On generalization bounds, projection profile, and margin distribution

Ashutosh Garg
Sariel Har-Peled
Dan Roth

University of Illinois, Urbana-Champaign
danr@cs.uiuc.edu
http://L2R.cs.uiuc.edu/~danr

Learning with high dimensional data

- Identifying phrase structure

  [$_{NP}$ He] [$_{VP}$ reckons] [$_{NP}$ the current account deficit] [$_{VP}$ will narrow] [$_{PP}$ to] [$_{NP}$ only # 1.8 billion] [$_{PP}$ in] [$_{NP}$ September]

- Information Extraction Tasks

  afternoon, Dr. Ab C will talk in Ms. De. F class..

- Prepositional Phrase Attachment

  buy shirt with sleeves, buy shirt with a credit card

- Context Sensitive Spelling Correction

  Illinois' bored of education board

Learning with high dimensional data

[$_{NP}$ He] [$_{VP}$ reckons] [$_{NP}$ the current account deficit] [$_{VP}$ will narrow] [$_{PP}$ to] [$_{NP}$ only # 1.8 billion] [$_{PP}$ in] [$_{NP}$ September]

afternoon, Dr. Ab C will talk in Ms. De. F class..

buy shirt with sleeves, buy shirt with a credit card

Illinois' bored of education board

Features include: (patterns of) words; POS tags; relational information (location; order;structure…)

In many of these problems dimensionality is 10^5 or more

Easiness of Learning

We learn well from relatively small number of examples in very high dimensional spaces? Should we believe it?

Some high dimensional problems are naturally constrained and become, effectively, low dimensional problems.

[Roth, Zelenko'00; Garg, Roth'01, Vempala'00]

In these cases, although learning is done in high dimension, generalization ought to depend on the true, lower dimensionality of the problem.

Not exploited by current theories

This work

Introduces a way to analyze learning in high dimension in a way that exploits the lower, effective dimensionality of the data.

Random projection methods are used to explicitly exploit the margin distribution

Exhibits generalization bounds the are (sometimes) realistic (< 0.5) for real problems in NLP and vision

Standard Bounds

VC dimension based bounds (hyperplanes)

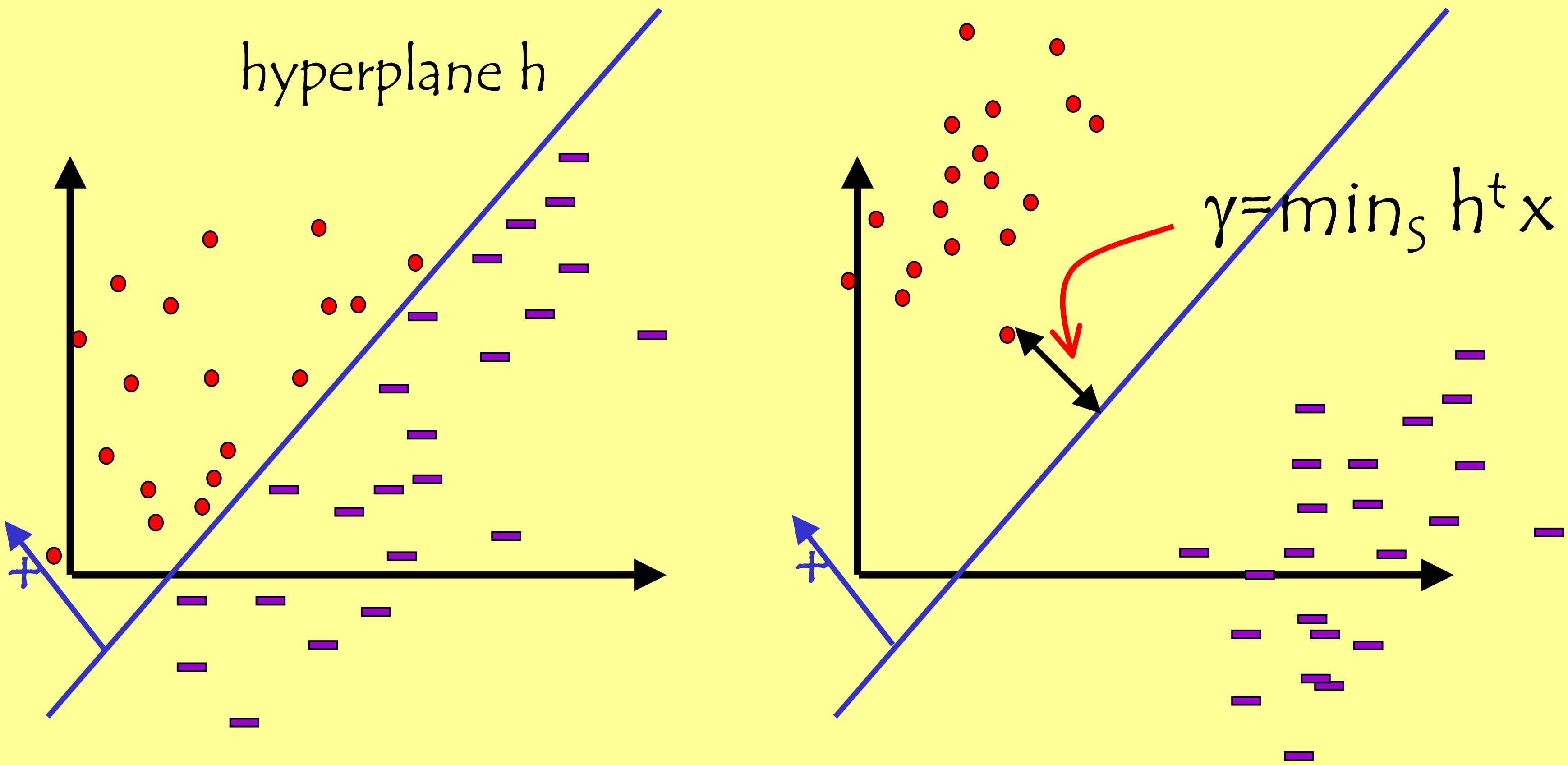
VC(n,m)

$$ERR_D \le ERR_S + \sqrt{[\underline{\boldsymbol{n}}(\ln(2\boldsymbol{m}/\boldsymbol{n})+1)-\ln(\delta/4)]/\underline{\boldsymbol{m}}}$$

Margin Based bounds (data dependent; γ – margin)

$$ERR_D \le ERR_S + (2/\boldsymbol{m})\left(\underline{(1/\gamma^2)}\log(32\boldsymbol{m})\log(8\boldsymbol{e}\boldsymbol{m}\gamma^2)+\log(8\boldsymbol{m}/\delta)\right)$$

