Machine Learning FAQ

What is the difference between filter, wrapper, and embedded methods for feature selection?

Wrapper methods measure the “usefulness” of features based on the classifier performance. In contrast, the filter methods pick up the intrinsic properties of the features (i.e., the “relevance” of the features) measured via univariate statistics instead of cross-validation performance. So, wrapper methods are essentially solving the “real” problem (optimizing the classifier performance), but they are also computationally more expensive compared to filter methods due to the repeated learning steps and cross-validation. The third class, embedded methods, are quite similar to wrapper methods since they are also used to optimize the objective function or performance of a learning algorithm or model. The difference to wrapper methods is that an intrinsic model building metric is used during learning. Let me give you a – off the top of my head – list of examples from these three categories.

Filter methods:

- information gain
- chi-square test
- fisher score
- correlation coefficient
- variance threshold

Wrapper methods:

- recursive feature elimination
- sequential feature selection algorithms
- genetic algorithms

Embedded methods:

- L1 (LASSO) regularization
- decision tree

(Note that I would count transformation and projection techniques such as Principal Component Analysis as a feature *extraction* approach, since we are projecting the data into a new feature space.) To give you a more hands-on illustration, let me pick one algorithm from each category and explain w

1). A Filter method Example: Variance Thresholds

Here, we simply compute the variance of each feature, and we select the subset of features based on a user-specified threshold. E.g., “keep all features that have a variance greater or equal to *x*” or “keep the the top *k* features with the largest variance.” We assume that features with a higher variance may contain more useful information, but note that we are not taking the relationship between feature variables or feature and target variables into account, which is one of the drawbacks of filter methods.

2). A Wrapper Method Example: Sequential Feature Selection

Sequential Forward Selection (SFS), a special case of sequential feature selection, is a greedy search algorithm that attempts to find the “optimal” feature subset by iteratively selecting features based on the classifier performance. We start with an empty feature subset and add one feature at the time in each round; this one feature is selected from the pool of all features that are not in our feature subset, and it is the feature that – when added – results in the best classifier performance. Since we have to train and cross-validate our model for each feature subset combination, this approach is much more expensive than a filter approach such as the variance threshold, which we discussed above.

3). An Embedded Method Example: L1 Regularization

L1 (or LASSO) regression for generalized linear models can be understood as adding a penalty against complexity to reduce the degree of overfitting or variance of a model by adding more bias. Here, we add a penalty term directly to the cost function,

```
regularized_cost = cost + regularization_penalty
```

In L1 regularization, the penalty term is

$$L1 : \lambda \Sigma^{k}_{i} w_i = \lambda \mathbf{w}_1,$$

where **w** is our *k-dimensional* feature vector. Through adding the L1 term, our objective function now becomes the minimization of the regularized cost, and since the penalty term grows with the value of the weight parameters (λ is just a free parameter to fine-tune the regularization strength), we can induce sparsity through this L1 vector norm, which can be considered as an intrinsic way of feature selection that is part of the model training step.