Announcements

- Final 7-8:15 PM, Wed. 12/15 here
- Q/A session 11-noon Mon. 12/13 2405SC
- Projects (for 4 credits) due Tue. 12/7
  - Code
  - Sample I/O (if it doesn't work, say so)
  - Paper discussing
    - What you did & why
    - What you learned
    - How you would do it differently given…

VC Dimension of a Concept Class

- Can be challenging to prove
- Can be non-intuitive
- Signum(sin($\omega \cdot x$)) on the real line
- Convex polygons in the plane

Learnability

- Often the hypothesis space (or concept class) is syntactically parameterized

  n-Conjuncts, k-DNF, k-CNF, m of n, MLP w/ k units,…
- The concept class is *PAC learnable* if there exists an algorithm whose running time grows no faster than polynomially in the natural complexity parameters: $1/\varepsilon$, $1/\delta$, others
- Clearly, polynomially-bounded growth in the minimum number of training examples is a necessary condition.

Suppose…

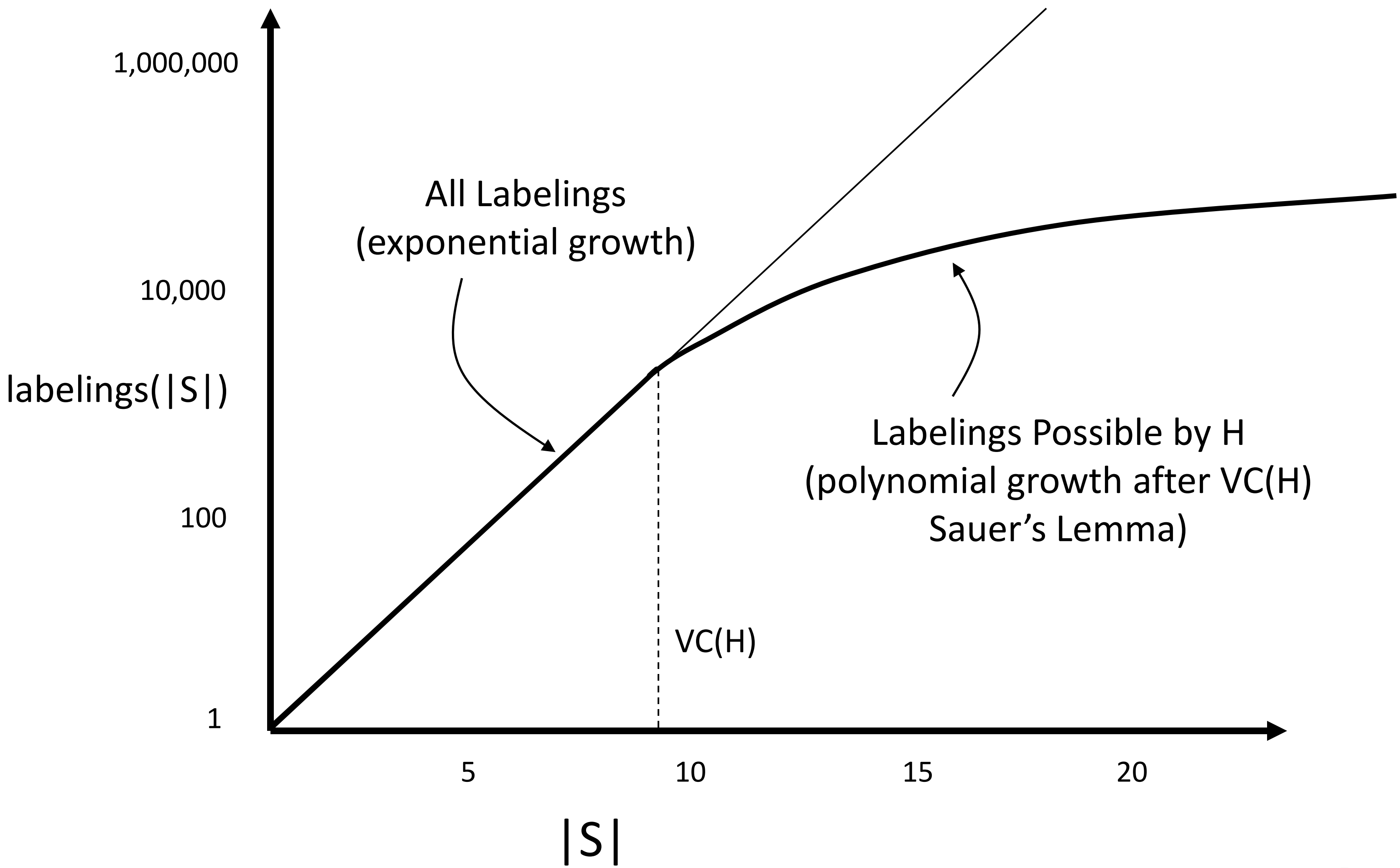
- All $h \in H$ are very low accuracy, say < 0.1% correct
- VC(H) is 100
- Training set S contains 80 labeled examples

What's the probability that an arbitrary h gets the first training example right?

What is the best some $h \in H$ can possibly do on all 80 elements of S?

