Statistical Machine Learning from Data
Statistical Learning Theory

Samy Bengio

IDIAP Research Institute, Martigny, Switzerland, and
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
bengio@idiap.ch
http://www.idiap.ch/~bengio

February 9, 2006
1. Data, Functions, Risk
2. The Capacity
3. Methodology
1. Data, Functions, Risk

2. The Capacity

3. Methodology
Available training data
- Let $Z_1, Z_2, \cdots, Z_n$ be an $n$-tuple random sample of an unknown distribution of density $p(z)$.
- All $Z_i$ are independently and identically distributed (iid).
- Let $D_n$ be a particular instance $= \{z_1, z_2, \cdots, z_n\}$.

Various forms of the data
- **Classification**: $Z = (X, Y) \in \mathbb{R}^d \times \{-1, 1\}$
  objective: given a new $x$, estimate $P(Y|X = x)$
- **Regression**: $Z = (X, Y) \in \mathbb{R}^d \times \mathbb{R}$
  objective: given a new $x$, estimate $E[Y|X = x]$  
- **Density estimation**: $Z \in \mathbb{R}^d$
  objective: given a new $z$, estimate $p(z)$
Learning: search for a good function in a function space $\mathcal{F}$

Examples of functions $f(\cdot; \theta) \in \mathcal{F}$:

- **Regression**:
  \[
  \hat{y} = f(x; a, b) = a \cdot x + b
  \]

- **Classification**:
  \[
  \hat{y} = f(x; a, b) = \text{sign}(a \cdot x + b)
  \]

- **Density estimation**
  \[
  \hat{p}(z) = f(z; \mu, \Sigma) = \frac{1}{(2\pi)^{|z|/2} \sqrt{|\Sigma|}} \exp \left( -\frac{1}{2} (z - \mu)^T \Sigma^{-1} (z - \mu) \right)
  \]
The Loss Function

Learning: search for a good function in a function space $\mathcal{F}$

Examples of loss functions $L : \mathcal{Z} \times \mathcal{F}$

- **Regression:**
  \[
  L(z, f) = L((x, y), f) = (f(x) - y)^2
  \]

- **Classification:**
  \[
  L(z, f) = L((x, y), f) = \begin{cases} 
  0 & \text{if } f(x) = y \\
  1 & \text{otherwise} 
  \end{cases}
  \]

- **Density estimation:**
  \[
  L(z, f) = - \log p(z)
  \]
The Risk and the Empirical Risk

Learning: search for a good function in a function space $\mathcal{F}$

- Minimize the Expected Risk on $\mathcal{F}$, defined for a given $f$ as
  \[ R(f) = \mathbb{E}_Z[L(z, f)] = \int_Z L(z, f) p(z) dz \]

- Induction Principle:
  - select $f^* = \arg\min_{f \in \mathcal{F}} R(f)$
  - problems: $p(z)$ is unknown, and we don’t have access to all $L(z, f)$!!

- Empirical Risk:
  \[ \hat{R}(f, D_n) = \frac{1}{n} \sum_{i=1}^n L(z_i, f) \]
The empirical risk is an unbiased estimate of the risk.

\[
E \left[ \hat{R}(f, D) \right] = E \left[ \frac{1}{n} \sum_{i=1}^{n} L(f, Z_i) \right]
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} E \left[ L(f, Z_i) \right], \quad Z_i \text{s are independent}
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} E \left[ L(f, Z) \right], \quad Z_i \text{s are identically distributed}
\]

\[
= \frac{1}{n} n E \left[ L(f, Z) \right]
\]

\[
= R(f)
\]
The empirical risk is an \textit{unbiased} estimate of the risk

\[
E \left[ \hat{R}(f, D) \right] = E \left[ \frac{1}{n} \sum_{i=1}^{n} L(f, Z_i) \right] \\
= \frac{1}{n} \sum_{i=1}^{n} E [L(f, Z_i)] , \quad Z_i \text{s are independent} \\
= \frac{1}{n} \sum_{i=1}^{n} E [L(f, Z)] , \quad Z_i \text{s are identically distributed} \\
= \frac{1}{n} n E [L(f, Z)] \\
= R(f)
\]
The empirical risk is an unbiased estimate of the risk.

