Machine learning: lecture 11

Tommi S. Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Topics

- Complexity and model selection
  - learning and VC dimension
  - structural risk minimization
- Complexity, compression, and model selection
  - description length
  - minimum description length principle

VC-dimension: review

- The complexity of a set of classifiers depends on how many different ways we can label n training points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with classifiers $h \in F$

  In other words, this is the number of distinct binary vectors

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

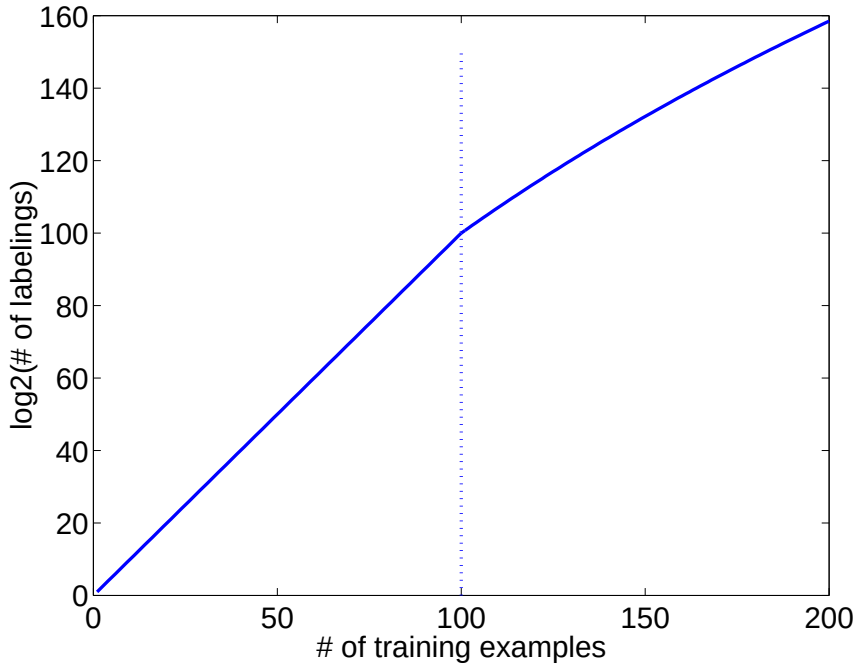
$$\begin{array}{llll} [& \text{-1} \quad \text{1} & \ldots & \text{1} \quad] \quad h_1 \\ [& \text{1} \quad \text{-1} & \ldots & \text{1} \quad] \quad h_2 \\ & & & \ldots \end{array}$$

  we get by trying out each $h \in F$ in turn. (the training points are chosen to maximize this number)

- VC-dimension is the largest number of points we can *shatter*, i.e., generate all possible labelings of the points

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

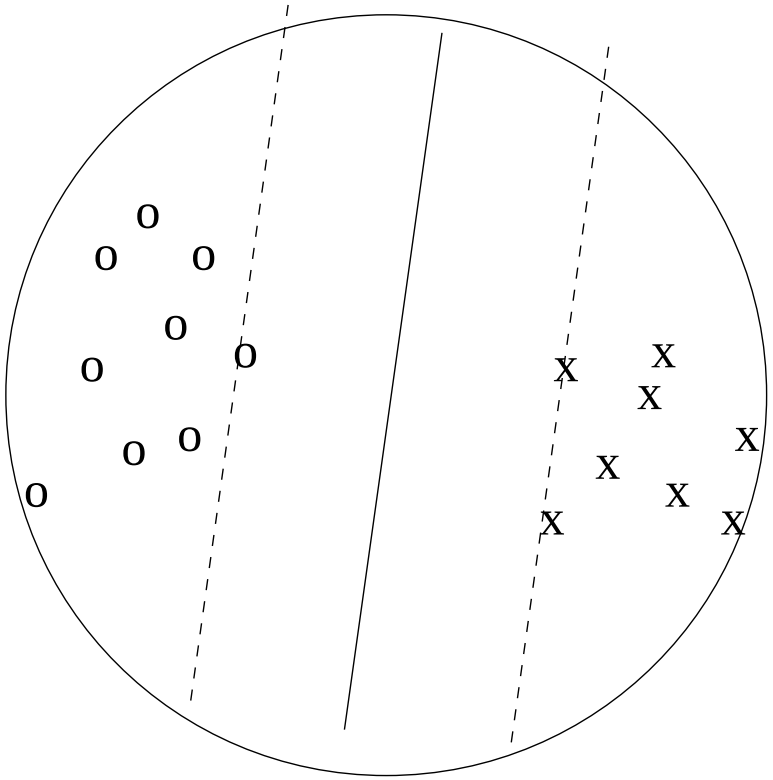
$$\left| \overbrace{\frac{1}{n} \sum_{i=1}^{n} \mathrm{Loss}(y_i, h(\mathbf{x}_i))}^{\text{Empirical loss}} - \overbrace{E\{ \mathrm{Loss}(y, h(\mathbf{x})) \}}^{\text{Expected loss}} \right|$$

