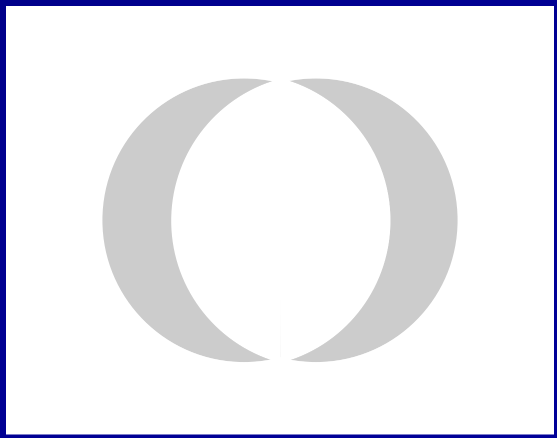
Statistical and Computational Learning Theory

- Fundamental Question: Predict Error Rates
  - Given:
    - The space H of hypotheses
    - The number and distribution of the training examples S
    - The complexity of the hypothesis $h \in H$ output by the learning algorithm
    - Measures of how well *h* fits the examples
    - etc.
  - Find:
    - Theoretical bounds on the error rate of *h* on new data points.

General Assumptions (Noise-Free Case)

- Assumption: Examples are generated according to a probability distribution D(**x**) and labeled according to an unknown function f: *y* = f(**x**)
- Learning Algorithm: The learning algorithm is given a set of *m* examples, and it outputs an hypothesis *h* $\in$ H that is <u>consistent</u> with those examples (i.e., correctly classifies all of them).
- Goal: *h* should have a low error rate ε on new examples drawn from the <u>same distribution</u> D.

$$error(h, f) = P_D[f(\mathbf{x}) \neq h(\mathbf{x})]$$



Probably-Approximately Correct Learning

- We allow our algorithms to fail with probability δ
- Imagine drawing a sample of *m* examples, running the learning algorithm, and obtaining *h*. Sometimes, the sample will be unrepresentative, so we only want to insist that $1 - \delta$ of the time, the hypothesis will have error less than ε. For example, we might want to obtain a 99% accurate hypothesis 90% of the time.
- Let $P^m_D(S)$ be the probability of drawing data set S of *m* examples according to D.

$$P_D^m\left[error(f,h) > \epsilon\right] < \delta$$

Case 1: Finite Hypothesis Space

- Assume H is finite
- Consider $h_1 \in$ H such that *error*(*h*,*f*) > ε. What is the probability that it will correctly classify *m* training examples?
- If we draw one training example, ($\mathbf{x}_1$,y_1), what is the probability that h_1 classifies it correctly?

  $P[h_1(\mathbf{x}_1) = y_1] = (1 - \varepsilon)$
- What is the probability that *h* will be right *m* times?

  $P^m_D[h_1(\mathbf{x}_1) = y_1] = (1 - \varepsilon)^m$

Finite Hypothesis Spaces (2)

- Now consider a second hypothesis h_2 that is also ε-bad. What is the probability that <u>either</u> h_1 or h_2 will survive the m training examples?

$$
\begin{aligned}
P^m_D[h_1 \vee h_2 \text{ survives}] &= P^m_D[h_1 \text{ survives}] + P^m_D[h_2 \text{ survives}] - P^m_D[\mathrm{h}_1 \wedge h_2 \text{ survives}] \\
&\leq P^m_D[h_1 \text{ survives}] + P^m_D[h_2 \text{ survives}] \\
&\leq 2(1 - \varepsilon)^m
\end{aligned}
$$