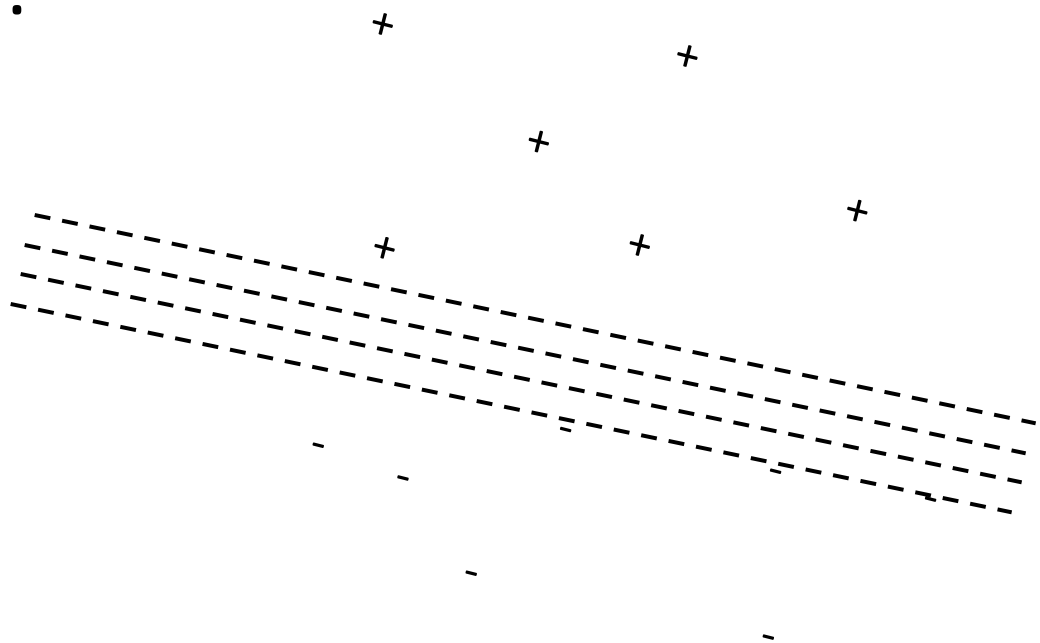
Will this h work well in general?

log(labelings) vs. |S|

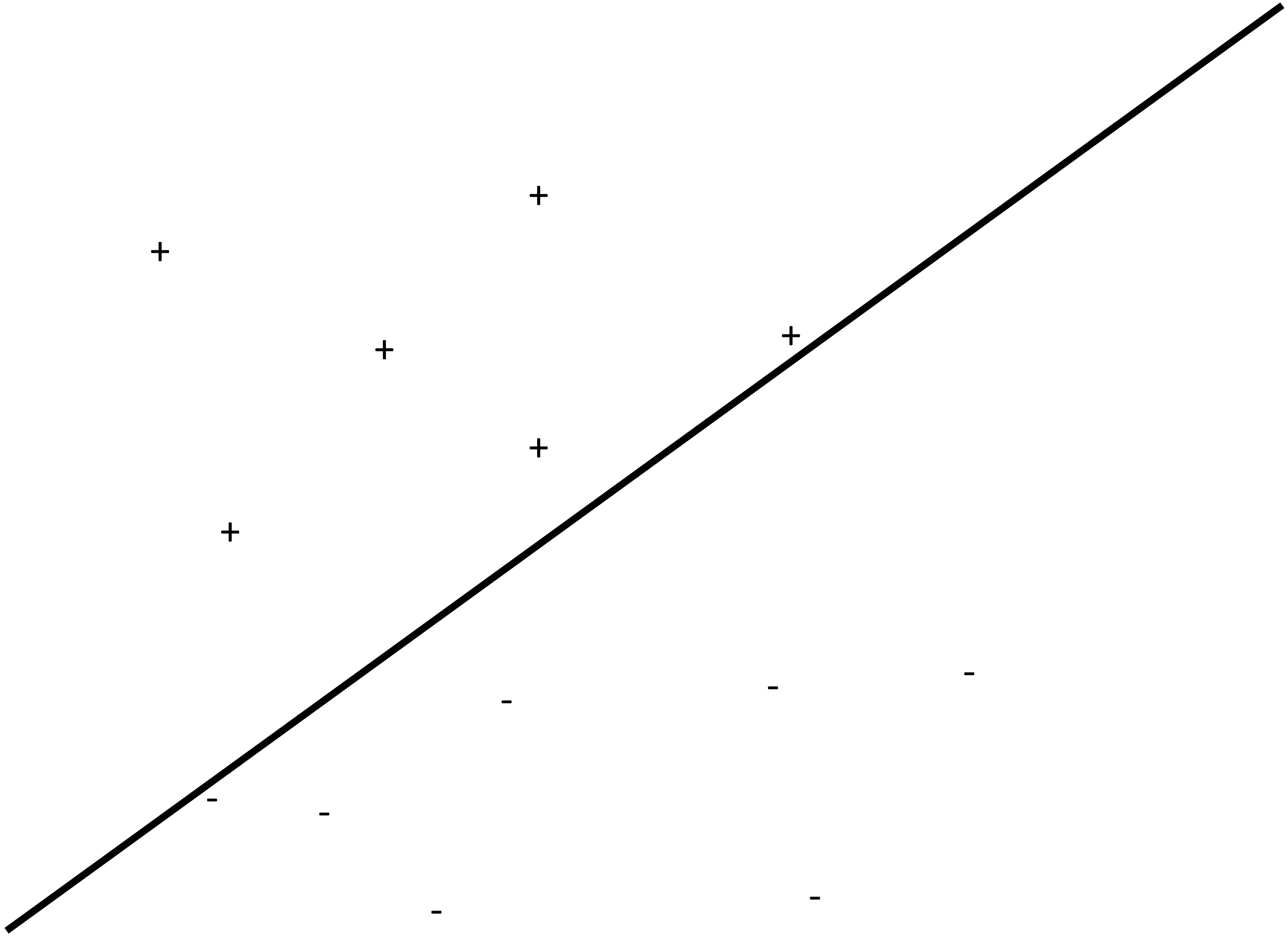


Back to Perceptrons
(linear threshold units, linear discriminators)