Intuition



Hard Problem

Easy Problem

Standard Bounds

VC dimension based bounds (hyperplanes)

VC(n,m)

$$\boldsymbol{ERR}_D \le \boldsymbol{ERR}_S + \sqrt{[\boldsymbol{n}(\ln(2\boldsymbol{m}/\boldsymbol{n})+1) - \ln(\delta/4)]/\boldsymbol{m}}$$

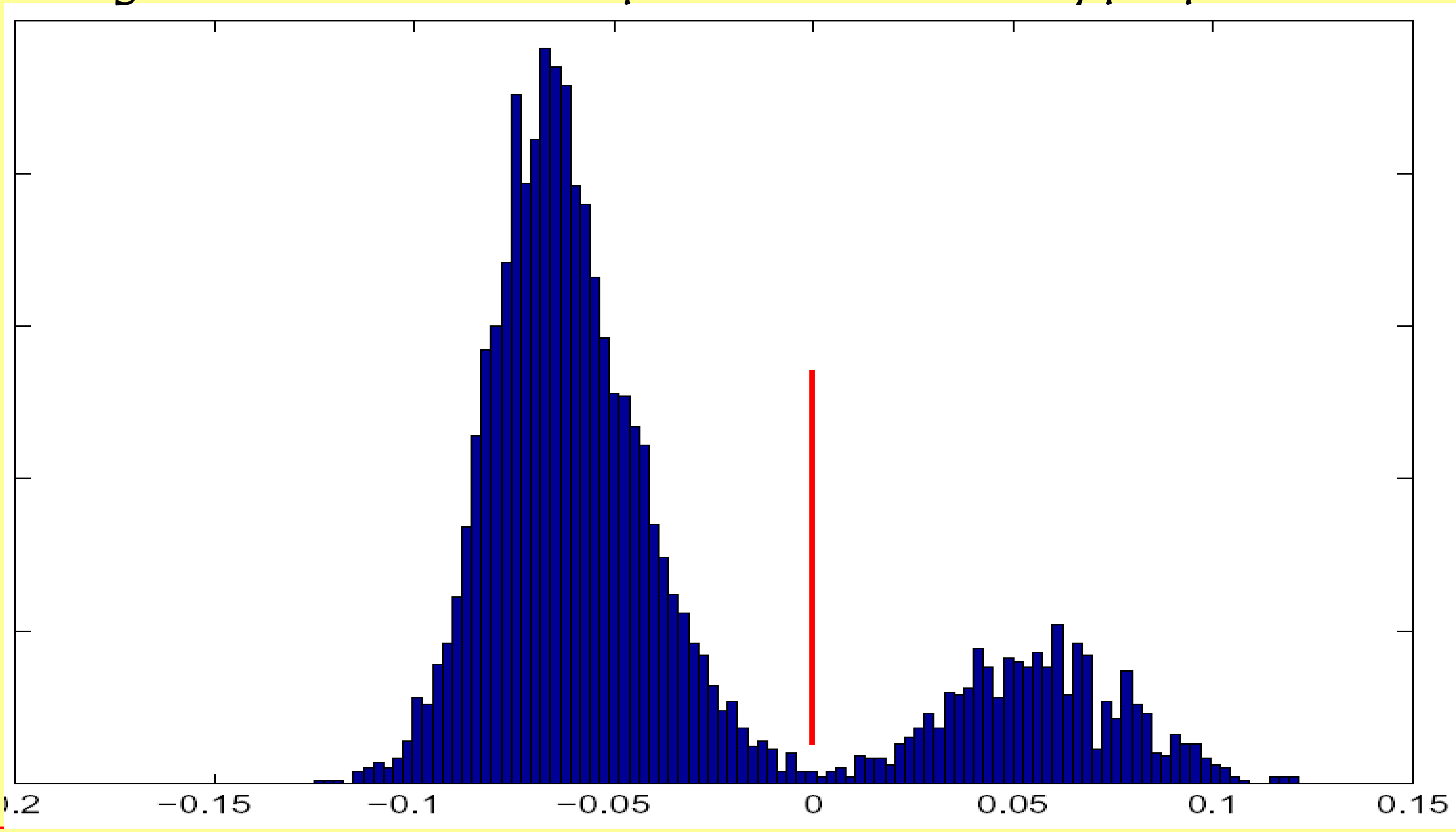
Margin Based bounds (data dependent; γ – margin)

$$\boldsymbol{ERR}_D \le \boldsymbol{ERR}_S + (2/\boldsymbol{m})\Big((1/\gamma^2)\log(32\boldsymbol{m})\log(8\boldsymbol{e}\boldsymbol{m}\gamma^2) + \log(8\boldsymbol{m}/\delta)\Big)$$

Typically: 1 **<<** VC bounds **<** Margin Based bound

Real Data

17,000 dimensional context sensitive spelling
Histogram of distance of points from the hyperplane



Standard Bounds

VC dimension based bounds (hyperplanes)

VC(n,m)

$$\boldsymbol{ERR_D} \leq \boldsymbol{ERR_S} + \sqrt{[\underline{\boldsymbol{n}}(\ln(2\boldsymbol{m}/\boldsymbol{n})+1)-\ln(\delta/4)]/\underline{\boldsymbol{m}}}$$

Margin Based bounds (data dependent; γ – margin)

$$\boldsymbol{ERR_D} \leq \boldsymbol{ERR_S} + (2/\boldsymbol{m})\left(\underline{(1/\gamma^2)}\log(32\boldsymbol{m})\log(8\boldsymbol{em}\gamma^2)+\log(8\boldsymbol{m}/\delta)\right)$$

