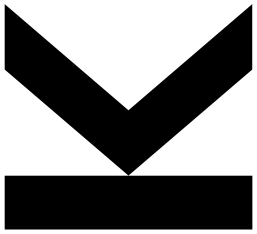
UNIT 6



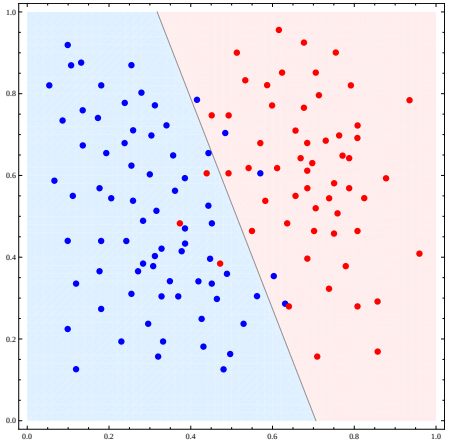
Feature Selection

JOHANNES KEPLER
UNIVERSITY LINZ

INTRODUCTION

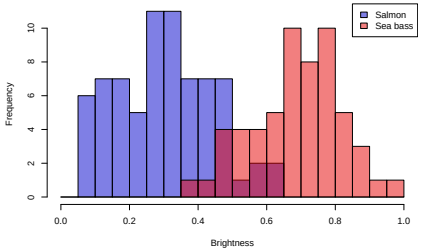
- Regardless of how many features are available for a given machine learning task, it is seldom known exactly which features are relevant and which are irrelevant or even misleading.
- *Feature selection*, also known as *variable selection* or *variable subset selection*, is concerned with determining a subset of features that are relevant to a given machine learning task.
- Feature selection can be carried out in a supervised and in an unsupervised fashion, both for supervised and unsupervised learning tasks.
- We will concentrate on *supervised machine learning* here.

FISH RECOGNITION EXAMPLE REVISITED



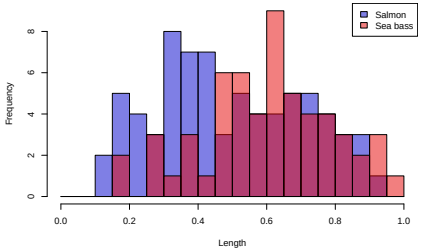
Two regular features (brightness & length):

- KNN ($k = 3$): CV-ACC $= 92.3\%$
- KNN ($k = 9$): CV-ACC $= 92.3\%$
- SVM (linear): CV-ACC $= 94.6\%$
- SVM (RBF kernel): CV-ACC $= 92.3\%$



Regular features + 4 noise features:

- KNN ($k = 3$): CV-ACC $= 86.9\%$
- KNN ($k = 9$): CV-ACC $= 90.0\%$
- SVM (linear): CV-ACC $= 94.6\%$
- SVM (RBF kernel): CV-ACC $= 92.3\%$



Regular features + 40 noise features:

- KNN ($k = 3$): CV-ACC $= 60.8\%$
- KNN ($k = 9$): CV-ACC $= 64.6\%$
- SVM (linear): CV-ACC $= 86.2\%$
- SVM (RBF kernel): CV-ACC $= 89.2\%$

MOTIVATION

- The previous example demonstrates that predictors are indeed impaired by irrelevant/noise features.
- How strong this impairment is depends on the method.
- The more irrelevant features there are, the more difficult the prediction becomes.
- The removal of irrelevant features, therefore, in any case, helps to improve/stabilize the prediction results.
- This is a notorious situation in many practical domains, in particular, in microarray data analysis.

CLASSIFICATION OF FEATURE SELECTION METHODS

Filter methods: select/rank features according to some statistical criteria without making use of any prediction method;

Wrapper methods: strategies that apply a predictor to evaluating candidate features subsets;

Feature selection during learning: some machine learning methods have an inherent feature selection component;

Feature selection after learning: some machine learning methods allow for pruning the least relevant features upon training;

FILTER METHODS: PREREQUISITES

Suppose we are given a data set $\mathbf{Z} = \{(\mathbf{x}^i, y^i) \mid i = 1, \ldots, l\}$ where the inputs $\mathbf{x}^i$ are real-valued vectors from $\mathbb{R}^d$ and the targets y^i are real-valued (which includes the case $y^i \in \{-1, +1\}$). Then we define ($j = 1, \ldots, d$):