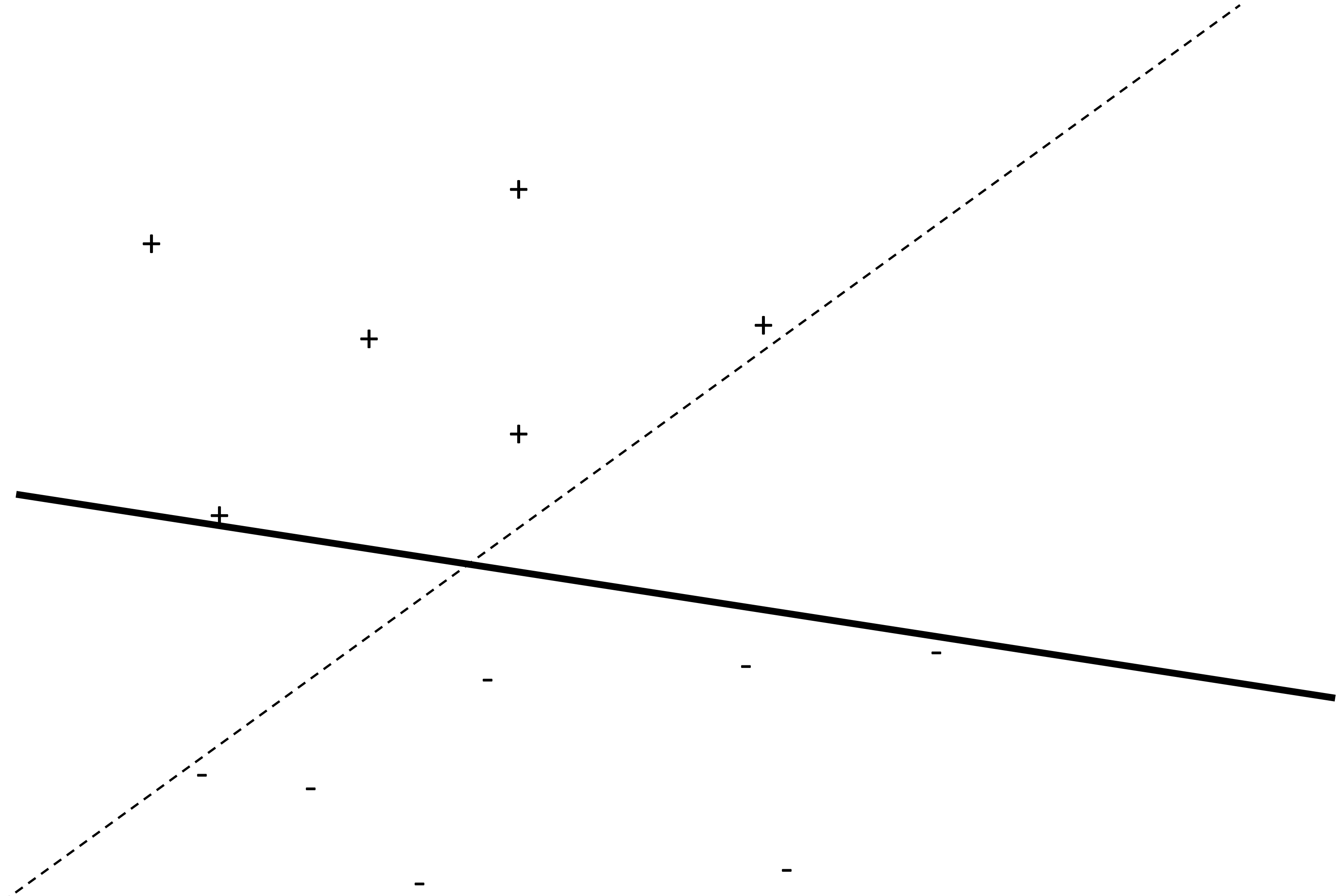
- If there is one perceptron, there are many
- Are some better?
- Is one best?
- Can we tell?
- Can we find it?