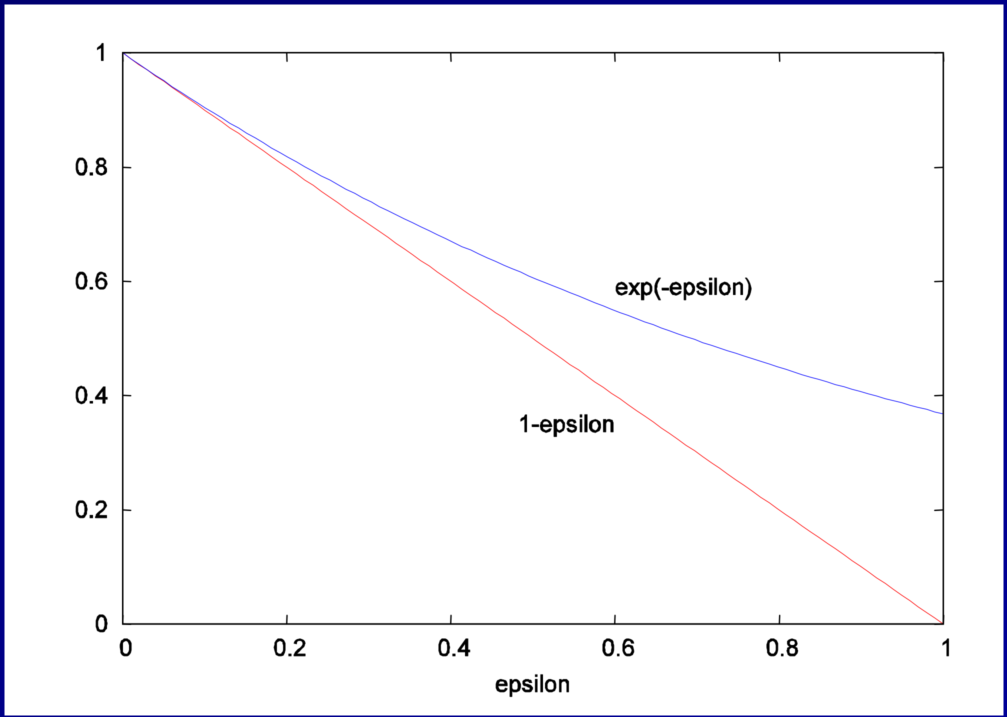
- So if there are k ε-bad hypotheses, the probability that <u>any one</u> of them will survive is $\leq$ k $(1 - \varepsilon)^m$
- Since $k < |H|$, this is $\leq |H|(1 - \varepsilon)^m$

Finite Hypothesis Spaces (3)

- Fact: When $0 \leq \varepsilon \leq 1$, $(1 - \varepsilon) \leq e^{-\varepsilon}$

  therefore

  $$|H|(1 - \varepsilon)^m \leq |H|\, e^{-\varepsilon m}$$



Blumer Bound

(Blumer, Ehrenfeucht, Haussler, Warmuth)

- Lemma. For a finite hypothesis space H, given a set of *m* training examples drawn independently according to D, the probability that there exists an hypothesis $h \in$ H with true error greater than ε consistent with the training examples is less than $|H|e^{-\varepsilon m}$.
- We want to ensure that this probability is less than δ.

  $$|H|e^{-\varepsilon m} \leq \delta$$

- This will be true when

$$m \geq \frac{1}{\epsilon}\left(\ln |H| + \ln \frac{1}{\delta}\right).$$

Finite Hypothesis Space Bound

- Corollary: If *h* $\in$ H is consistent with all *m* examples drawn according to D, then the error rate ε on new data points can be estimated as

$$\epsilon = \frac{1}{m}\left(\ln|H| + \ln\frac{1}{\delta}\right).$$

Examples

- Boolean conjunctions over *n* features.

  $|H| = 3^n$, since each feature can appear as x_j, $\neg x_j$, or be missing.

$$\epsilon = \frac{1}{m}\left(n \ln 3 + \ln \frac{1}{\delta}\right)$$

- k-DNF formulas:

$$(x_1 \wedge x_3) \vee (x_2 \wedge \neg x_4) \vee (x_1 \wedge x_4)$$

  There are at most $(2n)^k$ disjunctions, so

$$|H| \leq 2^{(2n)^k}$$

- for <u>fixed</u> *k*, this gives

$$\log_2 |H| = (2n)^k$$

- which is polynomial in *n*:

$$\epsilon = \frac{1}{m} O\left(n^k + \ln \frac{1}{\delta}\right)$$

Finite Hypothesis Space: Inconsistent Hypotheses

- Suppose that *h* does not perfectly fit the data, but rather that it has an error rate of ε_T. Then the following holds:

$$\epsilon <= \epsilon_T + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$$

