Machine learning: lecture 12

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

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

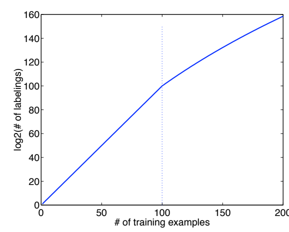
VC-dimension: review

Shattering: A set of classifiers F (e.g., linear classifiers) is said to shatter n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ if for any possible configuration of labels $y_1, \ldots, y_n$ we can find $h \in F$ that reproduces those labels.

VC-dimension: The VC-dimension of a set of classifiers F is the largest number of points that F can shatter (maximized over the choice of the n points).

Learning: We don't expect to learn anything until we have more than d_{VC} training examples and labels (this statement will be refined later on).

The number of labelings



$$n \le d_{VC}: \quad \text{\# of labelings } = 2^n$$

$$n > d_{VC}: \quad \text{\# of labelings } \le \left(\frac{en}{d_{VC}}\right)^{d_{VC}}$$

Learning and VC-dimension

- By essentially replacing $\log M$ in the finite case with the log of the number of possible labelings by the set of classifiers over n (really $2n$) points, we get an analogous result:

  Theorem: With probability at least $1-\delta$ over the choice of the training set, for all $h \in F$

$$\mathcal{E}(h) \le \hat{\mathcal{E}}_n(h) + \epsilon(n, d_{VC}, \delta)$$

where

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

Model selection

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

  Model 1, F_1 VC-dim $= d_1$
  Model 2, F_2 VC-dim $= d_2$
  Model 3, F_3 VC-dim $= d_3$

  where $F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots$
- **Model selection criterion:** find the model (set of classifiers) that achieves the lowest upper *bound* on the expected loss (generalization error):

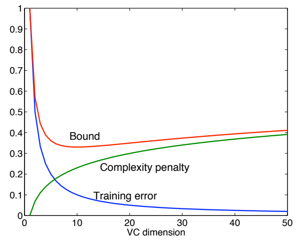
  Expected error $\le$ Training error $+$ Complexity penalty

Structural risk minimization

- We choose the model class F_i that minimizes the upper bound on the expected error:

$$\mathcal{E}(\hat{h}_i) \leq \hat{\mathcal{E}}_n(\hat{h}_i) + \sqrt{\frac{d_i(\log(2n/d_i) + 1) + \log(1/(4\delta))}{n}}$$

where $\hat{h}_i$ is the classifier from F_i that minimizes the training error.



Example

- Models of increasing complexity

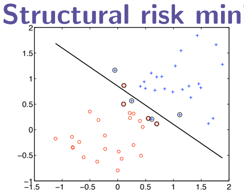
Model 1	$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))$
Model 2	$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^2$
Model 3	$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^3$
...	...

- These are nested, i.e.,

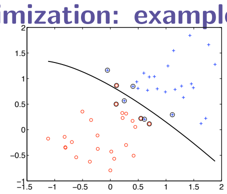
$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots$$

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

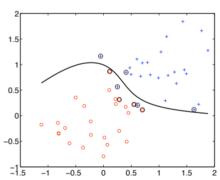
Structural risk minimization: example



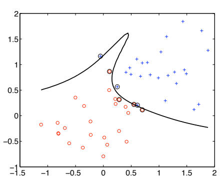
linear



2^{nd} order polynomial



4^{th} order polynomial



8^{th} order polynomial

Structural risk minimization: example cont'd

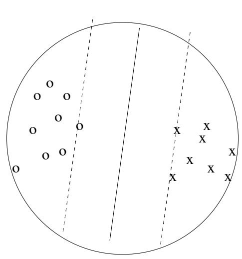
- Number of training examples $n = 50$, confidence parameter $\delta = 0.05$.

Model	d_{VC}	Empirical fit	$\epsilon(n, d_{VC}, \delta)$
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.

