Measures of complexity

- "Complexity" is a measure of a set of classifiers, not any specific (fixed) classifier
- Many possible measures
  - degrees of freedom
  - description length
  - Vapnik-Chervonenkis dimension

  etc.
- There are many reasons for introducing a measure of complexity
  - generalization error guarantees
  - selection among competing families of classifiers

VC-dimension: preliminaries

- **A set of classifiers F:**
  For example, this could be the set of all possible linear separators, where $h \in F$ means that

$$h(\mathbf{x}) = \text{sign}\left(w_0 + \mathbf{w}^T \mathbf{x} \right)$$

  for some values of the parameters $\mathbf{w}, w_0$.

VC-dimension: preliminaries

- **Complexity:** how many different ways can we label n training points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with classifiers $h \in F$?

  In other words, how many distinct binary vectors

  $$[h(\mathbf{x}_1)\ h(\mathbf{x}_2)\ \ldots\ h(\mathbf{x}_n)]$$

  do we get by trying each $h \in F$ in turn?

  $$\begin{array}{llll} [& \text{-1} & 1 \ \ldots & 1 \ ] \quad h_1 \\ [& 1 & \text{-1} \ \ldots & 1 \ ] \quad h_2 \\ & & & \ldots \end{array}$$

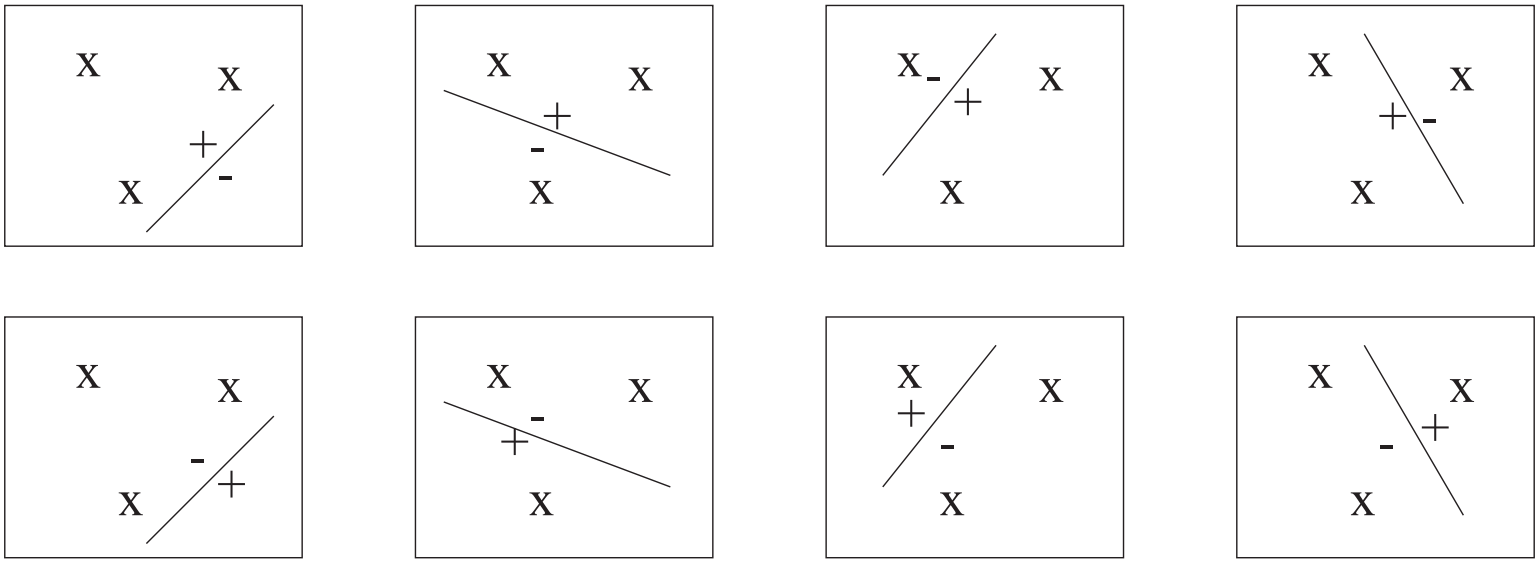
VC-dimension: shattering

- A set of classifiers F *shatters* n points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ if

$$[h(\mathbf{x}_1)\ h(\mathbf{x}_2)\ \ldots\ h(\mathbf{x}_n)],\quad h \in F$$

  generates all 2^n distinct labelings.

