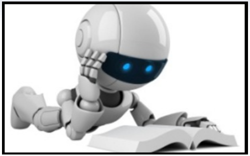
Machine learning - from theory to practice



Marcin Chrzaszcz[1,2]

November 22, 2013

[1] University of Zurich, [2] Institute of Nuclear Physics in Krakow

Lets start from a joke

Q: What is the difference between a physicist and a big pizza?

Lets start from a joke

Q: What is a difference between a physicist and a big pizza?
A: Pizza is enough to feed the full family.

What is machine learning

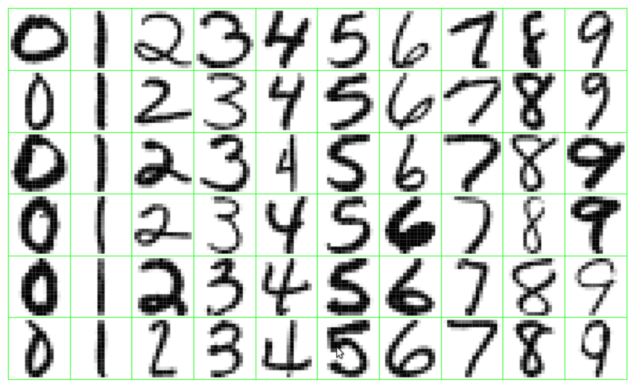
1. Machine learning:
   - It is a science about how to construct a system that can learn from data.

What is machine learning

1. Machine learning:
    - It is a science about how to construct a system that can learn from data.
2. To be less precise but more intuitive it helps you solve problems like:
    - Predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data.
    - Identify the numbers in a handwritten ZIP code, from a digitized image.
    - etc.

What is machine learning

A simple example:



Linear Models

- Let's assume we have a vector of inputs: $X^T = (X_1, X_2, ..., X_p)$.
- We predict the output of our machine/classifiers:

$$\widehat{Y} = \beta_0 + \sum_{j=1}^{p} \beta_j X_j = \sum_{j=0}^{p} \beta_j X_j = X^T \widehat{\beta} \tag{1}$$

- To fit this one could use the method of least squares:

$$RSS(\beta) = \sum_{j=1}^{n} (y_i - x_i^T \beta)^2 \tag{2}$$

- It's a quadratic function in β so minimum exists.

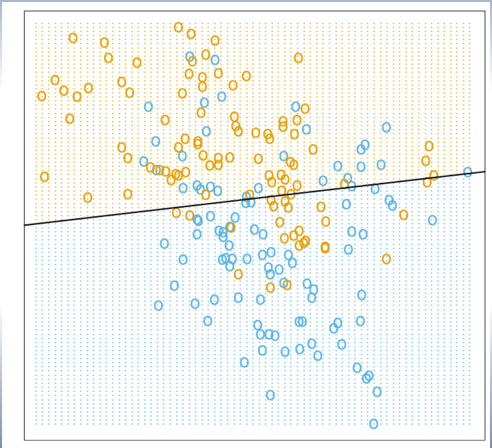
Linear Models - Example

Probably I have already managed to bore you, so let's look at an example:

- We have two pairs of simulated data: X_1, X_2
- A linear regression was fit to these data.
- Response $\widehat{Y}$ color coded:

$$\widehat{G}(\widehat{Y}) = \begin{cases} \textit{orange} & \text{if } \widehat{Y} \geq 0.5 \\ \textit{blue} & \text{if } \widehat{Y} < 0.5 \end{cases} \tag{3}$$

Linear Models - Example



- We see that in $\mathcal{R}^2$ space we used the boundary $x : x^T\widehat{\beta} = 0.5$
- There exists a number of points that have been misclassified on both sides.
- Looks like our linear model is not appropriate.

Linear Models - Example

We didn't say anything about the two test samples. The usual scenarios:

1. The training data in each class were generated from bivariate Gaussian distributions with uncorrelated components and different means.
2. The training data in each class came from a mixture of 10 low-variance Gaussian distributions, with individual means themselves distributed as Gaussian

I use Gaussian because it's easy to generate and has a nice interpretation.

Nearest-Neighbor Method

- Nearest-neighbor methods use those observations in the training set of k closest in input space to x to construct $\widehat{Y}$:

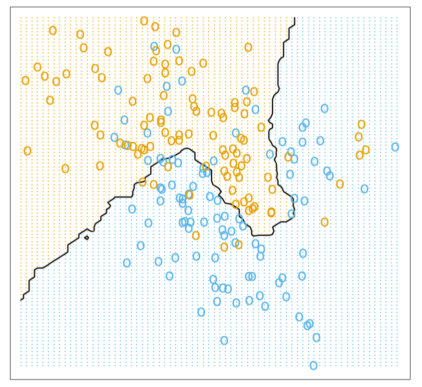
$$\widehat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i, \tag{4}$$

  where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample.