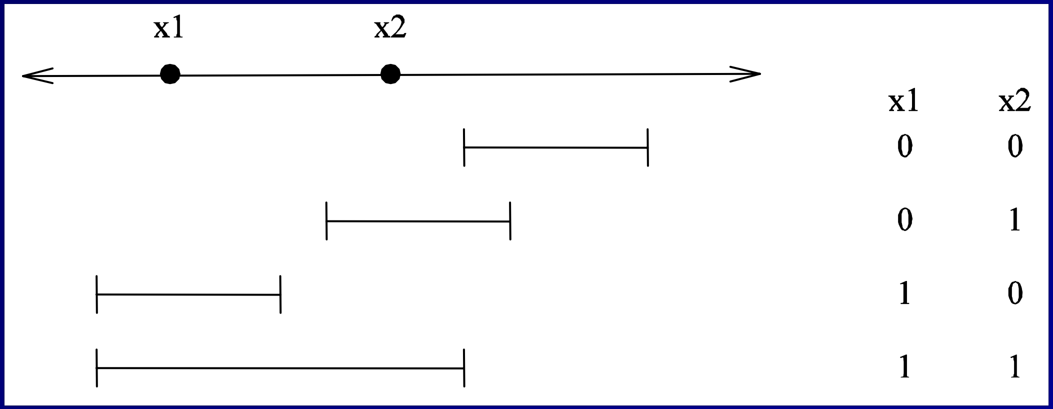
- This makes it clear that the error rate on the test data is usually going to be larger than the error rate ε_T on the training data.

Case 2: Infinite Hypothesis Spaces and the VC Dimension

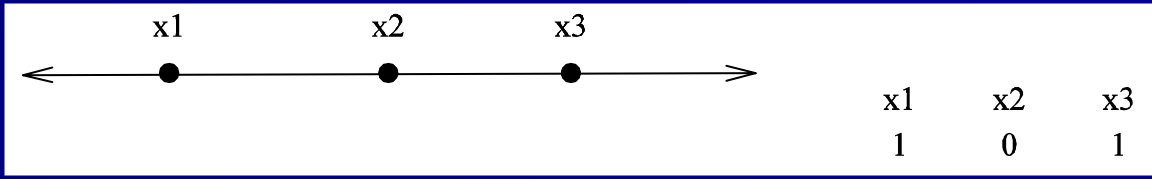
- Most of our classifiers (LTUs, neural networks, SVMs) have continuous parameters and therefore, have infinite hypothesis spaces
- Despite their infinite size, they have limited expressive power, so we should be able to prove something
- Definition: Consider a set of *m* examples S = $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$. An hypothesis space H can trivially fit *S*, if for every possible way of labeling the examples in S, there exists an $h \in$ H that gives this labeling. (H is said to "shatter" S)
- Definition: The Vapnik-Chervonenkis dimension (VC-dimension) of an hypothesis space H is the size of the largest set S of examples that can be trivially fit by H.
- For finite H, VC(H) $\leq \log_2 |H|$

VC-dimension Example (1)

- Let H be the set of intervals on the real line such that *h*(**x**) = 1 iff **x** is in the interval. H can trivially fit any pair of examples:

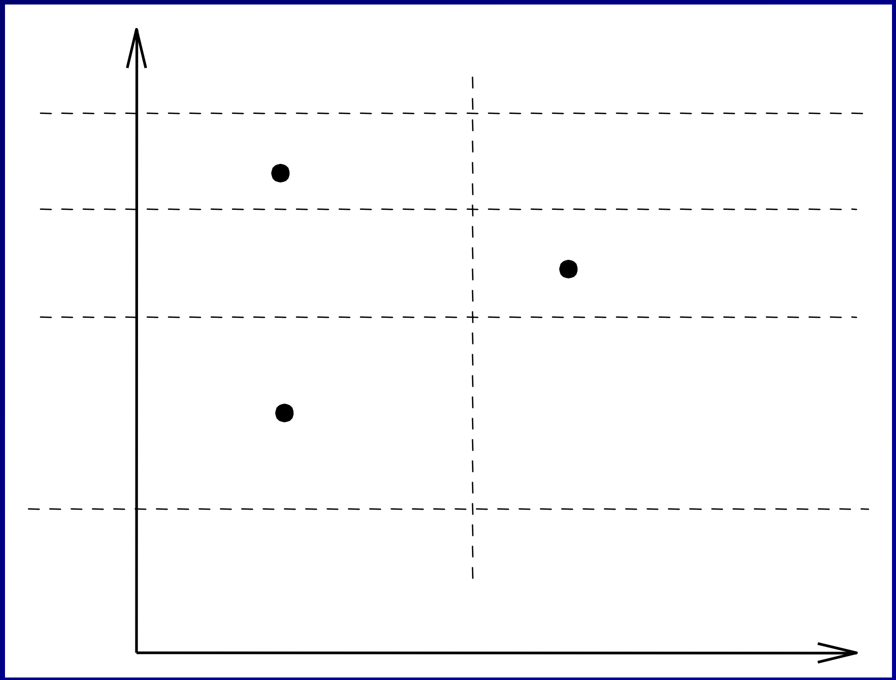


- However, H cannot trivially fit any triple of examples. Therefore the VC-dimension of H is 2



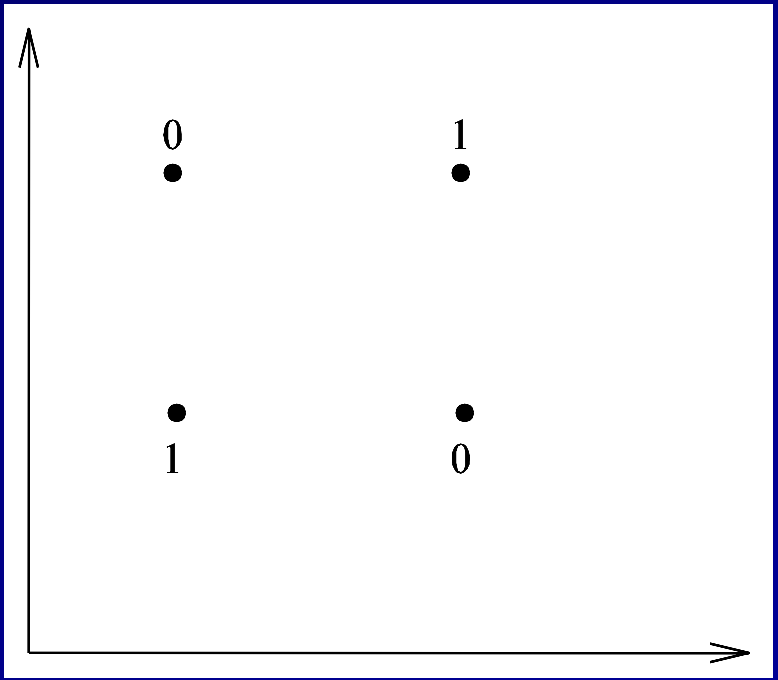
VC-dimension Example (2)

- Let H be the space of linear separators in the 2-D plane. We can trivially fit any 3 points.



VC-dimension Example (3)

- We cannot separate any set of 4 points (XOR). In general, the VC-dimension for LTUs in *n*-dimensional space is *n*+1. A good heuristic is that the VC-dimension is equal to the number of tunable parameters in the model (unless the parameters are redundant)



VC-dimension of Neural Networks

- The VC-dimension of a multi-layer perceptron network of depth s is

  $$VC \leq 2(n + 1)\, s\, (1 + \ln s)$$

- The exact value for sigmoid units is open, but probably larger

Error Bound for Consistent Hypotheses

- The following bound is analogous to the Blumer bound. If *h* is an hypothesis that makes no error on a training set of size *m*, and *h* is drawn from an hypothesis space H with VC-dimension *d*, then with probability 1 – δ, *h* will have an error rate less than ε if

$$m \geq \frac{1}{\epsilon}(4 \log_2(2/\delta) + 8d \log_2(13/\epsilon))$$

Error Bound for Inconsistent Hypotheses

- Theorem. Suppose H has VC-dimension *d* and a learning algorithm finds $h \in H$ with error rate ε_T on a training set of size *m*. Then with probability $1 - \delta$, the error rate ε on new data points is

$$\epsilon <= 2\epsilon_T + \frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)$$

- Empirical Risk Minimization Principle
  - If you have a fixed hypothesis space H, then your learning algorithm should minimize ε_T: the error on the training data. (ε_T is also called the "empirical risk")

