贺鑫(计算机科学与技术系研究生)，学号：MG1833027

Problem 1

(a): 聚类的目的是为了使得同类的样本尽可能相似，即同类样本距离尽可能近，等价于同类样本距离"类中心"样本尽可能近，即同类样本紧凑、聚成团。直观地需要去寻找能代表类i的"表示"(样本)μ_i，构造出各样本的类标记(group indicator)，得到式(5)的最优化形式。

(b): 首先从训练集$\{x_1, x_2, \ldots, x_M\}$中**随机选**出$K$个样本作为$\mu_1, \mu_2, \ldots, \mu_K$的初始值；

1. 固定$\mu_i(\forall\ 1 \leq i \leq K)$，运行**K近邻算法**得到样本$x_j$ 所属类标号λ：$\lambda = min\ arg_i \parallel \mu_i - x_j \parallel^2$. 令$\gamma_{\lambda j} = 1.\ \gamma_{ij} = 0\ \forall i \neq \lambda$.
2. 得到每个样本所属类别，固定γ_{ij}，对类i修改其代表μ_i为：$\mu_i = \sum_{j=1}^{M} \gamma_{ij} x_j / \sum_{j=1}^{M} \gamma_{ij}$.即将当前类的**平均向量**作为类的代表；
3. 回到步骤1迭代，直到$\forall\ \gamma_{ij}$不再变化。

(c): 将M个样本划分为K类，至多有M^K中划分方式。这是一个很大但是**有限**的数。算法每次迭代的过程中，总是基于old clustering进行更新，如果`new clustering`不同于`old clustering`，说明`new clustering`比`old clustering`有更小的类间距离。又由于算法在**有限域**中迭代，且环的大小只能为1(否则出现一个clustering比自身的类间距离小)，所以算法必定在有限步内终止。

Problem 2

(a): $min\ arg_\beta \sum_{i=1}^{n} \parallel x_i^T \beta - y_i \parallel^2$.

(b): $min\ arg_\beta \parallel \mathbf{X}\beta - y \parallel^2$.

(c): 忽略回归损失ϵ, 得$y = \mathbf{X}\beta$，由于$\mathbf{X}^T\mathbf{X}$可逆，可得：$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$.

(d): 当$d > n$时，$\mathbf{X}^T\mathbf{X}$为d by d的方阵，但$rank(\mathbf{X}^T\mathbf{X}) \leq n < d$，所以$\mathbf{X}^T\mathbf{X}$不可逆；

(e): 对损失函数加上$\lambda\beta^T\beta$后，优化函数倾向于参数β norm较小的模型，λ越大，β norm越趋近与0.

(f): $min\ arg_\beta \parallel \mathbf{X}\beta - y \parallel^2 + \lambda\beta^T\beta$.

(g): 当$\mathbf{X}^T\mathbf{X}$不可逆时，说明数据量体现数据的复杂性，我们需要添加关于数据分布的额外假设来达到唯一解。从贝叶斯的角度来看，如何我们假设数据服从正态分布，最小化对数似然函数即为添加了L^2正则项的岭回归。

参考：PRML第一章1.2.5节。

(h): 当$\lambda = 0$时，相当于正常的线性回归问题，参考(c)小问的解。当$\lambda = \infty$时，$\beta = 0$.

(i): 我认为不能将超参λ作为优化函数直接的对象，因为λ是用来控制模型复杂程度的，如果将λ作为损失函数的一部分，直接进行优化，不仅λ在损失函数中表示的含义不明，同时没有验证集的存在，导致无法预知模型是否过拟合，这将失去引入λ的意义。

Problem 3

(ab): 为节省空间，将(a)、(b)计算出来的数据整合到一个表格中，表格数据如下：

Index	Label	Score	Precision	Recall	AUC-PR	AP
0			1.0000	0.0000		
1	1	1.0	1.0000	0.2000	0.2000	0.2000
2	2	0.9	0.5000	0.2000	0.0000	0.0000
3	1	0.8	0.6667	0.4000	0.1167	0.1333
4	1	0.7	0.7500	0.6000	0.1417	0.1500
5	2	0.6	0.6000	0.6000	0.0000	0.0000
6	1	0.5	0.6667	0.8000	0.1267	0.1333
7	2	0.4	0.5714	0.8000	0.0000	0.0000
8	2	0.3	0.5000	0.8000	0.0000	0.0000
9	1	0.2	0.5556	1.0000	0.1056	0.1111
10	2	0.1	0.5000	1.0000	0.0000	0.0000
					0.6906	0.7278