Typically: 1 **<<** VC bounds **<** Margin Based bound

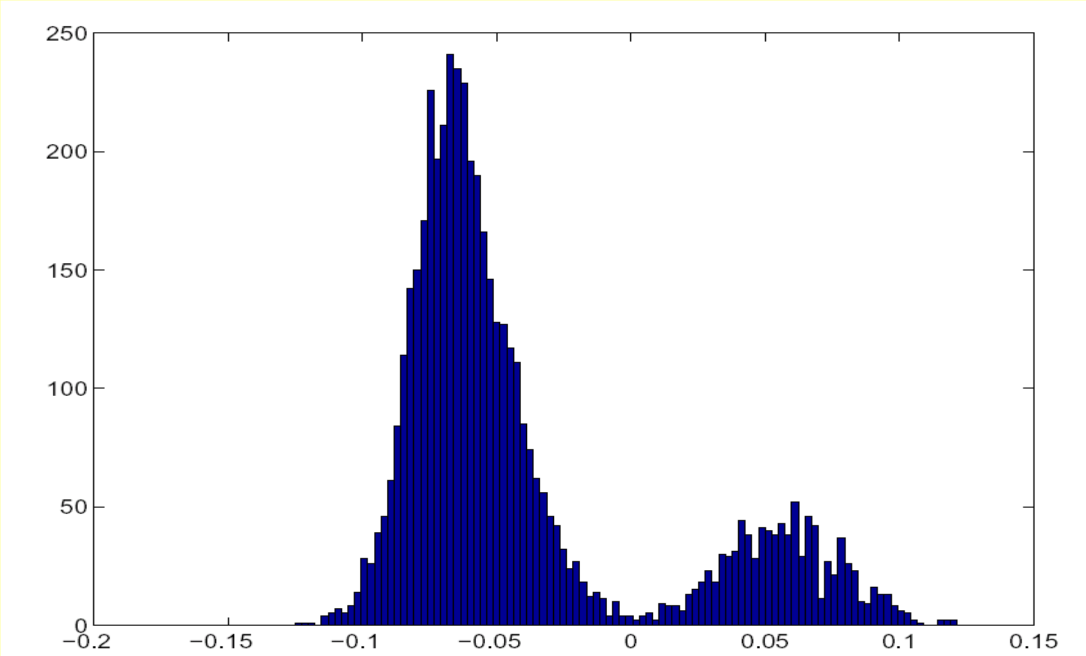
Value of bounds: algorithmic insight; model selection

This work

Even for:

17,000 dimensional

context sensitive spelling



Can get bounds

that are < 0.5,

using a 1000-5000 examples.

Key Idea: Projection Profile (I)

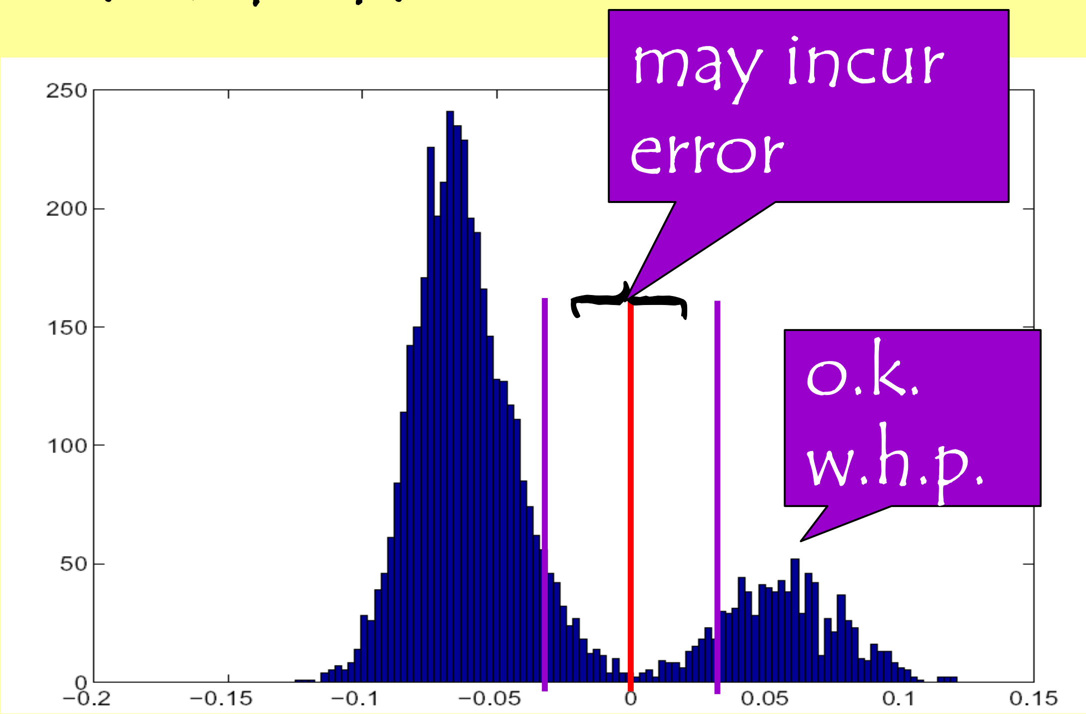
Learn a Hyperplane h from sample S, in high dimension n

Analysis: Project S and h randomly to low dimension (k)

w.h.p (k,S): small distortion of distances.

(Johnson-Lindenstraus)

⇒ Small error in the lower dimension



Key Idea: Projection Profile (II)

Expected amount of error introduced in projection captured by: $a_k(D,h) = \int_{x \in D} u(x) dD$

where: $u_k(x) = \min\left\{ \exp\left(-\frac{k}{|l(x)|} \right), \frac{1}{kl^2(x)}, 1 \right\}$ $l(x) = h^t x$

the profile:

$$P(D,h) = \left(a_1(D,h), a_2(D,h), \ldots a_k(D,h), \ldots \right)$$

Key Idea: Projection Profile (II)

Expected amount of error introduced in projection captured by: $a_k(D,h) = \int_{x \in D} u(x)\,dD$

where: $u_k(x) = \min\left\{ \exp\left(-\frac{k}{|l(x)|} \right), \frac{1}{kl^2(x)}, 1 \right\}$ $\quad l(x) = h^t x$

the profile:

$$P(D,h) = \left(a_1(D,h), a_2(D,h), \ldots a_k(D,h), \ldots \right)$$

gives the tradeoff between dimensionality and accuracy

Resulting bound:

$$ERR_D \le ERR_S + \min_k \left\{ \underline{\hat{u}_k} + \underline{VC(k,m)} \right\}$$

Rest of the talk

- Some details
  - Random projection
  - Random projection for classification
  - Projection profile of a sample
- Analysis
- Future/Questions

Random Projection

Random Matrix: $R[k \times n]$ with $r_{ij} \sim N(0,1/k)$

$x \in \Re^n$, $x' = Rx \in \Re^k$

Theorem[JohnsonLindenstraus 84]:

$u,v \in \Re^n$; $[u',v'] = R[u,v]$, projections to $\Re^k$. For any c

$$\boxed{\Pr\left[(1-c) \le \frac{||u'-v'||^2}{||u-v||^2} \le (1+c)\right] \ge 1 - e^{-c^2k/8}}$$

where the probability is over the selection of the random matrix R.

