Experiments

Introduction

In the experiments below, we study the ability of classification models to predict the phenotype of the user based on their genotype. That is, using information about the genetic profile of a patient we will predict if a patient has the disease or not (in our experiments, the disease is Melanoma).

Dataset Creation

We generate the dataset we will work with as follows. First, we extract lgen and assoc files using plink software. From the extracted assoc file, we create a new assoc file based on p-values. In our experiments, we keep SNPs with p-value less or equal to 10^-4. Using this data, we create an$M \times N$ matrix, where M is the number of patients and N is the number of SNPs. For each patient-SNP entry in the matrix we give a value 0, 1 or 2. The values are generated from the lgen and assoc files. We check the two alleles of patients from lgen file and compare them with allele1 of assoc file. We give the value 2 when both of lgen' s file alleles are the same as the assoc' s file allele1, value 1 when one of them is the same, and value 0 when neither of them are the same. The matrix will be used as input to the classification models. We will use either all SNPs or a subset of them selected according to some feature-selection technique.

Feature Selection

In our experiments, we use the correlation coefficient of two SNPs in order to select features, and reduce the $M \times N$matrix to a $M \times D$matrix ($D < N$). The correlation coefficient is a metric that measures the dependence of two variables X and Y and it is defined as $\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y}) / \sqrt{\Sigma_{i=1}^{n}(x_i - \bar{x})^2 (y_i - \bar{y})^2}$

The correlation coefficient takes values between -1 and 1, where 1 means perfect correlation, while -1 means negative correlation. Value of 0 implies independence.

We compute the correlation coefficient for all $\binom{N}{2}$, where N is the number of SNPs. We then divide the SNPs into two categories, the low correlation category, and the high correlation category. For a chosen threshold $\theta = 0.7$ in low correlation, belong SNPs which do not have correlation coefficient higher or equal to θ with any other SNP. The rest of SNPs belong to high correlation category. For example, the i-th SNP S_i belongs to the low correlation category if $CORR(S_i, S_j) < 0.7$ for all SNPs S_j ($1 \leq j \leq N$). In this way, we want extract SNPs that behave in a very unique way. Another reason is to avoid the classification overfitting.

To extract the final set of features we will use for the classification, we extract the low-correlation features from the patients (the positive class), and the low-correlation features from the control cases (the negative class).

Evaluation metrics

In our experiments, we use accuracy, recall, precision, F1-score, and AUC to measure classifiers' prediction ability.

- **TRUE NEGATIVES/POSITIVES**

True negative measures the proportion of negatives that are correctly identified as such (e.g. the healthy people who are correctly identified as not having the condition).

True positive measures the proportion of positives that are correctly identified as such (e.g. the sick people who are correctly identified as having the condition).

- **FALSE NEGATIVES/POSITIVES**

False negative measures the proportion of negatives that are wrongly identified as such (e.g. the healthy people who are wrongly identified as not having the condition).

False positive measures the proportion of positives that are wrongly identified as such (e.g. the sick people who are wrongly identified as having the condition).

- **CONFUSION MATRIX**

In predictive analytics, a confusion matrix is a table with two rows and two columns that reports the number of *false positives*, *false negatives*, *true positives*, and *true negatives*. This allows more detailed analysis than mere proportion of correct classifications (accuracy). Accuracy is not a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (that is, when the numbers of observations in different classes vary greatly).

- **RECALL AND PRECISION:**

In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been). In a classification task, a precision score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labeled correctly) whereas a recall of 1.0 means that every item from class C was labeled as belonging to class C (but says nothing about how many other items were incorrectly also labeled as belonging to class C). Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other.

- **AUC (Area Under the Curve):**

The area under the receiver operating characteristic (ROC) curve, known as the AUC, is widely used to estimate the predictive the accuracy of distributional models derived from presence–absence species data. As the output of the different modeling techniques that use binary data as dependent variables produces continuous probabilities of presence (P), where P and 1 – P represent the degree to which each case is a member of one of the two events, a threshold is needed to predict class membership. Thus, the cases above this threshold would be predicted as presences and the remaining cases would be absences. Comparing these binary transformed probabilities with the validation presence-absence data set enables the estimation of four different fractions in a two-by-two confusion matrix: the correctly predicted positive fraction or sensitivity; the correctly predicted negative fraction or specificity;

