Computational Intelligence: Methods and Applications

Lecture 34
Applications of information theory and selection of information

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

Some applications of info theory

Information theory has many applications in different CI areas.
Here only a few applications are mentioned, to data visualization, discretization and selection of features.

Information gain has already been used in decision trees (ID3, C4.5) to define the gain of information by making a split: for feature *A*, used to split node S into left S_l, and right S_r sub-node, with classes $\boldsymbol{\omega}=(\omega_1 \ldots \omega_K)$

$$G(\boldsymbol{\omega}, \mathrm{A}|\mathrm{S}) = H(\boldsymbol{\omega}\,|\,\mathrm{S}) - \frac{|\mathrm{S}_l|}{|\mathrm{S}|} H(\boldsymbol{\omega}\,|\,\mathrm{S}_l) - \frac{|\mathrm{S}_r|}{|\mathrm{S}|} H(\boldsymbol{\omega}\,|\,\mathrm{S}_r)$$

with
$$H(\boldsymbol{\omega}\,|\,\mathrm{S}) = -\sum_{i=1}^{K} P(\omega_i\,|\,\mathrm{S}) \lg_2 P(\omega_i\,|\,\mathrm{S})$$

= information in the class distribution $\boldsymbol{\omega}$ for vectors in the node S.
Information is zero if all samples in the node are from one class (log 1 = 0, and 0 log 0 = 0), and reaches maximum $H(\mathrm{S})=\lg_2 K$ for uniform distribution in *K* equally probable classes, $P(\omega_i|\mathrm{S})=1/K$.

Model selection

How complex should our model be? For example, what size of a tree, how many functions in a network, and what degree of the kernel?

Crossvalidation is a good method but on a huge data it is costly.

Another way to optimize model complexity is to measure the amount of information necessary to specify the model and its errors.
Simpler models make more errors, complex need longer description.

Minimum Description Length for model + errors (very simplified).

General intuition: learning is compression, finding simple models, regularities in the data (ability to compress). Therefore estimate:

L(M) = how many bits of information are needed to transmit the model.

L(D|M) = how many bits to transmit information about data, given M.

Minimize L(M)+L(D|M).
Data correctly handled need not be transmitted.
Estimations of L(M) are usually nontrivial but approximations work well.

More on model selection

Many criteria for information-based model selection have been devised in computational learning theory, two best known are:

AIC, Akaike Information Criterion.
BIC, Bayesian Information Criterion.

The goal is to predict, using training data, which model has the best potential for accurate generalization.

Although model selection is a popular topic applications are relatively rare and selection via crossvalidation is commonly used.

Models may be trained by max. mutual information of outputs/classes:

$$Y_j^{(i)} = g_j\left(\mathbf{X}^{(i)};\mathbf{W}\right) \Rightarrow \max_{\mathbf{W}} MI\left(\boldsymbol{\omega};Y\left(\mathbf{W}\right)\right)$$

This may in general be any non-linear transformation, for example implemented via basis set expansion methods.

Visualization via max MI

Linear transformation that maximizes mutual information between a set of class labels and a set of new input features Y_i, i=1..$d' < d$ is:

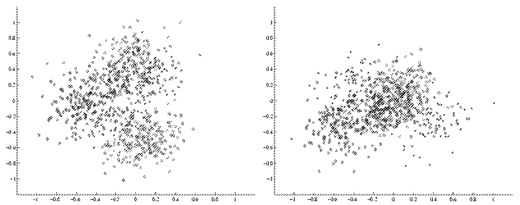
$$\mathbf{W} = \arg\max_{\mathbf{W}} MI\left(Y;\boldsymbol{\omega}\right); \quad \mathbf{Y}^{(i)} = \mathbf{W}^{\mathrm{T}}\mathbf{X}^{(i)}$$

Here W is a d' x d dimensional matrix of mixing coefficients.

Maximization proceeds using gradient-based iterative methods.

Left – FDA view of 3 clusters, right – linear MI view with better separation

FDA does not perform well if distributions are multimodal, separating the means that may be lead to overlapping clusters.

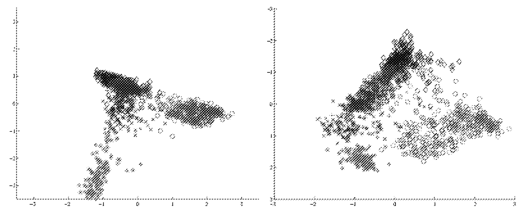


More examples

Torkkola (Motorola Labs) has developed the MI-based visualization; see more ex at: http://members.cox.net/torkkola/mmi.html

Example: the Landsat Image data contain 36 features (spectral intensities of 3x3 submatrix of pixels in 4 spectral bands) are used to classify 6 type of land use; 1500 samples used for visualization.

