机器学习导论
习题五

141210016, 刘冰楠, bingnliu@outlook.com

2017 年 5 月 31 日

1 [25pts] Bayes Optimal Classifier

试证明在二分类问题中，但两类数据同先验、满足高斯分布且协方差相等时，LDA 可产生贝叶斯最优分类器。

Solution.
We prove that under the presuppositions stated in the problem, classification criteria of LDA and Bayes optimal classifier are the same.

The presuppositions are:

Equal Prior The number of samples in two classes are the same: $|D_{c_1}| = |D_{c_2}|$;

Gauss Distribution We use multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ when calculating likelihood;

Equal Covariance We use the same covariance matrix $\boldsymbol{\Sigma}$ in two gauss distribution. Additionally, we assume sample covariances of two classes are close.

1.1 Bayes optimal classifier

Bayes optimal classifier satisfies, $\forall \mathbf{x} \in D$,

$$h^*(\mathbf{x}) = \min_{c \in Y} R(c, \mathbf{x}), \tag{1.1}$$

where $R(c, \mathbf{x})$ is Bayesian risk:

$$R(c, \mathbf{x}) = \sum_{i=1}^{N} f(c_i, \mathbf{x}) P(c_i \mid \mathbf{x}). \tag{1.2}$$

Here we assume $f(c, \mathbf{x})$ is 0-1 loss, then Eq.(1.1) is equivalent to

$$h^*(\mathbf{x}) = \max_{c \in Y} P(c \mid \mathbf{x}). \tag{1.3}$$

Under **assumption 1: equal prior**, Eq.(1.3) is equivalent to

$$h^*(\mathbf{x}) = \max_{c \in Y} P(\mathbf{x} \mid c). \tag{1.4}$$

Under **assumption 2: normal distribution**, Eq.(1.4) is equivalent to

$$h^*(\mathbf{x}) = \max_{c \in Y} \phi_k(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}), \tag{1.5}$$

where $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ is mean vector and covariance matrix respectively, and ϕ_k is probability density function (PDF) for k-dimension multivariate normal distribution:

$$\phi_k(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \tag{1.6}$$

Then we use Maximum Likelihood Estimation (MLE) to estimate parameters of normal distribution. The log likelihood is:

$$\mathcal{LL}(D_c \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}) = \sum_{\mathbf{x}_i \in D_c} \ln \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}). \tag{1.7}$$

After some algebra, we get the estimates:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_c &= \frac{1}{|D_c|} \sum_{\mathbf{x} \in D_c} \mathbf{x} \\ \hat{\boldsymbol{\Sigma}} &= \frac{1}{|D_c|} \sum_{\mathbf{x} \in D_c} (\mathbf{x} - \hat{\boldsymbol{\mu}}_c)(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}} \end{aligned} \tag{1.8}$$

Now training of the Bayesian classifier is completed. Then for a new sample $\mathbf{x}$, **prediction** c_{pred} is

$$c_{\text{pred}} = \begin{cases} c_1, & \phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}) > \phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}) \\ c_2, & \phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}) < \phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}). \end{cases} \tag{1.9}$$

Define $F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}})$ as:

$$F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}) = \ln \frac{\phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}})}{\phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}})}. \tag{1.10}$$

Then to compare $\phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}})$ with $\phi_k(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}})$ is equivalent to compare $F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}})$ with 0.

Simplify Eq.(1.10):

$$\begin{aligned} F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}) &= \ln \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_1)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_1)\right) / \sqrt{(2\pi)^k|\hat{\boldsymbol{\Sigma}}|}}{\exp\left(-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_2)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_2)\right) / \sqrt{(2\pi)^k|\hat{\boldsymbol{\Sigma}}|}} \\ &= \ln \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_1)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_1)\right)}{\exp\left(-\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_2)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_2)\right)} \\ &= -\frac{1}{2}\left[(\mathbf{x}-\hat{\boldsymbol{\mu}}_1)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_1) - (\mathbf{x}-\hat{\boldsymbol{\mu}}_2)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_2)\right] \\ &= \mathbf{x}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2) - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2) \\ &= (\mathbf{x} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2))^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2). \end{aligned} \tag{1.11}$$

Therefore we classify $\mathbf{x}$ into c_1 when $F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}) > 0$ or c_2 when $F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}) < 0$.

