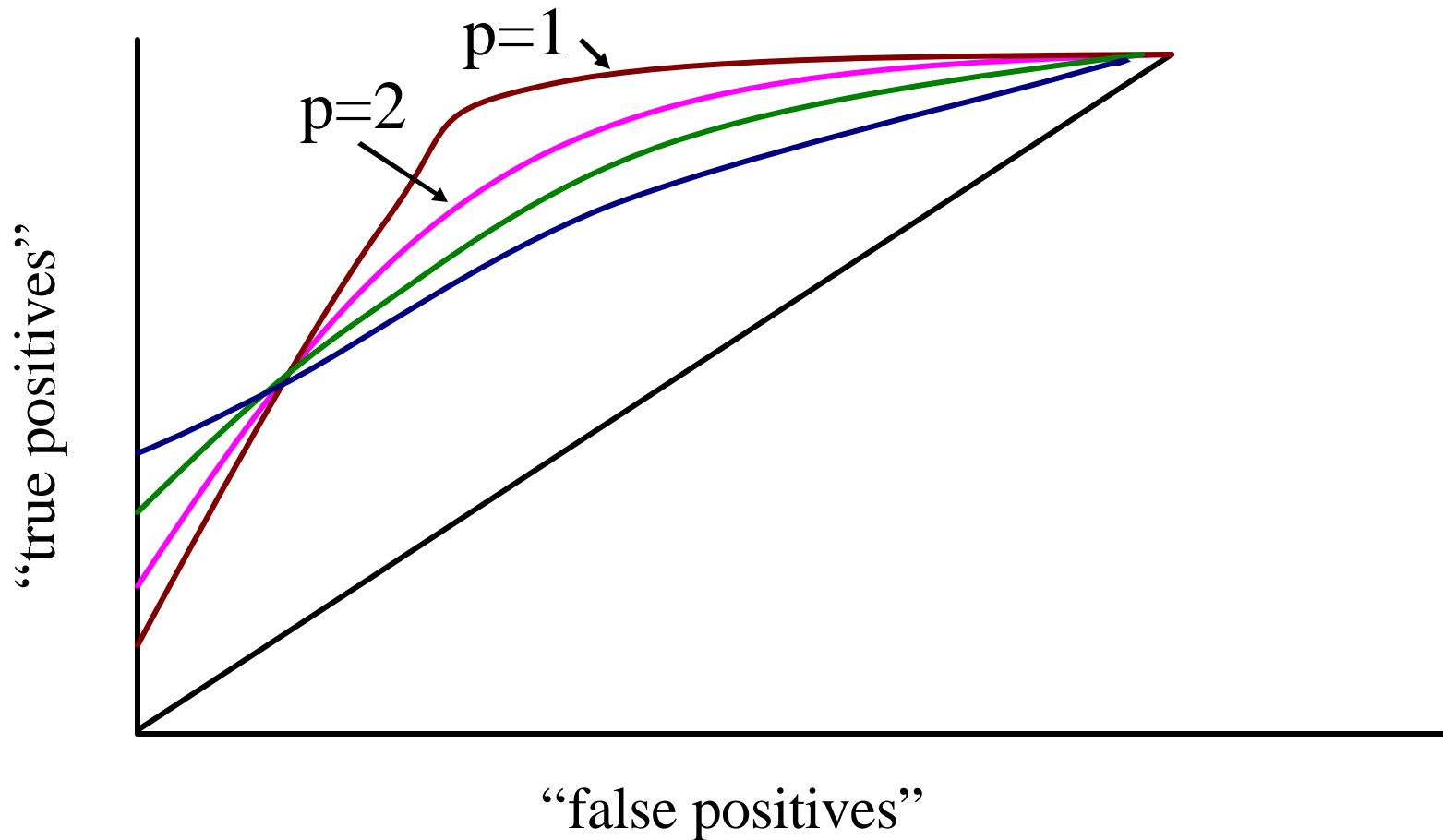
(d) $d_{t+1,ik} = d_{t,ik} \exp[(-\mathbf{M}\mathbf{l}_t)_{ik}] / z_t$ for $i = 1, \dots, I$ $k = 1, \dots, K$

where z_t is a normalization factor.

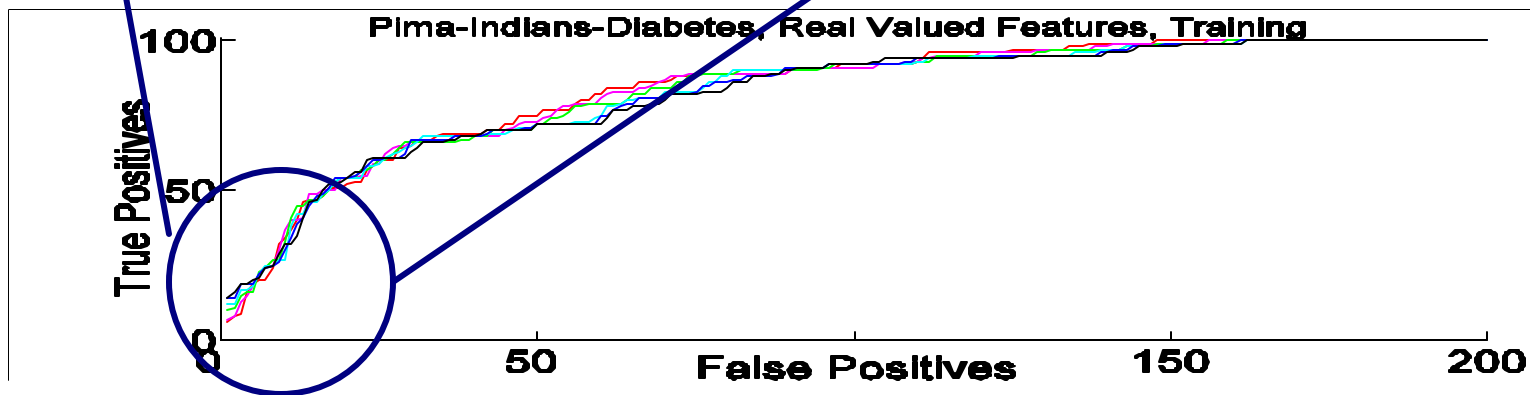
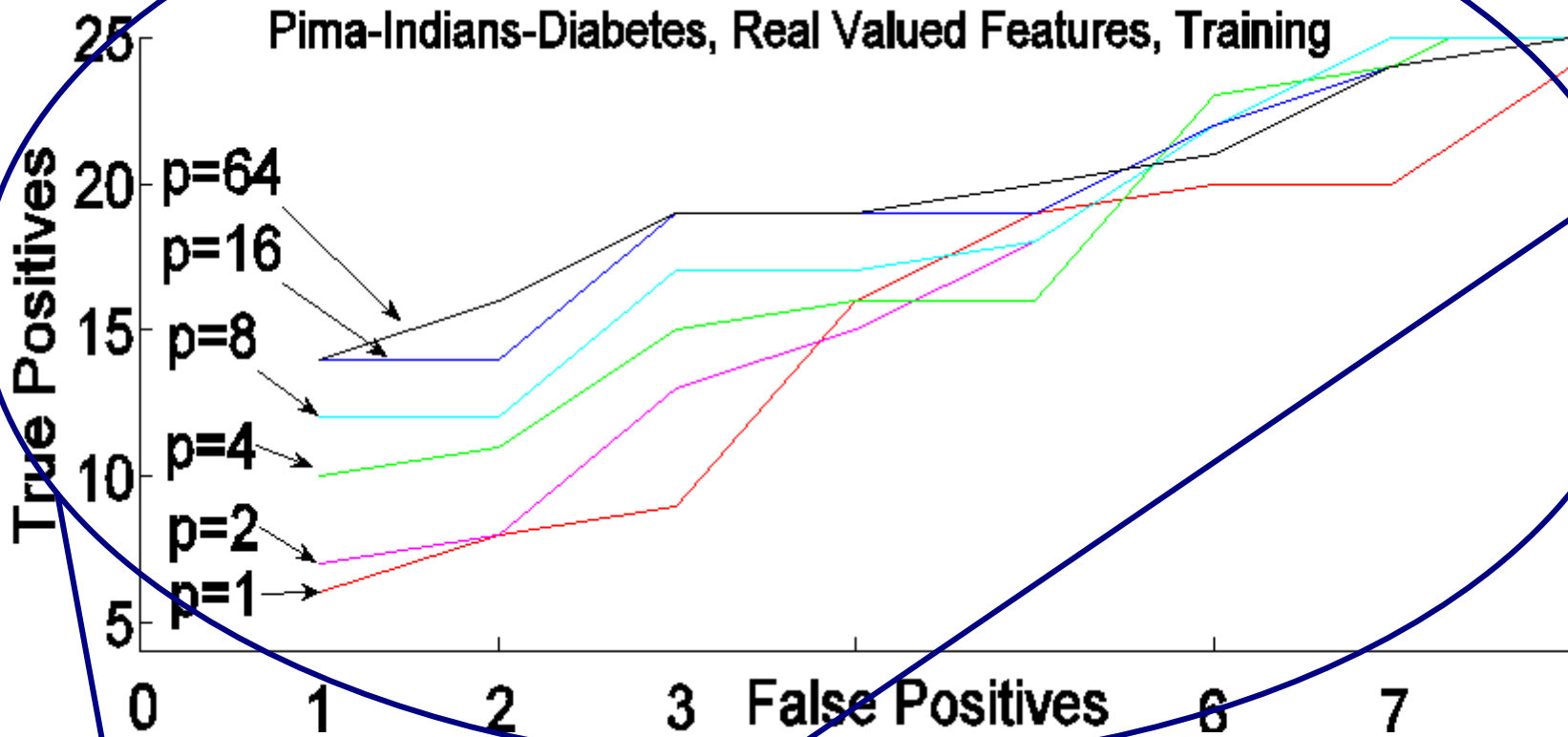
Output : $\mathbf{l}_{t_{\max}}$