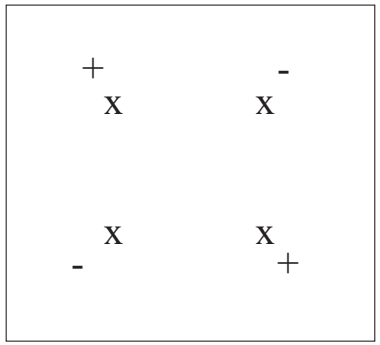
- Example: linear decision boundaries shatter (any) 3 points in 2D



  but not any 4 points...

VC-dimension: shattering cont'd

- We cannot shatter 4 points in 2D with linear separators
  For example, the following labeling



  cannot be produced with any linear separator

- More generally: the set of all d-dimensional linear separators can shatter exactly $d + 1$ points

VC-dimension

- *The VC-dimension d_{VC} of a set of classifiers F is the largest number of points that F can shatter*
- This is a combinatorial concept and doesn't depend on what type of classifier we use, only how "flexible" the set of classifiers is

  Example: Let F be a set of classifiers defined in terms of linear combinations of m **fixed** basis functions

  $$h(\mathbf{x}) = \text{sign}\,(\, w_0 + w_1\phi_1(\mathbf{x}) + \ldots + w_m\phi_m(\mathbf{x})\,)$$

  d_{VC} is at most $m+1$ regardless of the form of the fixed basis functions.

Learning and VC-dimension

- We learn something only after we no longer can shatter the training points (have more than d_{VC} training examples)

  Rationale: suppose we have n training examples and labels $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ and $n < d_{VC}$. Does the training set constrain our prediction for $\mathbf{x}_{n+1}$?

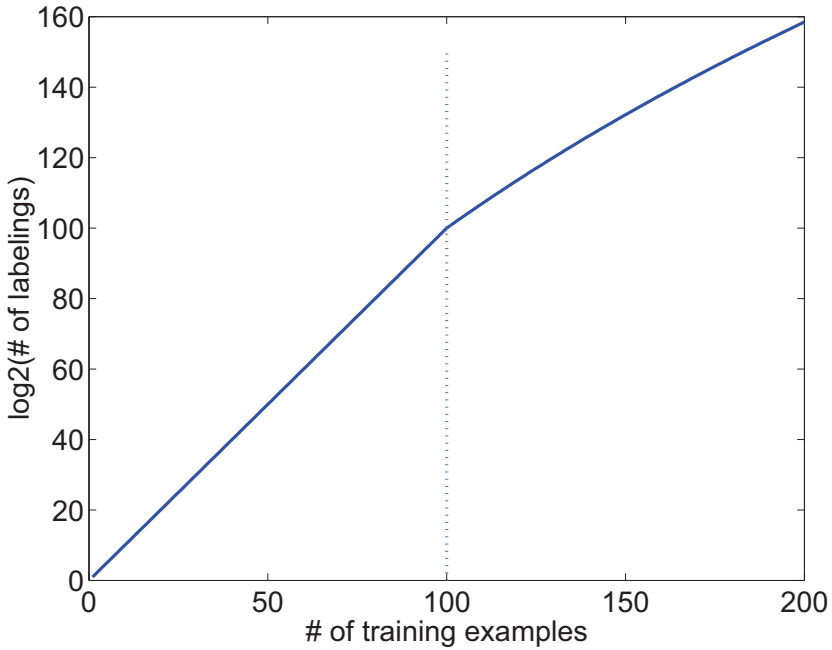
  Because we expect to be able to shatter $n+1$ points ($\leq d_{VC}$) it follows that we can find $h_1, h_2 \in F$, both consistent with training labels, but

$$h_1(\mathbf{x}_{n+1}) = 1, \quad h_2(\mathbf{x}_{n+1}) = -1$$

  We therefore cannot determine which label to predict for $\mathbf{x}_{n+1}$.

