Τίτλος

Συγγραφέας

Σαλτερής Γεώργιος

Μουλόπουλος Αντώνιος

Διπλωματική Εργασία

Επιβλέπων: Π. Τσαπάρας

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

Abstract

Recently, genome-wide association studies have substantially expanded our knowledge about genetic variants that influence the susceptibility to complex diseases. Although standard statistical tests for each single-nucleotide polymorphism (SNP) separately are able to capture main genetic effects, different approaches are necessary to identify SNPs that influence disease risk jointly or in complex interactions. Experimental and simulated genome-wide SNP data provided by medical department of university of Ioannina afforded an opportunity to analyze the applicability and benefit of several machine learning methods .

Acknowledgements

TABLE OF CONTENTS

Introduction

Fundamental Definitions

Before deep any further into technical issues lets start from the beginning.

As you may know every person has 22 chromosomes each of these chromosomes contains genes and the variations of this genes called alleles.

A gene is a locus of DNA which is made up of nucleotides. The transmission of genes to an organism's offspring is the basis of the inheritance of phenotypic traits. A gene is composed of sequences of DNA nucleotide base pairs (A,T,C and G).

An allele is a variant form of a given gene. Sometimes, different alleles can result in different observable phenotypic traits, such as different pigmentation. We encounter often the terms "Minor Allele ", "Major Allele " and "Risk Allele" .

Major and Minor alleles simply refer to the frequency with which an allele is found in a given population: a Minor allele is one that is expresses less often than a Major one.

On the other hand "risk alleles" are alleles that we know is responsible for commons diseases and usually defined by the minor allele.

Common risk alleles are often detected by genome-wide association studies(GWAS). GWAS are a type of case-control study in which people with the condition being studied are compared to similar people without the condition. Each person's complete set of DNA, or genome, is surveyed by examining a strategically selected area of genetic markers, called single nucleotide polymorphisms (SNPs). The goal is to discover new "risk alleles" responsible for certain diseases.

We often used for various calculation P-values and Z-scores. P-value helps us determine the significance of our result, all hypothesis tests ultimately use a p-value to weigh the strength of the evidence. The p-value is a number between 0 and 1 and interpreted in the following way:

- A small p-value (typicaly<= 0.05) indicates strong evidence of our hypothesis
- A large p-value indicates weak evidence

On the other hand Z-score is a numerical measurement of value's relationship to the mean in a group of values. If a Z-score is 0, it represents the score is identical to the mean score .

Introduction

Introduction to PLINK

PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, large-scale analyses in a computationally efficient manner.

The focus of PLINK is purely on analysis of genotype/phenotype data, plink is being developed by Shaun Purcell whilst at the center for Human Genetic Research(CHGR), Massachusetts General Hospital (MGH), and the Broad Institute of Harvard & MIT.

We use plink in order to manipulate our initial data files and export data files we can read in order to continue our analysis .

Data

Initial Data Sets

In our disposal we have 22 Chromosomes of (number of peoples) persons, for each chromosome we have a .bed, .fam, and .bim file .

- **BIM:**

  Bim file has 6 columns . From left to right each column has the number of chromosome the file refers to, the SNP code , the SNP position , the base-pair, the minor allele, and the major Allele

- **FAM:**

  FAM has also 6 columns and each column contains from left to right: Family ID, patient ID(in our data sets we have not any family relations so the code in the 2 first columns is the same), Id of the father, Id of the mother, genre of the patient, and phenotype information

- **BED:**

  BED file contains all the genetic information and is under 16-bit coding.

We also have in our disposal a file with phenotype information of 4980 patient which use for our analysis as test and train.

Data

Data Manipulation

BED File contains all the important pieces of information but because of his coding we cannot export any useful conclusion out of it, so the main reason we use plink is to get the ASSOC and LGEN file which use in our training.

ASSOC

To get the ASSOC we first run on the terminal the command " plink –bfile chr(number of chromosome) –allow-no-sex –out chr(number of chromosome) –1 –pheno (phenotype file) –assoc".

