Introduction to Data Mining: Assignment #1
Summer 2020

Due: Mar 14th, 23:59:59 CST (UTC +8).

1. Machine Learning Problems

(a) Choose proper word(s) from

A) *Supervised Learning* B) *Unsupervised Learning* C) *Not Learning*
D) *Classification* E) *Regression* F) *Clustering* G) *Dimensionality Reduction*

to describe the following tasks.

1) Automatically group thousands of art paintings by similar artistic styles.
2) Play sudoku[1] by searching the whole action space to find the possible solution.
3) Recognize handwritten digits by looking for the most similar image in a large dataset of labeled digit images, then use its label as result.
4) Visualize very high dimensional data in 2D or 3D space.
5) Based on former patients' records, predict the success rate of a surgery for a new patient.
6) Given thousands of peoples' names and sexes, decide a new person's name is male or female.
7) Discover communities of people in a social network.
8) Using historical stock prices, predict stock price in the future.
9) Represent image as a well chosen 64 bits integer, so that similar images will be represented as integers with small hamming distance.

(b) True or False: "To fully utilizing available data resource, we should use all the data we have to train our learning model and choose the parameters that maximize performance on the whole dataset." Justify your answer.

2. Bayes Decision Rule

(a) Suppose you are given a chance to win bonus grade points:

> There are three boxes. Only one box contains a special prize that will grant you 1 bonus points. After you have chosen a box B_1 (B_1 is kept closed), one of the two remaining boxes will be opened (called B_2) such that it **must not** contain the prize (note that there is at least one such box).
>
> Now you are are given a second chance to choose boxes. You can either stick to B_1 or choose the only left box B_3. What is your best choice?

[1] `https://en.wikipedia.org/wiki/Sudoku`

(i) What is the prior probability of B_1 contains prize, $\mathrm{P}(B_1 = 1)$?

(ii) What is the likelihood probability of B_2 does not contains prize if B_1 contains prize, $\mathrm{P}(B_2 = 0|B_1 = 1)$?

(iii) What is the posterior probability of B_1 contains prize given B_2 does not contain prize, $\mathrm{P}(B_1 = 1|B_2 = 0)$?

(iv) According to the Bayes decision rule, should you change your choice or not?

(b) Now let us use bayes decision theorem to make a two-class classifier. Please refer the codes in the *bayes_decision_rule* folder and main skeleton code is *run.m/run.ipynb*. There are two classes stored in *data.mat*. Each class has both training samples and testing samples of 1-dimensional feature $\mathbf{x}$.

(i) Finish the calculation of likelihood of each feature given particular class(in *likelihood.m/likelihood.py*). And calculate the number of misclassified test samples(in *run.m/run.ipynb*) using maximum likelihood decision rule. Show the distribution of $\mathrm{P}(x|\omega_i)$, and report the test error.

(ii) Finish the calculation of posterior of each class given particular feature(in *posterior.m/posterior.py*). And calculate the number of misclassified test samples(in *run.m/run.ipynb*) using optimal bayes decision rule. Show the distribution of $\mathrm{P}(\omega_i|x)$, and report the test error.

(iii) There are two actions $\{\alpha_1, \alpha_2\}$ we can take, with their loss matrix below. Show the minimal total risk ($R = \sum_x \min_i R(\alpha_i|x)$) we can get.

$\lambda(\alpha_i\|\omega_j)$	$j=1$	$j=2$
$i=1$	0	1
$i=2$	2	0

3. Gaussian Discriminant Analysis and MLE

Given a dataset $\{(\mathbf{x}^{(i)}, y^{(i)}) \mid \mathbf{x} \in \mathbb{R}^2, y \in \{0,1\}, i = 1, ..., m\}$ consisting of m samples. We assume these samples are independently generated by one of two Gaussian distributions:

$$p(\mathbf{x}|y=0) = N(\mu_0, \Sigma_0) = \frac{1}{2\pi\sqrt{|\Sigma_0|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_0)^T \Sigma_0^{-1}(\mathbf{x}-\mu_0)}$$
$$p(\mathbf{x}|y=1) = N(\mu_1, \Sigma_1) = \frac{1}{2\pi\sqrt{|\Sigma_1|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_1)^T \Sigma_1^{-1}(\mathbf{x}-\mu_1)}$$

the prior probability of y is

$$p(y) = \phi^y (1-\phi)^{1-y} = \begin{cases} \phi & y = 1 \\ 1-\phi & y = 0 \end{cases}.$$

The code of this section is in the *gaussian_discriminant* folder.

(a) Given a new data point $\mathbf{x} = (x_1, x_2)$, calculate the posterior probability

$$p(y = 1|\mathbf{x}; \phi, \mu_0, \mu_1, \Sigma_0, \Sigma_1).$$

To simplify you calculation, let $\Sigma_0 = \Sigma_1 = \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I, \phi = \frac{1}{2}, \mu_0 = (0, 0)^T, \mu_1 = (1, 1)^T$. What is the decision boundary?

