Multinomial Classification of Leukemia Subtypes

Naozumi Hiranuma
University of Washington
hiranumn@cs.washington.edu

1 Introduction

Leukemia is a type of cancer caused by defects in blood forming tissues (bone marrow, etc). The most notable characteristic of this disease is an abnormally elevated level of white blood cells. The 10-year survival rate of leukemia still sits around 33%. Leukemia can be classified into numerous subtypes depending on where in the differentiation process of hematopoietic stem-cells defects take place. Knowing which subtype of leukemia a patient has is very important for proper treatment.

Gene expression data is one of the most popular areas to apply machine learning in Biology. A cancer patient usually has a gene expression profile that is very different from a healthy person or a patient with a different type of cancer: cancerous defects can affect multiple biological pathways, altering the gene expression of a patient in unique manner. One can turn this into a classification problem that maps gene profile data to a type of cancer. In this project, I am implementing multinomial classification algorithms to classify gene profile data of leukemia patients into subtypes.

2 Dataset

The data used in this project comes from the MILE (Microarray Innovations in Leukemia) study: the MILE data contains the gene expression profiles of 1237 patients with 17726 features (genes). These samples are already normalized and classified into 4 subtypes (aml:Acute Myeloid Leukemia, CLL:Chronic Lymphocytic Leukemia, cALL:childhood acute lymphoblastic leukemia, mds:Myelodysplastic Syndromes) using existing standard diagnostic technologies. I am using 80% of the data as a training set. The raw data set can be found at the NCBI website under Gene Expression Omnibus Accession No. GSE13204.

3 Algorithm Pipeline

Given the potentially excess number of features, the multinomial classification process in this project contained two phases; feature selection and multionmial classification with the selected features.

3.1 Feature Selection

Three feature selection algorithms were considered

3.1.1 LASSO

5-fold cross validation with L1-regularized logistic regression was performed. The L1-regularized logistic regression was conducted using Scikit-Learn Python module. For each binary classifier for a pair of two subtypes, I found a set of nonzero features that result in the lowest sum of errors, which is defined as $\frac{1}{N}\sum_i |y_i - P(y_i|x_i)|$. The union of the selected features of the binary classifiers were used as a feature set for the following multinomial classification phase.

3.1.2 Variable Ranking

Variable ranking works by ordering variables (features) based on a particular distance metric against the true labels and picking the first n variables as a set of selected features. In this particular project, I used the cosign similarity between a feature vector and the true labels for the distance metric. Then, for each binary classifier, I conducted 5-fold cross validation with deceasing n until a sudden increase in a prediction error rate was observed. SVM was used as a binary classifier in this particular implementation. Again, the union of the selected features were used for the following multinomial classification phas. This algorithm is implemented under **feature_selection.py**.

3.1.3 Recursive Feature Elimination

Recursive feature elimination works by recursively eliminating features that are associated with weights with a low absolute value. In this particular set up, at each recursive step, I conducted 5-fold cross validation and removed bottom 10% of the features that had low weight values. We are justified to do this because the data is normalized. I conducted recursive feature elimination on all possible binary classifiers, and took the union of the selected features as a final output. SVM was used in this particular project. This recursive feature elimination algorithm is implemented under **feature_selection.py**.

3.2 Multinomial Classification

Three multinomial classification algorithms were considered.

3.2.1 1 vs 1 All Pair-wise Approach

The 1-vs-1 all pairwise approach works in the following way. In a training phase, a binary classifier was trained for every pair of classes. Given n classes, this resulted in $\frac{n(n-1)}{2}$ binary classifiers. For this project, I used support vector machine and L2-regularized logistic regression, which were adapted from our homework assignment. The implementation of the binary classifiers can be found in **svm.py** and **logistic_regression.py**.

The classification process takes place in the following way. Let H_{ij} denote a binary classifier for class i and j. Given a data point x, run x through all classifiers H_{ij}. In the case of logistic regression, for every binary classifier, $P(Y = i|x)$ votes were added to class i and $P(Y = j|x)$ votes were added to class j. In the case of support vector machine, 1 vote was added to class i if $predictoin < 0$, 1 vote was added to class j if $prediction > 0$, and 0.5 votes were added to both i and j if $prediction = 0$. Finally, a class with the highest votes were output as prediction. The implementation of this approach can be found under **multiclass.py**

3.2.2 1 vs Rest Approach

Let C be a set of n classes and $i \in C$. The 1-vs-Rest approach works by training n binary classifiers that classifies data points into i and $C - i$, where $i = 1...n$. Similar to the 1-vs-1 approach, I used L2-regularized logistic regression and support vector machine for the binary classifiers. The classification process takes place as follows. Let H_j be a binary classifier that classifies a data point into class j or $C - j$. Then, $\hat{y} = argmax_y H_y$. The implementation can be found under **multiclass.py**.