the falsely predicted positive fraction (commission errors); and the falsely predicted negative fraction (omission errors). These four scores, and other measures of accuracy derived from the confusion matrix, such as the proportion of correct predictions (correct classification rate) and Cohen's kappa (Cohen, 1960), all depend on the discrimination threshold. In order to overcome the supposed subjectivity in the threshold selection process, the ROC curve plots sensitivity as the function of commission error (1 – specificity) as the threshold changes. The calculation of the area under this curve (the AUC score) provides a single-number discrimination measure across all possible ranges of thresholds. This discrimination measure is equivalent to the non-parametric Wilcoxon test (Hanley &McNeil, 1982), in which the rank of all possible pairs for presence and absence assigned probabilities are compared. ROC curves were developed during World War II to assess the performance of radar receivers in signal detection (to estimate the trade-off between hit rates and false alarm rates), and were subsequently adopted in biomedical applications, mainly for comparing the performance of diagnostic tests (Pepe, 2000). Inspite of its wide use and its generally good performance (Bradley,1997), a lot of research effort has recently been devoted to the calculation of AUC scores variations. This is being done to provide a measure of variance or to estimate the AUC's statistical significance (Provost & Fawcett, 2001; Fawcett, 2004; Schröder, 2004; Ferri et al., 2005; Forman & Cohen, 2005). However, some authors have begun to criticize the indiscriminate use of AUC as the standard measurement of accuracy in distribution models. In particular, Austin (2007) warns that 'reliance on AUC as a sufficient test of model success needs to be re-examined'. Agreeing with this concern, we examined some of the characteristics of this measure that question its reliability as a comparative measure of accuracy between model results. We also evaluated its general usefulness in distribution predictive modeling.

- **Accuracy:**

Accuracy is also used as a statistical measure of how well a binary classification test correctly identifies or excludes a condition. That is, the accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined.

- **F score:**

In a statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

Classifiers

Bernoulli Classifier

- **Bernoulli**

In probability theory and statistics, the Bernoulli distribution, named after Swiss scientist Jacob Bernoulli, is the probability distribution of a random variable which takes the value 1 with probability p and the value 0 with probability q=1-p — i.e., the probability distribution of any single experiment that asks a yes-no question; the question results in a boolean-valued outcome, a single bit of information whose value is success/yes/true/one with probability p and failure/no/false/zero with probability q. It can be used to represent a coin toss where 1 and 0 would represent "head" and "tail" (or vice versa), respectively. In particular, unfair coins would have p\neq 0.5. The Bernoulli distribution is a special case of the binomial distribution where a single experiment/trial is conducted (n=1). It is also a special case of the two-point distribution, for which the outcome need not be a bit, i.e., the two possible outcomes need not be 0 and 1.

The decision rule for Bernoulli naive Bayes is based on

$$P(X \mid C_k) = \prod_{i=1}^{n} p_{ki}^{(x_i)}(1 - p_{ki})^{(1-x_i)}$$

where p_{ki} is the probability of class C_k generating the term x_i.

Classifiers

SVM Classifier

- **Support Vector Machines(SVM)**

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Linear

We are given a training dataset of **n** points of the form
$(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_n}, y_n)$
where the y_i are either 1 or −1, each indicating the class to which the point $\overrightarrow{x_i}$ belongs. Each $\overrightarrow{x_i}$ is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\overrightarrow{x_i}$ for which y_i from the group of points for which y_i=-1, which is defined so that the distance between the hyperplane and the nearest point $\overrightarrow{x_i}$ from either group is maximized.

Any hyperplane can be written as the set of points $\overrightarrow{x_i}$satisfying
$\vec{w} * \vec{x} - b = 0$, where $\vec{w}$ is the (not necessarily normalized). This is much like Hesse normal form, except that$\vec{w}$ is not necessarily a unit vector. The parameter $b/\|\vec{w}\|$ determines the offset of the hyperplane from the origin along the normal vector $\vec{w}$.

Hard-margin

If the training data are linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them. These hyperplanes can be described by the equations
$\vec{w} * \vec{x} - b = 1$
and
$\vec{w} * \vec{x} - b = -1$
Geometrically, the distance between these two hyperplanes is $2/\|\vec{w}\|$, so to maximize the distance between the planes we want to minimize $\|\vec{w}\|$. As we also have to prevent data points from falling into the margin, we add the following constraint: for each i either
$\vec{w} * \vec{x} - b \geq 1$ if $y_i = 1$
or
$\vec{w} * \vec{x} - b \leq -1$ if $y_i = -1$
These constraints state that each data point must lie on the correct side of the margin. This can be rewritten as:
$y_i(\vec{w} * \vec{x} - b) \geq 1$, for all 1 <= i <= n. (1)
We can put this together to get the optimization problem:
"Minimize$\|\vec{w}\|$ subject to $y_i(\vec{w} * \vec{x} - b) \geq 1$, for i = 1...n "
The $\|\vec{w}\|$ and b that solve this problem determine our classifier, $\vec{x} \rightarrow sgn(\vec{w} * vexx - b)$.
An easy-to-see but important consequence of this geometric description is that the max-margin hyperplane is completely determined by those $\overrightarrow{x_i}$ which lie nearest to it. These $\overrightarrow{x_i}$ are called "support vectors."