\[
E \left[ \hat{R}(f, D) \right] = E \left[ \frac{1}{n} \sum_{i=1}^{n} L(f, Z_i) \right] \\
= \frac{1}{n} \sum_{i=1}^{n} E \left[ L(f, Z_i) \right], \quad Z_is \ are \ independent \\
= \frac{1}{n} \sum_{i=1}^{n} E \left[ L(f, Z) \right], \quad Z_is \ are \ identically \ distributed \\
= \frac{1}{n} n E \left[ L(f, Z) \right] \\
= R(f)
\]
The empirical risk is an unbiased estimate of the risk

\[ E \left[ \hat{R}(f, D) \right] = E \left[ \frac{1}{n} \sum_{i=1}^{n} L(f, Z_i) \right] \]

\[ = \frac{1}{n} \sum_{i=1}^{n} E [L(f, Z_i)] , \quad Z_i s \text{ are independent} \]

\[ = \frac{1}{n} \sum_{i=1}^{n} E [L(f, Z)] , \quad Z_i s \text{ are identically distributed} \]

\[ = \frac{1}{n} nE [L(f, Z)] \]

\[ = R(f) \]
The empirical risk is an **unbiased** estimate of the risk.

\[
E \left[ \hat{R}(f, D) \right] = E \left[ \frac{1}{n} \sum_{i=1}^{n} L(f, Z_i) \right] \\
= \frac{1}{n} \sum_{i=1}^{n} E \left[ L(f, Z_i) \right], \quad Z_i \text{ are independent} \\
= \frac{1}{n} \sum_{i=1}^{n} E \left[ L(f, Z) \right], \quad Z_i \text{ are identically distributed} \\
= \frac{1}{n} n E \left[ L(f, Z) \right] \\
= \frac{1}{n} n E \left[ L(f, Z) \right] \\
= R(f)
\]
The empirical risk is an **unbiased** estimate of the risk

\[
E \left[ \hat{R}(f, D) \right] = E \left[ \frac{1}{n} \sum_{i=1}^{n} L(f, Z_i) \right]
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} E [L(f, Z_i)], \quad Z_i \text{s are independent}
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} E [L(f, Z)], \quad Z_i \text{s are identically distributed}
\]

\[
= \frac{1}{n} n E [L(f, Z)]
\]

\[
= R(f)
\]
Empirical Risk Minimization

The principle of empirical risk minimization (ERM):

\[ f^*(D_n) = \arg \min_{f \in \mathcal{F}} \hat{R}(f, D_n) \]

Consistency of the principle of ERM

Let \( f^* = \arg \min_{f \in \mathcal{F}} R(f) \) and \( f^*(D_n) = \arg \min_{f \in \mathcal{F}} \hat{R}(f, D_n) \)

We say that the principle of ERM is consistent for \( \mathcal{F} \) if

\[ R(f^*(D_n)) \xrightarrow{P} R(f^*) \quad \text{as} \quad n \to \infty \]

and

\[ \hat{R}(f^*(D_n), D_n) \xrightarrow{P} R(f^*) \quad \text{as} \quad n \to \infty \]
Consistency of the ERM

\[ \inf_{f} R(f) \]

Expected Risk

Empirical Risk
The Risk and the Training Error

- Training error:

\[ \hat{R}(f^*(D_n), D_n) = \min_{f \in \mathcal{F}} \hat{R}(f, D_n) \]

- Is the training error a biased estimate of the risk? YES.

\[ E[R(f^*(D_n)) - \hat{R}(f^*(D_n), D_n)] \geq 0 \]

- The solution \( f^*(D_n) \) found by minimizing the training error is better on \( D_n \) than on any other set \( D'_n \) drawn from \( p(z) \).
Bounding the Risk

Can we bound the difference between the training error and the generalization error?

\[ |R(f^*(D_n)) - \hat{R}(f^*(D_n), D_n)| \leq ? \]

- Answer: under certain conditions on \( F \), yes.
- These conditions depend on the notion of capacity \( h \) of \( F \).
1 Data, Functions, Risk

2 The Capacity

3 Methodology
The Capacity

- The capacity $h(\mathcal{F})$ is a measure of its size, or complexity.