1.2 LDA classifier

According to Eq.(3.33) in textbook 3.4, we can calculate the *within-class scatter matrix* $\mathbf{S}_w$:

$$
\begin{aligned}
\mathbf{S}_w &= \sum_{i=1}^{2} \sum_{\mathbf{x}\in D_{c_i}} (\mathbf{x}-\hat{\boldsymbol{\mu}}_i)(\mathbf{x}-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}} \\
&= \sum_{i=1}^{2} |D_{c_i}| \frac{1}{|D_{c_i}|} \sum_{\mathbf{x}\in D_{c_i}} (\mathbf{x}-\hat{\boldsymbol{\mu}}_i)(\mathbf{x}-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}} \\
&= \sum_{i=1}^{2} |D_{c_i}| \hat{\boldsymbol{\Sigma}}_i \\
&= 2|D_{c_1}|\hat{\boldsymbol{\Sigma}} \\
&= |D|\hat{\boldsymbol{\Sigma}},
\end{aligned} \tag{1.12}
$$

where we have implicitly used **assumptions 1** and **assumptions 2**.

According to Eq.(3.39) in textbook 3.4, we can calculate *projection matrix* $\mathbf{w}$:

$$
\begin{aligned}
\mathbf{w} &= \mathbf{S}_w^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2) \\
&= |D|^{-1}\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2).
\end{aligned} \tag{1.13}
$$

Now training of the LDA model is completed. Then for a new sample $\mathbf{x}$, we investigate whether $\mathbf{x}$ is closer to $\hat{\boldsymbol{\mu}}_1$ or $\hat{\boldsymbol{\mu}}_2$, or equivalently consider projection of $\mathbf{x} - (\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2)/2$ on $\mathbf{w}$, i.e. **prediction** c_{pred} is

$$
c_{\text{pred}} = \begin{cases} c_1, & (\mathbf{x} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2))^{\mathrm{T}}\mathbf{w} > 0 \\ c_2, & (\mathbf{x} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2))^{\mathrm{T}}\mathbf{w} < 0. \end{cases} \tag{1.14}
$$

Note $(\mathbf{x} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2))^{\mathrm{T}}\mathbf{w} = (\mathbf{x} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2))^{\mathrm{T}}|D|^{-1}\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)$. Compare this with Bayesian classifier, which is $F_{\text{Bayes}}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \hat{\boldsymbol{\Sigma}}) = (\mathbf{x} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2))^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)$, we found that they always have **the same sign** ($|D| > 0$).
Therefore, the LDA model will always have the same classification with the Bayesian optimal classifier.

□

2 [25pts] Naive Bayes

考虑下面的 400 个训练数据的数据统计情况，其中特征维度为 2（$\mathbf{x} = [x_1, x_2]$），每种特征取值 0 或 1，类别标记 $y \in \{-1, +1\}$。详细信息如表1所示。

根据该数据统计情况，请分别利用直接查表的方式和朴素贝叶斯分类器给出 $\mathbf{x} = [1, 0]$ 的测试样本的类别预测，并写出具体的推导过程。

Solution.

表 1: 数据统计信息

x_1	x_2	$y=+1$	$y=-1$
0	0	90	10
0	1	90	10
1	0	51	49
1	1	40	60

2.1 Refer to the table directly

We do not make any simplification assumptions. And we **do not need to use the Bayes Rule** because we can directly estimate *posterior probability* from the table:

$$
\begin{aligned}
P(\mathbf{x}=[1;0] \mid y=+1) &= \frac{51}{51+49} = 51\% \\
P(\mathbf{x}=[1;0] \mid y=-1) &= \frac{49}{51+49} = 49\%.
\end{aligned}
\tag{2.1}
$$

Therefore the classification result is: $y=+1$.

2.2 Naive Bayes

We assume that all features are mutually independent, thus simplifying joint distribution to univariate distribution.

First we calculate *prior*, see Table.(2):

表 2: Prior Probability (Estimated)

y = +1	y = -1
0.6775	0.3225

Next we count $P(x=i \mid y=j)$ for $i=0,1$ and $j=0,1$, see Table.(3) and Table.(4):

表 3: Likelihood Count for x_1 in Naive Bayes

x_1	$y=+1$	$y=-1$
0	180	20
1	91	109

表 4: Likelihood Count for x_2 in Naive Bayes

x_2	$y=+1$	$y=-1$
0	141	59
1	130	70

Then we calculate *likelihood*, see Table.(5) and Table.(6) (here we **didn't use *Laplace Correction*** since it is **not needed** for this specific dataset and we do not want to introduce **extra noise** into the dataset.)