Mean target: $\bar{y} = \frac{1}{l} \sum_{i=1}^{l} y^i$

Mean of j-th feature: $\bar{x}_j = \frac{1}{l} \sum_{i=1}^{l} x_j^i$

Standard deviation of target:

$$\sigma_y = \sqrt{\frac{1}{l} \sum_{i=1}^{l} (y^i - \bar{y})^2}$$

Standard deviation of j-th feature:

$$\sigma_j = \sqrt{\frac{1}{l} \sum_{i=1}^{l} (x_j^i - \bar{x}_j)^2}$$

Now suppose $y^i \in \{-1, +1\}$ and that l^+ samples are positive and l^- samples are negative. Then, analogously, we can compute means and standard deviations of values in the j-th column/feature separately for positive and negative samples:

- $\bar{x}_j^+$: mean of j-th feature for positive samples;
- $\bar{x}_j^-$: mean of j-th feature for negative samples;
- σ_j^+: standard deviation of j-th feature for positive samples;
- σ_j^-: standard deviation of j-th feature for negative samples;

FILTER CRITERIA

Pearson's correlation coefficient:

$$\frac{\sum_{i=1}^{l}(x_j^i - \bar{x}_j) \cdot (y^i - \bar{y})}{\sigma_j \cdot \sigma_y}$$

Signal-to-noise ratio (aka Golub's criterion):

$$\frac{\bar{x}_j^+ - \bar{x}_j^-}{\sigma_j^+ + \sigma_j^-}$$

Fisher criterion:

$$\frac{(\bar{x}_j^+ - \bar{x}_j^-)^2}{(\sigma_j^+)^2 + (\sigma_j^-)^2}$$

t-statistic:

$$\frac{\bar{x}_j^+ - \bar{x}_j^-}{\sqrt{\frac{(\sigma_j^+)^2}{l^+} + \frac{(\sigma_j^-)^2}{l^-}}}$$

FILTER METHODS

- Other filter criteria:
  - Mann-Whitney-Wilcoxon statistic (corresponds to AUC)
  - Mutual information
  - ...
- Filter methods evaluate a filter criterion for all features and rank the features according to the criterion.
- To make an actual selection, a cut-off threshold must be chosen. The choice of this threshold is crucial. For the methods presented above, some rigorous statistical procedures are available to adjust the threshold according to some given significance level.
- Since filter methods treat features independently, multiple testing correction must be applied.

WRAPPER METHODS

All wrapper methods rely on a given prediction method and evaluate feature subsets by estimating the prediction performance by cross validation or a validation set.

Exhaustive search: all possible 2^d feature subsets are evaluated. Since the number of subsets grows exponentially with the dimension, this is only possible for small numbers of features.

Forward selection: starts with an empty feature set and iteratively adds the feature that gives the best increase in performance; this is done until adding another feature would lead to decreasing performance.

Backward selection: starts with all features and iteratively removes the feature the removal of which gives the best increase in performance; this is done until removing another feature would lead to decreasing performance.

WRAPPER METHODS (cont'd)

Random mutation hillclimbing: starts with a random feature set and then, repeatedly, selects a random feature that is added or removed from the present feature set. This "mutation" is carried out if it leads to an increase in performance.

... (e.g. beam search, simulated annealing, genetic algorithms)

Forward and backward selection can both lead to sub-optimal feature sets (i.e. local minima of the search procedure), particularly if there are non-linear dependencies between features.

FEATURE SELECTION DURING TRAINING

Rule-based methods: use if-then rules and logical clauses to represent classifiers; they are inherently aimed at selecting the best clauses (thereby, the best features); this class includes decision and regression trees (ID3, CART, MARS, C4.5, random forests), inductive logic programming, etc.

Regularization: some machine learning method are or can be augmented with a regularization term that is geared towards the selection of few relevant features. Examples:

- Potential Support Vector Machine (P-SVM)
- Linear models with L_1 regularization (LASSO)

LINEAR MODELS WITH FEATURE SELECTION?

- Linear models determine weights for each feature. These weights (resp. their absolute values) can also be understood as measures of importance/relevance.
- However, the weights of standard linear models are seldom zero, and standard learning algorithms do not include a well-defined strategy for actually selecting a subset of relevant features.
- The L_2 regularization term (sum of squares of weights) used in standard SVMs, in neural networks with weight decay, and in so-called *ridge regression* favors small weights, but does not actually push them to 0.