The “P-Norm Push” Algorithm ∞

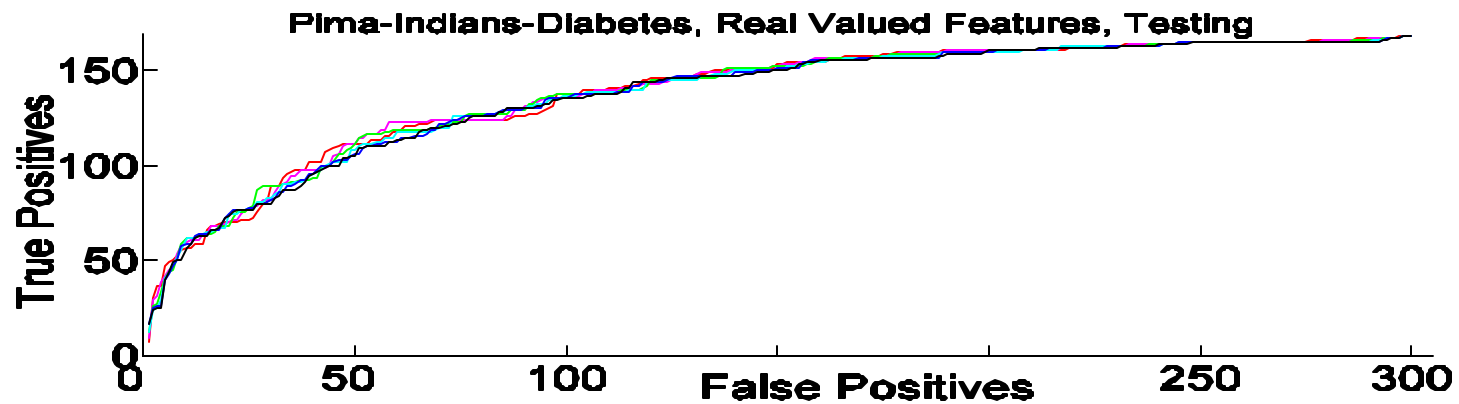
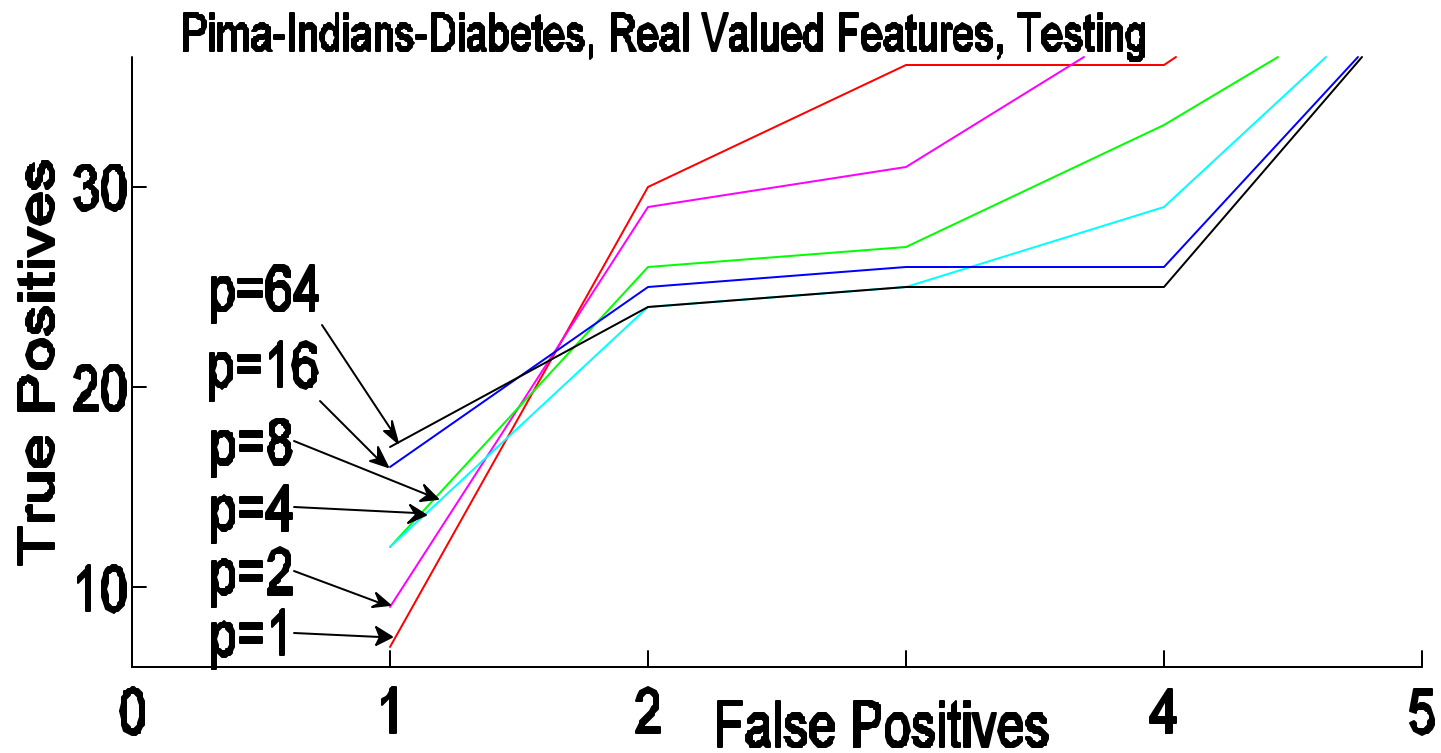
Experimentally, what do we expect?