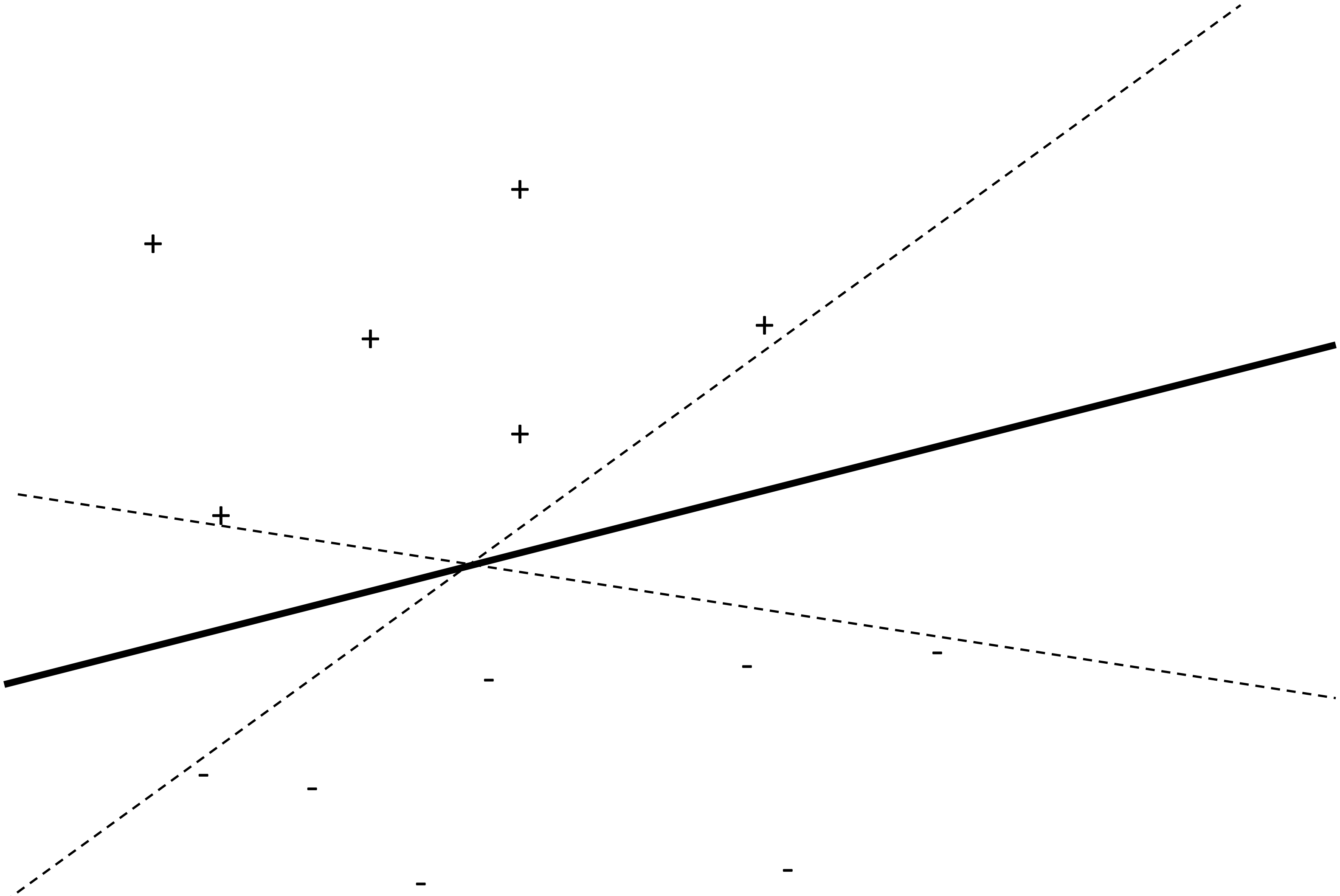
What's the Best Separating Hyperplane?



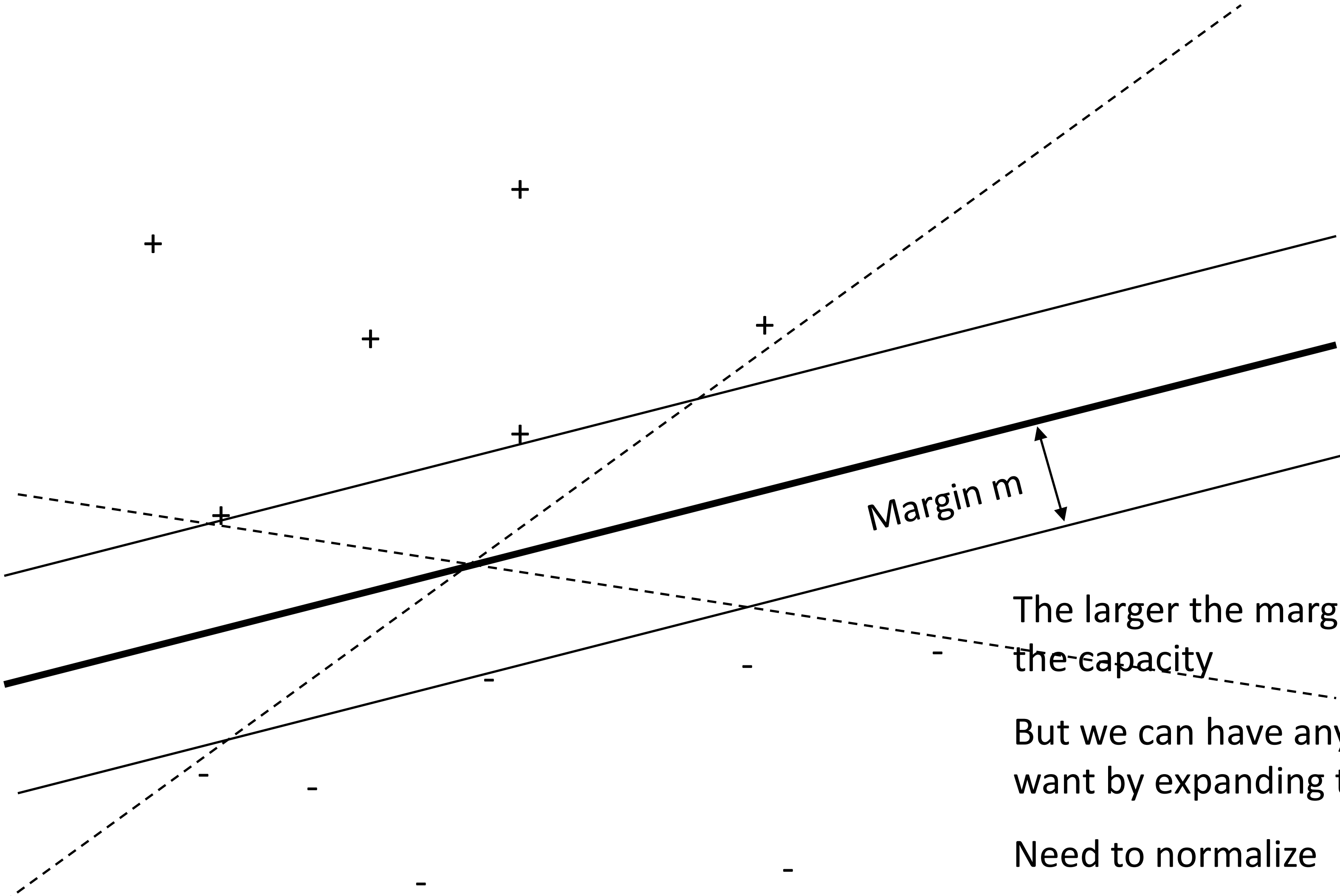
What's the Best Separating Hyperplane?