Case 3: Variable-Sized Hypothesis Spaces

- A fixed hypothesis space may not work well for two reasons
  - Underfitting: Every hypothesis in H has high ε_T. We would like to consider a larger hypothesis space H' so we can reduce ε_T
  - Overfitting: Many hypotheses in H have $\varepsilon_T = 0$. We would like to consider a smaller hypothesis space H' so we can reduce *d*.
- Suppose we have a nested series of hypothesis spaces:

$$H_1 \subseteq H_2 \subseteq \ldots \subseteq H_k \subseteq \ldots$$

with corresponding VC dimensions and errors

$$d_1 \leq d_2 \leq \ldots \leq d_k \leq \ldots$$

$$\varepsilon^1_T \geq \varepsilon^2_T \geq \ldots \geq \varepsilon^k_T \geq \ldots$$

Structural Risk Minimization Principle (Vapnik)

- Choose the hypothesis space H_k that minimizes the combined error bound

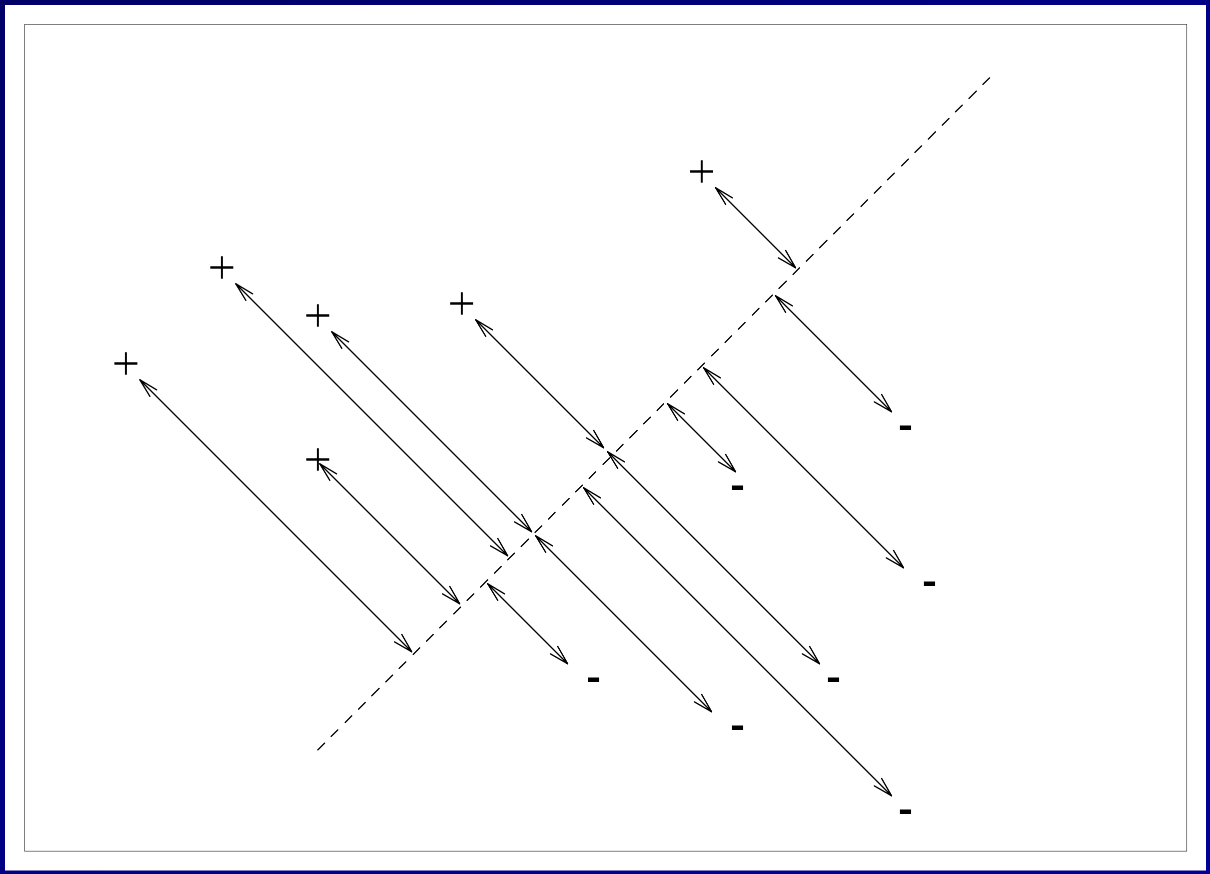
$$\epsilon <= 2\epsilon_T^k + \frac{4}{m}\left(d_k \log\frac{2em}{d_k} + \log\frac{4}{\delta}\right)$$

