Stability Analysis Report: Gene Expression Classifiers

Terek Arce

Original: December 8, 2015
Updated: August 23, 2018

Introduction

Machine learning algorithms (MLAs) are commonly used in gene expression prediction. In this way, given a set of training data with corresponding classes (e.g. disease states), a new gene expression can be classified. Stability of a MLA refers to how well it performs if input data is perturbed. A robust algorithm does not produce a wildly different results for very small changes to the input data. This report describes the stability of five common MLAs used in gene expression classification. A k-nearest neighbor (kNN), support vector machine (SVM), random forest (RF), naive-bayes (NB) and network-based (NBC) classifier were tested under cross-stable and sub-stable testing.

Methods

In this section the experimental setup and implementation is described in detail. The dataset used in testing is described and reasoning for its use provided. In addition to the dataset used, additional datasets are described, for which scripts are included in the base program. Finally, the feature extraction method and MLAs used are described, along with the testing metrics.

Datasets

Stability tests were run against a single lung cancer gene expression dataset obtained from the National Center for Biotechnology (NCBI) Gene Expression Omnibus (GEO). In addition, tow other datasets and their corresponding scripts are provided in the program for future study. All datasets use Affymetrix chips for collecting over 50,000 genes, which are then feature selected for further study.

Lung cancer dataset The lung cancer dataset (GSE19804) is taken from Lu et al. [2010]. It consists of two classes: a normal class with 60 samples and a cancerous class with 60 samples . This dataset was chosen to test classifier accuracy when predicting well separated and distinct classes. The assumption is that selected genes with cancer are sufficiently distinct from those that are non-cancerous.

Colon cancer dataset The colon cancer dataset (GSE39582) is taken from Marisa et al. [2013]. The dataset consists of six classes of colon cancer, which are treated as subclasses for purposed of this study. Subclass 1 contains 116 samples, subclass 2 contains 104 samples, subclass 3 contains 75 samples, subclass 4 contains 59 samples, subclass 5 contains 152 samples, and subclass 6 contains 60 samples. This dataset was chosen to test the accuracy of the classifiers when classes are not well separated and distinct. The assumption is that the selected genes will contain overlapping genes that are involved in the development of colon cancer.

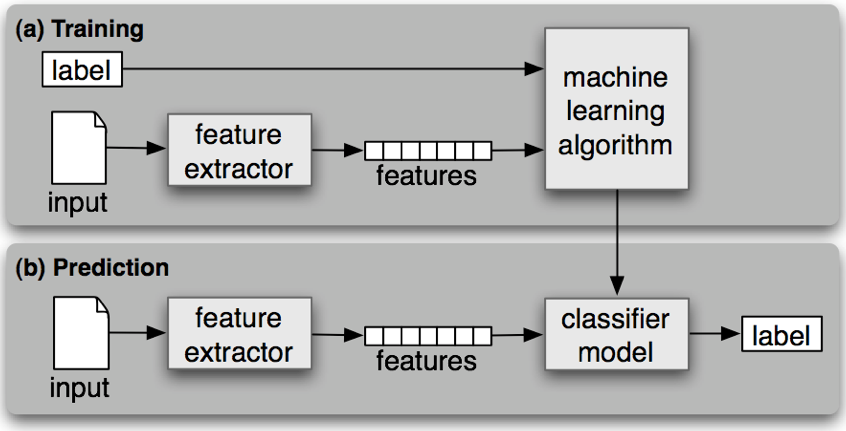

Figure 1: Components of a typical classifier.

Breast cancer dataset The breast cancer dataset (GSE27562) is taken from LaBreche et al. [2011]. The dataset consists of three classes: malignant tumors containing 47 samples, benign tumors containing 27 samples, and a normal class containing 21 samples. This dataset was chosen for its assumed difficulty in distinction between classes. The benign and malignant tumors are assumed to have a large set of overlapping genes, while the normal class is assumed to be sufficiently distinct.

Feature Extraction

Not all genes are relevant for training MLAs. It is common practice to select a subset of genes from the samples for training classifiers with. This helps to improve the runtime and increase the accuracy of results by preventing over-fitting and under-fitting of the data. For most studies, between 50 and 300 genes are normally selected. For this study, the top $k = 50$ genes from the dataset were selected and ranked according to their χ^2 value [Pearson, 1900]. The χ^2 test measures dependence between variables. This allows features that do not contribute to the class to be eliminated.

