

# Assignment #1

## 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. B F 1) Automatically group thousands of art paintings by similar artistic styles.
2. C 2) Play sudoku<sup>1</sup> by searching the whole action space to find the possible solution.
3. A D 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. B G 4) Visualize very high dimensional data in 2D or 3D space.
5. A E 5) Based on former patients' records, predict the success rate of a surgery for a new patient.
6. A D 6) Given thousands of peoples' names and sexes, decide a new person's name is male or female.
7. B F 7) Discover communities of people in a social network.
8. B E 8) Using historical stock prices, predict stock price in the future.
9. A E 9) Represent image as a well chosen 64 bits integer, so that similar images will be represented as integers with small hamming distance.

False  
(overfitting)

(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 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?

(i) What is the **prior probability** of  $B_1$  contains prize,  $P(B_1 = 1)$ ?  $P(B_1=1)=1/3$

<sup>1</sup><https://en.wikipedia.org/wiki/Sudoku>

- (ii) What is the **likelihood probability** of  $B_2$  does not contains prize if  $B_1$  contains prize,  $P(B_2 = 0|B_1 = 1)$ ?  $P(B_2=0|B_1=1) = 1$
- (iii) What is the **posterior probability** of  $B_1$  contains prize given  $B_2$  does not contain prize,  $P(B_1 = 1|B_2 = 0)$ ?  $P(B_1=1|B_2=0)=P(B_2=0|B_1=1)*P(B_1=1)/P(B_2=0)=1*1/3/1=1/3$   
 特别要注意因为每次主持人都打开没有bonus的盒子, 所以其实 $P(B_2=0)=1$
- (iv) According to the Bayes decision rule, should you change your choice or not?  $P(B_1=1|B_2=0)+P(B_3=1|B_2=0)=1$  所以 $P(B_1=1|B_2=0)=1/3$
- (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*. 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*). And calculate the number of misclassified test **samples**(in *run.m*) using **maximum likelihood decision rule**. Show the distribution of  $P(x|\omega_i)$ , and report the test error.  
 $\text{likelihood error} = 64$
- (ii) Finish the calculation of **posterior** of each class given particular feature(in *posterior.m*). And calculate the number of misclassified test samples(in *run.m*) using **optimal bayes decision rule**. Show the distribution of  $P(\omega_i|x)$ , and report the test error.  
 $\text{posterior error} = 47$
- (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.  
 $\text{risk} = 2.44$

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

### 3. Gaussian Discriminant Analysis and MLE

类似于二分类问题,  $\mathbf{x}$ 是feature,  $y$ 是class

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

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

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

likelihood: 每一类别的feature的分布

the prior probability of  $y$  is

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

先验概率

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**

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

$$\begin{aligned} P(y=1|\mathbf{x}) &= P(\mathbf{x}|y=1) * P(y=1) / P(\mathbf{x}) \\ &= P(\mathbf{x}|y=1) * P(y=1) / [P(\mathbf{x}|y=0) * P(y=0) + P(\mathbf{x}|y=1) * P(y=1)] \\ &= \frac{1/(2\pi) * e^{-1/2 * [(x_1-1)^2 + (x_2-1)^2]}}{1/(2\pi) * e^{-1/2 * [(x_1-1)^2 + (x_2-1)^2]} * 0.5 + 1/(2\pi) * e^{-1/2 * [(x_1-1)^2 + (x_2-1)^2]} * 0.5} \end{aligned}$$

To simplify your 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?  $P(x|y=0)P(y=0) = P(x|y=1)P(y=1)$  时的  $x$  的值为 decision boundary  $P(y=0)=P(y=1)=0.5$   
 $P(x|y=0)=1/(2\pi)*e^{-1/2*(x^2+x^2)}$   $P(x|y=1)=1/(2\pi)*e^{-1/2*((x-1)^2+(x-1)^2)}$   
 所以得出  $x_1+x_2=1$  时为 decision boundary

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

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

$$p(y=k) = \phi_k, \text{ where } \sum_{k=1}^K \phi_k = 1$$

$P(y=k|x) = P(x|y=k)*P(y=k)/P(x)$   
 $= P(x|y=k)*P(y=k)/[P(x|y=0)*P(y=0)+P(x|y=1)*P(y=1)+\dots+P(x|y=k)*P(y=k)]$

Then we can assign each data point to the class with the highest posterior probability. Your task is to finish *gaussian\_pos\_prob.m*, that compute the posterior probability of given datasets  $X$  under the extended model. See the comments in *gaussian\_pos\_prob.m* 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* and the plot of your result in your homework report.

当两者的posterior probability特别相近，即二分类时两者差小于阈值时，就是分界线

- (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  $\{(\mathbf{x}^{(i)}, y^{(i)}) \mid y \in \{0, 1\}, i = 1, \dots, m\}$ , what is the maximum likelihood estimation of  $\phi, \mu_0$  and  $\mu_1$ ? (Optionally, you are encouraged to compute the MLE for all the other parameters  $\Sigma_0, \Sigma_1$ , and generalize to the  $K$ -class gaussian model. This will be challenging but rewarding<sup>2</sup>.)

参考PPT  $\phi=1/k$   $\mu_0$ :  $y=0$  的时候的数据  $x_0$  的平均值  $\mu_1$ :  $y=1$  的是偶的数据  $x_1$  的平均值  $\sigma$  也是相类似的 都是和类中的分布有关 并且用MLE分别为平均值和方差

## 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 processing and lacks of some useful data structure, TA has written some Python scripts to transform email texts to numbers that MATLAB can read from. The skeleton code is *run.m* (in *text\_classification* folder).

<sup>2</sup>You may want to look back when we learn GMM in the future.

In this assignment, instead of following TA's Python scripts and *run.m*, 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,  $P(word_i = N | SPAM) = P(word_i = N | \text{document-type}_j) P(\text{document-type}_j | SPAM)$  should hold, where  $\text{document-type}_j$  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 naive bayes cares (we will learn PLSA in the near future). For simplification, in training we discard the documents information and mix all the words to generate  $P(word_i | SPAM)$  and  $P(word_i | HAM)$  denoting the possibility for a word in SPAM/HAM email to be  $word_i$ . Therefore  $P(word_i = N | SPAM) = P(word_i | SPAM)^N$ .
  - 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  $\frac{P(word_i | SPAM)}{P(word_i | HAM)}$  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<sup>3</sup>:

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

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

TP=1093 FP=28  
FN=31 TN=2983  
precision=97.50% recall=97.24%

<sup>3</sup>Positive and negative often substitutes as predictions in two labels problem. They are usually defined implicitly by common sense. 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 athletes taking doping as possible and otherwise negative. In TP/FP/TN/FN terminology, T stands for 'True' which means predict is the same as label while F stands for 'False'. And P stands for 'Positive' and N stands for 'Negative'. [http://en.wikipedia.org/wiki/Precision\\_and\\_recall](http://en.wikipedia.org/wiki/Precision_and_recall)

- (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.

spam filter: recall should be good at distinguish spam or ham

airport: precision should be good at predict every bombs. Finding bombs is more important than distinguish bombs and drugs.

---

Please submit your homework report to <http://assignment.zjulearning.org:8081/> in pdf format, with all your code in a zip archive.

If you have any problems in understading the homework, feel free to contact TA.