AUC-PR和AP都是计算PR曲线下方面积，只不过采用不同的近似方式。所以AUC-PR和AP相似

(c): 以下表格列出`9-th`和`10-th`交换之前和交换之后AUC-PR和AP第9、10行以及总和的数据对比：

	Old AUC-PR	Old AP	New AUC-PR	New AP
9	0.1056	0.1111	0.0000	0.0000
10	0.0000	0.0000	0.0944	0.1000
	0.6906	0.7278	0.6794	0.7167

(d)下面是计算P-R、AUC-PR、AP的Python代码：

```
labels = [1, 2, 1, 1, 2, 1, 2, 2, 1, 2]
exchanged_labels = labels[ :-2] + [2, 1]

TP, NP = 0, 0
points = [(1.0, 0.0)]
AUC_PR, AP_area = 0, 0
print("precision\tscore\tAUC-PR \t  AP")
for i in range(10):
    if exchanged_labels[i] == 1:
        TP += 1
    else:
        NP += 1
    precision, recall = TP / (TP + NP), TP / 5
    points.append((precision, recall))
    trapezoidal_area = 0.5 * (points[-1][1] - points[-2][1]) * (points[-1][0]
+ points[-2][0])
    AP = (points[-1][1] - points[-2][1]) * points[-1][0]
    # print("precision: %.4f \tscore: %.4f\t AUC-PR: %.4f\t AP: %.4f" %
(precision, recall, trapezoidal_area, AP))
    print(" %.4f\t\t%.4f\t%.4f\t%.4f" % (precision, recall, trapezoidal_area,
AP))
```

```
    AUC_PR += trapezoidal_area
    AP_area += AP

print("\t\t\t\t\t%.4f\t%.4f" % (AUC_PR, AP_area))
```

运行效果为：

```
precision score   AUC-PR    AP
 1.0000   0.2000  0.2000  0.2000
 0.5000   0.2000  0.0000  0.0000
 0.6667   0.4000  0.1167  0.1333
 0.7500   0.6000  0.1417  0.1500
 0.6000   0.6000  0.0000  0.0000
 0.6667   0.8000  0.1267  0.1333
 0.5714   0.8000  0.0000  0.0000
 0.5000   0.8000  0.0000  0.0000
 0.5556   1.0000  0.1056  0.1111
 0.5000   1.0000  0.0000  0.0000
                  0.6906  0.7278
```

Problem 4

(a): 将$f(\mathbf{x};D)=\frac{1}{k}\sum_{i=1}^{k}y_{nn}(i)$代入Equ. 28可得：

$$E[(y-f(\mathbf{x};D))^2]=(F(x)-E_D[f(\mathbf{x};D)])^2+E_D[(f(\mathbf{x};D)-E_D[f(\mathbf{x};D)])^2]+\sigma^2.$$

其中 bias为：$(F(x)-E_D[f(\mathbf{x};D)])^2$. Variance为：$E_D[(f(\mathbf{x};D)-E_D[f(\mathbf{x};D)])^2]$.

(b): $E[f]=E[\frac{1}{k}\sum_{i=1}^{k}y_{nn}(i)]=\frac{1}{k}\sum_{i=1}^{k}E[y_{nn}(i)]$

(c):
$E[(y-f(\mathbf{x};D))^2]=(F-E[\frac{1}{k}\sum_{i=1}^{k}y_{nn}(i)])^2+E[(\frac{1}{k}\sum_{i=1}^{k}y_{nn}(i)-E[\frac{1}{k}\sum_{i=1}^{k}y_{nn}(i)])^2]+\sigma^2$
(d): (c)中第二项为variance项，当$k\to n$时，方差项为整个数据集的方差，当$k=1$时，方差为单个样本的方差(0)。

(e): (c)中第一项为平方bias项，当$k\to n$时，每个预测样例$\mathbf{x}$的预测结果均为训练集中类别最多的标记，可见当采用$K-NN$作为分类器时，方差与分类器的k取值相关；

Problem 5

(a): 要证明$\|x\|=\|Gx\|$，只需证明$x^Tx=(Gx)^T(Gx)=x^TG^TGx$.由于G为正交单位矩阵，所以$G^TG=GG^T=\mathbf{I}$,所以$\|x\|=\|Gx\|$, $\|x\|=\|G^Tx\|$.