Case 4: Data-Dependent Bounds

- So far, our bounds on ε have depended only on ε_T and quantities that could be computed prior to training
- The resulting bounds are "worst case", because they must hold for all but $1 - \delta$ of the possible training sets.
- Data-dependent bounds measure other properties of the fit of *h* to the data. Suppose S is not a worst-case training set. Then we may be able to obtain a tighter error bound

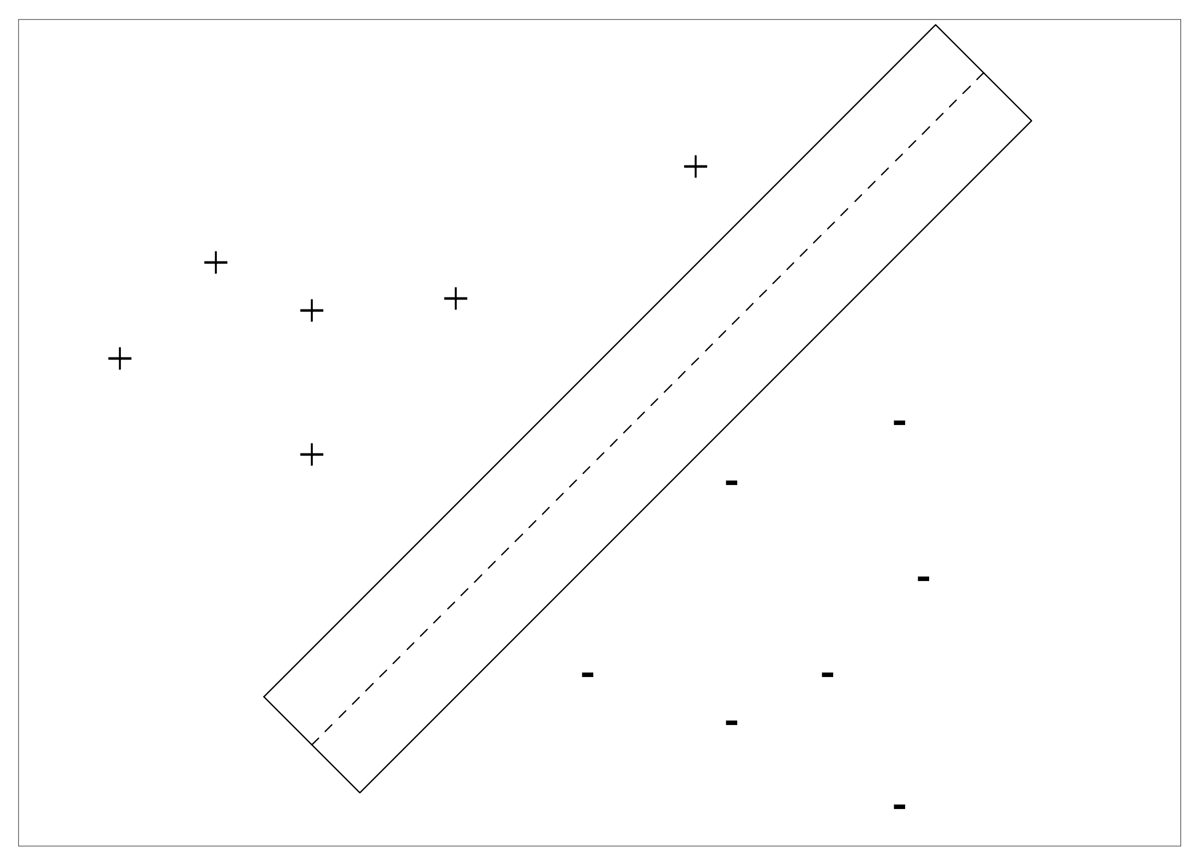
Margin Bounds

- Suppose $g(\mathbf{x})$ is a real-valued function that will be thresholded at 0 to give $h(\mathbf{x})$: $h(\mathbf{x}) = \mathrm{sgn}(g(\mathbf{x}))$. The <u>functional margin</u> γ of g on training example $\langle \mathbf{x}, y \rangle$ is $\gamma = y g(\mathbf{x})$. The margin with respect to the whole training set is defined as the minimum margin over the entire set: $\gamma(g,S) = \min_i y_i\, g(\mathbf{x}_i)$