Plan

- Project a sample and the hyperplane
- Bound empirical error in the projected space (k)

Random Projection:A Classification Version

Lemma:

h: n-dimensional classifier, $x \in \Re^n$; $||h||=||x||=1$, $l(x)=h^T x$

The probability of misclassifying x due to the random projection R, is

$$\boxed{\boldsymbol{P}\left[\mathrm{sgn}(\boldsymbol{h}^T \boldsymbol{x}) \neq \mathrm{sgn}(\boldsymbol{h}'^T \boldsymbol{x}')\right] \leq \min\left\{\exp\left(-\frac{\boldsymbol{l}^2(\boldsymbol{x})\boldsymbol{k}}{8(2+|\boldsymbol{l}(\boldsymbol{x})|)^2}\right), \frac{1}{\boldsymbol{k}\boldsymbol{l}^2(\boldsymbol{x})}, 1\right\}}$$

Intuition: (A Classification Version of RP)

$$P\left[\operatorname{sgn}(h^T x) \neq \operatorname{sgn}(h'^T x')\right] \leq \exp\left(-\frac{l^2(x)k}{|8(2+|l(x)|)^2}\right)$$

Since $||h||=||x||=1$, $l=h^T x$ $l'=h'^T x'$

we have $\underline{||h-x||^2} = ||h||^2 + ||x||^2 - 2h^T x = \underline{2-2l}$

$$||h'-x'||^2 = ||h'||^2 + ||x'||^2 - 2l'$$

JL: With probability at least 1-exp(c^2 k/8)

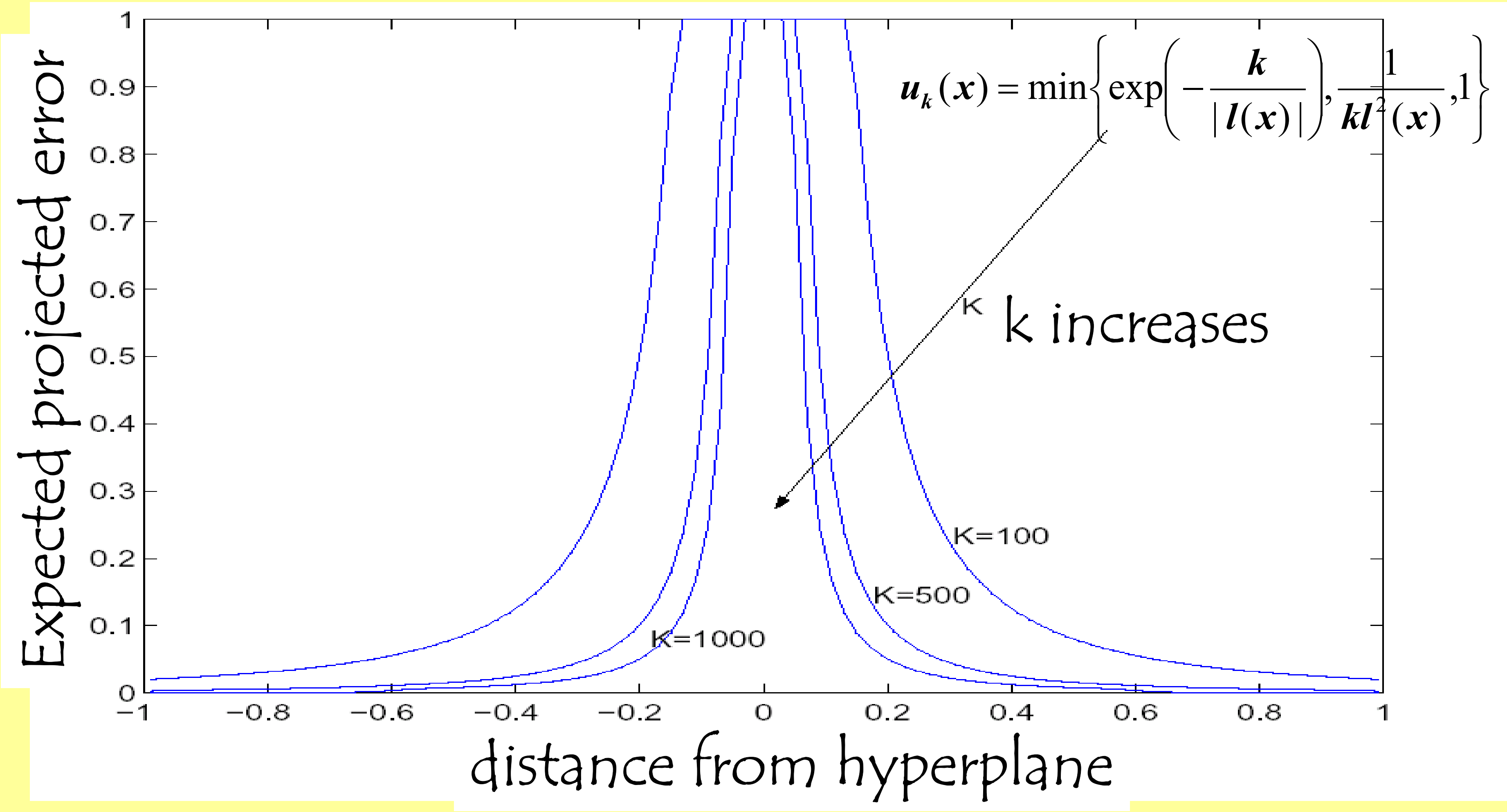
$$(1-c)\,||h||^2 \leq ||h'||^2 \leq (1+c)\,||h||^2,$$

$$(1-c)\,||x||^2 \leq ||x'||^2 \leq (1+c)\,||x||^2$$

$$(1-c)\,||h-x||^2 \leq ||h'-x'||^2 \leq (1+c)\,||h-x||^2.$$

Can find c in JL so that l and l' have same sign.