UCI Data - Pima Indians Diabetes Real - **Training**

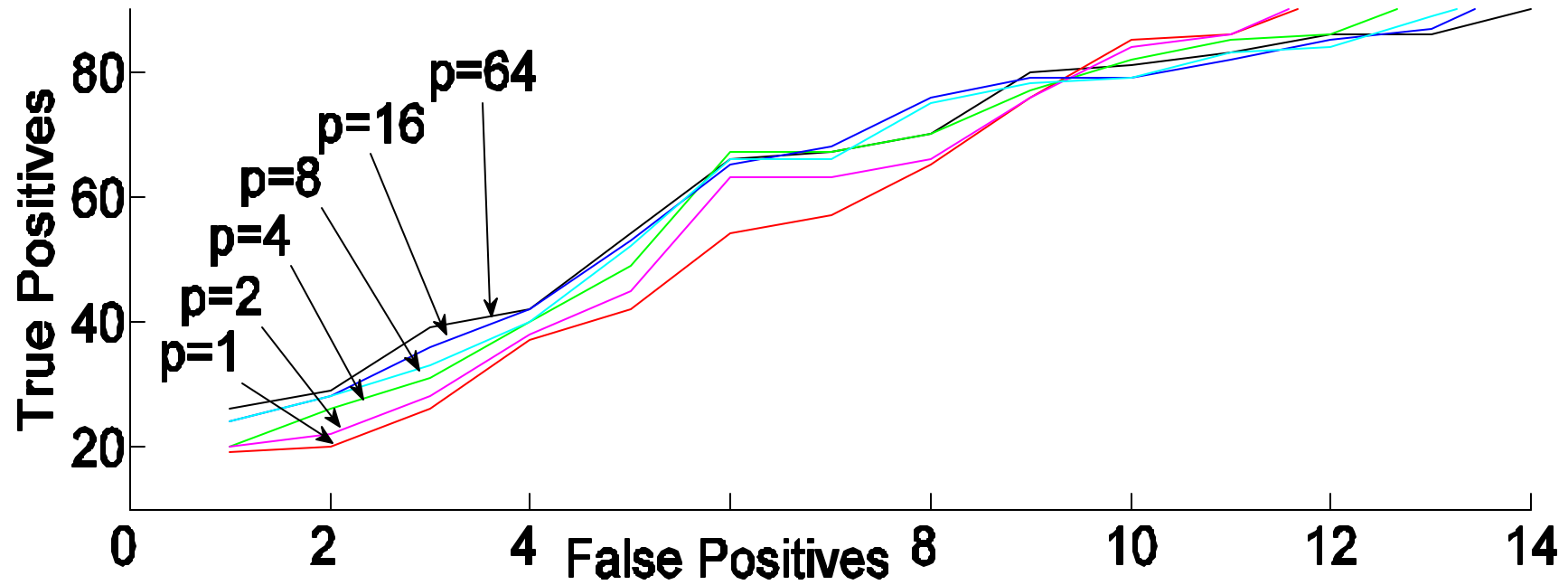


UCI Data - Pima Indians Diabetes Real - **Testing**

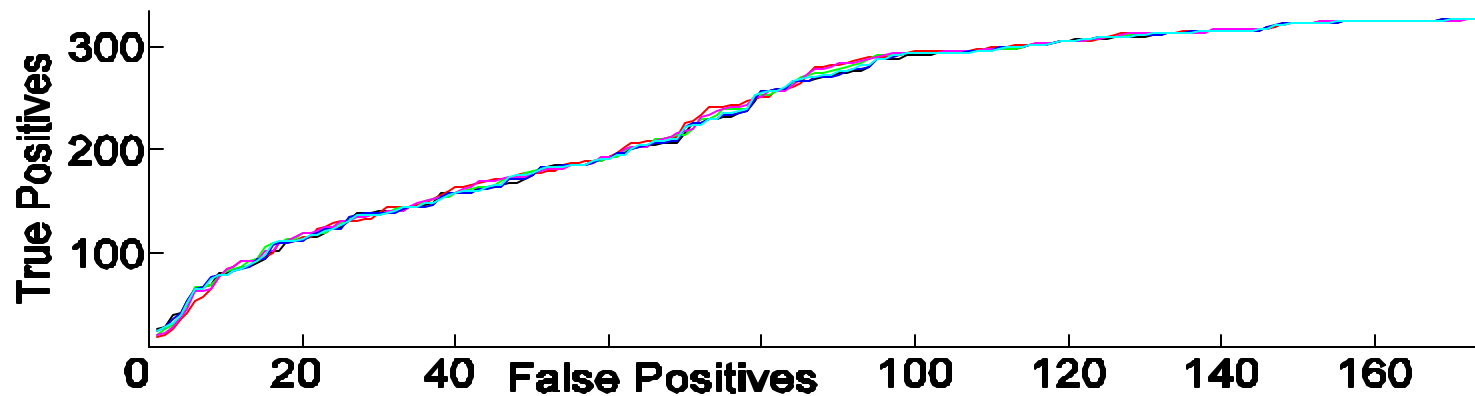


UCI Data - Tic Tac Toe - **Training**

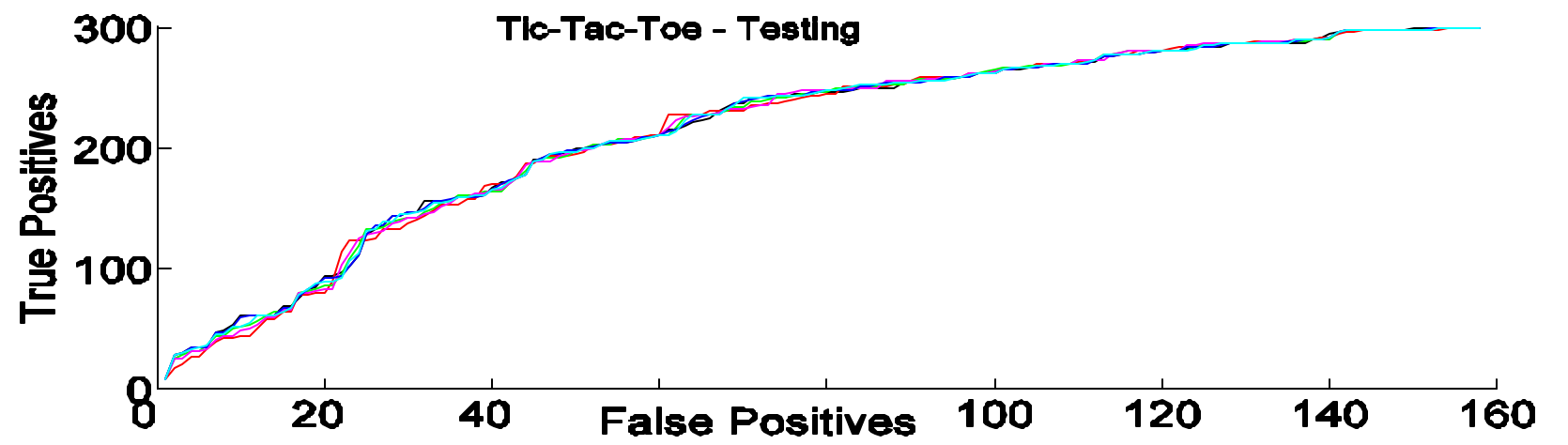
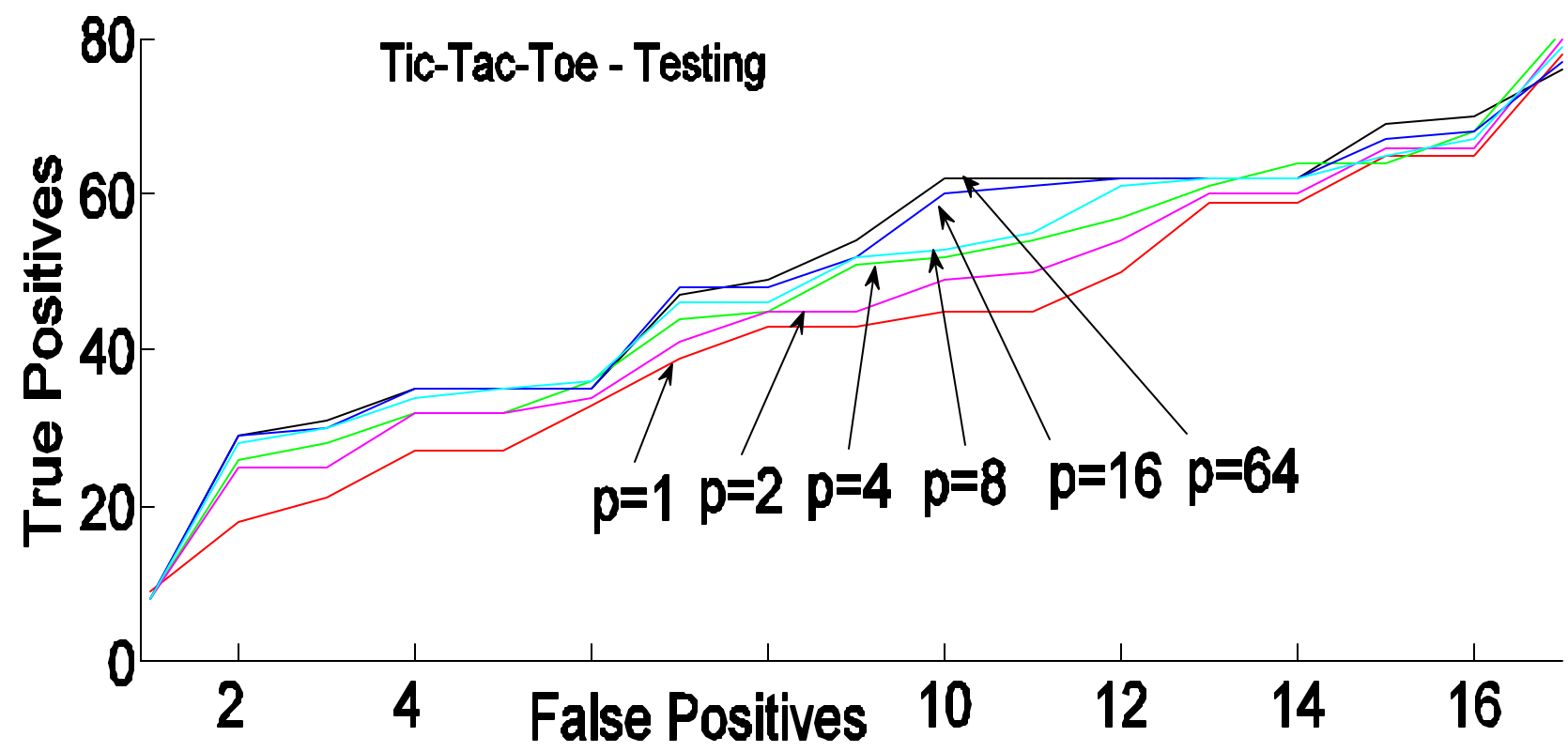
Tic-Tac-Toe - Training