Complexity and margin

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



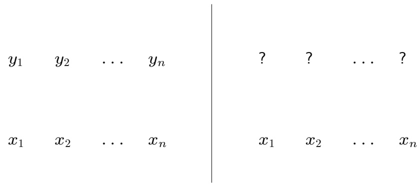
- The set of separating hyperplanes which attain margin γ or better for examples within a sphere of radius R has VC-dimension bounded by $d_{VC}(\gamma) \leq R^2/\gamma^2$

Topics

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

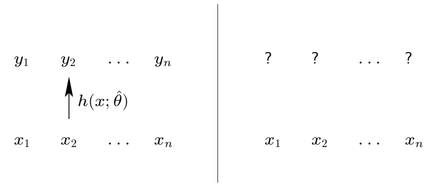
Data compression and model selection

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



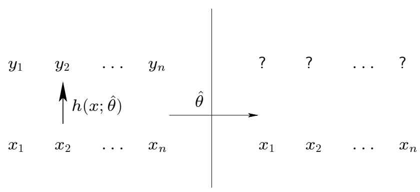
Data compression and model selection

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



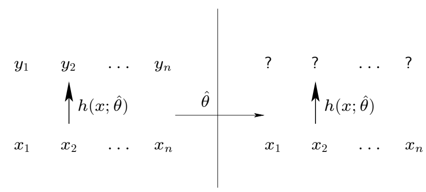
Data compression and model selection

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



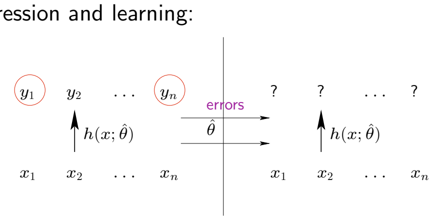
Data compression and model selection

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



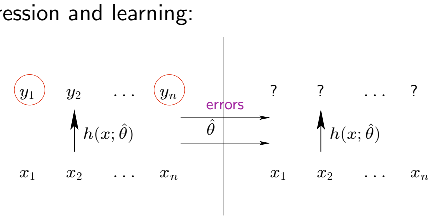
Data compression and model selection

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



Data compression and model selection

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



The receiver already knows
– input examples, models we consider

Need to communicate
– model class, parameter estimates, prediction errors

Compression and sequential estimation

- We don't have to communicate any real valued parameters if we setup the learning problem sequentially

$y_1 \quad y_2 \quad \ldots \quad y_n$	$? \quad ? \quad \ldots \quad ?$
$x_1 \quad x_2 \quad \ldots \quad x_n$	$x_1 \quad x_2 \quad \ldots \quad x_n$

Compression and sequential estimation

- We don't have to communicate any real valued parameters if we setup the learning problem sequentially

$y_1 \quad y_2 \quad \ldots \quad y_n$	$? \quad ? \quad \ldots \quad ?$
$\uparrow h(x;\theta_0)$	
$x_1 \quad x_2 \quad \ldots \quad x_n$	$x_1 \quad x_2 \quad \ldots \quad x_n$

θ_0 : default parameter values

Compression and sequential estimation

- We don't have to communicate any real valued parameters if we setup the learning problem sequentially

$y_1 \quad y_2 \quad \ldots \quad y_n$	$? \quad ? \quad \ldots \quad ?$
$\uparrow h(x;\hat{\theta}_1)$	
$x_1 \quad x_2 \quad \ldots \quad x_n$	$x_1 \quad x_2 \quad \ldots \quad x_n$

θ_0 : default parameter values
$\hat{\theta}_1$: based on θ_0 and (x_1, y_1)

Compression and sequential estimation

- We don't have to communicate any real valued parameters if we setup the learning problem sequentially

$y_1 \quad y_2 \quad \ldots \quad y_n$	$? \quad ? \quad \ldots \quad ?$
$\uparrow h(x;\hat{\theta}_1)$	
$x_1 \quad x_2 \quad \ldots \quad x_n$	$x_1 \quad x_2 \quad \ldots \quad x_n$

θ_0 : default parameter values
$\hat{\theta}_1$: based on θ_0 and (x_1, y_1)

Compression and sequential estimation

- We don't have to communicate any real valued parameters if we setup the learning problem sequentially

$y_1 \quad y_2 \quad \ldots \quad y_n$	$? \quad ? \quad \ldots \quad ?$
$\uparrow h(x;\hat{\theta}_{n-1})$	
$x_1 \quad x_2 \quad \ldots \quad x_n$	$x_1 \quad x_2 \quad \ldots \quad x_n$

θ_0 : default parameter values
$\hat{\theta}_1$: based on θ_0 and (x_1, y_1)
...
$\hat{\theta}_{n-1}$: based on θ_0 and $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$