(b): $\|\mathbf{X}\|_F=\sqrt{tr(\mathbf{X}\mathbf{X}^T)}$. $\|G^T\mathbf{X}G^T\|=\sqrt{tr(G^T\mathbf{X}\mathbf{X}^TG)}$.

只需证明$tr(A)=tr(G^TAG)$，其中$\mathbf{X}\mathbf{X}^T=A$.

由`(a)`知，$tr(A)=tr(G^TA)=tr(AG)$，所以$tr(A)=tr(G^TA)=tr(G^TAG)$.

(c): 将$off(\mathbf{X})$作为损失函数，当最小化损失函数时，$off(\mathbf{X}) \to 0$，即非对角线元素可视为0，即将对称矩阵$\mathbf{X}$**对角化**了。由于X为实对称矩阵，所以$\mathbf{X}$的**奇异值为特征值的平方**，**奇异向量为特征向量**。所以只要我们能将$\mathbf{X}$对角化，正交矩阵J中的列即为$\mathbf{X}$的特征向量，对角化后的主对角线的非0元素即为奇异值(特征值的平方)。

(d): 令$\mathcal{J}(i,j,\theta)$为Givens旋转矩阵，由于$\mathcal{J}(p,q,\theta)$只会改变矩阵$\mathbf{X}$中a_{pp}、a_{qq}、a_{pq}、a_{qp}四个元素，为了书写简便，将$\mathcal{J}^T\mathbf{X}\mathcal{J}$简写成:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} b_{pp} & 0 \\ 0 & b_{qq} \end{bmatrix}$$

其中$c=\cos\theta, s=\sin\theta$.

可得：$b_{pq}=0=a_{pq}(c^2-s^2)-(a_{pp}-a_{qq})cs \ldots\ldots (1)$.

令$t=\frac{c}{s}=\tan\theta, \tau=\frac{a_{qq}-a_{pp}}{2a_{pq}}$,可将$(1)$式化简为：$t^2+2\tau t-1=0 \ldots\ldots (2)$.

从(2)式中解出$t(\tan\theta)$.

由$s^2+c^2=1, t=\frac{s}{c}$，可得：$c=1/(\sqrt{1+t^2})$. $s=ct$.代入$\mathcal{J}$中即可得到$J(p,q,\theta)$.

(e): 为书写简便，将Jacobi迭代方法的第二步赋值操作写成：$B=\mathcal{J}^T A\mathcal{J}$.

问题等价于$off(B)^2 < off(A)^2$是否成立。

首先: $off(A)^2 = \| A \|_F^2 - \sum_{i=1}^n a_{ii}^2$. 由`(b)`可知：$\|B\|_F = \| A\|_F$.

$off(B)^2 = \| B \|_F^2 - \sum_{i=1}^n b_{ii}^2 = \| A \|_F^2 - \sum_{i\neq p,q} b_{ii}^2 - (b_{pp}^2+b_{qq}^2)$

$= \| A \|_F^2 - \sum_{i\neq p,q} a_{ii}^2 - (a_{pp}^2+2a_{pq}^2+a_{qq}^2)$

$= off(A) - 2a_{pq}^2 < off(A)^2$.

注：也可直接理解为：$off(B)$只是将$off(A)$的a_{pq}、a_{qp}置为零。

所以Jacobi 方法将会迭代地降低$off(\mathbf{X})$.

(f): $\mathbf{X}$中至多有$N=n(n-1)/2$对非零非对角线元素对. 在当前迭代过程中，假设a_{pq}为绝对值最大的非对角线元素，则：$off(A)^2 \leq 2Na_{pq}^2$.

由于$off(B)^2 = off(A)^2 - 2a_{pq}^2$,所以我们有：$off(B)^2 \leq (1-\frac{1}{N})off(A)^2$.

易知存在常数c使得每次迭代$off(B)^2 \leq c * off(A)^2$.即Jacobi下降速度至少为线性。因为Jacobi算法必定收敛。