  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.,

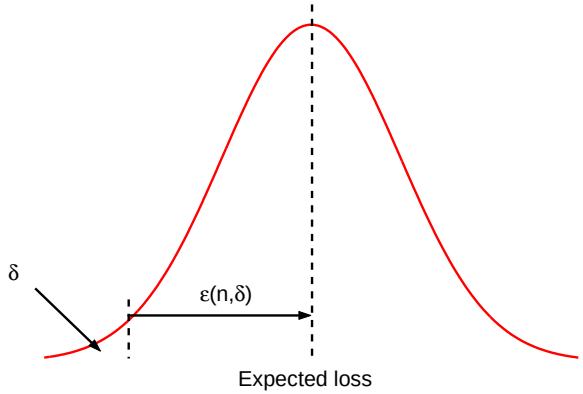
$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots$$

  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):

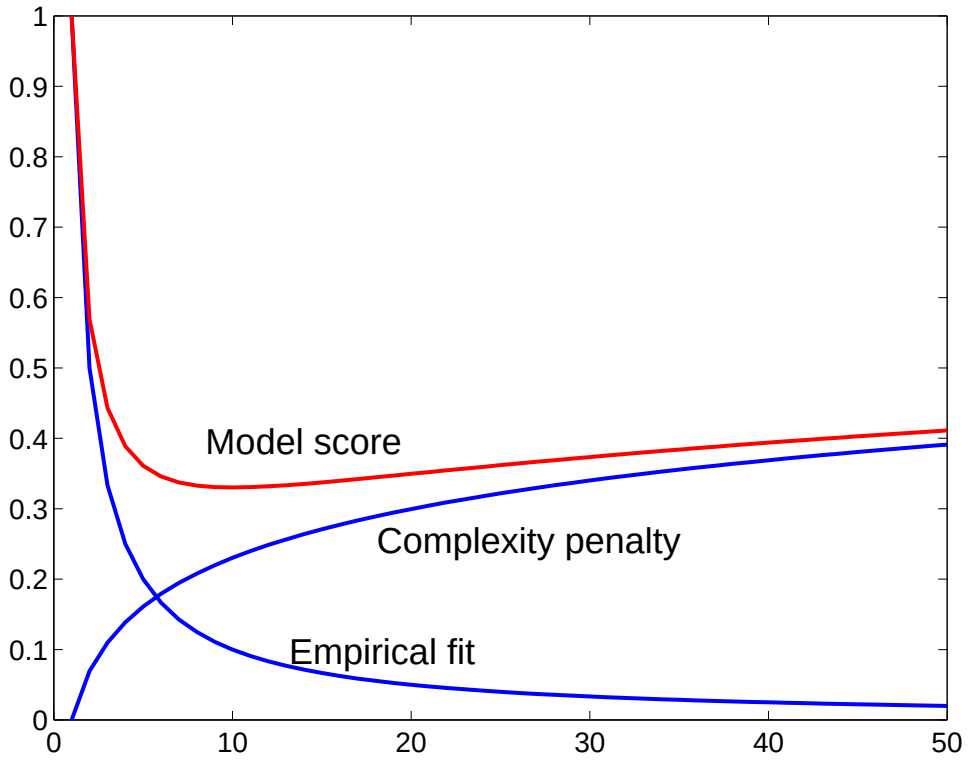
$$\epsilon(n, \delta, d) = \sqrt{\frac{d_{VC}(\log(2n/d_{VC}) + 1) + \log(1/(4\delta))}{n}}$$