Soft-margin

To extend SVM to cases in which the data are not linearly separable, we introduce the "hinge loss" function,
$max(0,1 - y_i(\vec{w} * \vec{x} - b))$
This function is zero if the constraint in (1) is satisfied, in other words, if $\overrightarrow{x_i}$ lies on the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.

We then wish to minimize

$$[1/n\sum_{i=1}^{n} \quad max(0,1-y_i(\vec{w}*\vec{x}-b))]+\lambda\|\vec{w}\|^2$$

where the parameter λ, determines the tradeoff between increasing the margin-size and ensuring that the $\vec{x_i}$ lie on the correct side of the margin. Thus, for sufficiently small values of λ, the soft-margin SVM will behave identically to the hard-margin SVM if the input data are linearly classifiable, but will still learn if a classification rule is viable or not.

Classifiers

Linear Logistic Regression Classifier

- **Linear Logistic Regression**

In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. This article covers the case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases, where the dependent variable has more than two outcome categories may be analyzed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression. In the terminology of economics, logistic regression is an example of a qualitative response/discrete choice model.

Logistic regression was developed by statistician David Cox in 1958.The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the probability of a given outcome by a specific percentage.

Fields and example applications

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by Boyd et al. using logistic regression. Many other medical scales used to assess the severity of a patient have been developed using logistic regression. Logistic regression may be used to predict whether a patient has a given

disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.). Another example might be to predict whether an American voter will vote Democratic or Republican, based on age, income, sex, race, state of residence, votes in previous elections, etc. The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product. It is also used in marketing applications such as prediction of a customer's propensity to purchase a product or halt a subscription, etc. In economics, it can be used to predict the likelihood of a person's choosing to be in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

The logistic regression can be understood simply as finding the β parameters that best fit:

$$y = \begin{cases} 1\beta_0 + \beta_1\chi + \varepsilon > 0 \\ 0 \text{ else} \end{cases}$$

where ε is an error distributed by standard logistic distribution.

The associated latent variable is $y = \beta_0 + \beta_1\chi + \varepsilon$. The error term ε is not observed, and so the y is also an unobservable, hence termed “latent”. Unlike ordinary regression, however, the β parameters cannot be expressed by any direct formula of the y and x values in the observed data. Instead they are to be found by an iterative search process, usually implemented by software program, that finds the maximum of a complicated “likelihood expression” that is a function of all of the observed y and x values.

Results

For each classifier, we use a 10-fold cross-validation technique. Cross-validation divides the dataset into ten different sets. We run a classifier ten times, each time we use the nine sets for data for training and the remaining one for testing.

We use correlation technique, we explain below and we extract the SNPs belongs to low correlation category and to high correlation category. We give also, all SNPs as a data training in classifiers. We divide the patients into two sets, the case-patients and control-patients. After, using the correlation technique we extract the SNPs belongs to low correlation category for the case-patients’ set. We do the same for control-patients’ set. We create two new sets, we use as data training in our classifiers, the

union and the intersection of the low category. We also make our data of patients' balance. The initial patients are 4980, 1018 are cases and the rest 3962 are controls. We keep 1018 patients who are cases and from 3962 who are controls we keep only 1018 and we do the same work we describe below. We choose 2020 SNPs in random and test the to classifiers. We choose 2020 because the results we have, are with about 2020 SNPs.

The results are represented in the tables below. The first column is reported to one of the experiments we explained above. The next 5 columns are for the evaluation metrics we use to evaluate the classifiers. The last one column is the number of SNPs we extract and use in the experiment. From the results we come to conclusion that low correlation category of SNPs, on Bernoulli classifier gives the best results.