Margin Bounds: Key Intuition

- Consider the space of real-valued functions G that will be thresholded at 0 to give H. This space has some VC dimension *d*. But now, suppose that we consider "thickening" each $g \in G$ by requiring that it correctly classify every point with a margin of at least γ. The VC dimension of these "fat" separators will be much less than *d*. It is called the fat shattering dimension: $fat_G(\gamma)$



Noise-Free Margin Bound

- Suppose a learning algorithm finds a $g \in G$ with margin $\gamma = \gamma(g,S)$ for a training set S of size m. Then with probability $1 - \delta$, the error rate on new points will be

$$\epsilon <= \frac{2}{m}\left(d \log \frac{2em}{d\gamma} \log \frac{32m}{\gamma^2} + \log \frac{4}{\delta}\right)$$

- where $d = \text{fat}_G(\gamma/8)$ is the fat shattering dimension of G with margin $\gamma/8$.
- We can see that the fat shattering dimension is behaving much as the VC dimension did in our error bounds

Fat Shattering using Linear Separators

- Let D be a probability distribution such that all points **x** drawn according to D satisfy the condition ||**x**|| $\leq$ R, so all points **x** lie within a sphere of radius R.
- Consider the functions defined by a unit weight vector:

  G = {g | g = **w** · **x** and ||**w**|| = 1}
- Then the fat shattering dimension of G is

$$\mathsf{fat}_G(\gamma) = \left(\frac{R}{\gamma}\right)^2$$

Noise-Free Margin Bound for Linear Separators

- By plugging this in, we find that the error rate of a linear classifier with unit weight vector and with margin γ on the training data (lying in a sphere of radius R) is

$$\epsilon <= \frac{2}{m}\left(\frac{64R^2}{\gamma^2}\log\frac{em\gamma}{8R^2}\log\frac{32m}{\gamma^2}+\log\frac{4}{\delta}\right)$$