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...)

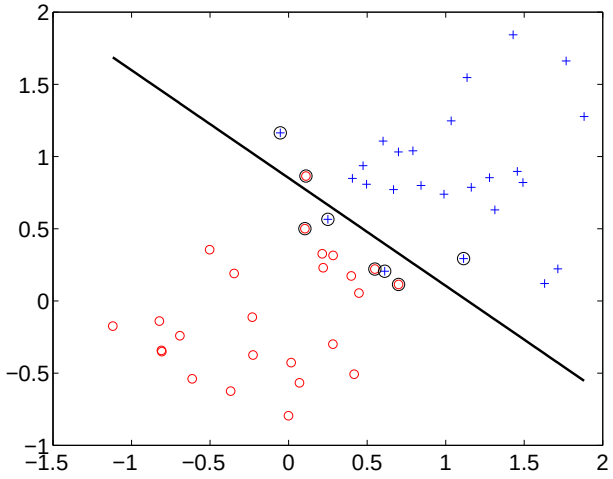
Structural risk minimization cont'd

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

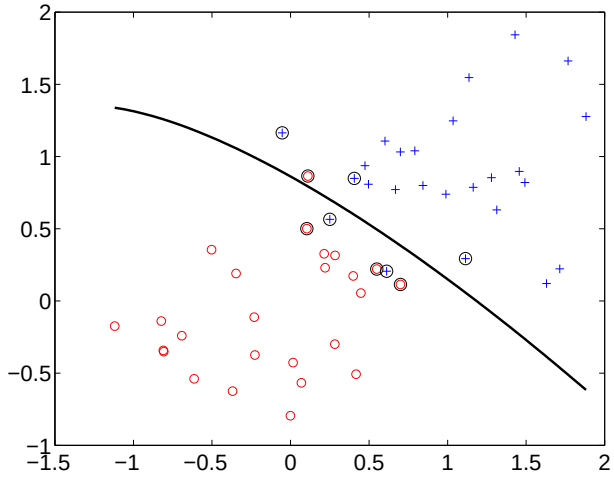


- We find the minimum of the model score (bound).