Unbalanced Data

Bernoulli Classifier

	Accuracy	**AUC**	**Recall**	**Precision**	**F_Score**	**SNPs**
Low Correlation Category	0,99	0,97	0,95	1	0,97	2106
High Correlation Category	0,68	0,66	0,61	0,35	0,44	3309
Low Union Category	0,91	0,80	0,61	0,99	0,75	1362
Low Intersection Category	0,98	0,97	0,95	1	0,97	2020
Random SNPs	0,77	0,75	0,71	0,47	0,56	2020
All SNPs	0,75	0,73	0,69	0,43	0,53	5415

Unbalanced Data

SVM Classifier

	Accuracy	AUC	Recall	Precision	F_Score	SNPs
Low Correlation Category	0,96	0,90	0,82	0,98	0,89	2106
High Correlation Category	0,87	0,78	0,63	0,73	0,67	3309
Low Union Category	0,91	0,81	0,64	0,94	0,76	1362
Low Intersection Category	0,96	0,90	0,82	0,97	0,89	2020
Random SNPs	0,92	0,85	0,74	0,88	0,80	2020
All SNPs	0,95	0,89	0,80	0,95	0,87	5415

Unbalanced Data

Linear Logistic Regression Classifier

	Accuracy	AUC	Recall	Precision	F_Score	SNPs
Low Correlation Category	0,93	0,84	0,70	0,94	0,80	2106
High Correlation Category	0,88	0,77	0,58	0,80	0,67	3309
Low Union Category	0,89	0,74	0,49	0,95	0,64	1362
Low Intersection Category	0,92	0,84	0,69	0,94	0,79	2020
Random SNPs	0,90	0,82	0,66	0,86	0,75	2020
All SNPs	0,92	0,85	0,72	0,91	0,80	5415

Balanced Data

Bernoulli Classifier

	Accuracy	AUC	Recall	Precision	F_Score	SNPs
Low Correlation Category	0,96	0,96	0,92	1	0,95	2070
High Correlation Category	0,64	0,64	0,59	0,66	0,62	3345
Low Union Category	0,77	0,77	0,55	0,99	0,71	1361
Low Intersection Category	0,94	0,94	0,89	1	0,94	1909
Random SNPs	0,69	0,69	0,64	0,71	0,67	2020
All SNPs	0,69	0,69	0,64	0,71	0,67	5415

Balanced Data

SVM Classifier

	Accuracy	AUC	Recall	Precision	F_Score	SNPs
Low Correlation Category	0,90	0,89	0,84	0,95	0,89	2070
High Correlation Category	0,77	0,77	0,73	0,80	0,76	3345
Low Union Category	0,82	0,82	0,72	0,90	0,80	1361
Low Intersection Category	0,89	0,89	0,83	0,95	0,88	1909
Random SNPs	0,83	0,83	0,78	0,87	0,82	2020
All SNPs	0,89	0,89	0,86	0,93	0,89	5415

Balanced Data

Linear Logistic Regression Classifier

	Accuracy	AUC	Recall	Precision	F_Score	SNPs
Low Correlation Category	0,86	0,87	0,81	0,91	0,86	2070
High Correlation Category	0,77	0,77	0,74	0,80	0,76	3345
Low Union Category	0,83	0,83	0,73	0,91	0,81	1361
Low Intersection Category	0,85	0,85	0,79	0,90	0,84	1909
Random SNPs	0,81	0,81	0,77	0,84	0,80	2020
All SNPs	0,87	0,87	0,83	0,90	0,86	5415

Top 30 Selected Features

We extract top 30 features based on coefficients using linear regression Linear Regression Logistic. From low correlation category, this category is the one with the best evaluation results of classifiers, we choose 30 SNPs with the highest coefficients in order to compare them with the SNPs you study.

SNP	Chromosome	Allele1	Allele 2
rs191440905	1	A	G
s544033664	1	T	C
rs66744254	1	TA	T
rs538543288	2	G	C
rs531308257	2	T	G
rs531949344	3	A	G
rs201380905	3	TA	T

rs532384961	4	T	C
rs569276476	4	G	C
rs193246249	4	G	A
rs186526104	4	A	G
rs189773785	4	G	A
rs7734488	5	G	A
rs573628213	6	C	A
rs146833935	7	A	C
rs192588608	7	T	C
rs528030187	9	C	T
rs117509963	9	T	C
rs187180589	12	T	C
rs183932003	12	A	C
rs184894937	14	A	G
rs193208238	14	C	T
rs557875920	14	G	A
rs117639881	15	G	A
rs546664832	15	G	T
rs535056530	15	T	G
rs530612130	17	A	G
rs141447005	19	A	G
rs577173142	21	A	G
rs184092415	22	A	G