Compression and sequential estimation

- We don't have to communicate any real valued parameters if we setup the learning problem sequentially

$y_1 \quad y_2 \quad \ldots \quad y_n$ $\xrightarrow{\text{errors}}$	$? \quad ? \quad \ldots \quad ?$
$\uparrow h(x;\hat{\theta}_{n-1})$	
$x_1 \quad x_2 \quad \ldots \quad x_n$	$x_1 \quad x_2 \quad \ldots \quad x_n$

θ_0 : default parameter values
$\hat{\theta}_1$: based on θ_0 and (x_1, y_1)
...
$\hat{\theta}_{n-1}$: based on θ_0 and $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$

  - we only need to communicate the model class (index) and prediction errors
  - but the answer depends on the sequential order

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms
- Suppose we define a model $P(y|x,\theta), \theta \in \Theta$ and prior $P(\theta)$, both known to the receiver

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms
- Suppose we define a model $P(y|x,\theta), \theta \in \Theta$ and prior $P(\theta)$, both known to the receiver

  We predict the first label according to

$$y_1|x_1 \quad : \quad P(y_1|x_1) = \int P(y_1|x_1,\theta)P(\theta)d\theta$$

  and update the prior (posterior)

$$P(\theta|D_1) = \frac{P(\theta)P(y_1|x_1,\theta)}{P(y_1|x_1)}$$

  where $D_1 = \{(x_1,y_1)\}$.

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms
- Suppose we define a model $P(y|x,\theta), \theta \in \Theta$ and prior $P(\theta)$, both known to the receiver

  We predict the second label according to

$$y_2|x_2 \quad : \quad P(y_2|x_2,D_1) = \int P(y_2|x_2,\theta)P(\theta|D_1)d\theta$$

  and again update the posterior

$$P(\theta|D_2) = \frac{P(\theta|D_1)P(y_2|x_2,\theta)}{P(y_2|x_2,D_1)}$$

  where $D_2 = \{(x_1,y_1),(x_2,y_2)\}$.

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms
- Suppose we define a model $P(y|x,\theta), \theta \in \Theta$ and prior $P(\theta)$, both known to the receiver

  Finally, we predict the last n^{th} label according to

$$y_n|x_n \quad : \quad P(y_n|x_n,D_{n-1}) = \int P(y_n|x_n,\theta)P(\theta|D_{n-1})d\theta$$

  where $D_{n-1} = \{(x_1,y_1),\ldots,(x_{n-1},y_{n-1})\}$.

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms
- Suppose we define a model $P(y|x,\theta), \theta \in \Theta$ and prior $P(\theta)$, both known to the receiver

  Our sequential prediction method defines a probability distribution over all the labels given the examples:

$$P(y_1|x_1)P(y_2|x_2,D_1)\cdots P(y_n|x_n,D_{n-1})$$

  This *does not* depend on the order in which we processed the examples.

Probabilistic sequential prediction

- To communicate the labels effectively we need to cast the problem in probabilistic terms
- Suppose we define a model $P(y|x,\theta), \theta \in \Theta$ and prior $P(\theta)$, both known to the receiver

  Our sequential prediction method defines a probability distribution over all the labels given the examples:

  $$P(y_1|x_1)P(y_2|x_2, D_1)\cdots P(y_n|x_n, D_{n-1})$$

  This *does not* depend on the order in which we processed the examples.

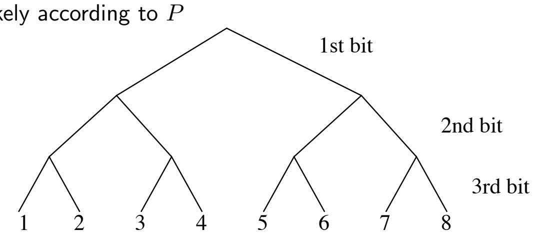
  $$= \int P(y_1|x_1,\theta)\cdots P(y_n|x_n,\theta)P(\theta)d\theta$$

  (Bayesian marginal likelihood)

Description length and probabilities

- It takes $-\log_2 P(y_1,\ldots,y_n)$ bits to communicate $y_1,\ldots,y_n$ according to distribution P.

  Example: suppose $y = 1,\ldots,8$ and each value is equally likely according to P



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