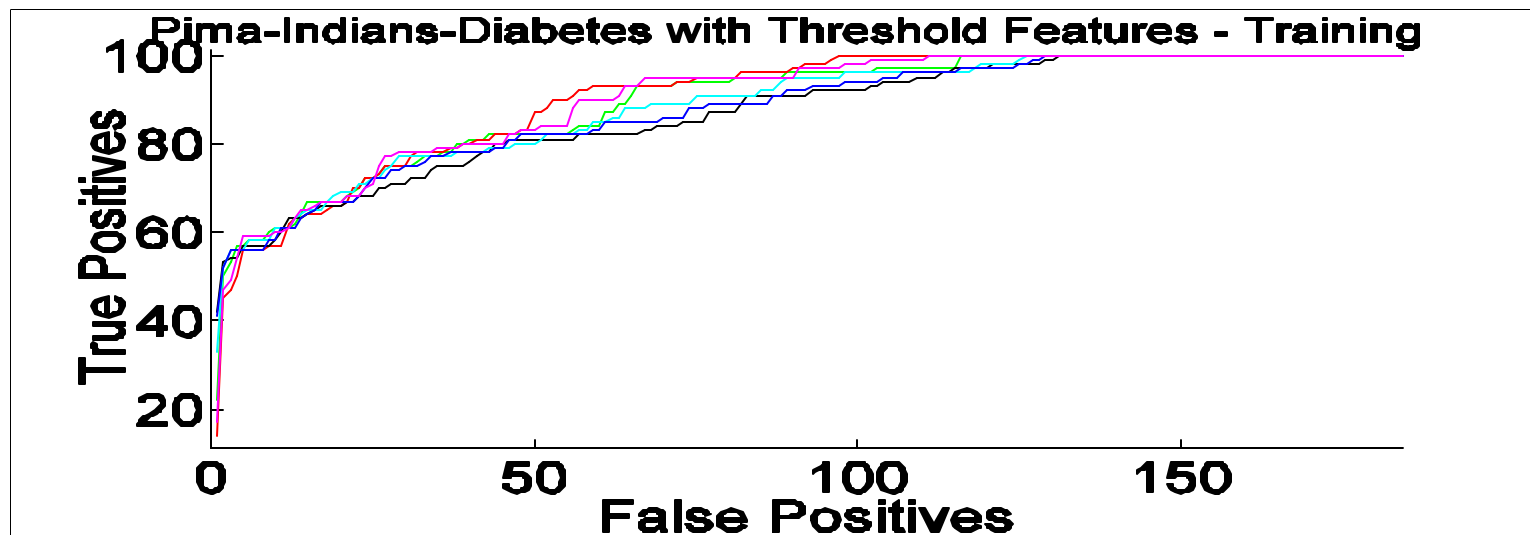
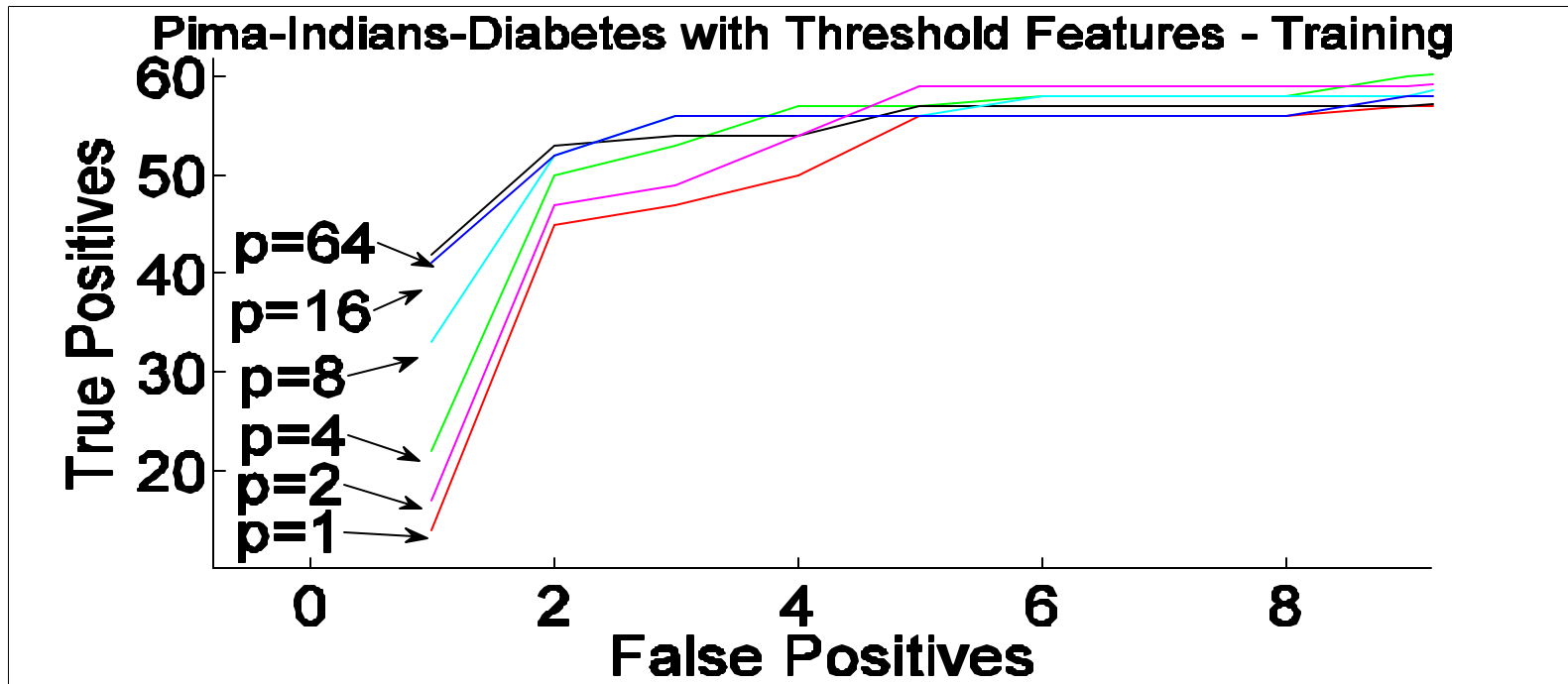
Tic-Tac-Toe - Training