Contribution of points to error



Projection Error for a Sample (I)

Definition (projection error):

Given a classifier h, a sample S, and a random matrix R, the classification error caused by R is defined by:

$$\mathrm{E}rr_{proj}(h, R, S) = \frac{1}{|S|} \sum_{x \in S} I(sign(h^T x) \neq sign(h'^T x')).$$

Lemma: With probability>1- δ (over the choice of R)

The projection error for sample S, |S|=m is bounded by:

$$\boldsymbol{Err_{proj}(h,R,S)} \leq \frac{1}{\boldsymbol{m}\delta} \sum_{1}^{\boldsymbol{m}} 3\exp\left(-\frac{\boldsymbol{l}^2(\boldsymbol{x})\boldsymbol{k}}{|8(2+|\boldsymbol{l}(\boldsymbol{x})|)^2}\right)$$

Proof idea

- Bound the expectation of the projection error with respect to the choice of the random matrix

$$\boldsymbol{E}[\boldsymbol{Err}_{proj}(\boldsymbol{h}, \boldsymbol{R}, \boldsymbol{S})]$$

- Use Markov inequality

Projection Error for a Sample (II)

Can now establish: The difference between the classification performance on two samples in high dimension is similar to difference in low dimension

Lemma:

S_1, S_2 be two samples in $\Re^n$, $|S_1|=|S_2|=m$;

S'_1, S'_2 the projected sets. Then, with probability $>1-2\delta$

$$P[|Err(h,S_1)-Err(h,S_2)|>\varepsilon] < P[|Err(h',S'_1)-Err(h',S'_2)|>\rho]$$

Where $\rho=\varepsilon-Err(h,S_1)-Err(h,S_2)$

Final Bound

Using Vapnik's doubling trick –

- once on the n dimensional data and
- once on the projected data, can now bound

$$\Pr[\sup_{h \in H} | \bar{E}rr(h) - Err(h, S_1) | > \varepsilon]$$

To yield the final bound.

Emprical error

Random Proj. error

VC at dimension k

$$\boxed{ERR_D \leq ERR_S + \min_k \{\hat{u}_k + VC(k, m)\}}$$

Analysis

- The expected probability of error for a k-dimensional image of x of distance l(x) = from an n-dimensional hyperplane:

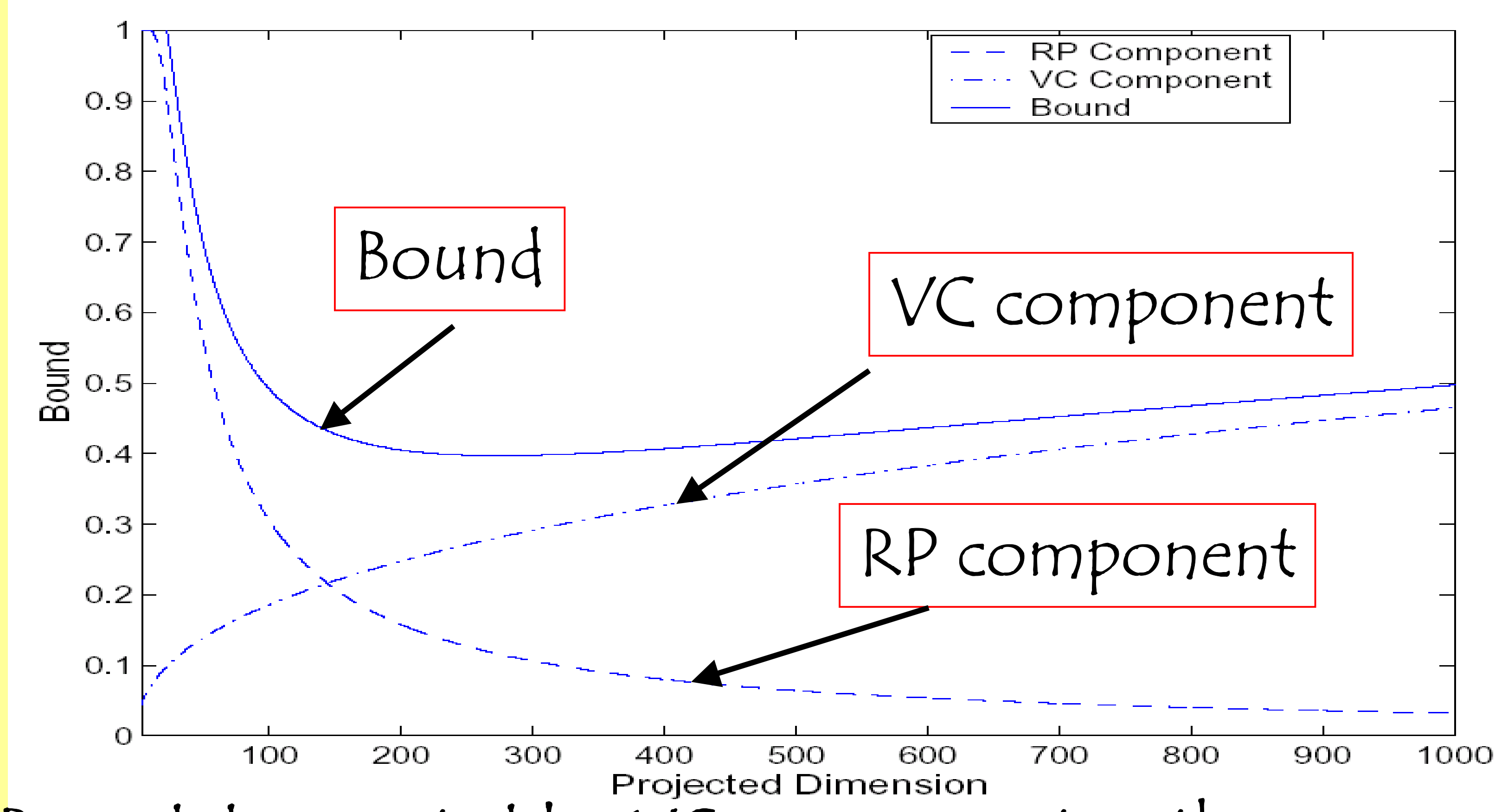
$$\boxed{\min\left\{\exp\left(-\frac{\boldsymbol{l}^2(\boldsymbol{x})\boldsymbol{k}}{8(2+|\boldsymbol{l}(\boldsymbol{x})|)^2}\right),\frac{1}{\boldsymbol{k}\boldsymbol{l}^2(\boldsymbol{x})},1\right\}}$$

- Given a probability distribution over the instance space, can compute the distribution over the margin

$$\boxed{\int_{\boldsymbol{x}\in \boldsymbol{D}}\min\left\{\exp\left(-\frac{\boldsymbol{l}^2(\boldsymbol{x})\boldsymbol{k}}{8(2+|\boldsymbol{l}(\boldsymbol{x})|)^2}\right),\frac{1}{\boldsymbol{k}\boldsymbol{l}^2(\boldsymbol{x})},1\right\}}$$