Learning and VC-dimension

- We don't really learn anything until after we have more than d_{VC} training examples



- The number of labelings that the set of classifiers can generate over n points increases sub-exponentially after $n > d_{VC}$ (in this case $d_{VC} = 100$)

Learning and VC-dimension

- When the VC-dimension is finite, the probability (over the choice of the training set) that we would find *any* $h \in F$ for which the difference

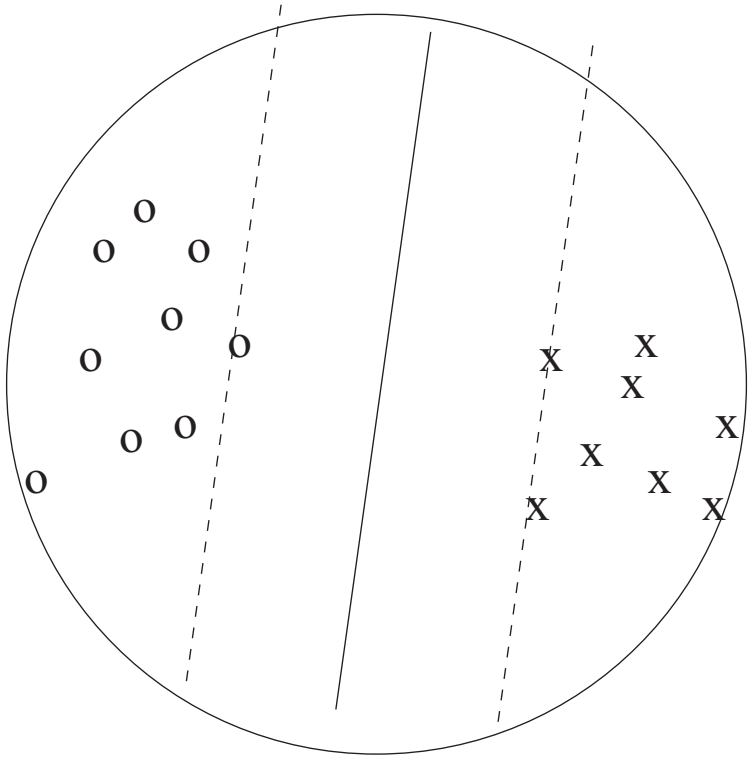
$$\left| \overbrace{\frac{1}{n} \sum_{i=1}^{n} \mathrm{Loss}(y_i, h(\mathbf{x}_i))}^{\text{Empirical loss}} - \overbrace{E\{\, \mathrm{Loss}(y, h(\mathbf{x}))\, \}}^{\text{Expected loss}} \right|$$

  is large goes down *exponentially* fast as a function of the size of the training set n. Here $\mathrm{Loss}(y, h(\mathbf{x})) = 1$ if $y \neq h(\mathbf{x})$ and zero otherwise (so called zero-one loss)

- This result holds for **any** underlying probability distribution from which the examples and the labels are generated

Extensions: complexity and margin

- The number of possible labelings of points with large margin can be dramatically less than the (basic) VC-dimension



- The set of separating hyperplaces which attain margin γ or better for examples within a sphere of radius R has VC-dimension bounded by $d_{VC}(\gamma) \leq R^2/\gamma^2$

Model selection

- We try to find the model with the best balance of complexity and the fit to the training data
- Ideally, we would select a model from a nested sequence of models of increasing complexity

  Model 1 d_1
  Model 2 d_2
  Model 3 d_3

  where $d_1 \leq d_2 \leq d_3 \leq \ldots$

- Basic model selection criterion:

$$\text{Criterion} = \text{(empirical) score} + \text{Complexity penalty}$$

Structural risk minimization

- In structural risk minimization we define the models in terms of VC-dimension (or refinements)

  Model 1 $d_{VC} = d_1$
  Model 2 $d_{VC} = d_2$
  Model 3 $d_{VC} = d_3$

  where $d_1 \leq d_2 \leq d_3 \leq \ldots$

- The selection criterion: lowest upper *bound* on the expected loss

  $$\text{Expected loss} \leq \text{Empirical loss} + \text{Complexity penalty}$$

Example

- Models of increasing complexity

  Model 1 $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))$
  Model 2 $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^2$
  Model 3 $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^3$
  ... ...

- These are nested, i.e.,

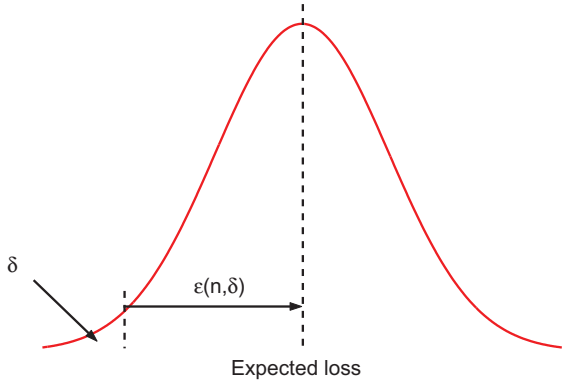
$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots$$

  where F_k refers to the set of possible decision boundaries that the model k can represent.