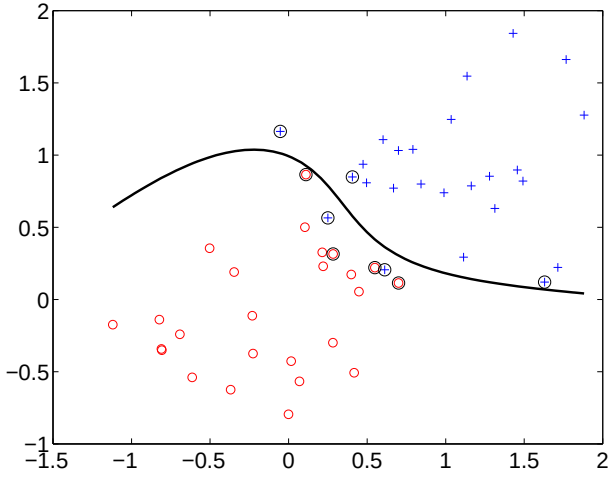
Structural risk minimization: example



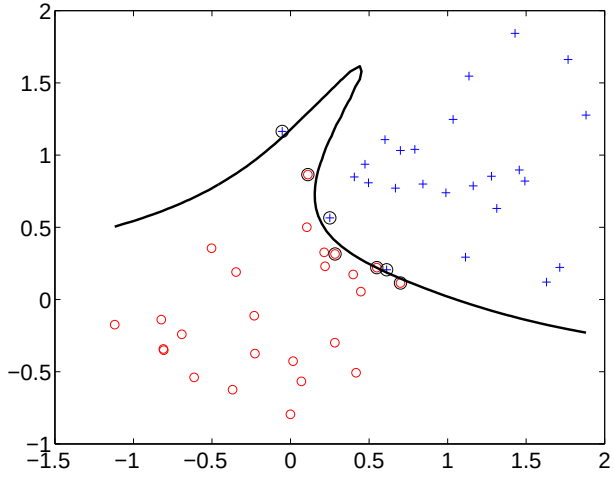
linear



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.

Topics

- Complexity, compression, and model selection
  - description length
  - minimum description length principle

Model selection and data compression

- We can alternatively view model selection as a problem of finding the best way of communicating the available data

  We have to communicate both the data and the method that we used to compress the data

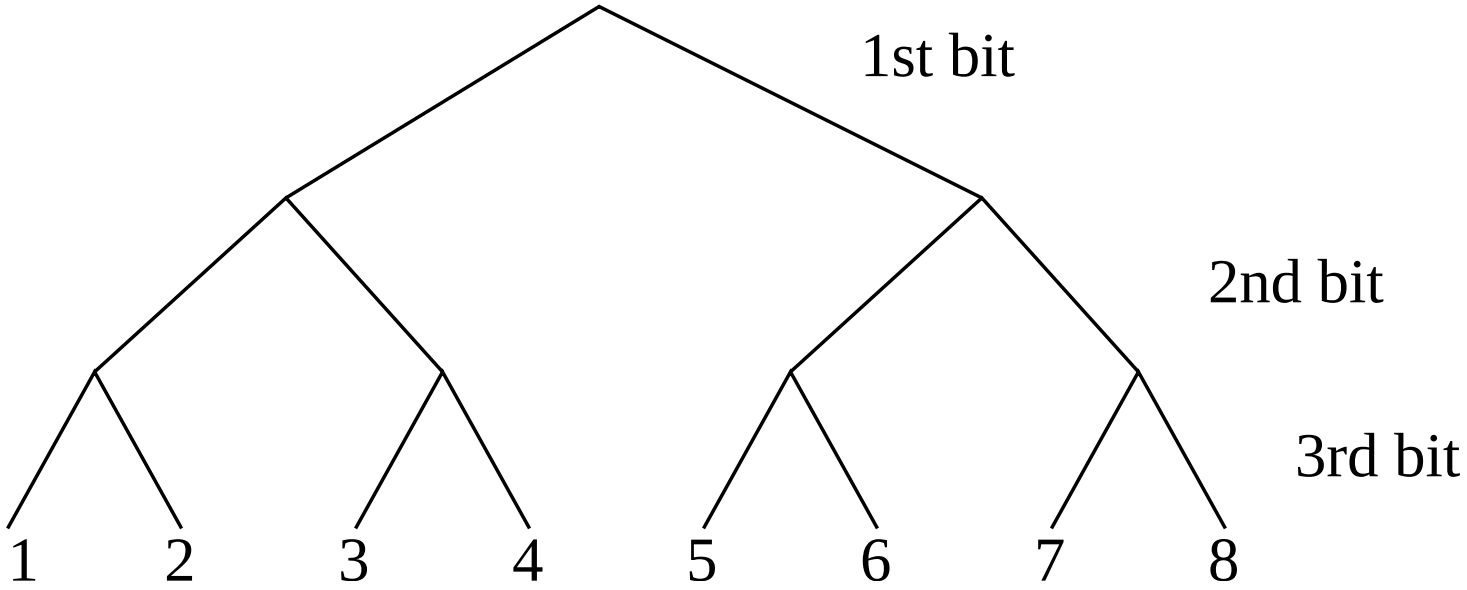
  The communication cost in bits depends on how well the model can predict the data as well as how hard it is to describe the model itself (complexity)