- E.g., if l ~N(0.3,0.1) can compute this analytically

Generalization Bound, 1 ~N

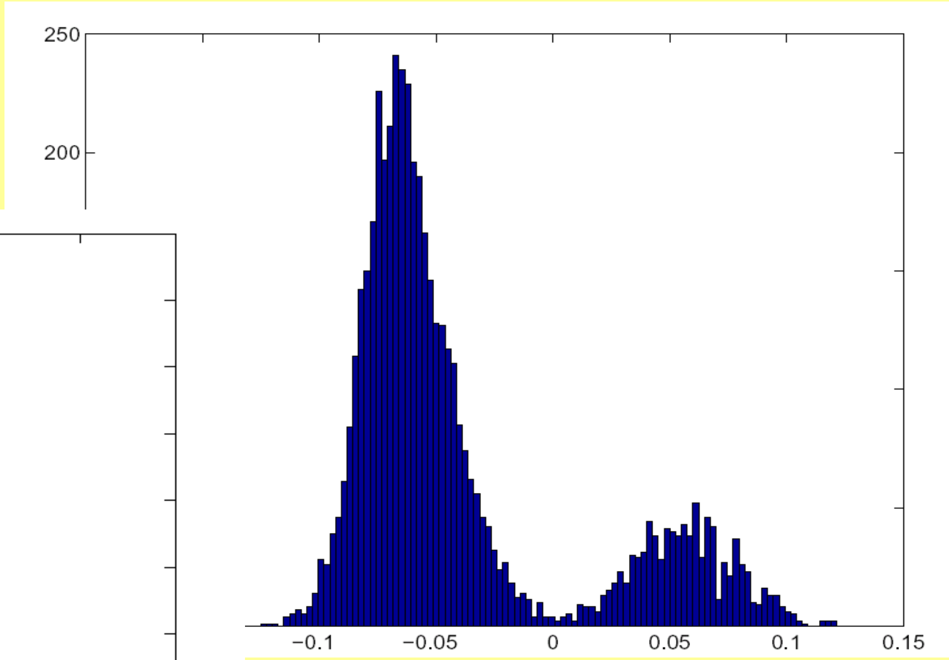


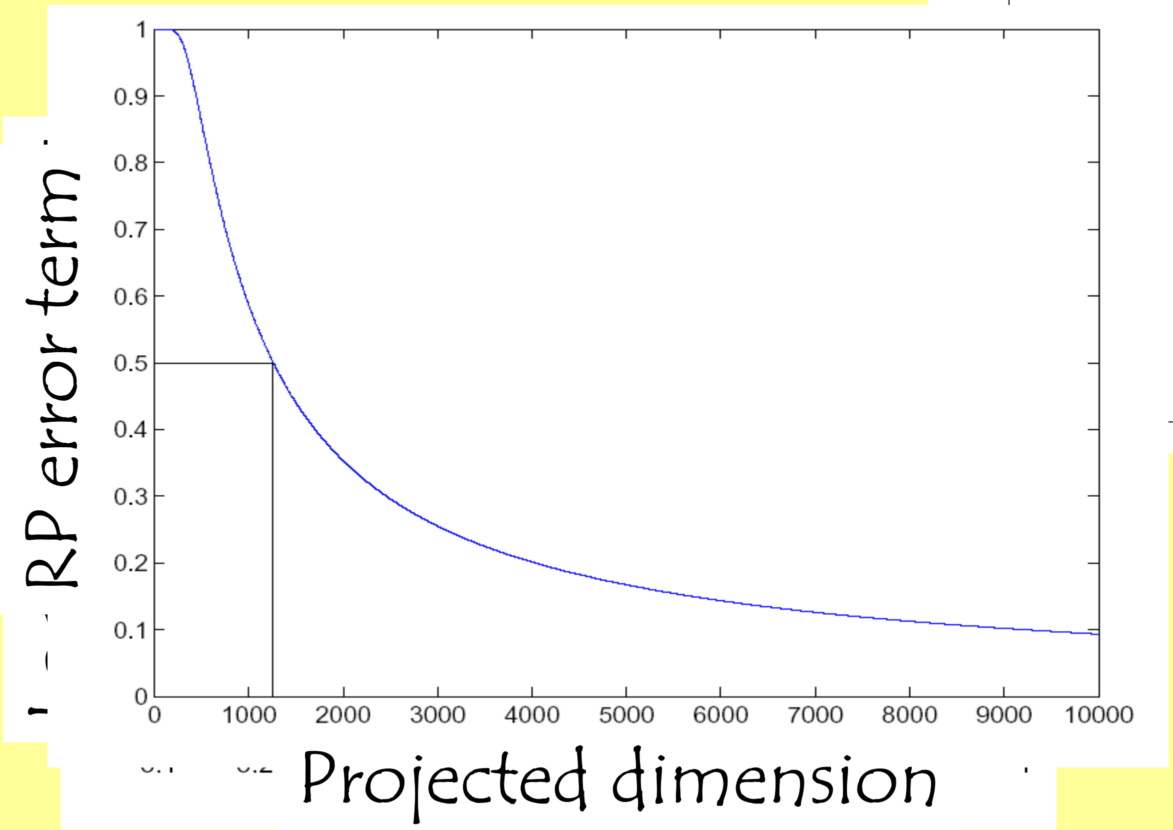
Bound dominated by VC component in the projected dimension

Real Data (I)