- Still need to derive the criterion...

Bounds on expected loss

- For simplicity, let's look at a single fixed classifier $h(\mathbf{x})$ and n training points



With probability at least $1-\delta$ over the choice of the training set

$$\overbrace{E\{\,\mathrm{Loss}(y,h(\mathbf{x}))\,\}}^{\text{Expected loss}} \;\leq\; \overbrace{\frac{1}{n}\sum_{i=1}^{n}\mathrm{Loss}(y_i,h(\mathbf{x}_i))}^{\text{Empirical loss}} + \overbrace{\epsilon(n,\delta)}^{\text{sampling penalty}}$$

- For the bound to be valid uniformly for all classifiers in the set F, we have to include the VC-dim

Structural risk minimization

- Finite VC-dimension gives us some guarantees about how close the empirical loss is to the expected loss
  With probability at least $1-\delta$ over the choice of the training set, for all $h \in F_k$

$$\overbrace{E\{\,\mathrm{Loss}(y, h(\mathbf{x}))\,\}}^{\text{Expected loss}} \leq \overbrace{\frac{1}{n}\sum_{i=1}^{n} \mathrm{Loss}(y_i, h(\mathbf{x}_i))}^{\text{Empirical loss}} + \overbrace{\epsilon(n, \delta, d_k)}^{\text{Complexity penalty}}$$

  where

$$\begin{aligned} d_k &= \text{VC-dimension of model (set of hypothesis) } k \\ \delta &= \text{Confidence parameter (probability of failure)} \end{aligned}$$

- We find model k that has the lowest bound on the expected loss

Structural risk minimization cont'd

- For our zero-one loss (classification error), we can derive the following complexity penalty (Vapnik 1995):

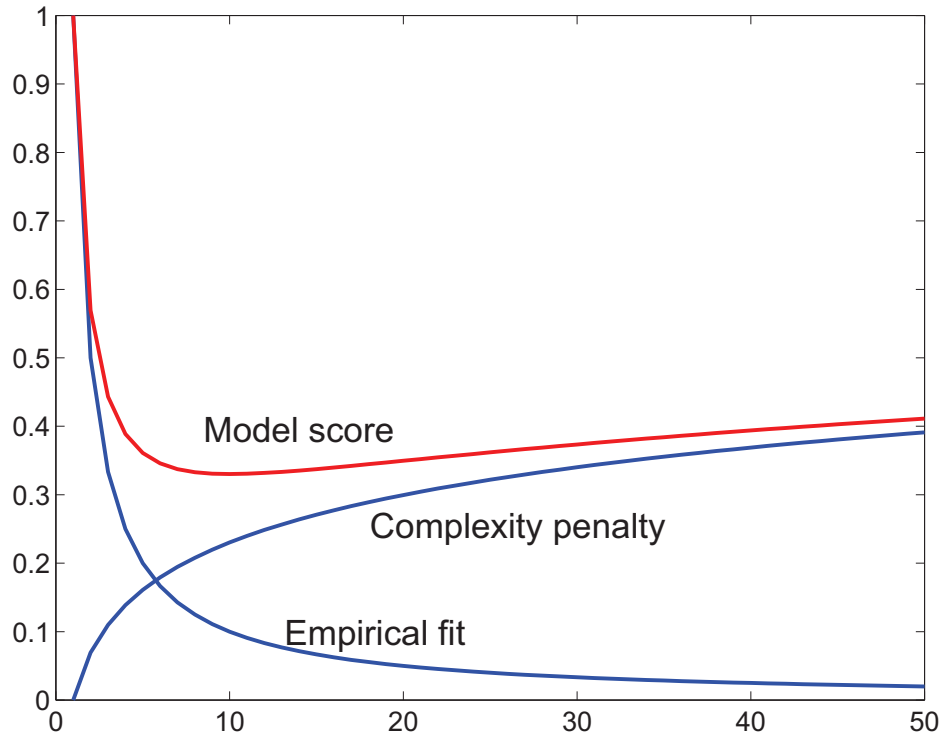
$$\epsilon(n, \delta, d) = \sqrt{\frac{d_{VC}(\log(2n/d_{VC}) + 1) + \log(1/(4\delta))}{n}}$$

1. This is an increasing function of d_{VC}
2. Increases as δ decreases
3. Decreases as a function of n

(this is not the only choice...)