Machine Learning Algorithms

The choice of machine learning algorithm determines the model constructed for use in classification. Five MLAs are tested in the experiment: kNN, SVM, RF, NB and NBC.

k-Nearest Neighbors Unlike other classifiers, kNN classifiers do not construct a internal model of the class. Instead, samples are classified by a majority vote of its k-nearest neighbors. For purposes of this study, $k = 1$. The nearest neighbor is determined by the Euclidean distance between the expressions.

Support Vector Machine Support vector machines construct hyperplanes to separate features, such that distance is maximized between class data points. The classifier uses the hyperplanes to determine a new samples class by observing which side of the partition the new sample falls under.

RF Random forest classifiers build multiple decision trees, where internal nodes are the splitting criterion and leaf nodes are the classes. The new sample's class is determined by aggregating the results over all the decision trees.

NB Naive Bayes classifiers creates rules based on Bayes theorem to determine the probability and likelihood of a new sample falling into each of several classes. It classifies the sample into the class with the greatest combined probability.

NBC The Network-based classifier builds a function for each gene, expressing its value in terms of its closely correlated neighbors. A correlation cutoff of $\epsilon = 0.8$ is used is all testing. When a new sample is to be classified, the function is used to determine a theoretical gene expression for each class. The error (root mean square error) is determined between the sample's actual expression and theoretical expression, with the sample being classified under the class which results in the smallest error.

Stability Tests

To determine the stability of a classifier two types of tests were performed: cross-stable and sub-stable tests. These tests were chosen, as they describe stability errors that commonly arise in biological experimentation and collection of gene expression data.

Cross-stable Cross-stable testing refers to changing each gene in the testing data by a percentage amount of its original value. Let g_i denote the expression level of the i^{th} gene in a sample. Let p be a percentage by which we wish to alter a genes value. For our tests, $p = \{0.05, 0.10, ..., 0.95, 1.00\}$. For each gene, g_i, we calculate a new gene expression, g_i', where $g_i' = choice(g_i - (g_i * p), g_i + (g_i * p))$. The $choice(x, y)$ function returns either x or y. This test is intended to represent instrument calibration errors, where the error is propagated across all sample genes collected.

Sub-stable Sub-stable testing refers to changing a subset of the genes in the testing data to random values. Let G represent the set of genes expressions in a sample. A subset of genes from G is denoted as G', such that the number of genes in G' is a percentage of the original number of genes in G. Let p represent the set of percentage, where $p = \{0.05, 0.10, ..., 0.95, 1.00\}$. The values of the subset of genes are altered to be $random(max(G), min(G))$, where $max(G)$ and $min(G)$ is the maximum and minimum of the gene expressions over all test samples respectively, and $random(x, y)$ chooses a random value between x and y. The determination of G' is the key step for testing. Three methods of selecting G' are tested: random, χ^2 and greedy. In the random selection strategy, G' is determined by picking random genes, with each incremental percentage subset containing the previous set of genes. In the χ^2 selection strategy, G' is determined by picking the genes in order of their χ^2 values. Lastly, in the greedy selection strategy, the subset G' is grown one gene at a time by selecting the gene that results in the lowest accuracy (after running the classifier) from the feature selected set.

Cross-validation To ensure the validity and generalizable nature of the results, k-fold cross-validation is performed. Briefly, the dataset is partitioned into k subsets, with $k-1$ subsets used to train the classifier and the remaining subset used for testing. The average of the accuracies is reported for each classifier. During cross-validation, only testing data is perturbed. All tests are for $k = 10$.

Code

All code is written in Python and R and targeted for concurrent systems of 2 or more cores. All code also comes in a notebook format for iPython users. The notebooks contain more information on how details of the code. The program consists of two parts. First, the user imports GSE data using a combination of the SIT.R (Series Import Tool) program and custom GSE scripts. The Series Import Tool allows for fast import of GSE data. Once imported, a custom R script is written to properly format data for running the main analysis program. Examples of import scripts for GSEs are provided. The main program, GSA.py, performs analysis of classifiers, outputing a final graph. Python packages for χ^2, kNN, SVM, RF and NB are utilized for testing machine learning methods. The NBC method is implemented according to Ay et al. [2014]. All stability tests are self-implemented. The code is available on GitHub.