  $$\begin{aligned}\text{Total \# of bits} &= \text{bits to describe the data given the model} \\ &\quad + \text{bits to describe the model}\end{aligned}$$

Description length

- How many bits do we need to communicate in order to specify the outcome of a random variable with 8 possible values?

  Assume $P(y = 1) = \ldots = P(y = 8) = 1/8$



  We need $-\log_2 P(y) = -\log_2(1/8) = 3$ bits to describe each value y.

Description length cont'd

- How many bits do we need to describe

  0111111111110111001111111111111110111111

  If we assume that the bits $\{y_i\}$ in the sequence are independent random draws from P, where $P(y = 1) = 0.5$, then

$$\sum_{i=1}^{40} (-\log_2 P(y_i)) = 40\text{bits}$$

  If we assume instead that $P(y = 1) = 0.9$, then

$$\sum_{i=1}^{40} (-\log_2 P(y_i)) \approx 22\text{bits}$$

- What we assume matters a great deal.

Description length cont'd

- We can also describe outcomes conditionally, i.e., determine the number of bits we need to specify y given $\mathbf{x}$

$$\begin{array}{cccc} y_1 & y_2 & y_3 & y_4 & \dots \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \dots \end{array}$$

  Assuming the labels are generated from a conditional distribution $P(y|\mathbf{x}, \theta)$, we need

$$\sum_i (-\log_2 P(y_i|\mathbf{x}_i, \theta))$$

  bits to describe the outcomes (labels).

- The actual number of bits will vary considerably as a function of the parameters θ.

Description length cont'd

- We can of course find $\hat{\theta} \in \Theta$ (the maximum likelihood parameter estimate) that minimizes the number of bits needed to describe the data

$$\sum_i \left(-\log_2 P(y_i | \mathbf{x}_i, \hat{\theta}) \right)$$

  but the minimizing $\hat{\theta}$ depends on the data...

Description length cont'd

- In addition to describing the data using $\hat{\theta}$, which costs us

$$\sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \hat{\theta}) \right) \text{ bits,}$$

  we have to describe or communicate $\hat{\theta}$.

$$\texttt{total DL} = \texttt{DL of data using } \hat{\theta} + \texttt{DL of } \hat{\theta}$$

- The description length of the parameters $\hat{\theta}$ depends on the model (the set of distributions we are considering)
  - the more choices we have, the more bits it takes to describe any specific selection

How to describe the parameters

- We need to encode the parameters up to a finite precision $\delta_k = 1/2^k$, i.e., use k significant bits (we assume here that the precision is the same for all parameters)
- With the help of a prior density $p(\theta)$, it takes us roughly speaking

$$-\log_2 \left(p(\theta_{\delta_k}) \delta_k^d \right)$$

bits to describe any finite precision choice θ_{δ_k}. Here d is the dimensionality of the parameter vector θ.

How to describe the parameters cont'd

- We also need to communicate our choice of precision or k. This takes us

$$\log_2^*(k) \quad = \quad \log_2(k) + \log_2 \log_2(k) + \ldots$$

  bits.

Description length

- The total description length – bits needed to communicate the data – is given by the minimum of

$$\sum_i \left(-\log_2 P(y_i|\mathbf{x}_i, \theta_{\delta_k})\right) - \log_2 \left(p(\theta_{\delta_k})\delta_k^d\right) + \log_2^*(k)$$

  where the minimization is taken with respect to finite precision choices θ_{δ_k} as well as the number of significant bits k.