- Classification:

  The capacity $h(\mathcal{F})$ is the largest $n$ such that there exist a set of examples $D_n$ such that one can always find an $f \in \mathcal{F}$ which gives the correct answer for all examples in $D_n$, for any possible labeling.

- Example: for the set of linear functions ($y = w \cdot x + b$) in $d$ dimensions, the capacity is $d + 1$.

- Regression and density estimation: capacity exists also, but more complex to derive (for instance, we can always reduce a regression problem to a classification problem).
Bounding the Risk

Bound on the expected risk:

- Let \( \tau = \sup L - \inf L \).
- \( \forall \eta \) we have

\[
P \left( \sup_{f \in \mathcal{F}} |R(f) - \hat{R}(f, D_n)| \leq 2\tau \sqrt{\frac{h \left( \ln \frac{2n}{h} + 1 \right) - \ln \frac{\eta}{9}}{n}} \right) \geq 1 - \eta
\]

- With \( h \) the capacity of \( \mathcal{F} \) and \( n \) the number of training examples in \( D_n \).
Structural Risk Minimization - Fixed $n$

Bound on the Expected Risk

Confidence Interval

Empirical Risk

$h$
Consistency - Fixed $h$

$\inf_{f} R(f)$

Expected Risk

Empirical Risk

$n$
The generalization error can be decomposed into 3 parts:

- **the bias**: due to the fact that the set of functions $\mathcal{F}$ does not contain the optimal solution,

- **the variance**: due to the fact that if we had been using another set $D'_n$ drawn from the same distribution $p(Z)$, we would have obtained a different solution,

- **the noise**: even the optimal solution could be wrong! (for instance if for a given $x$ there are more than one possible $y$)
The Bias-Variance Dilemma (Graphical View)

- Variance
- Bias
- Noise
- Optimal solution
- Set of functions
- Solution obtained with training set 1
- Solution obtained with training set 2
- The size of the set of functions depends on the size of the training set.
Intrinsic dilemma: when the capacity $h(\mathcal{F})$ grows, the bias goes down, but the variance goes up!

Bias–Variance Dilemma
We have seen that learning = searching in a set of functions

This set should not be too small (underfitting)

This set should not be too large (overfitting)

One solution: regularization

Penalize functions $f$ according to a prior knowledge

For instance, penalize functions that have large parameters

$$f^*(D_n) = \arg\min_{f \in F} \hat{R}(f, D_n) + H(f)$$

with $H(f)$ a function that penalizes according to your prior

For example, in some models:
small parameters $\rightarrow$ simpler solutions $\rightarrow$ less capacity
Early Stopping

- Another method for regularization: **early stopping**.
- Works when training is an **iterative process**.
- Instead of selecting the function that minimizes the empirical risk on $D_n$, we can do:
  - divide your training set $D_n$ into two parts
    - **train set** $D^{tr} = \{z_1, z_2, \ldots, z_{tr}\}$
    - **validation set** $D^{va} = \{z_{va+1}, z_{tr+2}, \ldots, z_{tr+va}\}$
    - $tr + va = n$
  - let $f^t(D^{tr})$ be the current function found at iteration $t$
  - let $\hat{R}(f^t(D^{tr}), D^{va}) = \frac{1}{va} \sum_{z_i \in D^{va}} L(z_i, f^t(D^{tr}))$
  - stop training at iteration $t^*$ such that
    $$t^* = \arg \min_t \hat{R}(f^t(D^{tr}), D^{va})$$
  - and return function $f(D_n) = f^{t^*}(D^{tr})$
1. Data, Functions, Risk
2. The Capacity
3. Methodology
First: **identify the goal!** It could be

1. to give the best model you can obtain given a training set?
2. to give the expected performance of a model obtained by empirical risk minimization given a training set?
3. to give the best model and its expected performance that you can obtain given a training set?

- If the goal is (1): use need to do **model selection**
- If the goal is (2), you need to estimate the **risk**
- If the goal is (3): use need to do both!