(b) An extension of the above model is to classify K classes by fitting a Gaussian distribution for each class, i.e.

$$p(\mathbf{x}|y = k) = N(\mu_k, \Sigma_k) = \frac{1}{2\pi\sqrt{|\Sigma_k|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1} (\mathbf{x}-\mu_k)}$$
$$p(y = k) = \phi_k, \text{ where } \sum_{k=1}^{K} \phi_k = 1$$

Then we can assign each data points to the class with the highest posterior probability. Your task is to finish *gaussian_pos_prob.m/gaussian_pos_prob.py*, that compute the posterior probability of given datasets X under the extended model. See the comments in *gaussian_pos_prob.m/gaussian_pos_prob.py* for more details.

(c) Now let us do some field work – playing with the above 2-class Gaussian discriminant model. For each of the following kind of decision boundary, find an appropriate tuple of parameters $\phi, \mu_0, \mu_1, \Sigma_0, \Sigma_1$. Turn in the code *run.m/run.ipynb* and the plot of your result in your homework report.

(i) A linear line.

(ii) A linear line, while both means are on the same side of the line.

(iii) A parabolic curve.

(iv) A hyperbola curve.

(v) Two parallel lines.

(vi) A circle.

(vii) An ellipsoid.

(viii) No boundary, i.e. assigning all samples to only one label.

(d) Given a dataset $\{(\mathbf{x}^{(i)}, y^{(i)}) \mid y \in \{0, 1\}, i = 1, \cdots, m\}$, what is the maximum likelihood estimation of ϕ, μ_0 and μ_1? (Optionally, you are encouraged to compute the MLE for all the other parameters Σ_0, Σ_1, and generalize to the K-class gaussian model. This will be challenging but rewarding[2].)

4. Text Classification with Naive Bayes

In this problem, you will implement a text classifier using Naive Bayes method, i.e., a classifier that takes an incoming email message and classifies it as positive (spam) or negative (not-spam/ham). The data are in *hw1_data.zip*. Since MATLAB is not good at text

[2]You may want to look back when we learn GMM in the future.

processing and lacks of some useful data structure, TA has written some Python scripts to tranform email texts to numbers that MATLAB can read from. The skeleton code is *run.m/run.ipynb*(in *text_classification* folder).

In this assignment, instead of following TA's Python scripts and *run.m/run.ipynb*, you can use any programming language you like to build up a text classifier barely from email texts. You are more encouraged to finish the assignment in this way, since you will get better understanding of where the features come from, what is the relationship between label, emails and words, and other details.

Here are some tips you may find useful:

i) **Relationship between words, document and label**. Theoretically, $P(word_i = N|\text{SPAM}) = P(word_i = N|\text{document-type}_j)P(\text{document-type}_j|\text{SPAM})$ should hold, where document-type_j is the type of the document e.g. a family email will have more words about family members and house, a work email will have more words about bussiness and a game advertising email will have words like "play now". But we can not include all the document types (a not big enough data set) and that is not what naiye bayes cares(we will learn PLSA in the near future). For simplification, in traning we discard the documents information and mix all the words to generate $P(word_i|\text{SPAM})$ and $P(word_i|\text{HAM})$ denoting the possibility for a word in SPAM/HAM email to be $word_i$. Therefore $P(word_i = N|\text{SPAM}) = P(word_i|\text{SPAM})^N$.

ii) **Training**. Remember to add Laplace smoothing.

iii) **Testing**. When you compute $p(\mathbf{x}|y) = \prod_i p(x_i|y)$, you may experience floating underflow problem. You can use logarithm to avoid this issue..

(a) It is usually useful to visualize you learnt model to gain more insight. List the top 10 words that are most indicative of the SPAM class by finding the words with highest ratio $\frac{P(word_i|\text{SPAM})}{P(word_i|\text{HAM})}$ in the vocabulary.

(b) What is the accuracy of your spam filter on the testing set?

(c) True or False: a model with 99% accuracy is always a good model. Why? (Hint: consider the situation of spam filter when the ratio of spam and ham email is 1:99).

(d) With following confusion matrix[3]:

	Spam(label)	Ham(label)
Spam(predict)	TP	FP
Ham(predict)	FN	TN

[3]Positive and negative often substitutes as predictions in two labels problem. They are usually defined implicitly by common sence. A xxx-detector will use positive for the presence of xxx and negative for the absence of xxx, eg doping detection in the Olympics will mark atheles taking doping as posible and otherwise negative. In TP/FP/TN/FN terminology, T stands for 'True' which means predict is the same as label whihe F stans for 'False'. And P stands for 'Positive' and N stands for 'Negative'. `http://en.wikipedia.org/wiki/Precision_and_recall`

compute the precision and recall of your learnt model, where precision $= \frac{tp}{tp+fp}$, recall $= \frac{tp}{tp+fn}$

(e) For a spam filter, which one do you think is more important, precision or recall? What about a classifier to identify drugs and bombs at airport? Justify your answer.

Please submit your homework report to at `http://courses.zju.edu.cn:8060/course/11827/` in pdf format, with all your code in a zip archive.