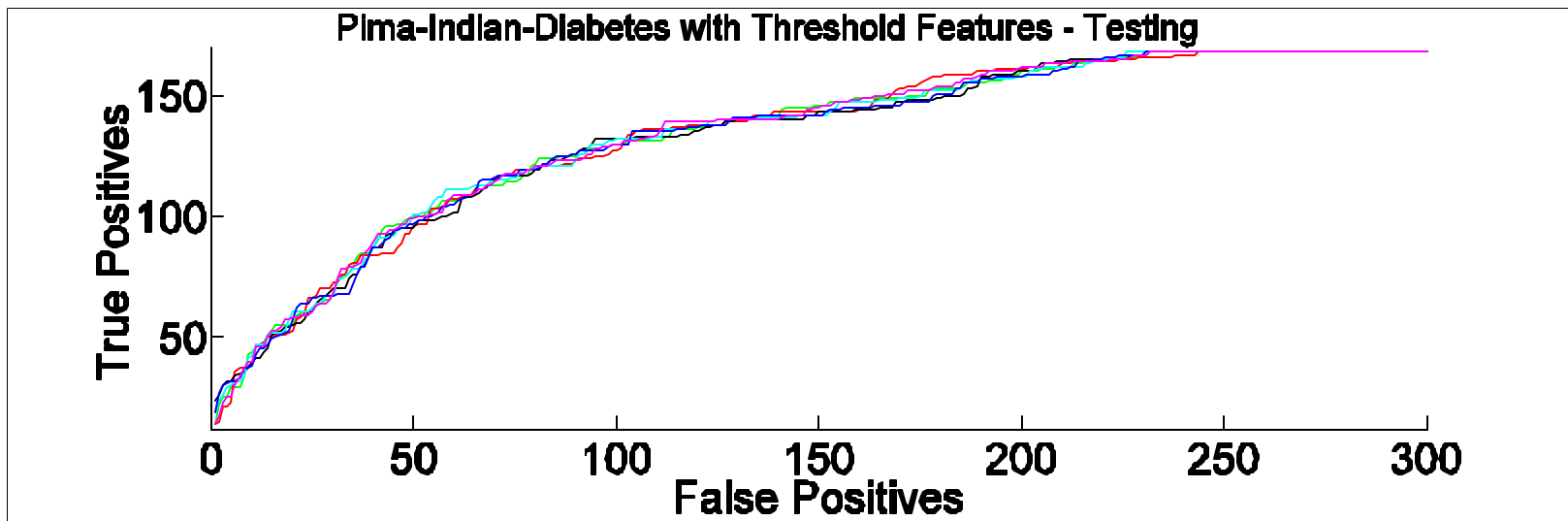
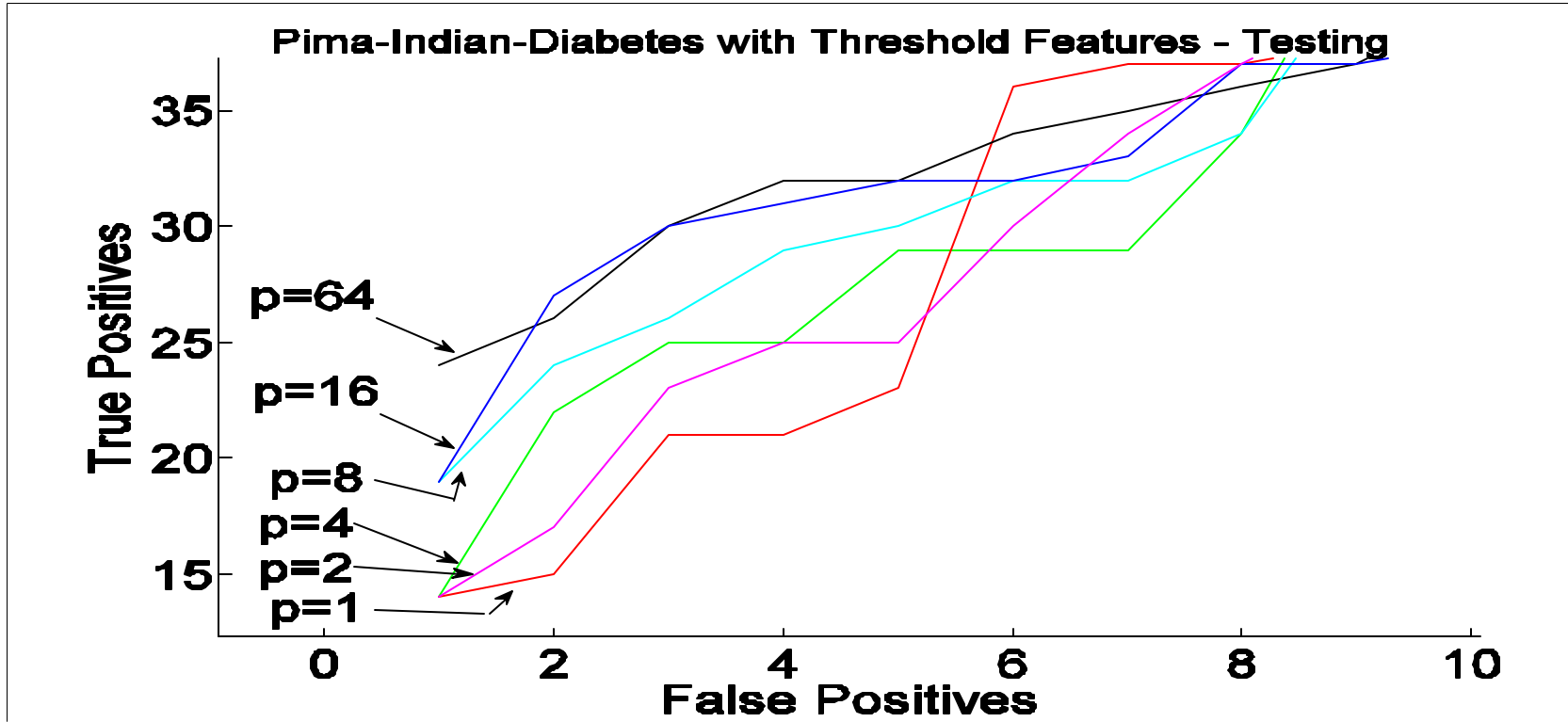
UCI Data - Tic Tac Toe - **Testing**



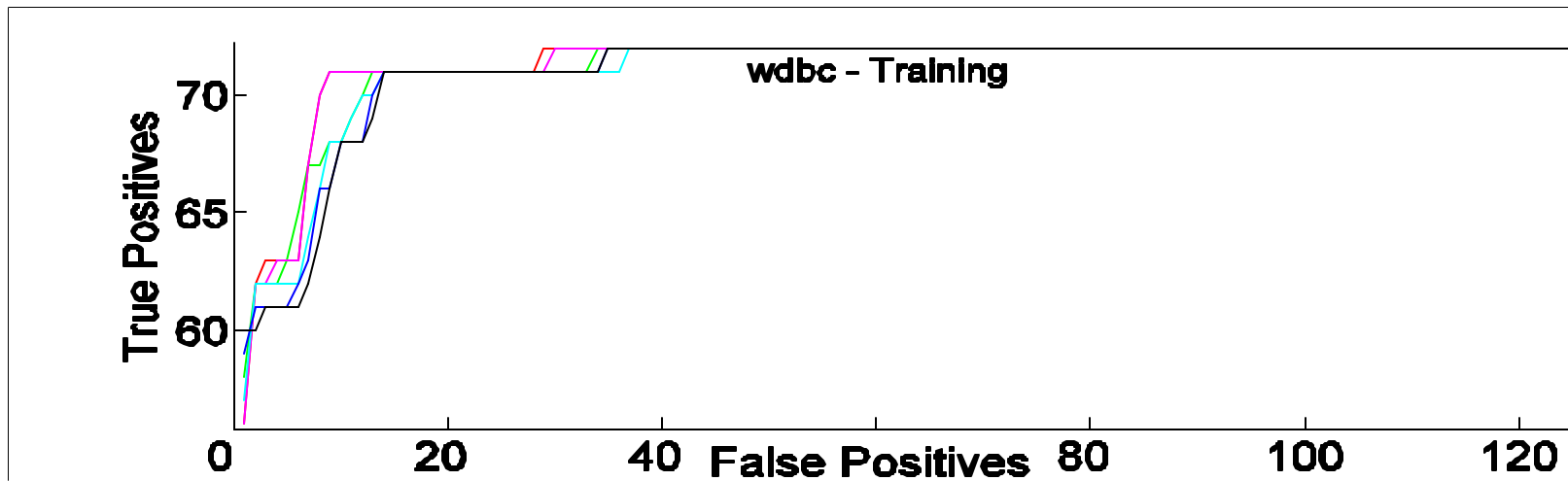
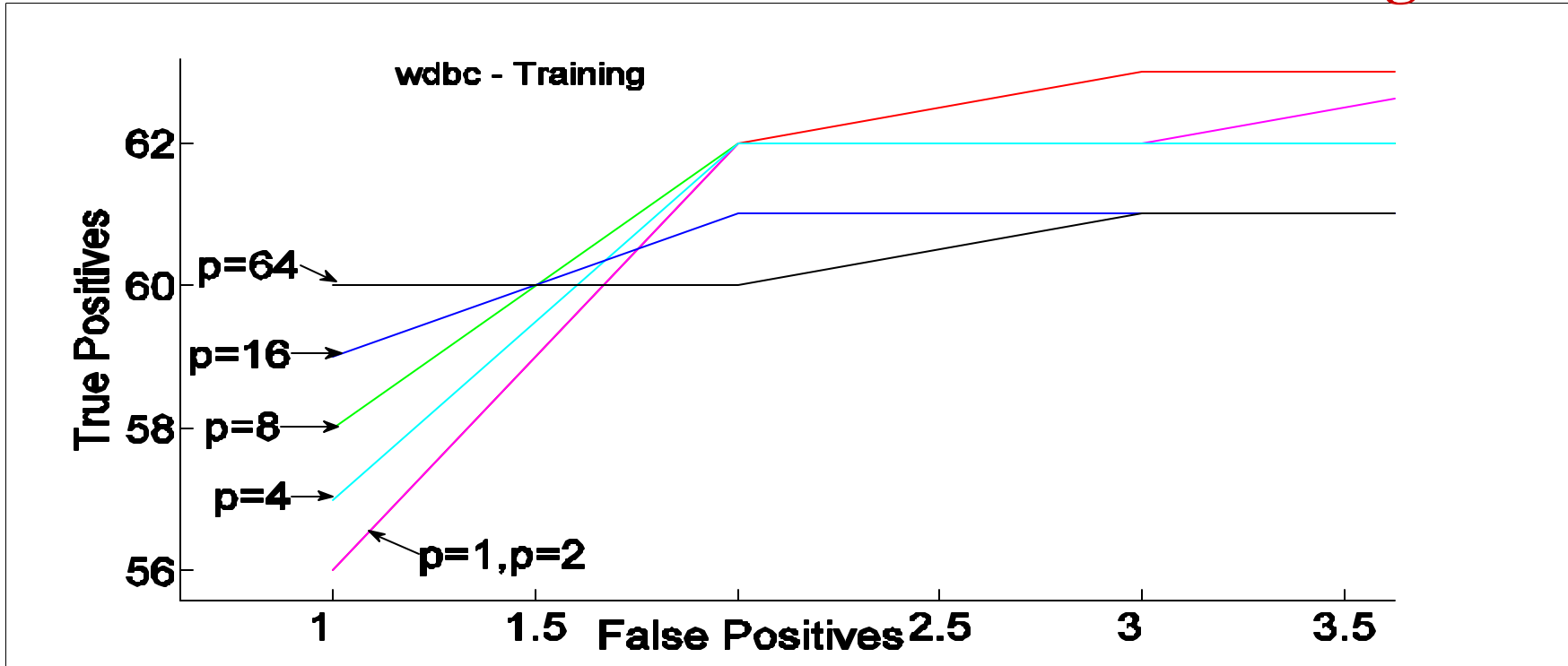
UCI Data – Pima Indians Diabetes Threshold - **Training**