There are various methods that can be used for either risk estimation or model selection:

- **simple validation**
- **cross validation** (k-fold, leave-one-out)
Select a family of functions with hyper-parameter $\theta$

Divide your training set $D_n$ into two parts

- $D^{tr} = \{z_1, z_2, \cdots, z_{tr}\}$
- $D^{va} = \{z_{tr+1}, z_{tr+2}, \cdots, z_{tr+va}\}$
- $tr + va = n$

For each value $\theta_m$ of the hyper-parameter $\theta$

- select $f_{\theta_m}^* (D^{tr}) = \arg \min_{f \in F_{\theta_m}} \hat{R}(f, D^{tr})$

- estimate $R(f_{\theta_m}^*)$ with $\hat{R}(f_{\theta_m}^*, D^{va}) = \frac{1}{va} \sum_{z_i \in D^{va}} L(z_i, f_{\theta_m}^*(D^{tr}))$

- select $\theta_m^* = \arg \min_{\theta_m} R(f_{\theta_m}^*)$

- return $f^*(D_n) = \arg \min_{f \in F_{\theta_m^*}} \hat{R}(f, D_n)$
Model Selection - Cross-validation

- Select a family of functions with **hyper-parameter** $\theta$
- Divide your training set $D_n$ into $K$ distinct and equal parts $D^1, \ldots, D^K$
- For each value $\theta_m$ of the hyper-parameter $\theta$
  - For each part $D^k$ (and its counterpart $\bar{D}^k$)
    - select $f_{\theta_m}^*(\bar{D}^k) = \arg\min_{f \in \mathcal{F}_{\theta_m}} \hat{R}(f, \bar{D}^k)$
    - estimate $\hat{R}(f_{\theta_m}^*(\bar{D}^k))$ with
      $$\hat{R}(f_{\theta_m}^*(\bar{D}^k), D^k) = \frac{1}{|D^k|} \sum_{z_i \in D^k} L(z_i, f_{\theta_m}^*(\bar{D}^k))$$
  - estimate $R(f_{\theta_m}^*(D_n))$ with
    $$\frac{1}{K} \sum_k R(f_{\theta_m}^*(\bar{D}^k))$$
- select $\theta_m^* = \arg\min_{\theta_m} R(f_{\theta_m}^*(D))$
- return $f^*(D_n) = \arg\min_{f \in \mathcal{F}_{\theta_m}^*} \hat{R}(f, D_n)$
Estimation of the Risk - Validation

- Divide your training set $D_n$ into two parts
  - $D^{tr} = \{z_1, z_2, \ldots, z_{tr}\}$
  - $D^{te} = \{z_{tr+1}, z_{tr+2}, \ldots, z_{tr+te}\}$
  - $tr + te = n$
- select $f^*(D^{tr}) = \arg \min_{f \in F} \hat{R}(f, D^{tr})$
  
  (this optimization process could include model selection)
- estimate $R(f^*(D^{tr}))$ with
  $$\hat{R}(f^*(D^{tr}), D^{te}) = \frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*(D^{tr}))$$
Estimation of the Risk - Cross-validation

- Divide your training set $D_n$ into $K$ distinct and equal parts $D^1, \ldots, D^K$.
- For each part $D^k$
  - let $\bar{D}^k$ be the set of examples that are in $D_n$ but not in $D^k$.
  - select $f^*(\bar{D}^k) = \arg\min_{f \in \mathcal{F}} \hat{R}(f, \bar{D}^k)$

  *(this process could include model selection)*

- estimate $R(f^*(\bar{D}^k))$ with
  $$\hat{R}(f^*(\bar{D}^k), D^k) = \frac{1}{|D^k|} \sum_{z_i \in D^k} L(z_i, f^*(\bar{D}^k))$$

- estimate $R(f^*(D_n))$ with
  $$\frac{1}{K} \sum_k R(f^*(\bar{D}^k))$$

- When $k = n$: leave-one-out cross-validation.
Estimation of the Risk and Model Selection