- Ignoring all of the log terms, this says we should try to minimize

$$\frac{R^2}{m\gamma^2}$$

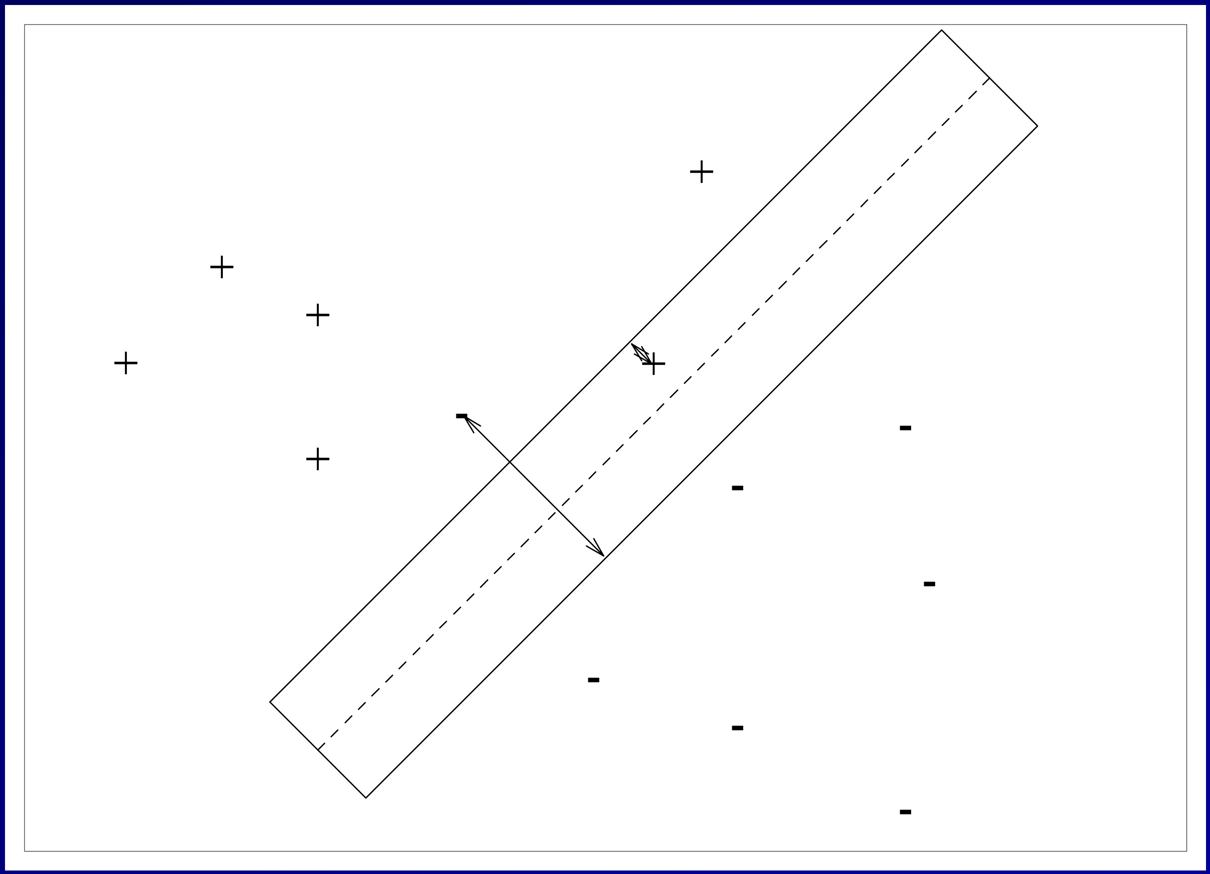
- R and *m* are fixed by the training set, so we should try to find a *g* that maximizes γ. This is the theoretical rationale for finding a maximum margin classifier.

Margin Bounds for Inconsistent Classifiers (soft margin classification)

- We can extend the margin analysis to the case when the data are not linearly separable (i.e., when a linear classifier is not consistent with the data). We will do this by measuring the margin on each training example
- Define $\xi_i = \max\{0, \gamma - y_i\, g(\mathbf{x}_i)\}$

  ξ_i is called the <u>margin slack variable</u> for example $\langle \mathbf{x}_i, y_i \rangle$
- Note that $\xi_i > \gamma$ implies that $\mathbf{x}_i$ is misclassified by *g*.
- Define $\xi = (\xi_1, \ldots, \xi_m)$ to be the <u>margin slack vector</u> for the classifier *g* on training set S

Soft Margin Classification (2)



$$\xi_i = \max\{0, \gamma - y_i\, g(\mathbf{x}_i)\}$$

Soft Margin Classification (3)

- Theorem. With probability $1 - \delta$, a linear separator with unit weight vector and margin γ on training data lying in a sphere of radius R will have an error rate on new data points bounded by

$$\epsilon <= \frac{C}{m}\left(\frac{R^2 + \|\xi\|^2}{\gamma^2}\log^2 m + \log\frac{1}{\delta}\right)$$

- for some constant C.
- This result tells us that we should
  - maximize γ
  - minimize $||\xi||^2$
  - but it doesn't tell us how to tradeoff among these two (because C may vary depending on γ and ξ)
- This will give us the full support vector machine

Statistical Learning Theory: Summary

- There is a 3-way tradeoff between ε, *m*, and the complexity of the hypothesis space H.
- The complexity of H can be measured by the VC dimension
- For a fixed hypothesis space, we should try to minimize training set error (empirical risk minimization)
- For a variable-sized hypothesis space, we should be willing to accept some training set errors in order to reduce the VC dimension of H_k (structural risk minimization)
- Margin theory shows that by changing γ, we continuously change the effective VC dimension of the hypothesis space. Large γ means small effective VC dimension (fat shattering dimension)
- Soft margin theory tells us that we should be willing to accept an increase in $||\xi||^2$ in order to get an increase in γ.
- We will be able to implement structural risk minimization within a single optimizer by having a dual objective function that tries to maximize γ while minimizing $||\xi||^2$