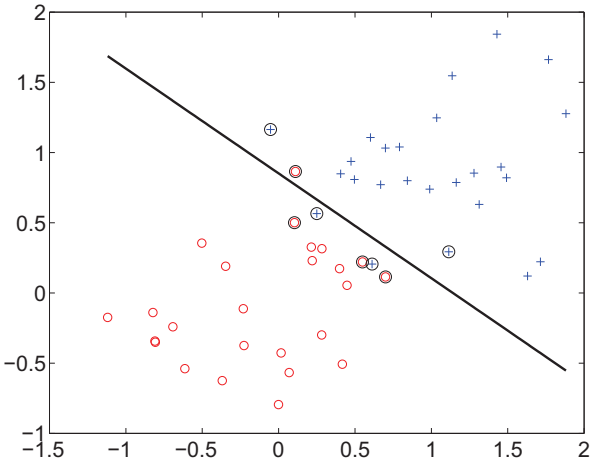
Structural risk minimization cont'd

- Competition of terms...
  1. Empirical loss decreases with increasing d_{VC}
  2. Complexity penalty increases with increasing d_{VC}

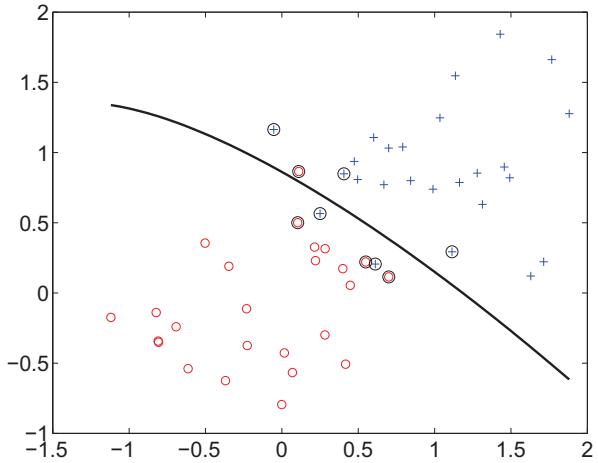


- We find the minimum of the model score (bound).

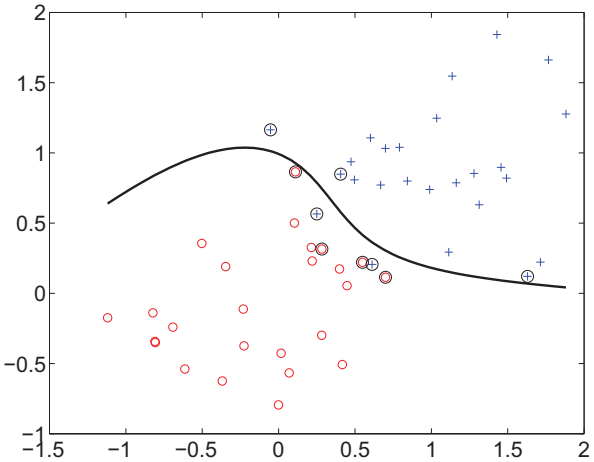
Structural risk minimization: example



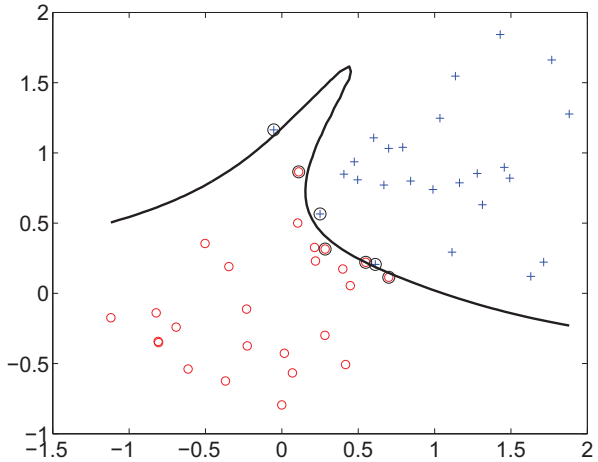
linear



2^{nd} order polynomial



4^{th} order polynomial



8^{th} order polynomial

Structural risk minimization: example cont'd

- Number of training examples $n = 50$, confidence parameter $\delta = 0.05$.

Model	d_{VC}	Empirical fit	Complexity penalty $\epsilon(n, \delta, d_{VC})$
1^{st} order	3	0.06	0.5501
2^{nd} order	6	0.06	0.6999
4^{th} order	15	0.04	0.9494
8^{th} order	45	0.02	1.2849

- Structural risk minimization would select the simplest (linear) model in this case.