GenePattern

WeightedVoting Documentation

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

Summary: The weighted voting algorithm makes a weighted linear combination of relevant “marker” or “informative” features 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 features) is accomplished either 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 or by reading in a list of user provided features. The class predictor is uniquely defined by the initial set of samples and markers. 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 feature x. To predict the class of a test sample y, each feature 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 $\text{sign}(S_x V_x)$. 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 can tested on a separately specified test set. Additionally, the model can be saved and used subsequently on additional test sets.

The table below summarizes the different options available and which parameters are required depending on the option selected.

Parameter	**Train** create a predictive model from a training dataset	**Test with saved model** run a saved model on a new test dataset	**Train/Test** create a model on training data and run it on test data
train.filename	Required	No	Required
train.class.filename	Required	No	Required
saved.model.filename	No	Required	No
test.filename	No	Required	Required
test.class.filename	No	Required	Required
num.features or feature.list.filename	Required	No	Required
model.file	Required	No	Required
pred.results.file	No	Yes	Yes

Parameters

Name	Description
train.filename	training data file name - .gct, .res, .odf type = Dataset

	ignored if a saved model (saved.model.filename) is used
train.class.filename	class file for training data - .cls ignored if a saved model (saved.model.filename) is used
saved.model.filename	input Weighted Voting model file - .odf type = Weighted Voting Prediction Model
model.file	name of output KNN model file - .odf type = Weighted Voting Prediction Model
test.filename	test data file name - .gct, .res, .odf type = Dataset
test.class.filename	class file for test data - .cls
num.features	number of signal-to-noise selected features if feature list filename is not specified
feature.list.filename	features to use for prediction
pred.results.file	name of prediction results output file – .odf type = Prediction Results

References:

- Golub T.R., Slonim D.K., et al. “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring,” Science, 531-537 (1999).

- Slonim, D.K., Tamayo, P., Mesirov, J.P., Golub, T.R., Lander, E.S. (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.

Return Value:

1. if test data is supplied, a file containing the prediction results
2. if training data is specified, a file containing the saved prediction model

Platform dependencies:

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