CHR	**Chromosome code**
SNP	**Variant identifier**
BP	**Base-pair coordinate**
A1	**Allele 1 (usually minor)**
F_A	**Allele 1 frequency among cases**
F_U	**Allele 1 frequency among controls**
A2	**Allele 2**
CHISQ	**Allelic test chi-square statistic. *Not present with 'fisher'/'fisher-midp' modifier.***
P	**Allelic test p-value**
OR	**odds(allele 1 | case) / odds(allele 1 | control)**

Table 1

The ASSOC file exists in the above format where the CHR represents the Chromosome Code the we have the Variant identifier, Base-pair coordinate, Allele 1 (usually the minor one), his Frequency among cases , his frequency among controls , Allele 2, Allelic test chi-square, P-value, and the odds.

LGEN

We also get the LGEN file which contains The IID of the patient his SNP and minor and major allele. The LGEN file exists in the following format:

Family ID
Within-family ID
SNP
Allele 1
Allele 2

Table2

After we get the LGEN file we encoding the SNPs for each patient along with the ASSOC file.

If the allele in the LGEN file is the same as both the Minor and Major in ASSOC we then give it the value 2 , if it’s same with one of it we give him the value 1, finally if its not same with either we give him the value 0.

Classifier Methods

SNPs Selection

First, we calculate the correlation of SNPs and divide them into two categories the low correlation and the high correlation category. Correlation is a metric that shows us the dependence of two variables. Correlation formula of two SNPs is defined as COR(X,Y) = COV(X,Y) / VAR(X)*VAR(Y) = $\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})/\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2(y_i-\bar{y})^2}$. In low correlation, belong SNPs which do not have correlation equal or higher to 0.7 with no one SNP. The rest of SNPs belong to high correlation category. Properties os Correlation:

- $-1 \leq COR(X,Y) \leq 1$
- COR(X,Y) = COR(Y,X)
- COR(X,X) = 1

Classifier Methods

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

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

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

● 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 $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$ where the y_i are either 1 or −1, each indicating the class to which the point $\vec{x}_i$ belongs. Each $\vec{x}_i$ is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\vec{x}_i$ 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 $\vec{x}_i$ from either group is maximized.

Any hyperplane can be written as the set of points $\vec{x}_i$ 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, $\vec{x} \mapsto sgn(\vec{w} * vex\, x - b)$.

An easy-to-see but important consequence of this geometric description is that the max-margin hyperplane is completely determined by those $\vec{x}_i$ which lie nearest to it. These $\vec{x}_i$ 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 $\vec{x}_i$ 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

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

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.

Classifier Methods

Linear Logistic Regression Classifier

● Linear Logistic Regression

In statistics, logistic regression, or logit regression, or logit model[1] 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.[4] 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 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 train and the remaining one as data test. We extract results about accuracy, AUC, recall, precision and F measure. Accuracy is the percentage o success prediction. Recall is given by true positives / (true positives + false negatives) and precision is given by true positives / (true positives + false positives). True positive is the predict about a patient that has the illness, and really he has it. False positive is the predict about a patient that has the illness and in fact, he does not have it. A false negative is the predict about a patient that has not the illness and in fact, has it.

Results

Bernoulli Results

	Accuracy	AUC	Recall	Precision	F_Score
Low	0,99	0,96	0,93	1,0	0,96
High	0,68	0,65	0,60	0,34	0,44
All	0,75	0,73	0,70	0,43	0,54

Results

SVM Results

	Accuracy	AUC	Recall	Precision	F_Score
Low	0,96	0,80	0,83	0,97	0,89
High	0,87	0,78	0,64	0,72	0,67
All	0,95	0,90	0,82	0,96	0,88

Results

Linear Logistic Regression Results

	Accuracy	AUC	Recall	Precision	F_Score
Low	0,93	0,84	0,70	0,94	0,80
High	0,88	0,77	0,60	0,78	0,68
All	0,93	0,85	0,73	0,91	0,81