3.2.3 Softmax Regression

Softmax regression is a generalized version of logistic regression where it aims to classify data points into multiple classes. Given a set of classes C, softmax regression works by picking a pivot class k and training $|C| - 1$ binary logistic classifier that classifies data points into k or $c \in (C - k)$. Knowing that $ln\frac{P(y=c|x)}{P(y=k|x)} = w_c \cdot x$ for all $c \in (C - k)$, we can calculate $P(Y|x)$ as follows:

$$P(Y = c|x) = \frac{exp(w_{c0} + \sum_{i=1}^{k} w_{ci}x_i)}{1 + \sum_{c=1}^{C-1} exp(\sum_{i=1}^{k} w_{ci}x_i)}$$

$$P(Y = k|x) = \frac{1}{1+\sum_{c=1}^{C-1} exp(\sum_{i=1}^{k} w_{ci} x_i)}$$

I used batch gradient descent for my softmax regression implementation. The full implementation of softmax regression can be found under **softmax_regression.py**

4 Implementation

The Python code in this project was mostly implemented by myself including softmax regression, 1v1 multiclass classification, 1-vs-Rest multiclass classification, recursive feature elimination, variable ranking, and cross-validation. The outside modules used were as follows; the Numpy module (of course) and Scikit-learn L1 logistic regression used only in the LASSO feature selection section. L2 logistic regression and support vector machine were adapted from the code written in the assignments.

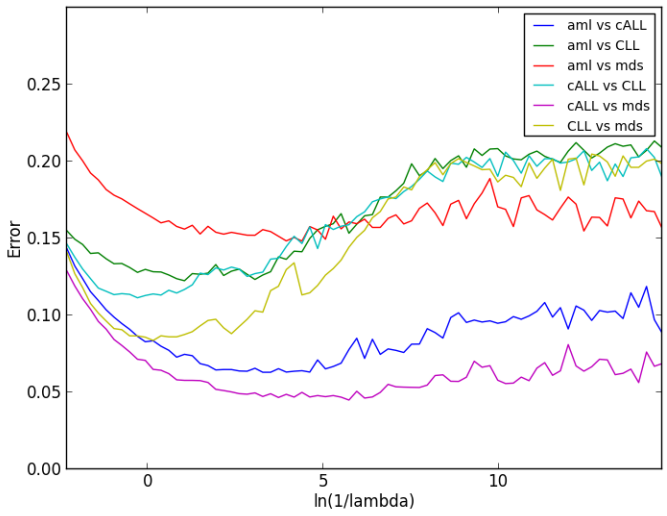
5 Previous Studies

As a part of the MILE project, Haferlach et al. conducted multionomial classification in the following way; For each pair of classes, they trained a support vector machine and selected the top 100 features based on the absolute values of the trained weights (the variable ranking method). Then, they conducted 1v1 all pairwise classification with support vector machine using the union of the selected features for the binary SVMs. With this method, they achieved the accuracy of 88.1%.

The motivation of this project is to further improve the prediction accuracy using the machine learning techniques learned in CSE546. For example, Haferlach et al. did not justify why they selected 100 features for each binary classifiers. What really should have been done is cross-validation tests to find what the right number of features to be selected.