L_1 REGULARIZATION AND LASSO

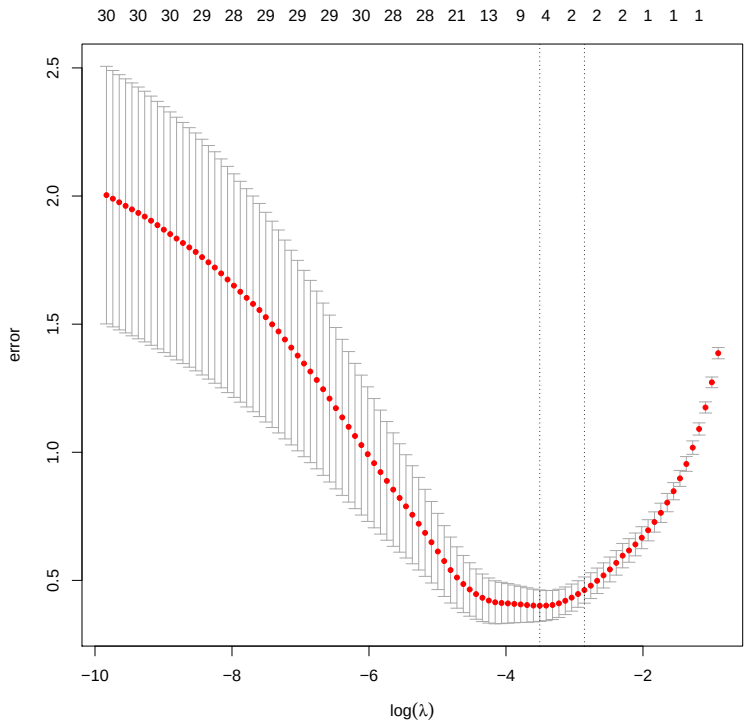
A common strategy is to add an L_1 *regularization term* to the learning objective (with λ being the regularization parameter that controls the strength of the influence of the regularization term):

$$\lambda \cdot \|\mathbf{w}\|_1 = \lambda \cdot \sum_{j=1}^{d} |w_j|$$

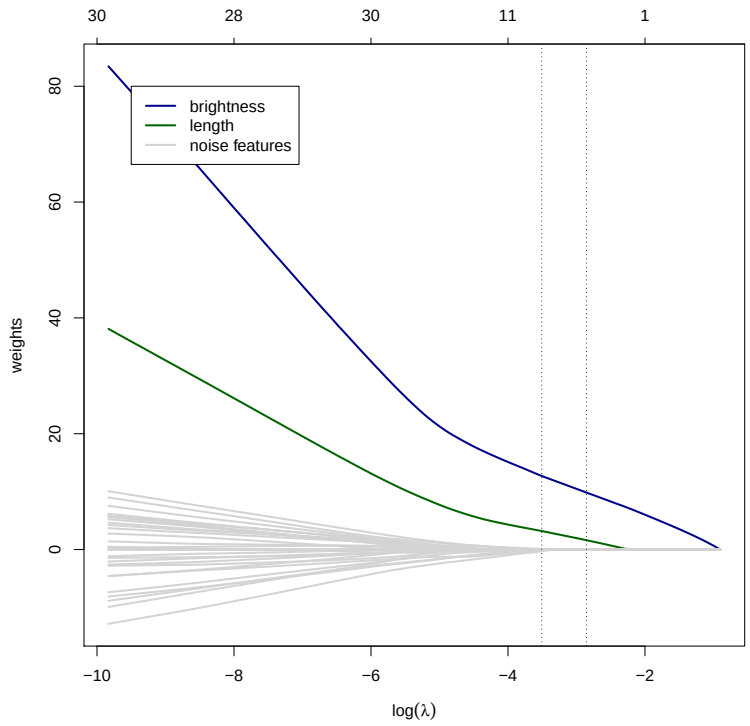
This L_1 regularization term tends to push weights to zero and, thereby, facilitates *feature selection during training*. The most prominent method is the *Least Absolute Shrinkage and Selection Operator (LASSO)*. Among groups of redundant (i.e. correlated) features, only the most prominent feature is selected.

LASSO EXAMPLE: FISH RECOGNITION (+ 40 noise features)

λ vs. cross validation performance:



λ vs. weights:



FEATURE SELECTION AFTER TRAINING

Neural networks: OBS and OBD pruning of input weights

Linear models (including linear support vector machines):

- Prune features with lowest squared weights (with a given cut-off threshold)
- Repeat iteratively: prune feature with lowest squared weight and re-train; this is called *recursive feature elimination*.

These methods can prune irrelevant features, but may also prune relevant, yet redundant, features.

A FINAL WARNING

- Never apply feature selection to the whole data set, since this will bias the selected features to the whole data set! This is a common error known as the *feature selection bias*.
- When using the test set method, apply feature selection only to the training set.
- When using cross validation, apply feature selection to each training set ($n - 1$ folds) separately.