What's the Best Separating Hyperplane?



What's the Best Separating Hyperplane?

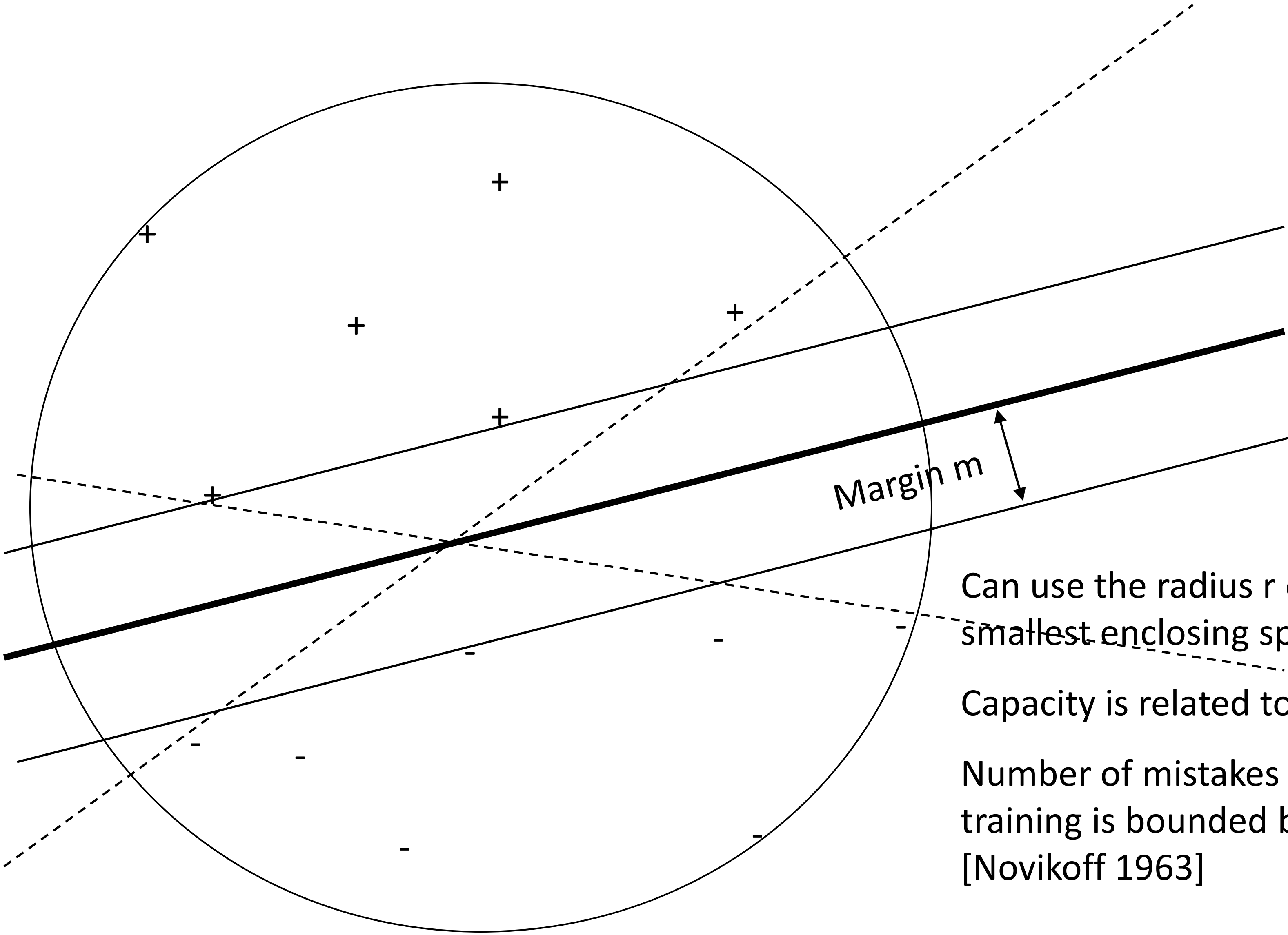


The larger the margin, the lower the capacity

But we can have any margin we want by expanding the space…

Need to normalize

What's the Best Separating Hyperplane?