17,000 context

Sensitive spelling

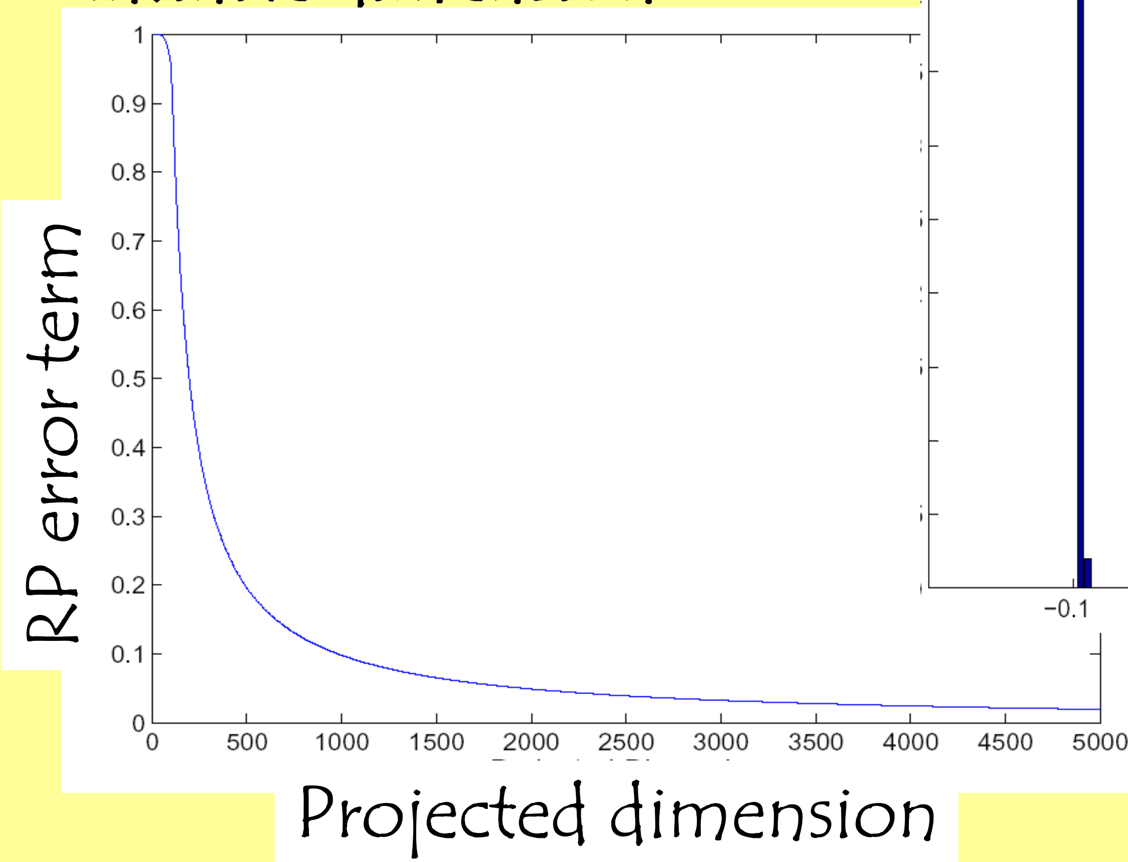


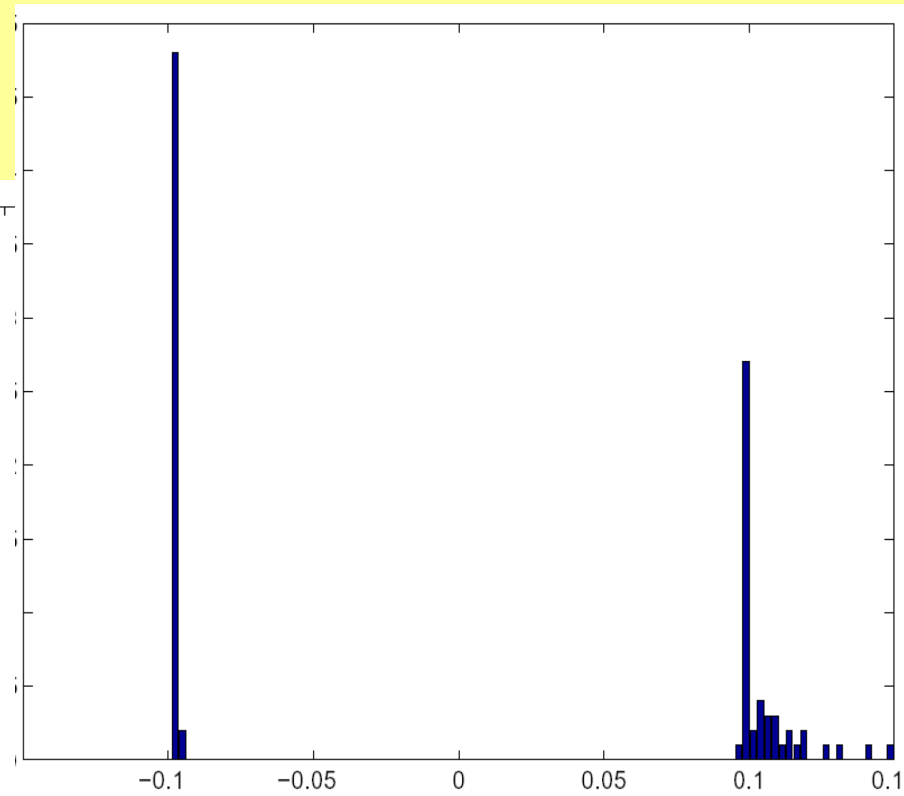


Real Data (II)

RBF kernel face detection

Infinite dimension





Conclusions

- Understanding learning in high dimensional spaces
- Analysis of error based on
  - Prediction preserving projection into low dimension
  - Standard VC argument at low dimension
- Projection profile

  depends on distribution of distance of points to hyperplane
- Gives informative bounds for some real world (very) high dimensional problems

- Algorithmic implications? Better than random proj. ?

