WeightedVotingXValidation Documentation

Module name: WeightedVotingXValidation
Description: Weighted Voting Classifier
Author: Ken Ross (Broad Institute), gp-help@broad.mit.edu

Summary: The weighted voting algorithm makes a weighted linear combination of relevant "marker" or "informative" genes obtained in the training set to provide a classification scheme for new samples. Target classes (classes 0 and 1) can be for example defined based on a phenotype such as morphological class or treatment outcome. The selection of classifier input features (marker genes) is accomplished by computing a signal-to-noise statistic $S_x = (\mu_0 - \mu_1)/(\sigma_0 + \sigma_1)$ where μ_0 is the mean of class 0 and σ_0 is the standard deviation of class 0. The class predictor is uniquely defined by the initial set of samples and marker genes. In addition to computing S_x, the algorithm also finds the decision boundaries (half way) between the class means: $B_x = (\mu_0 + \mu_1)/2$ for each gene x. To predict the class of a test sample y, each gene x in the feature set casts a vote: $V_x = S_x (G_{xy} – B_x)$ and the final vote for class 0 or 1 is sign(ΣV_x). The strength or confidence in the prediction of the winning class is $(V_{win} - V_{lose})/(V_{win} + V_{lose})$ (i.e., the relative margin of victory for the vote). Notice that this algorithm is quite similar to Naïve Bayes (see the appendix in Slonim et al. 2000).

The model is tested in the leave-one-out cross-validation mode by iteratively leaving one sample out and training a model on the remaining data and testing on the left out sample.

References:

- Golub TR, Slonim DK, et al. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," Science, 531-537 (1999).

- Slonim DK, Tamayo P, Mesirov JP, Golub TR, Lander ES. (2000) Class prediction and discovery using gene expression data. In Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB) 2000. ACM Press, New York, pp. 263–272.

Parameters:

Name	Description
data.filename:	data file name - gct, res, odf
class.filename:	class input file - cls
pred.results.file:	Prediction results output file – odf
feature.summary.file	Feature summary results output file - odf
num.features:	number of signal-to-noise selected features

Return Value:

1. Prediction features odf file.
2. Prediction results odf file.

Platform dependencies:

Task type:	Prediction
CPU type:	any
OS:	any
Java JVM level:	1.4
Language:	Java