- Let's assume a Euclidean metric and calculate and repeat the same example but with a new function.
- For example we can put $k = 15$ and $k = 1$.

Nearest-Neighbor Example

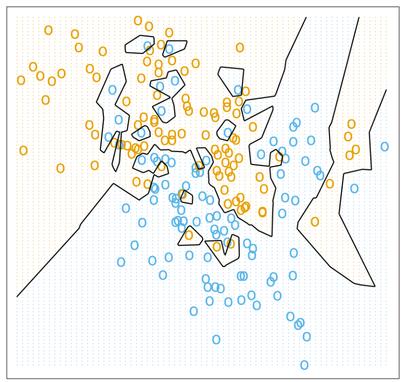
$k = 15$



- Fewer training observations are misclassified.
- This should not give you too much hope!
- See next example.

Nearest-Neighbor Example

$k = 1$



- No points are misclassified!
- Clearly this doesn't tell you anything about the real distribution.
- You should always check your methods on a testing sample.
- aka train on half of the data and apply classifier to the second half and see if they agree

Bias-Variance Tradeoff

- All methods I have described so far have a parameter that needs to be tuned.
- There are two competing forces.
- The trick is to balance the effect.
- Let's make an example based on Nearest-Neighbor.
- The test error ($Y = f(X) + \epsilon$):

$$EPE_k(x_0) = \sigma^2 + [f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_l)]^2 + \frac{\sigma^2}{k}, \tag{5}$$

where $\sigma = Var(\epsilon)$

Bias-Variance Tradeoff

- All the methods that I already described have a parameters that needs to be tuned.
- There are two competing forces.
- The trick is balance the effect.
- Let make an example based on Nearest-Neighbor.
- The test error ($Y = f(X) + \epsilon$):

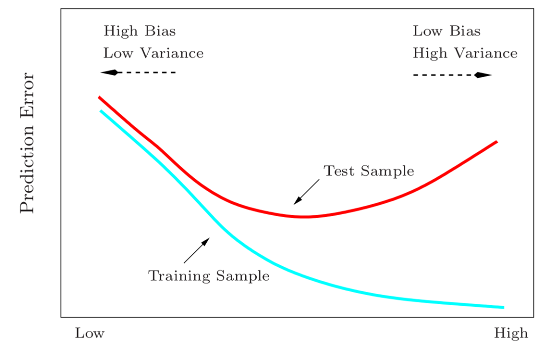
$$EPE_k(x_0) = \sigma^2 + \underbrace{[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_l)]^2}_{\text{bias}} + \underbrace{\frac{\sigma^2}{k}}_{\text{var}} \quad (5)$$

Bias-Variance Tradeoff

$$EPE_k(x_0) = \sigma^2 + \underbrace{[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_l)]^2}_{\text{bias}} + \underbrace{\frac{\sigma^2}{k}}_{\text{var}} \tag{5}$$

- Bias component tend to blow up a k increases.
- On the other hand the variance term decreases as k increases.
- We are basically balancing on the edge.

Bias-Variance Tradeoff



$$EPE_k(x_0) = \sigma^2 + \underbrace{[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_l)]^2}_{\text{bias}} + \underbrace{\frac{\sigma^2}{k}}_{\text{var}}$$

Other complex models

Other complex models include:

- Neural Networks
- Kernel Methods
- Sparse Kernel Methods
- Decision Trees
- Graphs
- Sampling methods
- Mix methods
- Principal Component Analysis
- Many others.

Neutral Networks

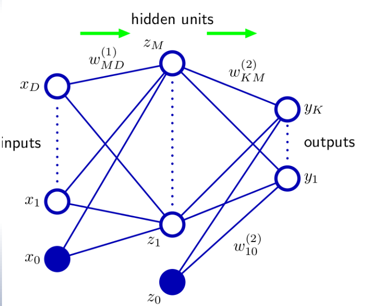
It's simple extension of the linear case:

$$Y(\mathbf{x},\mathbf{w}) = f\left(\sum_{j=1}^{M} w_j \Phi_j(\mathbf{x})\right), \tag{5}$$

where:
$\Phi_j(\mathbf{x})$ are basis functions,
$f(.)$ is a nonlinear activation function

Neutral Networks

In practice:



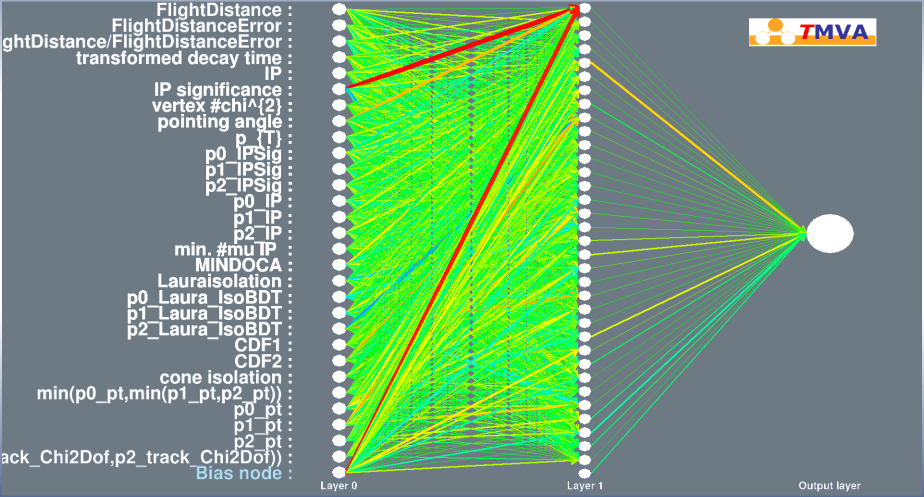
- Construct linear combinations: $a_j = w_{ji}x_i + w_{J0}$
- Each of the activations (a_j) we transform with non-linear function:

$$z_j = h(a_j)$$

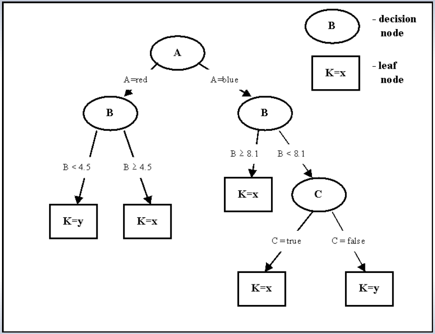
- Output of this function are called hidden units.
- $h()$ is usually a sigmoidal function.
- Then again you construct a linear combination of variables: $a_k = w_{ki}x_i + w_{k0}$ and again put inside the activation function.

Neutral Networks

A real world example:

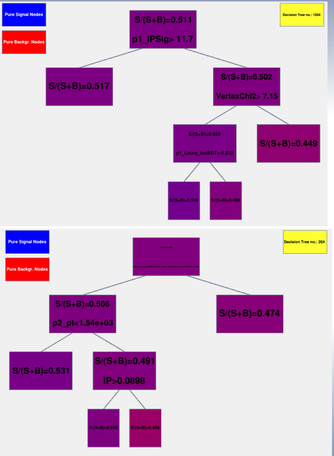


Decision trees



- Flow chart(First trees were calculated by hand)
- Decisions are dependent on previous step.
- Easy to use.
- Learning converges fast.
- Usually one trains 1k of trees for clarifier.

Decision trees

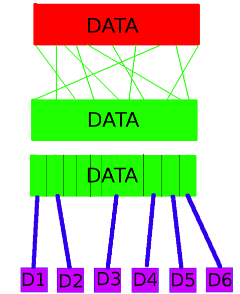


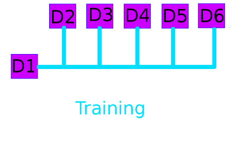
- Real example used in LHCb experiment.
- Search for $\tau \to 3\mu$.
- Trees are combined using unlikelihood.
- Most commonly used in HEP.

Folding

- When training one needs to use two samples: training and testing.
- Training sample can't be used for analysis because of biases.
- Normally one needs to throw some part of the data away just for training.
- When ones considers costs throwing away 10% of data is like throwing away 5M dollars a year
- Could we get that money/data back?

Folding



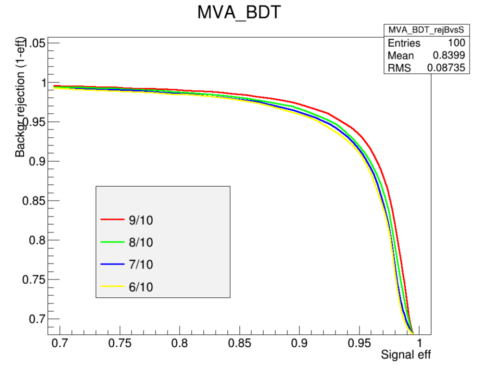


1. Reshuffling the events to guarantee the uniformity of the data.

2. Chopping in sub-samples.

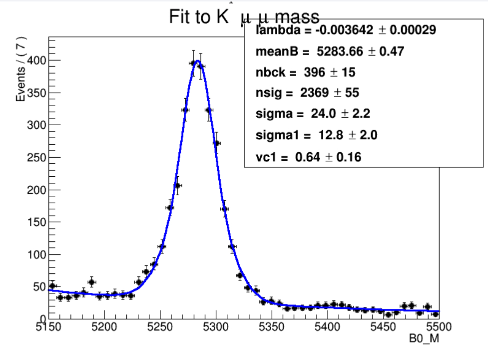
3. Training using n-1 sub-samples and applying the result on the remaining one (iteratively)
   Increase in the statistics used in the training (more stable MVA response), no bias in the result :-)

