Machine Learning

Ilya Narsky, Caltech
Lecture 4

Variable selection and dimension reduction methods.
How to throw away variables when you have too many?

• **Two groups of methods:**
  1. **Classifier-independent:** Pre-select variables before you feed them into a classifier. Selection is done by methods not related to your classifier of choice.
  2. **Classifier-specific:** Use your classifier to select variables. Obtain information on variable importance from your classifier.

  – **General wisdom:** Prefer classifier-specific methods over classifier-independent methods. Use classifier-independent methods only when your classifier requires it.
  
  – For example, k-NN methods do not work well in high-D. You need to select those few variables that are likely to give good classifier performance in low-D.
Why I like decision trees so much?

• …because they perform well in high-D space. You can feed hundreds of variables into BDT or RF without deteriorating the performance of your classifier.
  – Then you can rank their importance using tree-specific measures (to be discussed later today).
  – Then you can start removing least important ones until classification power drops significantly.

• The number of variables you can feed into your decision tree depends on the implementation. Sometimes you just have too many… need to pre-select variables.

• For Neural Nets, I would use an opposite approach:
  – Start with a few most powerful variables and add more as long as you can improve NN performance.
  – Again, need to pre-select variables.
Classifier-independent methods (besides visual inspection of 1D projections)
Correlation with class label

- Compute correlation between variable and class label

\[ \rho(X, Y) = \frac{E[(X - EX)(Y - EY)]}{\sqrt{VarX \cdot VarY}} \]

- Ok, try to fold the distribution to avoid symmetry:

\[ \rho(|X - EX|, Y) = \frac{E[|X - EX| \cdot (Y - EY)]}{\sqrt{VarX \cdot VarY}} \]

- Still not very reliable: can have both \( \rho(X, Y) \) and \( \rho(|X-EX|, Y) \) small for a powerful classification variable \( X \).
Correlations between variables

- Even if a variable strongly correlates with class label, you may not need it. This variable can also strongly correlate with other variables and you can get most information out of these other variables.
Separability of classes using scatter matrices

$\mu_k$ and $\Sigma_k$ – estimates of mean vector and covariance matrix for class $k$

$$\mu = \frac{\sum_{k=1}^{K} W_k \mu_k}{\sum_{k=1}^{K} W_k}$$

sample mean

$$\Sigma_W = \frac{\sum_{k=1}^{K} W_k \Sigma_k}{\sum_{k=1}^{K} W_k}$$

within-class covariance matrix

$$\Sigma_B = \frac{\sum_{k=1}^{K} W_k (\mu_k - \mu)(\mu_k - \mu)^T}{\sum_{k=1}^{K} W_k}$$

between-class covariance matrix

Maximize dissimilarity:

$$Tr \left\{ \Sigma_W^{-1} \Sigma_B \right\}$$

Several other measures of dissimilarity using scatter matrices are available.

Note: This algorithm works (at least in principle) for an arbitrary number of classes!
Separability of two samples: GOF test

• Define a goodness-of-fit measure to compute distance between two multivariate samples. One sample is signal and another one is background.

• This GOF measure, of course, should be able to handle high-D problems. GOF techniques in high-D is a subject of ongoing research.

• Apply the GOF test to the two samples to estimate the dissimilarity between them.

• See how much you can increase this dissimilarity by including more variables.
Classifier-specific methods
Decision Trees

Popular criteria used for splitting

(p = fraction of correctly classified events)

- $Q(p) = p$ purity
- $Q(p) = -2p(1-p)$ Gini index
- $Q(p) = p \log p + (1-p) \log(1-p)$ cross-entropy

Parent node with $W$ events and correctly classified $p*W$ events is split into two daughters nodes iff

$$WQ(p) < W_1Q(p_1) + W_2Q(p_2)$$

Measure improvement in figure-of-merit due to this split:

$$\Delta Q = W_1Q(p_1) + W_2Q(p_2) - WQ(p)$$
Variable importance for decision trees

• Simple approach – count decision splits on each variable
  – Example: Boosted Decision Trees typically require big leaves for optimal performance => splits are imposed on most important variables. The split count can indeed tell you something useful.
  – Sometimes there is not useful info in this approach. Imagine, for example, that you are running Random Forest and randomly choosing one variable for each decision split. You get the same number of splits for each variable.

• More clever ideas (both due to Breiman):
  – Estimate change in the optimization figure-of-merit due to splits on this variable
    \[ \mathcal{T}_d = \sum_{\text{internal tree nodes}} \Delta Q \cdot I(\text{split on } x^{(d)}) \]
  – Estimate change in FOM due to random permutations of this variable: Randomly permute values for variable \( d \) in the test sample and estimate decrease in FOM returned by decision tree
Ultimate approach (for any classifier)

• Try all or most combinations of variables for the chosen classifier and monitor the predictive power.

• Make only negative assumptions:
  – If one set of variables is close to optimal for one classifier, it is not necessarily optimal for another.
  – If one set of variables is optimal for a classifier with some parameters, it is not necessarily optimal for the same classifier with different parameters.
  – If one set of variables is optimal for one definition of “predictive power”, it is not necessarily optimal for another definition.

• Sooner or later, you will have to make a subjective call about what variables you want and what variables you don’t want in your data.
The End.
Meet again in 6 years.