6 Results

6.1 Feature Selection with LASSO

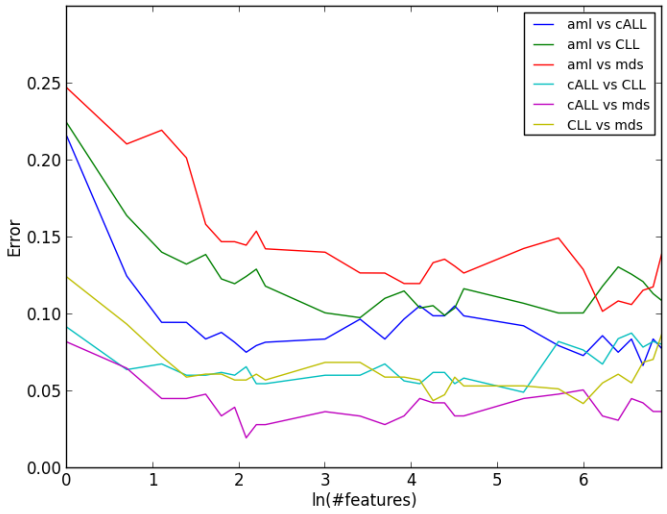


The graph above shows the result of the 5-fold cross validation test on each L1-regularized logistic classifier. The x axis is $ln(\frac{1}{\lambda})$ and y axis is the error defined as $\frac{1}{N}\sum_i |y_i - P(y_i|x_i)|$. As expected, the error values have U-shape with respect to λ. The features were selected for each binary classifier based on λ that resulted in the lowest sum of errors.

The union of the selected features resulted in a set of 2433 features. Of the 2433 selected features, 475 features were selected more than twice by different binary classifiers.

# times selected	0	1	2	3	4	5	6
# features	15293	1958	404	70	1	0	0

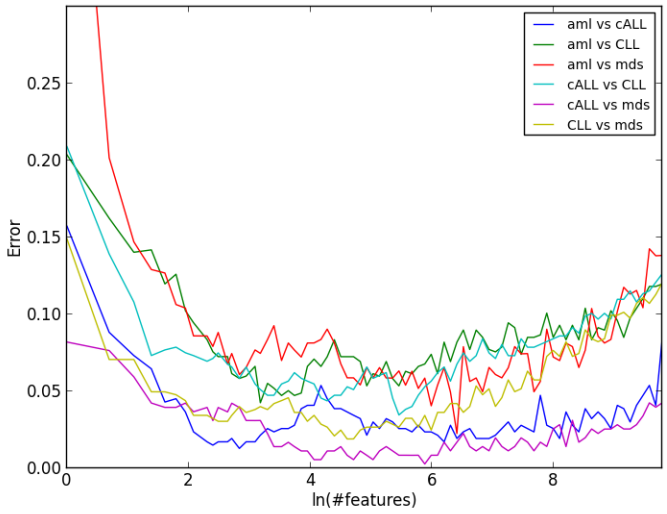
6.2 Feature Selection with Variable Ranking



The graph above shows the result of the 5-fold cross validation test on each svm. The x axis is the log-scaled number of features, and the y axis is the mean of squared errors (MSE) of the predictions, which is simply the rate of making wrong predictions. The MSE of each svm seemed to stabilize at $ln(3) = 20$ features. Therefore, the top 20 features with the highest consign similarities were taken. The union of the selected features resulted in a set of 112 features.

# times selected	0	1	2	3	4	5	6
# features	17614	104	8	0	0	0	0

6.3 Feature Selection with Recursive Feature Elimination



The graph above shows the result of the 5-fold cross validation test on each svm. Similar to the graph in 3.2.3, the x axis is the log-scaled number of features, and the y axis is the mean of squared errors (MSE) of the predictions. The MSE of each svm seemed to sharply increase when the algorithm reduced the number of features down to $ln(3) = 20$. Thus, the remaining 20 features were selected from each classifiers. The union of the 20 features from each svm resulted in a feature set of size 129.

# times selected	0	1	2	3	4	5	6
# features	17614	126	3	0	0	0	0

6.4 Multinomial Classification

The prediction accuracy of the combinations of the feature selection algorithms and multiclass classification algorithms are listed in the table below.

Accuracy	LASSO	Variable Ranking	Recursive Feature Elimination
1-vs-1 SVM	0.856	0.864	0.836
1-vs-1 Logistic Regression	0.864	0.880	0.884
1-vs-Rest SVM	0.82	0.852	0.796
1-vs-Rest Logistic Regression	0.872	0.880	0.888
Softmax Regression	0.856	0.876	0.868

7 Discussion & Conclusion

The results of variable ranking and recursive feature elimination showed that only 20 features from each binary classifiers are needed to achieve the same level of accuracy as Halferlach et al's report, which used the top 100 features from each binary classifiers. Although the LASSO method selected much more features than the other two feature elimination method, it did not improve the accuracy of the multi-class classification, while it suffered from longer computation time due to the large feature size.