- When you want both the best model and its expected risk.
- You then need to **merge** the methods already presented. For instance:
  - train-validation-test: 3 separate data sets are necessary
  - cross-validation + test: cross-validate on train set, then test on separate set
  - double-cross-validation: for each subset, need to do a second cross-validation with the $K - 1$ other subsets
- Other important methodological aspects:
  - **compare** your results with other methods!!!!
  - use statistical tests to **verify significance**
  - verify your model on more than one datasets
Train - Validation - Test

- Select a family of functions with hyper-parameter $\theta$
- Divide your training set $D_n$ into three parts $D^{tr}$, $D^{va}$, and $D^{te}$
- For each value $\theta_m$ of the hyper-parameter $\theta$
  - select $f^*_{\theta_m}(D^{tr}) = \arg \min_{f \in F_{\theta_m}} \hat{R}(f, D^{tr})$
  - let $\hat{R}(f^*_{\theta_m}(D^{tr}), D^{va}) = \frac{1}{va} \sum_{z_i \in D^{va}} L(z_i, f^*_{\theta_m}(D^{tr}))$
- select $\theta^*_m = \arg \min_{\theta_m} \hat{R}(f^*_{\theta_m}(D^{tr}), D^{va})$
- select $f^*(D^{tr} \cup D^{va}) = \arg \min_{f \in F_{\theta^*_m}} \hat{R}(f, D^{tr} \cup D^{va})$
- estimate $R(f^*(D^{tr} \cup D^{va}))$ with $\frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*(D^{tr} \cup D^{va}))$
Cross-validation + Test

- Select a family of functions with hyper-parameter $\theta$
- Divide your dataset $D_n$ into two parts: 
  
  \[ \text{a training set } D^{tr} \text{ and a test set } D^{te} \]

- For each value $\theta_m$ of the hyper-parameter $\theta$
  
  estimate $R(f^{*}_{\theta_m}(D^{tr}))$ with $D^{tr}$ using cross-validation

- select $\theta^*_m = \arg\min_{\theta_m} R(f^*_{\theta_m}(D^{tr}))$

- retrain $f^*(D^{tr}) = \arg\min_{f\in F_{\theta^*_m}} \hat{R}(f, D^{tr})$

- estimate $R(f^*(D^{tr}))$ with $\frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*(D^{tr}))$
Double Cross-validation

- Select a family of functions with hyper-parameter $\theta$
- **Divide** your training set $D_n$ into $K$ distinct and equal parts $D_1, \ldots, D^K$
- For each part $D^k$
  - **select** the best model $f^* (\tilde{D}^k)$ by cross-validation on $\tilde{D}^k$
  - estimate $R(f^*(\tilde{D}^k))$ with
    \[
    \hat{R}(f^*(\tilde{D}^k), D^k) = \frac{1}{|D^k|} \sum_{z_i \in D^k} L(z_i, f^*(\tilde{D}^k))
    \]
- **estimate** $R(f^*(D))$ with $\frac{1}{K} \sum_k R(f^*(\tilde{D}^k))$
- Note: this process only gives you an estimate of the risk, but not a model. If you need the model as well, you have to perform a separate model selection process!
Double Cross-validation

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The whole dataset is cut into 3 parts</td>
</tr>
<tr>
<td>2</td>
<td>The first 2 parts are cut into 3 parts then perform a 3-fold cross-validation to select the best hyper-parameter</td>
</tr>
<tr>
<td>3</td>
<td>The best hyper-parameter is used to retrain on the 2 original parts and test on the other one</td>
</tr>
<tr>
<td>4</td>
<td>... and do the same for each part to estimate risk</td>
</tr>
</tbody>
</table>
Beware of the Machine Learning Magic

Training Data

Random Tuning

Learning Algorithm

Excellent Results

Data, Functions, Risk
The Capacity
Methodology

Model Selection
Estimation of the Risk
Model Selection and Estimation of the Risk
The Machine Learning Magic
Beware of the Machine Learning Magic (con’t)

Lot’s of Time and Effort

Random Tuning

Learning Algorithm

Garbage Results

Random Algorithm Learning Lot’s of Time and Effort Garbage Results