Folding



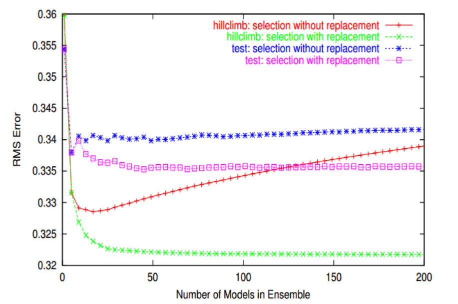
- Standard way to judge a classifier is to look on the ROC(Receiver operating characteristic) curve.
- Ones sees that not only one can use all data, but one gains with increasing number of folds.
- Simply statistical explanation. More data to train makes fits inside the classifiers more stable(less sensitive to fluctuations)
- One can tune the parameters of the classifier to "higher" values.

Folding



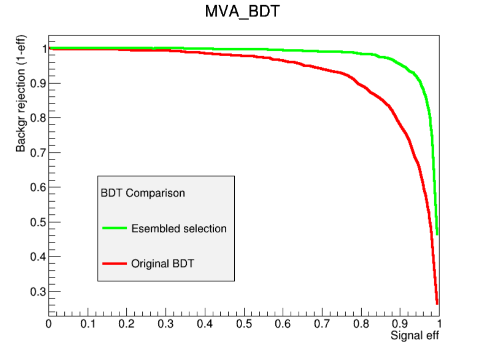
- Example from a recently studied channel: $B0 \to K^*\mu\mu$.
- Using folding one reduced background from 500 events to 400.
- Background are extremely dangerous for this analysis.

Ensemble Selection



- An ensemble is a collection of models whose predictions are combined by weighting or voting.
- Add to the ensemble the model in the library that maximizes the ensembles performance.
- Repeat Step 2 for fixed number of iterations until all models are used.
  1. In practice one can add all the classifiers to one single classifier.

Ensemble Selection



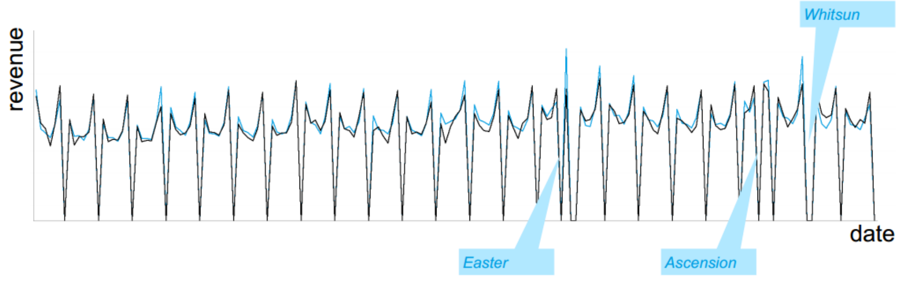
- One clearly gains with using this classifier.
- This is an extension to the Ensemble Selection for the search for $\tau \to 3\mu$.
- τ leptons are produced in one of the given modes:
  - $\mathrm{B} \to \tau X$
  - $\mathrm{B} \to \mathrm{D} \to \tau X$
  - $\mathrm{B} \to \mathrm{D_s} \to \tau X$
  - $\mathrm{D_s} \to \tau X$
  - $\mathrm{D} \to \tau X$
- One clearly gains using this approach :)

Nonscientific application

Replenishment for a grocery chain (24/7 SaaS operations)	Automation increased from 61% to 95%
Supply chain predictions (24/7 SaaS operations)	**> 620,000,000** predictions every day
Dynamic pricing for a major online shop	10% revenue increase after 4 weeks
Customer life cycle management	6% revenue increase within 3 months

Nonscientific application

- Revenue prediction for each individual store



Conclusions

- Machine learning is everywhere.
- One of the fastest developing branches in mathematics.
- Very profitable business :)
- Market is there, so maybe for a living apart of hard core mathematics one should think about putting some time into machine learning?