表 5: Likelihood for x_1 in Naive Bayes

x_1	$y=+1$	$y=-1$
0	0.6642	0.1550
1	0.3358	0.8450

表 6: Likelihood for x_2 in Naive Bayes

x_2	$y=+1$	$y=-1$
0	0.5203	0.4574
1	0.4797	0.5426

Finally we calculate *posterior probability* using the Bayes Rule:

$$
\begin{aligned}
P(y=+1 \mid \mathbf{x}=[1;0]) &= P(y=+1)P(\mathbf{x}=[1;0] \mid y=+1)/P(\mathbf{x}=[1;0]) \\
&\propto P(y=+1)P(\mathbf{x}=[1;0] \mid y=+1) \\
&= P(y=+1)P(x_1=1 \mid y=+1)P(x_2=0 \mid y=+1) \\
&\approx 0.1184 \\
P(y=-1 \mid \mathbf{x}=[1;0]) &= P(y=-1)P(\mathbf{x}=[1;0] \mid y=-1)/P(\mathbf{x}=[1;0]) \\
&\propto P(y=-1)P(\mathbf{x}=[1;0] \mid y=-1) \\
&= P(y=-1)P(x_1=1 \mid y=-1)P(x_2=0 \mid y=-1) \\
&\approx 0.1246
\end{aligned}
\tag{2.2}
$$

Therefore the classification result is: $y=-1$.

3 [25pts] Bayesian Network

贝叶斯网 (Bayesian Network) 是一种经典的概率图模型，请学习书本 7.5 节内容回答下面的问题：

(1) **[5pts]** 请画出下面的联合概率分布的分解式对应的贝叶斯网结构：

$$\Pr(A,B,C,D,E,F,G) = \Pr(A)\Pr(B)\Pr(C)\Pr(D|A)\Pr(E|A)\Pr(F|B,D)\Pr(G|D,E)$$

(2) **[5pts]** 请写出图3中贝叶斯网结构的联合概率分布的分解表达式。

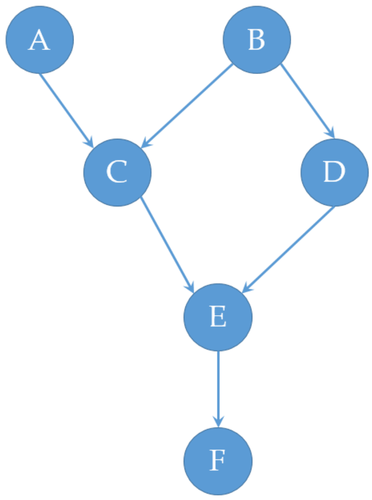

图 1: 题目 3-(2) 有向图

(3) **[15pts]** 基于第 (2) 问中的图3, 请判断表格7中的论断是否正确，只需将下面的表格填完整即可。

表 7: 判断表格中的论断是否正确

序号	关系	True/False	序号	关系	True/False
1	$A \perp\!\!\!\perp B$		7	$F \perp B \mid C$	
2	$A \perp B \mid C$		8	$F \perp B \mid C, D$	
3	$C \perp\!\!\!\perp D$		9	$F \perp B \mid E$	
4	$C \perp D \mid E$		10	$A \perp\!\!\!\perp F$	
5	$C \perp D \mid B, F$		11	$A \perp F \mid C$	
6	$F \perp\!\!\!\perp B$		12	$A \perp F \mid D$	

Solution.

Conditional independence implied by DAG:

$$x_s \perp\!\!\!\perp x_{\text{pred}(s)\setminus\text{pa}(s)} \mid x_{\text{pa}(s)} \tag{3.1}$$

and

$$P(x_1, x_2, \ldots, x_n \mid G) = \prod_{i=1}^{n} P(x_i \mid x_{\text{pa}(s)}). \tag{3.2}$$

Here n is the number of nodes in G, $x_{\text{pa}(s)}$ are x_s's parent nodes and $x_{\text{pred}(s)}$ are x_s's predecessors in topological order.