Results

The results of cross-stable and sub-stable tests is shown in Figure 2. As shown, kNN is most stable under cross-stable test conditions. This could be a result of test data being well separated. Counter-intuitively,

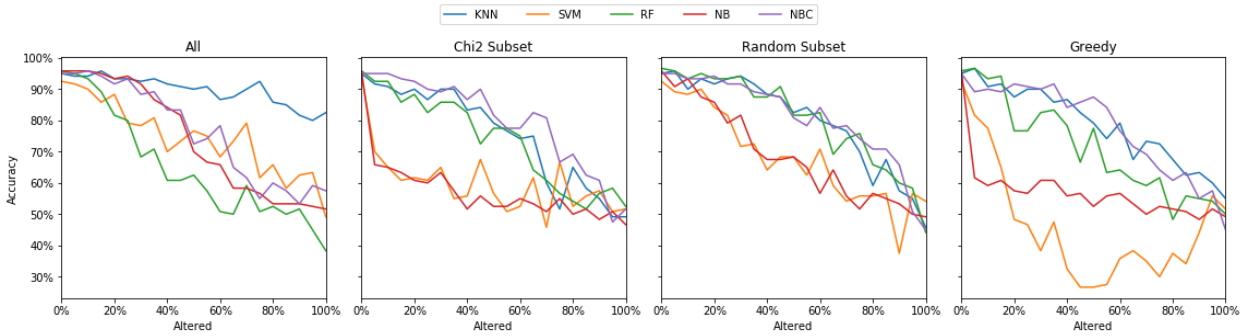

Figure 2: Results of cross-stable and sub-stable testing. The y-axis shows the accuracy given between lowest=0 and highest=100, while the x-axis shows the percentage altered for each gene in the test dataset. The All graph shows cross-stable test results, while the remaining graphs show sub-stable test results.

even when altering all genes, accuracy remained high. This is most likely a problem in the way the genes are modified. Because all genes are changed wither positively or negatively by the same percentage amount, relationships are maintained. This in combination with well separated data would produce consistently high accuracies. For the lung cancer dataset, other classifiers remained stable, however the RF algorithm performed the worst under cross-stable testing, with a sharp drop around 20% alteration. This could be due to the algorithm taking into consideration genes that are not correlated to the actual disease.

The results of sub-stable testing are shown in the last three graphs of Figure 2. Overall NB and SVM performed the worst under sub-stable testing. The heuristic greedy approach and χ^2 algorithm caused the greatest instability in SVM and NB algorithms. In stability under these two selection strategies caused accuracy drops to 50% with less than 5% of genes altered. This could reflect the instability and importance of single gene changes. Study of such genes could be important for future medical studies. There were no significant differences between kNN, RF and NBC under varying selection strategies, with theses showing greatest stability. These tests demonstrate that kNN performs the best under all conditions. The NBC, method is a good alternative, performing as well as others under cross-stable testing and remaining stable under sub-stable testing.

References

Ahmet Ay, Dihong Gong, and Tamer Kahveci. Network-based prediction of cancer under genetic storm. *Cancer informatics*, 13:CIN–S14025, 2014.

Heather G LaBreche, Joseph R Nevins, and Erich Huang. Integrating factor analysis and a transgenic mouse model to reveal a peripheral blood predictor of breast tumors. *BMC medical genomics*, 4(1):61, 2011.

Tzu-Pin Lu, Mong-Hsun Tsai, Jang-Ming Lee, Chung-Ping Hsu, Pei-Chun Chen, Chung-Wu Lin, Jin-Yuan Shih, Pan-Chyr Yang, Chuhsing Kate Hsiao, Liang-Chuan Lai, et al. Identification of a novel biomarker, sema5a, for non–small cell lung carcinoma in nonsmoking women. *Cancer Epidemiology and Prevention Biomarkers*, pages 1055–9965, 2010.

Laetitia Marisa, Aurélien de Reyniès, Alex Duval, Janick Selves, Marie Pierre Gaub, Laure Vescovo, Marie-Christine Etienne-Grimaldi, Renaud Schiappa, Dominique Guenot, Mira Ayadi, et al. Gene expression classification of colon cancer into molecular subtypes: characterization, validation, and prognostic value. *PLoS medicine*, 10(5):e1001453, 2013.

Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302): 157–175, 1900.