Left: FDA, right: MI, note violet/blue separation.



Movie 1: Reuters

Movie 2: Satimage

(local only)

Classification in the reduced space is more accurate.

Feature selection and attention

- **Attention**: basic cognitive skill, without attention learning would not have been possible. First we focus on some sensation (visual object, sounds, smells, tactile sensations) and only then the full power of the brain is used to analyze this sensation.
- Given a large database, to find relevant information you may:
  - discard features that do not contain information,
  - use weights to express their relative importance,
  - reduce dimensionality aggregating information, making linear or non-linear combinations of subsets of features (FDA, MI) – new features may not be so understandable as the original ones.
  - create new, more informative features, introducing new higher-level concepts; this is usually left to human invention.

Ranking and selection

- **Feature ranking**: treat each feature as independent, and compare them to determine the order of relevance or importance (rank).

$$X_{i1} \prec X_{i2} \prec X_{i3} \prec \ldots X_{id}$$

Note that:

Several features may have identical relevance, especially for nominal features. This is either by chance, or because features are strongly correlated and therefore redundant.

Ranking depends on what we look for, for example rankings of cars from the comfort, usefulness in the city, or rough terrain performance point of view, will be quite different.

- **Feature selection**: search for the best subsets of features, remove redundant features, create subsets of 1, 2, ... k-best features.

Filters and wrappers

- Can ranking or selection be universal, independent of the particular decision system used?

Some features are quite irrelevant to the task at hand.

- Feature **filters** are model-independent, universal methods based on some criteria to measure relevance, for information filtering.
  They are usually computationally inexpensive.
- **Wrappers** – methods that check the influence of feature selection on the result of particular classifier at each step of the algorithm.
  For example, LDA or kNN or NB methods may be used as wrappers for feature selection.
- Forward selection: add one feature, evaluate result using a wrapper.
- Backward selection: remove one feature, evaluate results.

NB feature selection

Naive Bayes assumes that all features are independent.
Results degrade if redundant features are kept.

Naive Bayes predicts the class and its probability. For one feature X_i

$$P(\omega_k | X_i) = P(X_i | \omega_k)\frac{P(\omega_k)}{P(X_i)} \qquad NB(X_i) = \max_k P(\omega_k | X_i)$$

$$\sum_{k=1}^{K} P(\omega_k | X_i) = 1 \Rightarrow P(X_i) = \sum_{k=1}^{K} P(X_i | \omega_k) P(\omega_k)$$

P(X) does not need to be computed if NB returns a class label only.
Comparing predictions with the desired class C(X) (or probability distribution over all classes) for all training data gives an error estimation when a given feature X_i is used for the dataset *D*

$$i_1 = \arg\min_i E(X_i | D) = \sum_{j=1}^{n} \left\| NB\left(\mathbf{X}_i^{(j)}\right) - C\left(\mathbf{X}^{(j)}\right) \right\|^2$$

NB selection algorithm

Forward selection, best first, but $O(n^2)$:

- start with a single feature, find X_{i1} that minimizes the NB classifier error rate; this is the most important feature; set $\mathbf{X}_s = \{X_{i1}\}$.
- Let $\mathbf{X}_s$ be the subspace of s features already selected; check all the remaining features, one after another, calculating probabilities (as products of factors) in the $\mathbf{X}_s + X_i$ subspace:

$$P(\omega_k | \mathbf{X}_s, X_i) = P(\mathbf{X}_s | \omega_k) P(X_i | \omega_k) \frac{P(\omega_k)}{P(\mathbf{X}_s, X_i)}$$

$$NB(\mathbf{X}_s, X_i) = \max_k P(\omega_k | \mathbf{X}_s, X_i),\ X_i \notin \mathbf{X}_s$$

$$i_{s+1} = \arg\min_i E(\mathbf{X}_s, X_i | D) = \sum_{j=1}^{n} \left\| NB\left(\mathbf{X}_s^{(j)}, X_i^{(j)}\right) - C\left(\mathbf{X}^{(j)}\right) \right\|^2$$

Set $\mathbf{X}_s$ <= $\{\mathbf{X}_s + X_{s+1}\}$ and repeat until the error stops decreasing.

For NB this forward selection wrapper approach works quite well.

NB WEKA example

First run WEKA with NB on data with larger number of attributes, for example colic or hypothyroid data.

Add wrapper selection using FilteredClassifier with the AttributeSelectionFilter

- Note that the number of attributes has decreased, while the accuracy has increased.
- Note that WEKA may run forward selection, backward selection, and bi-directional selection.
- Search is terminated after a user-specified number of nodes after expansion does not decrease error; a beam-search instead of the best-first-search is used.
  That means that if expansion of the best node does not improve results, second best is used and new features added, then third best, etc, avoiding local minima in the search process.