# Hw7: Converting multiclass to binary classification task
# Due: 11:45pm on Feb 25, 2010

The example files are under dropbox/09-10/572/hw7/examples/.

**Q1 (35 points):** Build a tool **q1.sh** that calls MaxEnt trainer/decoder in Mallet to handle *multiple* classes using **one-vs-all**, pretending that the trainer/decoder could handle only two classes.

- The command line is: q1.sh training_data test_data output_dir > acc_file
- training_data and test_data are in standard Mallet format: "instanceName goldClass f1 v1 f2 v2 ..." (e.g., **train.txt** and **test.txt**).
- acc_file is the accuracy file, and it has the same format as in Hw2-Hw5.
- output_dir is the output directory, which includes the following:
  - class_map (e.g., **ex1/class_map**): the file has the format "class-name class-index": class-name is the class name in the training_data, and class index is a number that starts from 1. For instance, the first class name in the training_data will have index 1, the second will have index 2, and so on. This file is used to determine what class name the class index "m" in m-vs-all (see below) refers to.
  - For each classifier (say m-vs-all, m is a class index), there should be a subdirectory called m-vs-all. For instance, if the training_data has three classes, there should be three subdirectories: 1-vs-all, 2-vs-all, and 3-vs-all. Each subdirectory should include the following files:
    - A training data file called **"train"**, which has the same format as training_data, and the goldClass in "train" is 1 or -1. This is the training file for the classifier m-vs-all.
    - Similarly, a test file called **"test"**.
    - A file called **"sys_output"** that contains the classification results when running the classifier m-vs-all on test_data. The format is the same as sys_output files in Hw2-Hw5; that is, the format is "instanceName goldClass c1 p1 c2 p2", where goldClass and $c_i$ are "1" or "-1", and $p_i$ is the probability $P(c_i \mid x)$ based on the classifier.
  - Under output_dir, there should be a file called **"final_sys_output"** (e.g., **ex1/final_sys_output**). This file has the format "InstanceName goldClassName [cn1 p1 cn2 p2 ...]", where $cn_i$ is a class name, and $p_i$ is the probability $P(class = 1 \mid x)$ when running the classifier for i-vs-all. The brackets "[...]" indicate that the $(cn_i, p_i)$ pairs in each line are sorted according to $p_i$ in descending order. Note that since $p_i$ comes from different classifiers, $\sum_i p_i$ is not necessarily equal to one.

**Q2 (35 points):** Build a tool **q2.sh** that calls MaxEnt trainer/decoder in Mallet to handle *multiple* classes using **all-pair**, pretending the trainer/decoder could handle only two classes.

- The command line is: q2.sh training_data test_data output_dir > acc_file

- The format of the files are the same as in Q1 except the following:
  - The subdirectory **m-vs-all** in Q1 becomes **m-vs-n** in Q2, where m and n are class indices (e.g., 1-vs-2, 1-vs-3).
  - The file **"final_sys_output"** has the format "InstanceName goldClassName sysClassName 1:2 $p_{1,2}$ 1:3 $p_{1,3}$ ... i:j $p_{i,j}$ ... [c1=n1 c2=n2 ...]", where $p_{i,j}$ is the probability that x belongs to class 1 (instead of -1) according to the classifier for the class pair (i,j). $n_i$ is the number of times class $c_i$ wins, and "[...]" indicates that the list of $c_i$=$n_i$ is sorted by the value of $n_i$ in descending order. An example file is in **ex2/final_sys_output**.
  - When there is a tie w.r.t. the number of games a class wins, choose the class with the smallest class index.

**Q3 (25 points)** Use **train.txt** and **test.txt** as the training and test data. Run Mallet and commands in Q1 and Q2 to fill out Table 1.

- Row 1 is the training and test accuracy when using Mallet commands to handle the data directly; that is, do not pretend that Mallet could handle only two classes.
- Row 2 is the accuracy when running q1.sh
- Row 3 is the accuracy when running q2.sh

Table 1: Training and test accuracy

| | training acc | test acc |
|---|---|---|
| Run Mallet directly | | |
| Run q1.sh (one-vs-all) | | |
| Run q2.sh (all-pairs) | | |

**Q4 (5 points)** What conclusions can you draw from Table 1?

**Submission:** Submit a tar file via CollectIt. The tar file should include the following.

- If your team has two people, please submit only one copy. In your note file, please list the names of team members.
- In your note file hw7.*, include your answers to Q3 and Q4, and any notes that you want the TA to read.
- The source code for Q1 and Q2.
- The output_dir created for Q3: q1_res/ is the output_dir when running q1.sh and q2_res/ is the one when running q2.sh.
- The acc_file should be put under q*_res/: e.g., q1_res/acc_file is the acc_file when running q1.sh.