机器学习导论
综合能力测试

Boardwell Nanjing University

2017 年 6 月 18 日

1 [40pts] Exponential Families

指数分布族(Exponential Families)是一类在机器学习和统计中非常常见的分布族, 具有良好的性质。在后文不引起歧义的情况下, 简称为指数族。

指数分布族是一组具有如下形式概率密度函数的分布族群:

$$f_X(x|\theta) = h(x)\exp\left(\eta(\theta)\cdot T(x) - A(\theta)\right) \tag{1.1}$$

其中, $\eta(\theta)$, $A(\theta)$以及函数$T(\cdot)$, $h(\cdot)$都是已知的。

(1) **[10pts]** 试证明多项分布(Multinomial distribution)属于指数分布族。

(2) **[10pts]** 试证明多元高斯分布(Multivariate Gaussian distribution)属于指数分布族。

(3) **[20pts]** 考虑样本集$\mathcal{D} = \{x_1, \cdots, x_n\}$是从某个已知的指数族分布中独立同分布地(i.i.d.)采样得到, 即对于$\forall i \in [1, n]$, 我们有$f(x_i|\boldsymbol{\theta}) = h(x_i)\exp\left(\boldsymbol{\theta}^{\mathrm{T}}T(x_i) - A(\boldsymbol{\theta})\right)$.

对参数$\boldsymbol{\theta}$, 假设其服从如下先验分布:

$$p_\pi(\boldsymbol{\theta}|\boldsymbol{\chi}, \nu) = f(\boldsymbol{\chi}, \nu)\exp\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\chi} - \nu A(\boldsymbol{\theta})\right) \tag{1.2}$$

其中, $\boldsymbol{\chi}$和ν是$\boldsymbol{\theta}$生成模型的参数。请计算其后验, 并证明后验与先验具有相同的形式。(**Hint**: 上述又称为“共轭”(Conjugacy),在贝叶斯建模中经常用到)

Solution.

(1): Multinomial distribution's probability mass function:

$$f(x|\mathbf{p}) = \frac{n!}{\prod_{i=1}^k x_i!}\prod_{i=1}^k p_i^{x_i}$$

Let $\mathbf{P} = \eta(\theta) = [lnp_1, lnp_2, ..., lnp_k]$, $\mathbf{X} = T(x) = [x_1; x_2; ...; x_k]$,$A(\theta) = 0$ and $h(x) = \frac{n!}{\prod_{i=1}^k x_i!}$, so that

$$\begin{aligned} f_X(x|\theta) &= h(x)\exp\left(\eta(\theta)\cdot T(x)\right) \\ &= h(x)\exp\left(\mathbf{P^T X}\right) \\ &= \frac{n!}{\prod_{i=1}^k x_i!}\prod_{i=1}^k p_i^{x_i} \end{aligned} \tag{1.3}$$

(2):
Multivariate Gaussian distribution's 's probability density function:

$$f(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{1}{2}k}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

Unfold the items in the exponent:

$$(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = \quad \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + 2\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \tag{1.4}$$

Considering that:

$$\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} = tr(\boldsymbol{\Sigma}^{-1}\mathbf{x}\mathbf{x}^T)$$

Let $h(x) = (2\pi)^{-\frac{1}{2}k}$, $\eta(\theta) = [\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}; -\frac{1}{2}\boldsymbol{\Sigma}^{-1}]$, $A(\theta) = \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{1}{2}ln|\boldsymbol{\Sigma}|$ and $T(x) = [\mathbf{x}; \mathbf{x}\mathbf{x}^T]$.
Back to the origin equation:

$$\begin{aligned} f_X(x|\theta) &= h(x)\exp\left(\eta(\theta)\cdot T(x) - A(\theta)\right) \\ &= (2\pi)^{-\frac{1}{2}k}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})) \end{aligned} \tag{1.5}$$

(3):
Because of the independence of x_i, we can obtain: $f(x_1, x_2, ..., x_n|\boldsymbol{\theta}) = \prod_{i=1}^n f(x_i|\boldsymbol{\theta})$. So that the posterior probability is:

$$P(\boldsymbol{\theta}|x_1, x_2, ..., x_n) = \frac{f(x_1, x_2, ..., x_n|\boldsymbol{\theta})p_\pi(\boldsymbol{\theta})}{P(\mathbf{x})}$$

Because of

$$P(\mathbf{x}) = \int f(x_1, x_2, ..., x_n|\boldsymbol{\theta})p_\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$$

we can see that $P(\mathbf{x})$ has nothing to do with $\boldsymbol{\theta}$. So we can record $P(\mathbf{x})$ as a constant factor $\frac{1}{C}$. So that the posterior probability is:

$$\begin{aligned} P(\boldsymbol{\theta}|x_1, x_2, ..., x_n) &= Cf(x_1, x_2, ..., x_n|\boldsymbol{\theta})p_\pi(\boldsymbol{\theta}|\boldsymbol{\chi}, \nu) \\ &= C\prod_{i=1}^n f(x_i|\boldsymbol{\theta})p_\pi(\boldsymbol{\theta}|\boldsymbol{\chi}, \nu) \\ &= C(\prod_{i=1}^n h(x_i))\exp\left(\boldsymbol{\theta}^{\mathrm{T}}\sum_{i=1}^n T(x_i) - nA(\boldsymbol{\theta})\right) f(\boldsymbol{\chi}, \nu)\exp\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\chi} - \nu A(\boldsymbol{\theta})\right) \\ &= Cf(\boldsymbol{\chi}, \nu)(\prod_{i=1}^n h(x_i))\exp\left(\boldsymbol{\theta}^{\mathrm{T}}(\sum_{i=1}^n T(x_i) + \boldsymbol{\chi}) - (n+\nu)A(\boldsymbol{\theta})\right) \end{aligned} \tag{1.6}$$