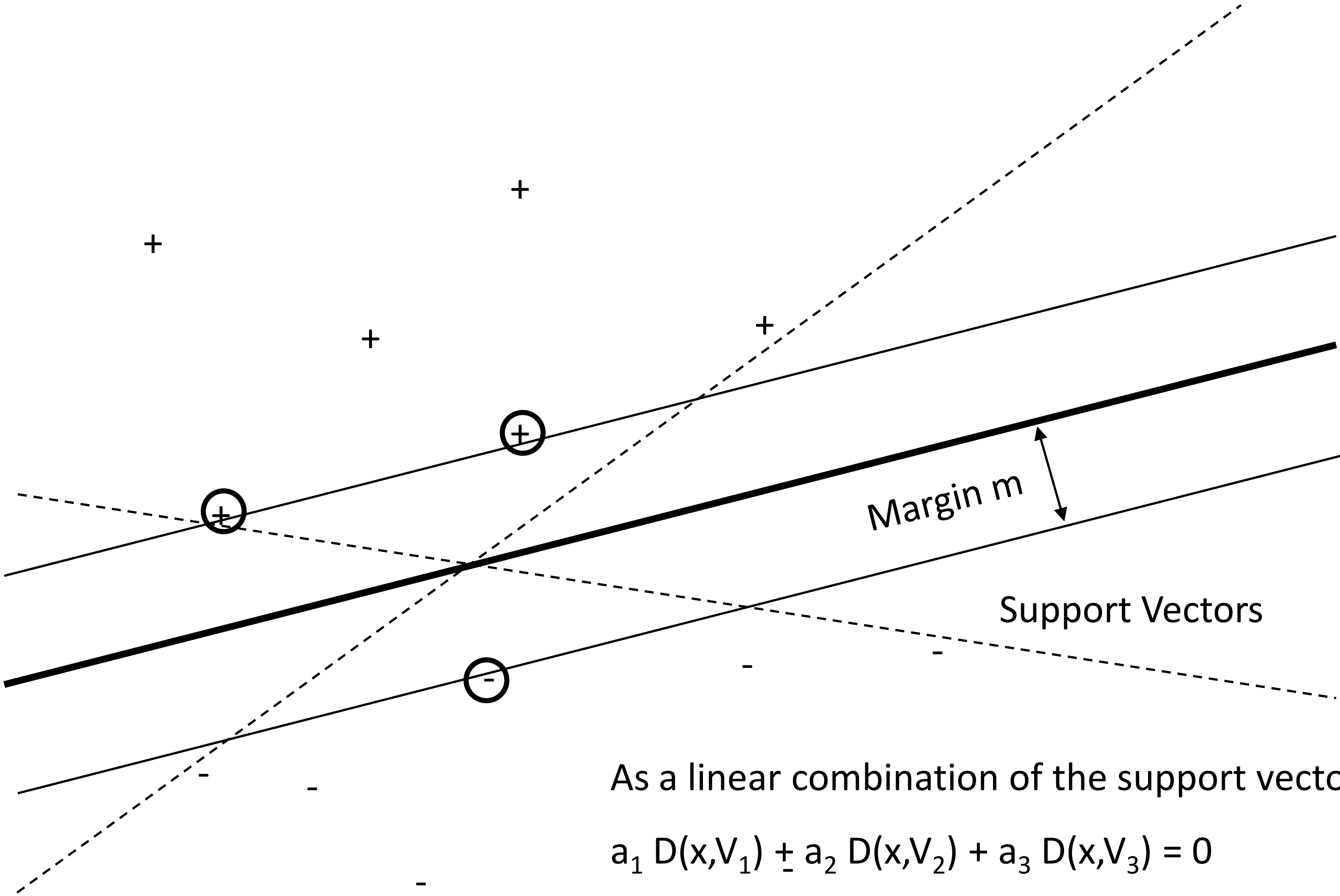


Can use the radius r of the smallest enclosing sphere

Capacity is related to r/m

Number of mistakes in training is bounded by $(r/m)^2$ [Novikoff 1963]

What's the Best Separating Hyperplane?



As a linear combination of the support vectors

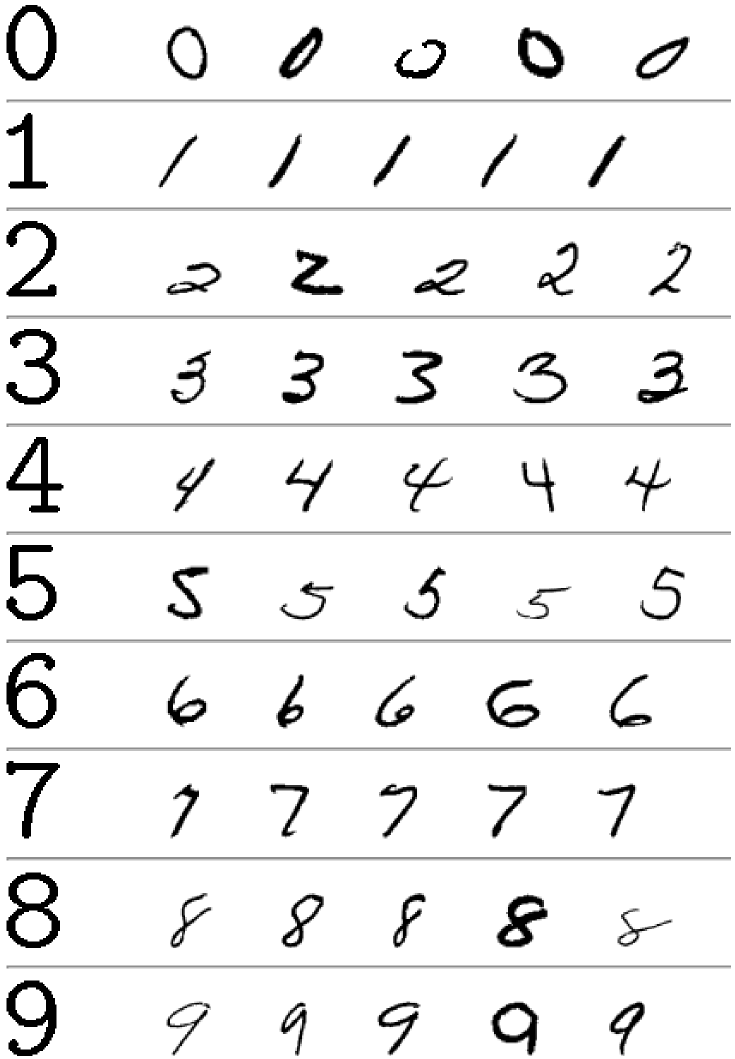
$a_1\, D(x,V_1) \pm a_2\, D(x,V_2) + a_3\, D(x,V_3) = 0$

