Introduction to Machine Learning **Fall Semester, 2014/15**

Project Guidelines

Submission

Submission Date: 8/3/15. Submission is in groups of up to **3** students.

The base grade will be computed as 101-*rank*, where *rank* is your relative rank amongst the other groups.

From this grade points will be removed for technical errors, lack of description, missing details in the report, or reports that don't adhere to the format. In some cases, points may be added for truly elegant and promising solutions.

Background

The DNA in our cells contains long chains of four chemical building blocks – adenine, thymine, cytosine, and guanine, abbreviated A, T, C, and G. More than 6 billion of these chemical bases, strung together in 23 pairs of chromosomes, exist in a human cell. These genetic sequences contain information that influences our physical traits, our likelihood of suffering from disease, and the responses of our bodies to substances that we encounter in the environment.

The genetic sequences of different people are remarkably similar. When the chromosomes of two humans are compared, their DNA sequences can be identical for hundreds of bases. But at about one in every 1,200 bases, on average, the sequences will differ. Differences in individual bases are by far the most common type of genetic variation. One person might have an A at that location, while another person has a G. These genetic differences are known as single nucleotide polymorphisms, or SNPs (pronounced "snips"). There are approximately 10 million SNPs estimated to occur commonly in the human genome. Each distinct "spelling" of a chromosomal region is called an allele, and a collection of alleles in a person's chromosomes is known as a genotype.

In the most common case, there are only two alleles for all population at each SNP position. Data describing the genotype data for individuals, often does not specify the bases explicitly. Instead, one allele (per position) is selected as a reference allele. Then, at that position, the number of non-reference alleles is presented: 0 if both alleles in that position, in the chromosome pair, were identical to the reference allele for that position; 1 if only one of them was the reference allele; and 2 if neither were the reference alleles.

Problem Specification

You are given the SNPs for $N_{train} = 600$ people. For each person, $S = 165229$ SNPs are given. In addition, you are given the SNPs for $N_{test} = 400$ different people. Of those, $K = 300$ SNPs are missing. You are required to impute (predict) the missing SNPs for the test individuals, given a classifier that you will train on the training individuals.

Data Description

All the data described below is zipped and is available at:

```
~schweiger/courses/ML/project.zip
```

The full data is available in two files: `train.txt` and `test.txt`. The can both be loaded with:

```
train = dlmread('train.txt');
test = dlmread('test.txt');
```

You can read a subset of these arrays with extra arguments for `dlmread`. For the relevant subsets, also see below.

The file `train.txt` contains an array of size $S \times N_{train} = 165229 \times 600$: It contains the genotype data for $N_{train} = 600$ individuals. There are $S = 165229$ SNPs in this data. Each line corresponds to a SNP, so therefore there are $S = 165229$ lines. The SNPs are scattered along the chromosomes, and generally there are gaps between them (measured in DNA bases). The SNPs are given in their order along the chromosomes.

Each line contains a list of $N_{train} = 600$ space-separated numbers. Each number may be $0, 1$ or 2, depending on the number of non-reference alleles we have seen for this person, for this position.

The file `test.txt` is identical to that of `train.txt`, with one difference. There are $K = 300$ SNPs which are missing; in their respective lines, instead of $0, 1, 2$, the values are -1. You are required to predict these values, for each one of the missing M SNPs, for each one of the N individuals. The indices of the missing SNPs is given in the array `missing`, specified below.

The file `positions.txt` contains the positions of all of the SNPs along the chromosomes. Each line corresponds to a SNP, and contains the chromosome on which the SNP resides, and its position (in bases) in the chromosome. Don't worry if you're not familiar with all the types of chromosomes. Using this information is by no means obligatory, but you may find it useful.

The rest of the data, described below, is available as a Matlab data file, and may be loaded with:

```
load dataforproject.mat
```

For your convenience, the array `missing` ($1 \times K = 1 \times 300$) contains the line numbers (1, not 0, being the number of the first index) of the missing SNPs.

In addition, the arrays `extracted_train, extracted_test` are of size $K \times 201 \times N_{train}/N_{test} = 300 \times 201 \times 600/400$ - for each one of the $K = 300$ missing SNPs, a matrix of the window of size 201 around the given SNPs (100 SNPs before and 100 SNPs after). You are not obligated to use these arrays, but this may be more convenient for some, and may save long loading times.

In summary:

- `train.txt` - size 165229×600 - training data
- `test.txt` - size 165229×400 - test data, with missing SNPs
- `positions.txt` - size 165229 - positions of the SNPs
- In `dataforproject.mat`:
  - `missing` - size 1×300 - indices of missing SNPs
  - `extracted_train` - size $300 \times 201 \times 600$ - extracted sections of SNPs around the missing SNPs
  - `extracted_test` - size $300 \times 201 \times 400$ - extracted sections of SNPs around the missing SNPs

What to Submit

A single zip file called `X.Y.Z.zip` (where `X`, `Y`, and `Z` are the ID numbers of the submitting students) containing the following:

1. A matrix `ytest` of size $K \times N_{test} = 300 \times 400$, of the predicted missing SNPs, saved as `ytest.mat`.

2. A fully documented Matlab code, which includes the script `go.m` that starts with loading the data (`load dataforproject.mat` etc.) and ends with saving your solution (`save ytest.mat ytest`).

3. A file called `readme.txt` or `readme.pdf` that explains your solution and also documents the other alternatives you have tried during the development process. For each alternative (including the submission one), please provide: (1) a script of the form `gogo_alt1.m` (2) the performance score you used during development in order to compare the alternatives. Make sure to detail what success rate you were using during development. You can use a table as the one below:

Script Name	Description Summary	Success Rate
`go.m`	PCA (RBF, gamma=0.1, dim=20) followed by AdaBoost. See section 2.1 of my report.	0.87
`go_alt1.m`	Feature standardization followed by polynomial SVM d=4. Positions were used for... See section 2.2 of my report.	0.56
...	...	...

You should submit according to the submission guidelines of the HW as published in the course web site. In additional to a hard copy of the files submitted to the checkers mailbox, submit your zip file to `ml.intro.2014@gmail.com`. The subject of the email should be "Final Project X.Y.Z").

You should also follow carefully the programming assignment guidelines. Any official and publicly available software package may be used as long as an exact reference is indicated and usage instructions are detailed. All other software must be original and included in your submission.