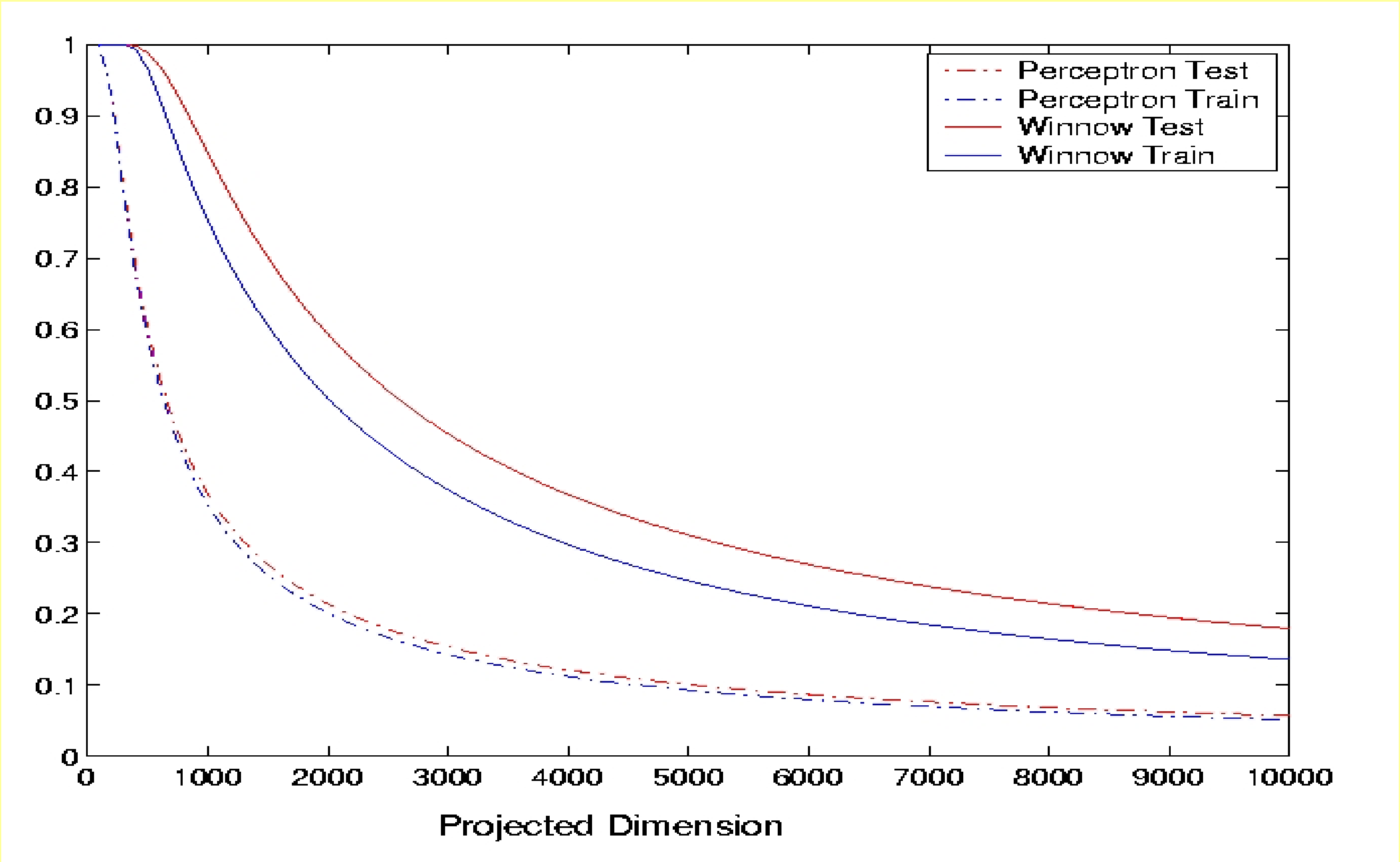
Puzzle

- Is it really the margin?

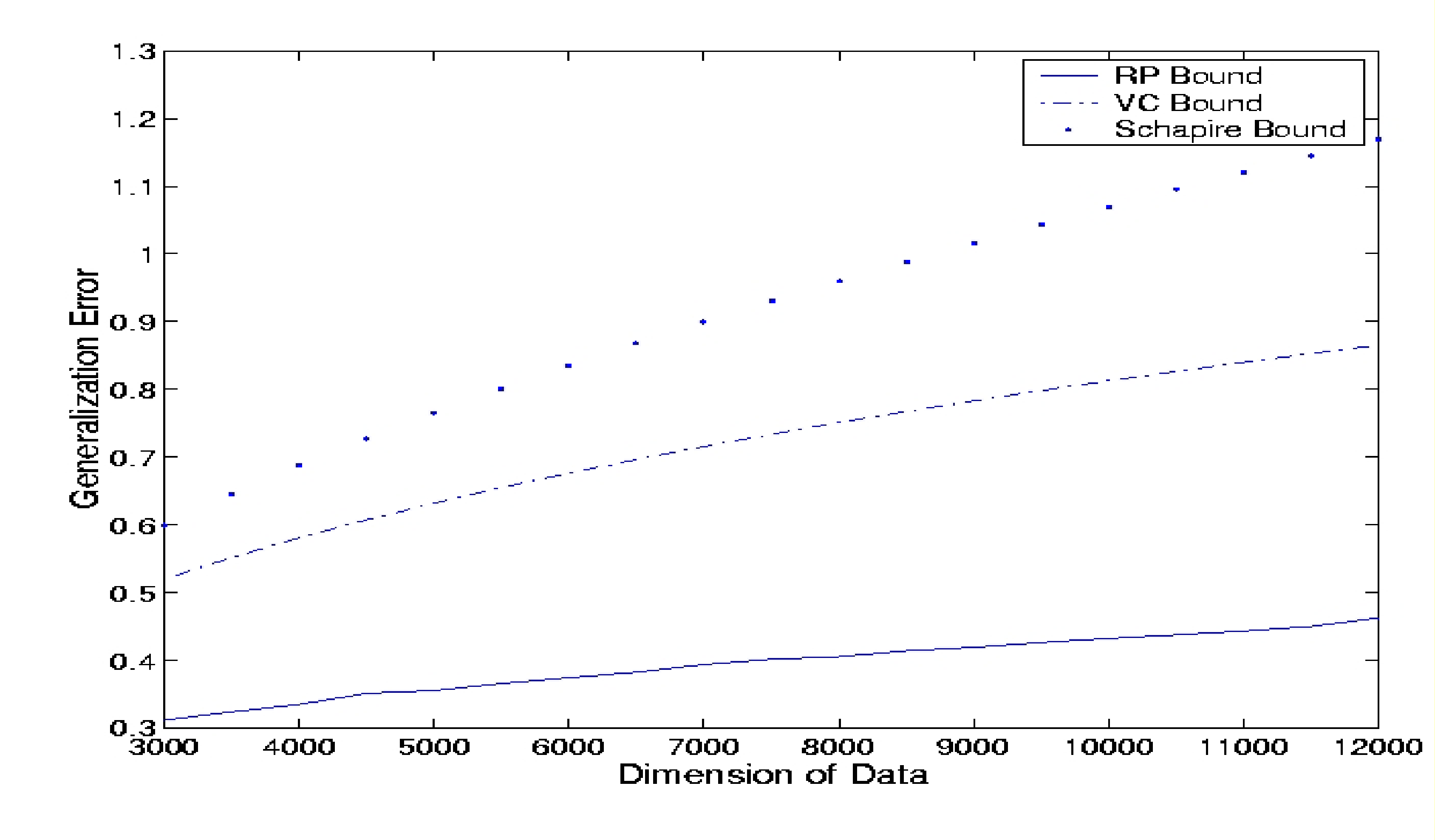
- Example: Winnow vs. Perceptron.
- Perceptron tries to maximize the margin; Winnow does not.
- Indeed, Winnow's margin distribution is worse.

- Yet, Winnow performs consistently better.

Puzzle


Perceptron Test
Perceptron Train
Winnow Test
Winnow Train
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Projected Dimension

Comparison



RP Bound
VC Bound
Schapire Bound
Generalization Error
Dimension of Data
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000

Real Generalization