$\sum a_i\, D(x,V_i) = 0$

Why are Large Margins Better?

- Classification is more robust / stable
  - Small changes to training examples
  - Re-sampling training data
  - New examples
- Lower expressiveness / capacity
  - The larger the margin
    - the fewer the hypothesis choices given the data
    - the less the symmetric difference of similar hypotheses
  - "Fat Shattering" dimension rather than
    Shattering & VC dimension
- Contingent on choice of distance metric also approximately measuring confidence
- What choices for distance? Many, but little guidance…
- We do not choose the margin – it is a learning bias
  - Possibly: Train, Measure margin, Calculate significance(?)
  - But bounds are loose; calculations cannot be trusted
  - Rather large margin considerations
    - suggest learning algorithms
    - forms a well-defined, well-motivated learning bias
  - Bias is parameterized by distance choice and (oddly) by training examples

Consider a Perceptron for Handwritten Digit Recognition



- Pixel input, e.g.:
- 32 x 32 x 8
- x = 1024 features / dimensions, each 256 values
- Generic ANNs work poorly
- Specially designed ANNs work very well
- Multi-class from binary
  - Ten index classifiers
  - All pairs w/ voting
  - Four base 2 encoders
  - Consider “3” vs. “6”
- Will a perceptron work well? Why?

What Determines the Maximum Margin Separator?

- Only the nearest / most constraining points (support vectors)
- A learner that finds them is called a Support Vector Machine (SVM)
- Finding them is a quadratic programming optimization problem
- There are efficient iterative solutions given certain conditions
- Note class density estimation is no longer necessary
- Maximizing margin minimizes risk, assuming…
- Many extensions
  - Noise, outliers, non-separable classes, imbalanced training…
  - Soft margins, margin distributions, asymmetric margins…

Kernel Spaces: Better Distance Metrics

- Instead of adding perceptron layers, choose a better distance metric
- What???
- Want 7's to be close, 8's to be close but far from 2's, etc.
- Image distance as combination of independent pixel distances does not work well
- What are we missing?
  - Pixels do not contribute independently
  - Must appreciate interactions among pixels

Kernel Methods

- Map to a new higher dimensional space
  - Can be very high
  - Can be infinite
- Kernel functions
  - Introduce high dimensionality
  - Computation is independent of dimensionality
  - Defined w/ dot product of input image vectors (information on the Cosine between image vectors)
- A kernel function defines a distance metric over space of example images

Mercer's Condition / Representer Theorem

- <Kernel matrix is positive semidefinite>
- The desired hyperplane can be represented as

$$\sum_{i=1}^{m} \alpha_i K(\mathbf{s}_i, \mathbf{x})$$

  Linear weighted sum of similarities to support vectors
- Kernel defines a distance metric
- The hypothesis space is represented efficiently by using some of the training examples – the support vectors

SVMs for Digit Images

- $K(x,y) = (x \cdot y)^3$ or $(x \cdot y + 1)^3$
- Dot product $\rightarrow$ scalar; cube it
  Consider how this works…
- Before 32^2 features (or about 10^3)
- Now ~ $(32^2)^3$ features (or about 10^9)
- New Feature = monomial = correlation among three pixels
- VC(lin sep) ~ # dimensions
- Overfitting problem?
  - Not if the margin is large
  - Monitor number of support vectors