3.1 Problem (1)

Just use Eq.(3.1) and (3.2). See Fig.(3.1) for the result:

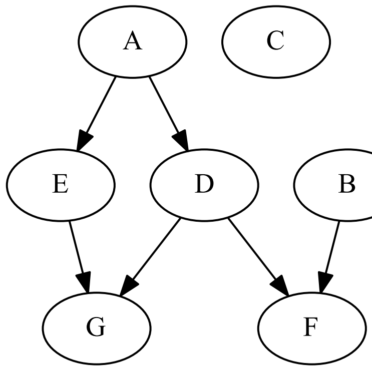

图 2: Problem 3-(1) DAG

3.2 Problem (2)

Just use Eq.(3.1) and (3.2). See Eq.(3.3):

$$P(A, B, C, D, E, F) = P(A)P(B)P(C \mid A, B)P(D \mid B)P(E \mid C, D)P(F \mid E). \tag{3.3}$$

3.3 Problem (3)

Use *D-separation* to get correct relationships. See Table.(8) for the answer:

表 8: 判断表格中的论断是否正确

序号	关系	True/False	序号	关系	True/False
1	$A \perp\!\!\!\perp B$	**T**	7	$F \perp B \mid C$	**F**
2	$A \perp B \mid C$	**F**	8	$F \perp B \mid C, D$	**T**
3	$C \perp\!\!\!\perp D$	**F**	9	$F \perp B \mid E$	**T**
4	$C \perp D \mid E$	**F**	10	$A \perp\!\!\!\perp F$	**F**
5	$C \perp D \mid B, F$	**F**	11	$A \perp F \mid C$	**F**
6	$F \perp\!\!\!\perp B$	**F**	12	$A \perp F \mid D$	**F**

4 [25pts] Naive Bayes in Practice

请实现朴素贝叶斯分类器，同时支持离散属性和连续属性。详细编程题指南请参见链接：`http://lamda.nju.edu.cn/ml2017/PS5/ML5_programming.html`.

同时，请简要谈谈你的感想。实践过程中遇到了什么问题，你是如何解决的？

Solution.

感想：

4.1 关于 Modeling

4.1.1 连续分布的选择

针对这次的连续数据，用的正态分布是不太合适的，因为离散的特征均大于等于零，而正态分布则是分布于整个横轴，可以考虑从-1 处截断的正态分布，或者卡方分布等。

4.1.2 两个概率的 trade-off

可以发现，预测时不计算先验和离散特征的概率，只用连续特征的概率，得到的精度是一样的。这是由于由于连续特征算得的 log 概率 (的绝对值) 非常大从而 dominate 了后验概率。只使用离散特征，得到精度 0.65 左右；只使用连续特征，得到精度 0.68 左右，若对连续特征算得的对数概率统一放缩到和离散特征一个 scale 上 (本数据可使用 $1e-5$)，则能得到 0.71 的精度。

4.1.3 std zero 的处理

对连续 feature 的 log 概率排序，发现最小的（绝对值最大的）就正好对应那些 std=0 的，事实对每一类，上 std=0 的占所有 feature 的一半以上，因而连续 feature 算得的 log 概率又是被这些 std=0 的 dominate

所以精度对于我们处理 std=0 的方式非常敏感

五个类别的 stds.max() 分别是，从每一类 2500 个 std 中取出的最大值是非常不 robust 的，无论是你说的某些特征限定的大波动，或者某个 feature 成为 oulier 或者 feature 中某个值成为 outlier，都会直接影响这个 max 的选取

而对五类，引入相同的 std 增量，就不会引入上述的这种不稳健性

4.2 关于 Algorithm

4.2.1 取对数遇到零

浮点数绝对值太小而被 round 到 0，进而影响取对数。

解决方案一：对所有要取对数的值加上一个小量，这个小量的合适取值可以:

```
from sys import float_info; SMALL_FLOAT = float_info.min
```

解决方案二：直接写 log 概率密度函数，我采用了这种解决方案。

4.2.2 定义一些常量

定义如 `LAPLACE_CORRECTION_ALPHA`，`STD_RATIO` 等常量。

4.2.3 性能优化

一般准则是预先建立好矩阵，尽量向量化。预测时需要对每一个 sample 循环，再对每一类循环，再对每一个 feature 循环。其中只有对 sample 循环的时候无法向量化，其余两个都可以向量化实现。这可提升性能 50 倍左右。对于 sample 的循环，虽然不可以向量化，但可以考虑并行 (本实现未做这个)。

不知道性能瓶颈的时候，用 `line_profiler` 和 `snackviz` 等工具进行 profile，再针对结果优化。

参考文献

[1] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.