Building a Classifier

Detailed Description

A classifier in Qlucore Omics Explorer consists of the classifier model and a variable list of the variables used in the model. A classifier also contains information about how the training data was normalized, and the accuracy of the classifier when evaluated.

During the build process, the program tests a number of different combinations of variables and classifier parameters in order to find the combination that gives the best classifier performance. This is done by training several classifier candidates using different variable sub sets and classifier parameters. Each candidate is trained using stratified k-fold cross-validation (where 'k' corresponds to the setting "Number of Inner Sets", see below). The parameter and variable combination that yields the best result is then used to train a final classifier. The final classifier is evaluated either using the training data or, preferably, using an independent data set.

1. Decide on how many and how large variable sets to test

   By default, 7 different variable sets will be tested. The sizes of the variable sets are evenly spaced on a log-scale, the start and stop values are calculated based on the total number of variables per sample:

   Total number of variables less than or equal to 400: Start value = 1, stop value = 0.1 * total number of variables.

   Total number of variables greater than 400: Start value = 10, stop value = 0.1 * total number of variables + 360.

   It is possible to override the default settings using the edit field "Number of Inner Sets (folds)" within the settings dialog in the Build Classifier Tab. It is also possible to set interval start, interval stop, and whether to use equal log-spacing or linear spacing when determining intermediate values for variable set size. The variable set sizes used during training is shown in the classifier build report after the completion of the build process.

2. Decide on which variables to include in the test sets The variables in each tested variable set are the 'n' highest ranked variables based on the statistical test selected (where 'n' denotes the number of variables to use in this variable set, see above). Variable candidates are the active variables remaining after variance filtering has been applied. Note that any statistical test applied by using the statistics dock window will not affect the variables used for building the classifier.

3. Decide on which classifier specific parameters to test.

   KNN: Per default, five different values for the parameter K (number of neighbors) is tested. For details, see Knn settings.
   SVM: Per default, seven different values for the parameter "cost" is tested. see SVM settings.
   Random Trees: Per default, three different values for "Max Trees" (250, 500 and 1000) are tested. see Random Trees settings.

4. Split the training data into inner sets

   Split the training data into a number of inner sets - random stratified folds, each fold containing

a number of samples from the training data. Per default, the number of inner folds to use are derived from the number of samples in the training data set; 3 folds if sample size is less than 32, 5 folds if sample size is between 32 and 60, otherwise 10 folds are used. The number of folds can be changed through "Number of inner sets (folds) edit field in the settings dialog in the build classifier tab. The folds are used to setup a cross-validation scheme in which the different classifier candidates are trained and tested. The term "inner set" is used to distinguish the sub-sets uses during training with those used during evaluation ("outer sets") when no independent data set is used.

5. Train and Evaluate all candidate classifiers

   For every combination of variable number and parameter, the following steps are taken:

   - For each training sub-set / test sub-set pair created through the cross-validation scheme:
     - Select the samples specified in the training sub-set from the main training data set.
     - Select the variables to use int he training subset based on the selected ranking.
     - Train a classifier using the training subset.
     - Apply the classifier to the test subset.
     - Calculate and store the accuracy score.

   - Once all sub-sets have been used for training/testing, calculate accuracy per key annotation and total accuracy for the classifier.

6. Select best classifier

   When all classifier candidates have been trained and tested, the classifier that yielded the highest accuracy is selected. If several classifier have the same accuracy, the classifier using the least number of variables is chosen. If both accuracy and number of variables are the same, the classifier parameter is used for determining which classifier that is selected.

7. Train Final Classifier

   Train the final classifier using all samples in the training data set and the variables and classifier parameters of the classifier selected during the previous step.

8. Evaluate Classifier

   The final classifier is evaluated, either using an independent data set, or using a cross-validation process, see Evaluation for details.

Selecting the Best Classifier

When evaluating and selecting the best classifier from all classifier candidates, the classifier with the best accuracy is chosen. If several classifiers with the same accuracy exist, the one using the fewest number of variables is selected. If the number of variables is the same, the classifier is selected on basis of its parameters.

1. Accuracy

   When evaluating a classifier, the result of applying the classifier to a test data set is compared to the key-data (i.e. the expected response). For each possible outcome (i.e. annotation value in the key annotation field), the percentage of correctly classified samples is calculated. The total accuracy is defined by the average of the "per-class"-accuracies. By using the per-class accuracies when calculating the total accuracy, the risk of not detecting potential problems

when using data with uneven class sizes is reduced.

2. Number of Variables

   A classifier able to predict using fewer variables is preferred.

3. Classifier Parameters

   - For KNN classifiers, a high value of "number of neighbors" (K) is preferred.
   - For SVM classifier, a low cost-parameter ("Cost") is considered better than a high cost-parameter.
   - For Random Trees, a low "number of trees" ("Max Trees") is considered better than a high number.

Evaluation

Qlucore Omics Explorer offers two methods for evaluating a built classifier: an independent test data set can be used for evaluation and it is also possible to evaluate the classifier using the training data set, by means of cross validation.

Using Independent Test Data

When a data-set has been selected in the Validation combo box in the Build Classifier Tab, that data-set is used for evaluation of the classifier. For the evaluation to be meaningful, the validation data set must have a key annotation that contains the same value names and values as the key annotation used for training (i.e. if the sample values are "Group 1" and "Group 2" in the training data set, they also have to be "Group 1" and "Group 2" in the validation data set). During evaluation, the validation data set is classified using the final classifier, and the classification result (the prediction) is compared to key annotation of the validation data set. Accuracy per class (i.e. percentage of correct pred ictions per annotation value) is calculated and stored in the classifier build report. A total score is also computed in the same way as when accuracy is calculated during the classification build process.

Using Outer Cross Validation Folds on Training Data Set

In situations where no independent data can be used for evaluation, it is possible to evaluate the likely behavior of the classifier by evaluating the classifier using the training data set. When evaluating using outer folds, the following procedure is followed:

1. The training data set is split into 'n' random stratified folds. The value 'n' is set to 5 per default, but can be changed through "Number of outer sets (folds)" edit field in the the settings dialog of the build classifier tab

2. The folds are used to form training sets and test sets in the same way as during the build classifier process. For each training set, a classifier is built using the normal build procedure described above. The classifier built is then evaluated against the test fold corresponding to the training data, and accuracy information is stored.

3. Once all outer sets have been used for building and evaluating the built classifiers, the final evaluation result is calculated and stored in the classifier build report.

Note that when evaluating the classifier using outer folds as described above, the classifier that was built is not used during the evaluation, instead the process itself is being evaluated, giving information about how likely the *process* is to produce a good quality classifier. (basically by simulating having a

number of test data sets and corresponding independent test sets and evaluate the outcome in these situations).