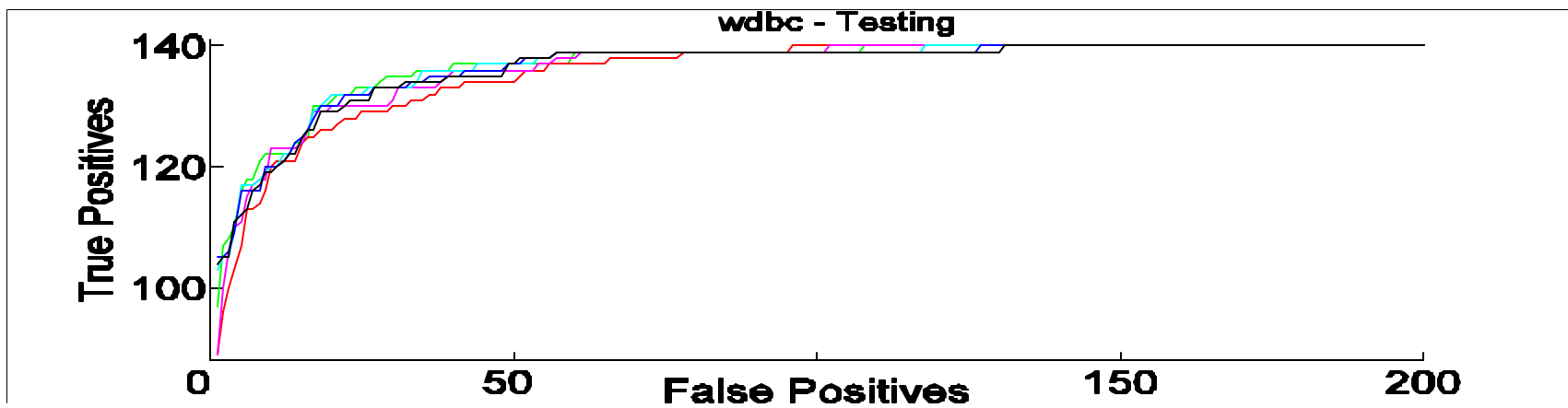
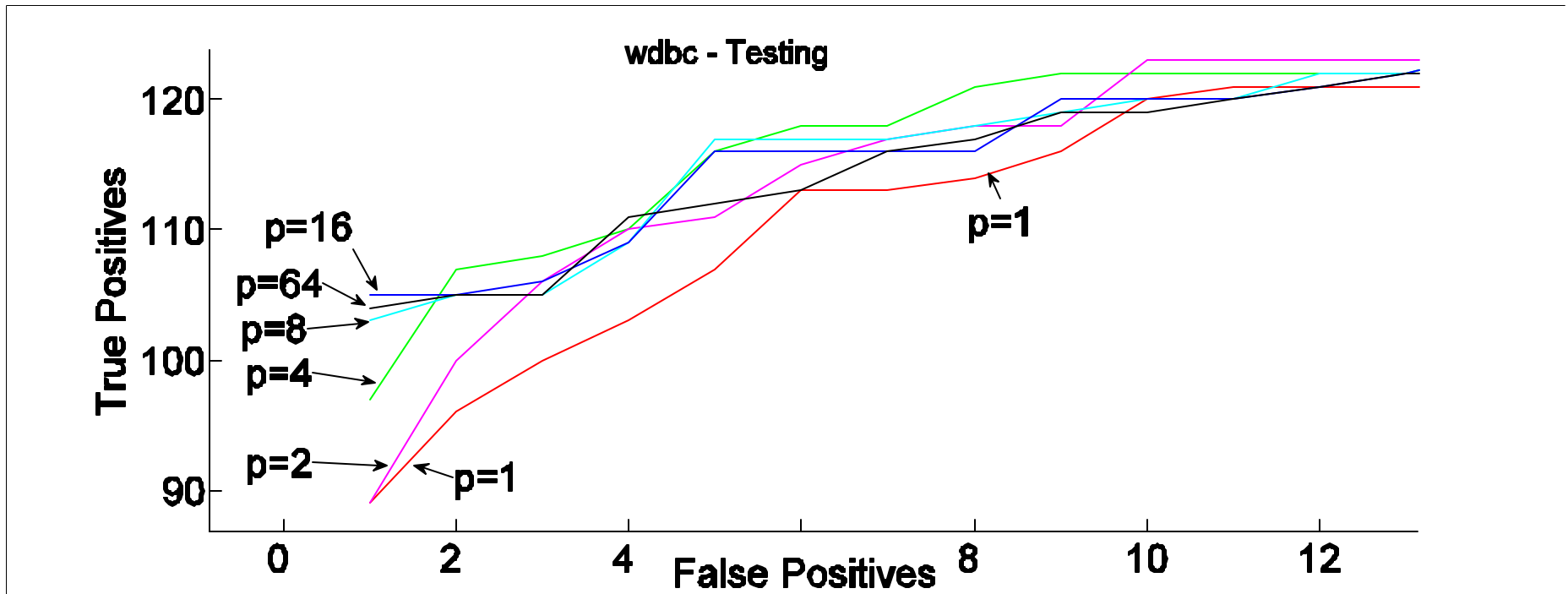
UCI Data - Pima Indians Diabetes Threshold - **Testing**



