

KNN Documentation

Module name: KNN

Description: K-nearest neighbors classification

Author: Joshua Korn, Joshua Gould (Broad Institute)

gp-help@broad.mit.edu

Summary: The k-nearest-neighbor algorithm classifies a sample by assigning it the label most frequently represented among the k nearest samples. There are many references for this type of classifier (with several of the early important papers listed below). No explicit model for the probability density of the classes is formed; each point is estimated locally from the surrounding points. Target classes for prediction (classes 0 and 1) can be defined based on a phenotype such as morphological class or treatment outcome.

The class predictor is uniquely defined by the initial set of samples and marker genes. The k-nearest-neighbor algorithm stores the training instances and uses a distance function to determine which k members of the training set are closest to an unknown test instance. Once the k-nearest training instances have been found, their class assignments are used to predict the class for the test instance by a majority 'vote'.

Our implementation of the k-nearest-neighbor algorithm allows the 'votes' of the k neighbors to be unweighted, weighted by the reciprocal of the rank of the neighbor's distance (e.g., the closest neighbor is given weight 1/1, next closest neighbor is given weight 1/2, etc.), or by the reciprocal of the distance. Either the cosine or euclidean distance measures can be used. The confidence is the proportion of votes for the winning class. 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	Test with saved model	Train/Test
	create a predictive	run a saved model on a	create a model on
	model from a	new test dataset	training data and run it
	training dataset		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
class.filename	No	Required	Required
num.features or	Required	No	Required
feature.list.filename			
weighting.type	No	Required	Required
distance.measure	No	Required	Required
model.file	Required	No	Required
pred.results.file	No	Yes	Yes

Parameters



Name	Description	
train.filename	training data file namegct, .res, .odf type = Dataset ignored if a saved model (saved.model.filename) is used	
train.class.filename	class file for training datacls ignored if a saved model (saved.model.filename) is used	
saved.model.filename	input KNN model fileodf type = KNN Prediction Model	
model.file	name of output KNN model fileodf type = KNN Prediction Model	
test.filename	test data file namegct, .res, .odf type = Dataset	
class.filename	class file for test datacls	
num.features	number of selected features if feature list filename is not specified	
feature.selection .statistc	statistic to use to perform feature selection	
min.std	minimum standard deviation if feature selection statistic includes min std option	
feature.list.filename	features to use for prediction	
num.neighbors	number of neighbors for KNN	
weighting.type	weighting type for neighbors	
distance.measure	distance measure	
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.
- Johns, M. V. (1961) An empirical Bayes approach to non-parametric two-way classification. In Solomon, H., editor, Studies in item analysis and prediction. Palo Alto, CA: Stanford University Press.
- Cover, T. M. and Hart, P. E. (1967) Nearest neighbor pattern classification, IEEE Trans. Info. Theory, IT-13, 21-27, January 1967.



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