One interesting question to ask is whether the selected set of the features (genes) matches our biological knowledge. More specifically, I looked at 6 genes that were selected by all three feature selection methods; EMID1, CFD, FLT3, FPGS, CCL23 and ITFG3. The functions of these genes in conjuction with leukemia is summarized down below.

- EMI Domain Containing Protein 1 (EMID1)
  - No documented function in leukemia.
- Complement Factor D (CFD)
  - CFD is known for playing a crucial role in humoral immune suppression against infectious agents.
  - Lin, Ying-Wei, and Peter D. Aplan reported that, CFD had 4.5 fold decrease in expression in precursor T-cell lymphoblastic lymphoma/leukemia (pre-T LBL) mice with NHD13 over-expressed.
- FMS-like tyrosine kinase-3 (FLT3)
  - FMS-like tyrosine kinase-3 is a gene that is responsible for the development of the hematopoietic and immune systems.
  - Levis, M., and D. Small. reported that FTK3 is usually abnormally expressed in AML patients.
  - Mutations in FLT3 is most likely involved in other hematopoietic diseases other than leukemia as well.
- Folypolyglutamate Synthase (FPGS)
  - Rots, Marianne G., et al. reported that FPGS is differently expressed in different kinds of leukemic cells. (Two-fold decrease of activity in T-ALL and AML than in c/preB-ALL).

- Chemokine ligand 23 (CCL23)
  - CCL23 is known for its overexpression in bone marrow and and peripheral blood cells in AML patients.
  - Patel, et al. reported that CCL23 is important for its inhibitory activity in hematopoietic cells.
  - Gong, Q., et al. reported that CCL23 in combination with PMA promoted the differentiation of a leukemic (U937) cells
- Integrin alpha FG-GAP repeat containing 3 (ITFG3)
  - No documented function in leukemia.

Quite convincingly, four out of the six genes already had scientific reports on its relationship with leukemia. EMID1 and ITFG3, the two genes without any documented association with leukemia, might be good candidates to look for new discovery on the mechanism of leukemia.

Finally and unfortunately, I was unable to make any significant improvement on the prediction accuracy of the multiclass classification on the MILES data. One interesting observation is that multiclass classification with support vector machine seemed to have lower prediction accuracy than that with l2 logistic regression. The highest accuracy was achieved by 1-vs-rest logistic regression with recursive feature elimination. This combination might be the most suitable algorithm pipeline to apply to a similar kind of data set.

8 References

1. Gong, Q., et al. "[Effect of CCL23/myeloid progenitor inhibitory factor 1 (MPIF-1) on the proliferation, apoptosis and differentiation of U937 cells]." Zhongguo shi yan xue ye xue za zhi/Zhongguo bing li sheng li xue hui= Journal of experimental hematology/Chinese Association of Pathophysiology 15.3 (2007): 496-500.
2. Guyon, Isabelle, and Andr Elisseeff. "An introduction to variable and feature selection." The Journal of Machine Learning Research 3 (2003): 1157-1182.
3. Haferlach, Torsten, et al. "Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the International Microarray Innovations in Leukemia Study Group." Journal of Clinical Oncology 28.15 (2010): 2529-2537.
4. Kohlmann, Alexander, et al. "An international standardization programme towards the application of gene expression profiling in routine leukaemia diagnostics: the Microarray Innovations in LEukemia study prephase." British journal of haematology 142.5 (2008): 802-807.
5. Khnl, Andrea, et al. "High BAALC expression predicts chemoresistance in adult B-precursor acute lymphoblastic leukemia." Blood 115.18 (2010): 3737-3744.
6. Levis, M., and D. Small. "FLT3: IT Does matter in leukemia." Leukemia 17.9 (2003): 1738-1752.
7. Lin, Ying-Wei, and Peter D. Aplan. "Gene expression profiling of precursor T-cell lymphoblastic leukemia/lymphoma identifies oncogenic pathways that are potential therapeutic targets." Leukemia 21.6 (2007): 1276-1284.
8. Patel, Vikram P., et al. "Molecular and functional characterization of two novel human CC chemokines as inhibitors of two distinct classes of myeloid progenitors." The Journal of experimental medicine 185.7 (1997): 1163-1172.
9. Rots, Marianne G., et al. "Role of folylpolyglutamate synthetase and folylpolyglutamate hydrolase in methotrexate accumulation and polyglutamylation in childhood leukemia." Blood 93.5 (1999): 1677-1683.