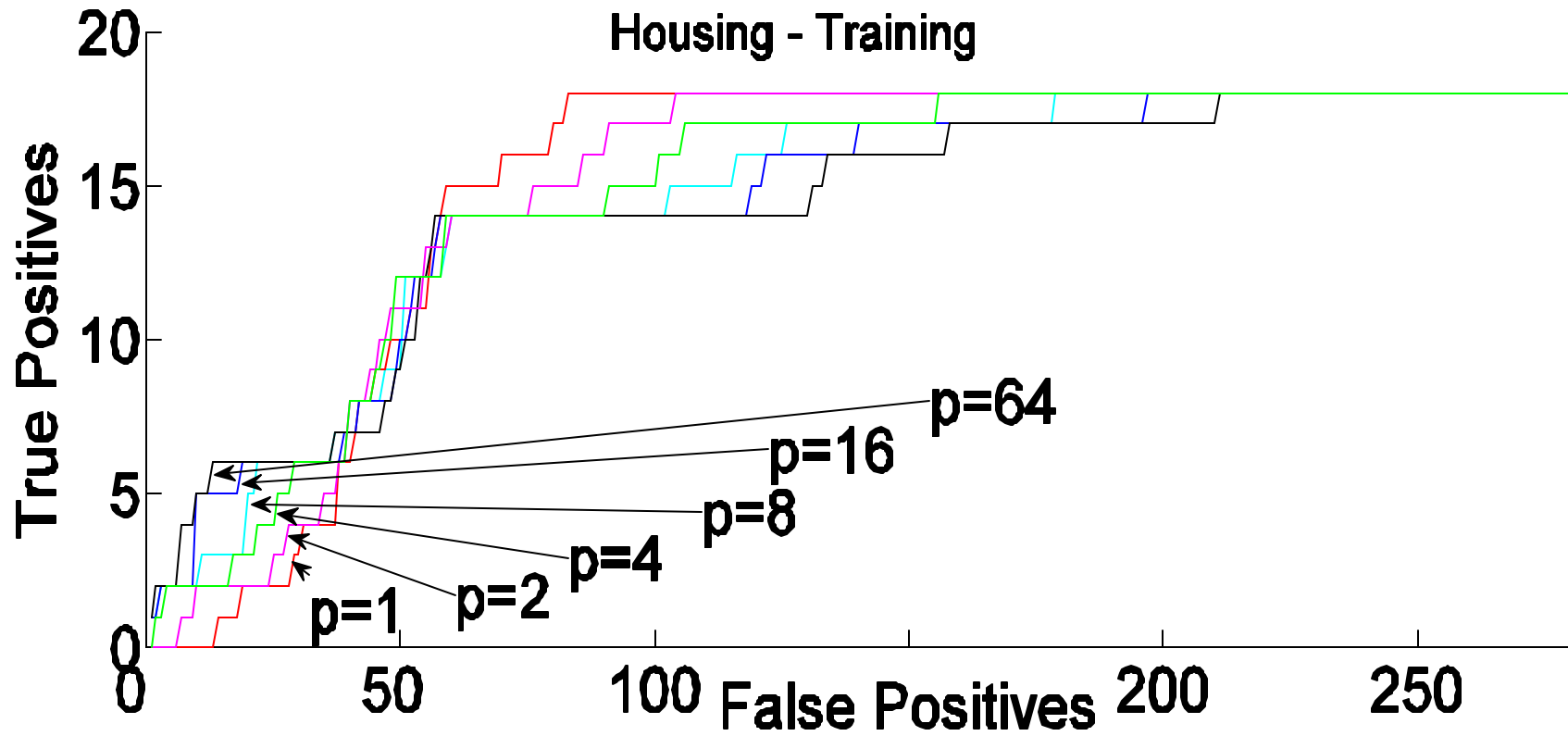
UCI Data -Wisconsin Breast Cancer- **Training**



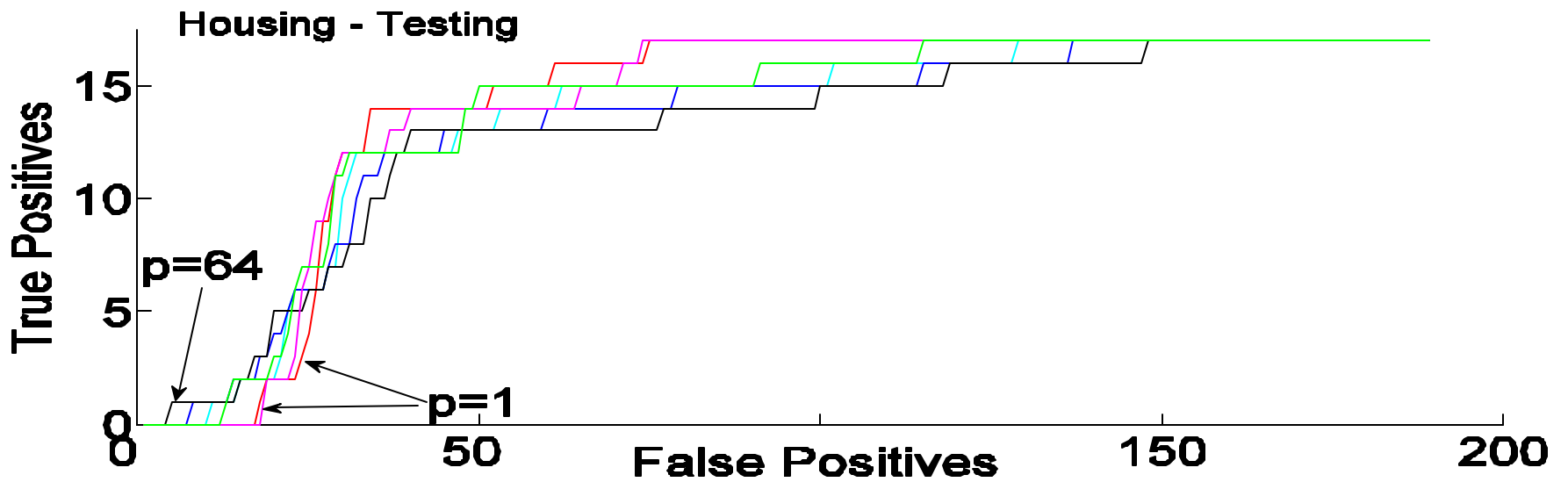
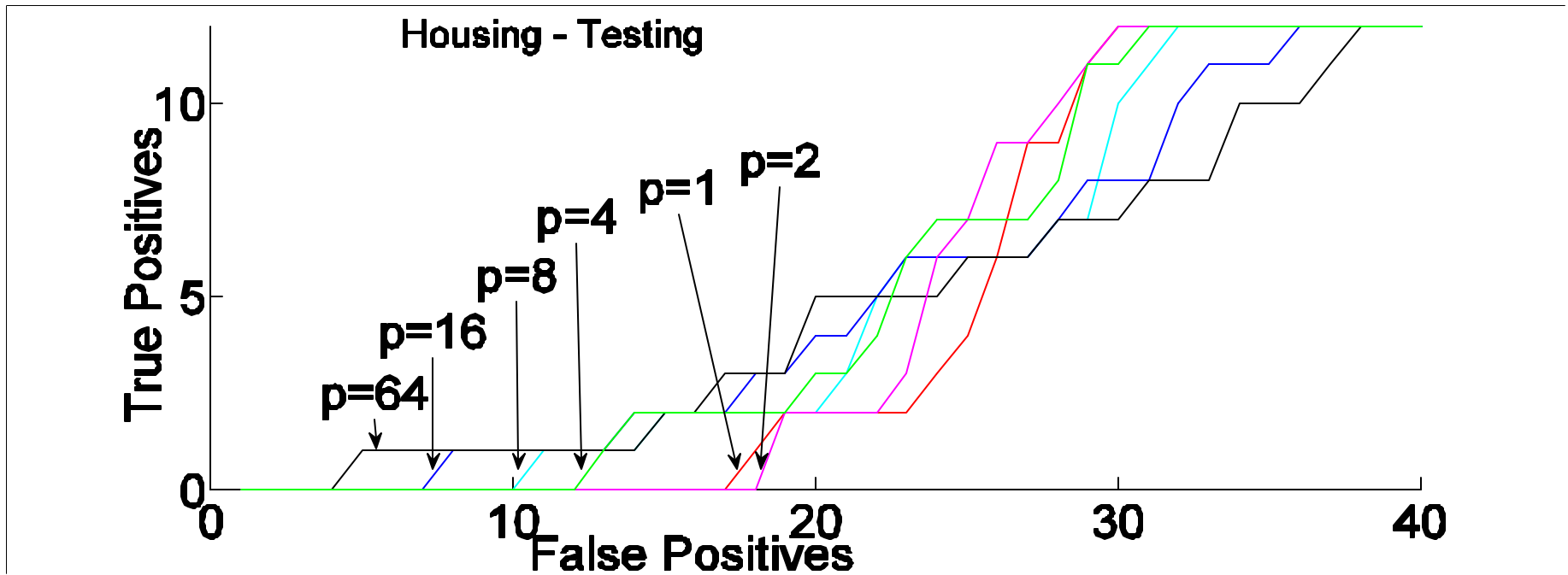
UCI Data -Wisconsin Breast Cancer- **Testing**



UCI Data - Boston Housing - **Training**



UCI Data - Boston Housing - **Testing**



Using P-Norm Ranking Algorithm for NLP

- Joint with Heng Ji and Ralph Grishman (NYU)
- Chinese Name Tagging
- Weighted crucial pairs formulation of P-Norm Objective

Heng Ji, Cynthia Rudin, and Ralph Grishman. Re-ranking Algorithms for Name Tagging. HLT/NAACL, 2006.