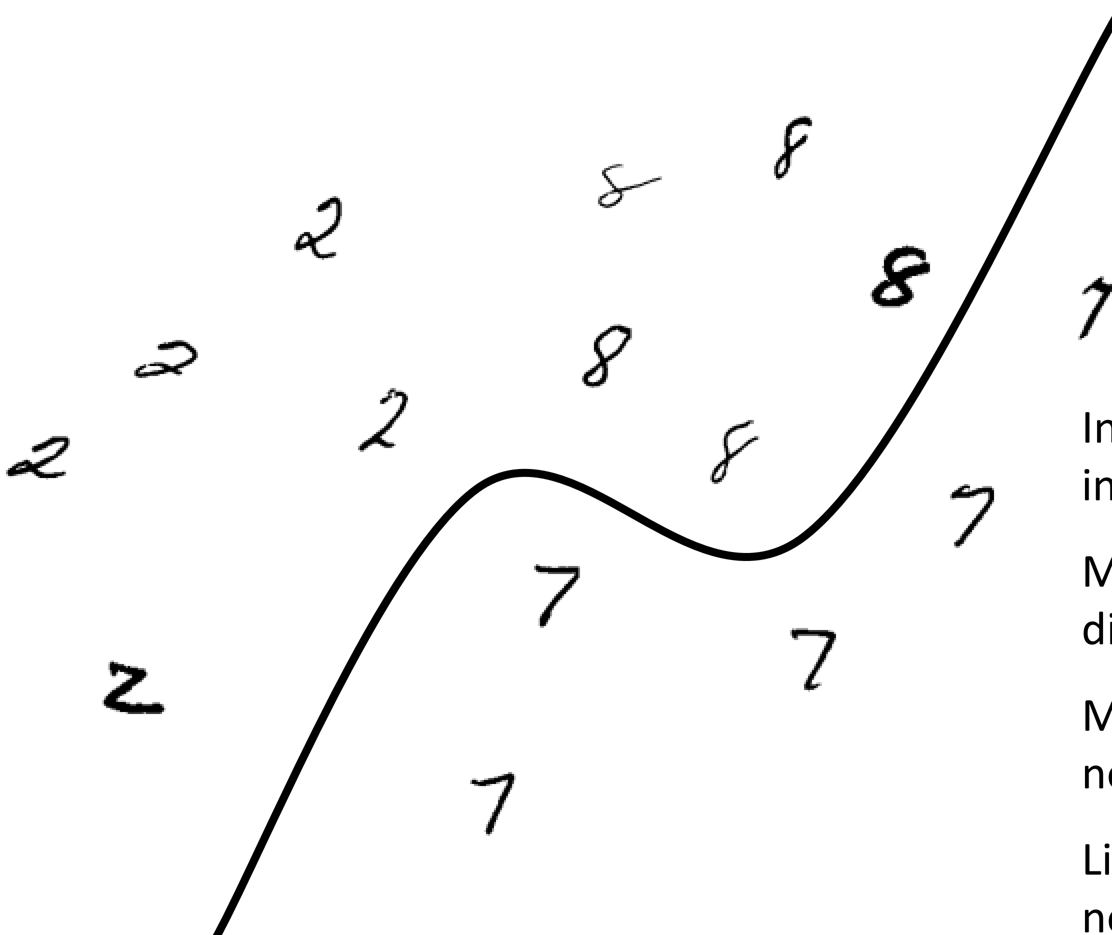
Distinguishing Handwritten Seven's vs. Two's and Eight's

Handwritten 32 x 32 gray scale pixels

Two's

Eight's

Seven's



Input feature space is inappropriate

Map inputs to a high-dimensional space

Many more features; nonlinear combinations

Linearly separable in the new space

Mercer Kernels

Usually start with a kernel rather than features

$(s \cdot x)^d$ Homogeneous polynomials

$(s \cdot x + 1)^d$ Complete polynomials

$\mathrm{Exp}(-||s - x||^2 / 2\,\sigma^2)$ Gaussian / RBF

$K + k$

$c \cdot K$

$K + c$

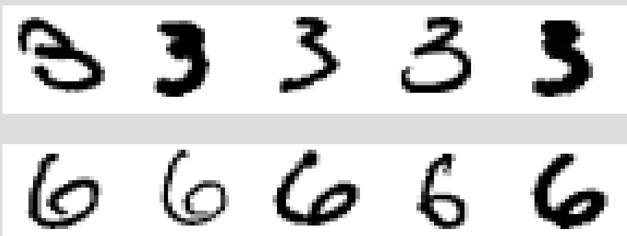
$K \cdot k$

Problems

SVMs & statistical learning generally

- Little information from each training example
  - Signal must show through the noise
  - Need many training examples
  - Thousands of are needed for handwritten digits
- Much information is ignored (weak bias vocabulary)
- Compare w/ humans
  - Novel simple shape of similar complexity
  - Master with several tens (perhaps a hundred) training examples
  - Exceedingly small non-fatigue error rate

Two Related Classification Problems



	No. examples	error
Humans	< 100 ?	negligible
SVMs	60000	1.2%

Two Related Classification Problems

a fixed permutation over pixels →

	No. examples	error
Humans	< 100 ?	negligible
SVMs	60000	1.2%

Two Related Classification Problems

	No. examples	error	No. examples	error
Humans	< 100 ?	negligible	NA	50%
SVMs	60000	1.2%	60000	1.2%

To an SVM these are the *same problem*
Apparently the SVM ignores information crucial to people

- Statistical machine learning
- Regularization – reduce the available expressiveness
- SVMs – large margin is a regularizer

Semi-Supervised Learning

- Access to
  - Some labeled training examples
  - Many more unlabeled examples
  - Often cheaper…
- Co-Training
  - Mitchell & Blum
  - Two different style learners
  - Train each on supervised set
  - Train each other on unsupervised examples
- Direct information: distribution density
- Transductive learning
  - Vapnik
  - Simpler problem than inductive (supervised) learning (?)
  - Added bias: Prefer confidence on unlabeled examples
  - Consider a SVM…

Unsupervised Learning
Clustering

- Only unlabeled examples
- Learn / Guess structure of the space
- Mixture modeling
- Many techniques
- K-means, metric space
  - Popular, Simple
  - Assume K random centers
  - Assign members to clusters given the centers
  - Re-compute the centers given the members
  - Repeat