Outline of Talk ∞

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

A convex objective exists which concentrates near the top

1) Deriving an Objective Function

Performs well on UCI data. Easy to implement.

2) The “P-Norm Push” Algorithm

Provide a theoretical performance guarantee?

3) A Generalization Bound

4) Uniqueness

Provide uniqueness of the minimizer in some sense?

A Generalization Bound \propto

With high probability:

$$R_{\text{true}} \leq R_{\text{emp},q} + \mathbf{e}$$

Want to minimize this,
but can't measure it

Can try to minimize this

Important factors:

number of positive examples I

number of negative examples K

parameter q

norm power p

some notion of the complexity of the hypothesis space $F: N(F, \mathbf{e})$

A Generalization Bound \propto

The “true” objective function for which our algorithm is designed:

$$\begin{aligned} R_{\text{true}}^p(f) &:= \left(\mathbf{E}_{\mathbf{x}_- \sim D_-} \left(\mathbf{E}_{\mathbf{x}_+ \sim D_+} \mathbf{1}_{[f(\mathbf{x}_+) - f(\mathbf{x}_-) \leq 0]} \right)^p \right)^{1/p} \\ &= \left\| \mathbf{P}_{\mathbf{x}_+ \sim D_+} (f(\mathbf{x}_+) - f(\mathbf{x}_-) \leq 0 \mid \mathbf{x}_-) \right\|_{L_p(X_-, D_-)} \end{aligned}$$

The empirical loss associated with $R_{\text{true}}^p(f)$ is:

$$R_{\text{emp}}^p(f) := \left(\frac{1}{K} \sum_{k=1}^K \left(\frac{1}{I} \sum_{i=1}^I \mathbf{1}_{[f(\mathbf{x}_i^+) - f(\mathbf{x}_k^-) \leq 0]} \right)^p \right)^{1/p}$$

like $R_{g,1}$

A more general notion, incorporating a “margin”:

$$R_{\text{emp},q}^p(f) := \left(\frac{1}{K} \sum_{k=1}^K \left(\frac{1}{I} \sum_{i=1}^I \mathbf{1}_{[f(\mathbf{x}_i^+) - f(\mathbf{x}_k^-) \leq q]} \right)^p \right)^{1/p}$$

A Generalization Bound \propto

Theorem 2 (Generalization Bound): For all $q > 0, \mathbf{e} > 0, f \in F$

$$P_{S_+ \sim D_+^I, S_- \sim D_-^K} [R_{\text{true}} \leq R_{\text{emp}, q} + \mathbf{e}] \\ \geq 1 - 2N \left(F, \frac{\mathbf{e}q}{8} \right) \left[\exp \left[-2 \left(\frac{\mathbf{e}}{4} \right)^{2p} K \right] + \exp \left[-\frac{\mathbf{e}^2}{8} I \right] \right]$$

Proof based on Rudin et al 05 and Rudin & Schapire 05, which was inspired by Koltchinskii and Panchenko 02, Cucker and Smale 00, Bousquet 02.

The bound says that generalization is possible!

Now we have a theoretical guarantee on performance!

Note: these kinds of bounds are usually proved with a symmetrization step, which does not work in our case. See (Agarwal et al. 04, Clemencon et al. 05, Usunier et al. 05)

The covering number approach fixes this. (Cucker and Smale 02, Rudin et al 05)

Outline of Talk ∞

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

A convex objective exists which concentrates near the top

1) Deriving an Objective Function

Performs well on UCI data. Easy to implement.

2) The “P-Norm Push” Algorithm

Provide a theoretical performance guarantee?

3) A Generalization Bound

4) Uniqueness

Provide uniqueness of the minimizer in some sense?

Uniqueness ∞

- Want to show that the minimizer of $F_p(\mathbf{I})$ is somehow unique.

$$\min_{\mathbf{I} \in \mathfrak{R}^n} F_p(\mathbf{I}) := \sum_{k=1}^K \left(\sum_{i=1}^I \exp(-f(\mathbf{x}_i^+) + f(\mathbf{x}_k^-)) \right)^p$$

$$\text{where } f(\mathbf{x}) = \sum_{j=1}^n \mathbf{I}_j h_j(\mathbf{x})$$

- Why is this tricky?
 - It's not unique! At least with respect to \mathbf{I} (with no assumptions allowed on the h_j 's.)
- How can I keep things finite and get uniqueness?
 - use $\left\{ \exp(-f(\mathbf{x}_i^+) + f(\mathbf{x}_k^-)) \right\}_{ik} \in \mathfrak{R}^{\text{IK}}$

Theorem 3 (Uniqueness)

Define

$$Q' = \left\{ \mathbf{q}' \in \mathfrak{R}_+^{IK} \left| \begin{array}{l} q'_{ik} = \exp(-f(\mathbf{x}_i^+) + f(\mathbf{x}_k^-)) \text{ for some } \mathbf{l}, \\ \text{where } f(\mathbf{x}) = \sum_k \mathbf{l}_j h_j(\mathbf{x}) \end{array} \right. \right\}$$

Then,

$$\mathbf{q}^* = \operatorname{argmin}_{\mathbf{q}' \in \operatorname{closure}(Q')} \sum_k \left(\sum_i q'_{ik} \right)^p$$

and this determines \mathbf{q}^* uniquely.

Proof based on convex duality for a class of Bregman distances
(Della Pietra, Della Pietra, Lafferty 2002), (Collins, Schapire, Singer 2002)

Proof relies on finding a “magic function” to define the Bregman distance.

$$\mathbf{j}(\mathbf{q}) := -\sum_{ik} q_{ik} g(q_{ik}, \mathbf{q}), \text{ where } g(q_{ik}, \mathbf{q}) := \ln \frac{q_{ik}}{p^{1/p} \left(\sum_{i'} q_{i'k} \right)^{(p-1)/p}}$$

(If $p=1$, this is the relative entropy between \mathbf{q} and $\mathbf{1}$. Phew!)

Outline of Talk ∞

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

A convex objective exists which concentrates near the top

1) Deriving an Objective Function

Performs well on UCI data. Easy to implement.

2) The “P-Norm Push” Algorithm

3) A Generalization Bound

Provide a theoretical performance guarantee?

4) Uniqueness

Provide uniqueness of the minimizer in some sense?

5) AdaBoost is a ranking algorithm

But Cynthia, AdaBoost is a **classification** algorithm...
Does it really do **ranking**?

Yes! And it's as good as RankBoost!
(non-separable case)

Cortes & Mohri, and Caruana & Niculescu-Mizil

AdaBoost is good for ranking (in addition to RankBoost).
(It tends to achieve a high AUC value.)

Consider this movie ranking rule:

$h_c =$ Movies that... are movies! $h_c=1$ for every movie.

Boogeyman, Spiderman, The Matrix, Titanic, Amelie

... everything satisfies this rule.

If we add it into the set of weak classifiers
(a very innocent assumption) and then run AdaBoost...

AdaBoost and RankBoost produce equally good solutions!

AdaBoost and Ranking ∞

- Define F-skew: it measures the imbalance of the loss between positive and negative examples.

$$F\text{-skew}(\mathbf{I}) := \sum_{i \in \text{positives}} e^{-(\mathbf{M}\mathbf{I})_i} - \sum_{k \in \text{negatives}} e^{-(\mathbf{M}\mathbf{I})_k}$$

AdaBoost and RankBoost Theorem (R, Cortes, Mohri, Schapire, 05)

Whenever the F-skew vanishes, AdaBoost converges to the minimum of RankBoost's objective function. (Non-separable case)

Corollary: *The F-skew vanishes whenever the constant hypothesis is included in the set of weak classifiers. So...*

Theorem (R, Cortes, Mohri, Schapire, 05)

Whenever the F -skew vanishes, AdaBoost and RankBoost converge to equally good AUC values. (nonseparable case.)

This is truly bizarre, but very good.

Thank you

Thanks to Rob Schapire, Eero Simoncelli, and Sinan Güntürk

Related Prior Work:

Dynamics of Boosting

(Rudin, Daubechies, Schapire: NIPS 04, JMLR 04)

Boosting Based on A Smooth Margin

(Rudin, Schapire, Daubechies: CoLT 04)

Margin-Based Ranking and Boosting Meet in the Middle

(Rudin, Cortes, Mohri, Schapire CoLT 05, Rudin and Schapire 05)

www.cns.nyu.edu/~rudin