So that we can find that the form of $P(\boldsymbol{\theta}|x_1, x_2, ..., x_n)$ is the same with its prior distribution $p_\pi(\boldsymbol{\theta}|\boldsymbol{\chi}, \nu)$. Use $f'(\boldsymbol{\chi}', \nu') = Cf(\boldsymbol{\chi}, \nu)\prod_{i=1}^n h(x_i)$, $\boldsymbol{\chi}' = \boldsymbol{\chi} + \sum_{i=1}^n T(x_i)$ and $\nu' = n + \nu$ to replace the position of origin variables and we can prove they follow the same distribution.

2 [40pts] Decision Boundary

考虑二分类问题, 特征空间$X \in \mathcal{X} = \mathbb{R}^d$, 标记$Y \in \mathcal{Y} = \{0, 1\}$. 我们对模型做如下生成式假设：

- attribute conditional independence assumption: 对已知类别, 假设所有属性相互独立, 即每个属性特征独立地对分类结果发生影响；
- Bernoulli prior on label: 假设标记满足Bernoulli分布先验, 并记$\Pr(Y = 1) = \pi$.

(1) **[20pts]** 假设$P(X_i|Y)$服从指数族分布, 即

$$\Pr(X_i = x_i|Y = y) = h_i(x_i) \exp(\theta_{iy} \cdot T_i(x_i) - A_i(\theta_{iy}))$$

请计算后验概率分布$\Pr(Y|X)$以及分类边界$\{x \in \mathcal{X} : P(Y = 1|X = x) = P(Y = 0|X = x)\}$. (**Hint**: 你可以使用sigmoid函数$\mathcal{S}(x) = 1/(1 + e^{-x})$进行化简最终的结果).

(2) **[20pts]** 假设$P(X_i|Y = y)$服从高斯分布, 且记均值为μ_{iy}以及方差为σ_i^2 (注意, 这里的方差与标记Y是独立的), 请证明分类边界与特征X是成线性的。

Solution.
(1):
Bernoulli distribution prior:

$$P(Y = y) = \pi^y (1 - \pi)^{1-y}$$

Because all attributes are conditional independent, so we can calculate that:

$$\begin{aligned} \Pr(X_1 = x1, X_2 = x_2, ..., X_d = x_d) &= \prod_{i=1}^{d} \Pr(X_i = x_i|Y = y) \\ &= \prod_{i=1}^{d} h_i(x_i) \exp(\theta_{iy} \cdot T_i(x_i) - A_i(\theta_{iy})) \end{aligned} \tag{2.1}$$

According the Bayes formula, we can calculate the posterior probability:
Note that $P(X)$ has nothing to do with Y just like the problem above, we can also use a factor $\frac{1}{C}$ to represent it.

$$\begin{aligned} Pr(Y|X) &= \frac{P(X|Y)P(Y)}{P(X)} \\ &= \frac{P(X|Y)P(Y)}{\sum_{Y=0}^{1} P(X|Y)P(Y)} \\ &= C(\prod_{i=1}^{d} h_i(x_i)) \exp(\sum_{i=1}^{d} \theta_{iy} \cdot T_i(x_i) - \sum_{i=1}^{d} A_i(\theta_{iy}))\pi^y (1 - \pi)^{1-y} \end{aligned} \tag{2.2}$$

So that

$$P(Y = 1|X = x) = C\pi(\prod_{i=1}^{d} h_i(x_i)) \exp(\sum_{i=1}^{d} \theta_{i,1} \cdot T_i(x_i) - \sum_{i=1}^{d} A_i(\theta_{i,1}))$$

$$P(Y=0|X=x) = C(1-\pi)(\prod_{i=1}^{d} h_i(x_i)) \exp(\sum_{i=1}^{d} \theta_{i,0} \cdot T_i(x_i) - \sum_{i=1}^{d} A_i(\theta_{i,0}))$$

Let $P(Y=1|X=x) = P(Y=0|X=x)$ and solve this equation:

$$\begin{aligned}
\pi \exp\left(\sum_{i=1}^{d}[\theta_{i,1} \cdot T_i(x_i) - A_i(\theta_{i,1})]\right) &= (1-\pi)\exp\left(\sum_{i=1}^{d}[\theta_{i,0} \cdot T_i(x_i) - A_i(\theta_{i,0})]\right) \\
ln(\pi) + \sum_{i=1}^{d}[\theta_{i,1} \cdot T_i(x_i) - A_i(\theta_{i,1})] &= ln(1-\pi) + \sum_{i=1}^{d}[\theta_{i,0} \cdot T_i(x_i) - A_i(\theta_{i,0})] \\
\sum_{i=1}^{d}[(\theta_{i,1}-\theta_{i,0})T_i(x_i) + A_i(\theta_{i,1}) - A_i(\theta_{i,0})] &= ln(\frac{1-\pi}{\pi}) \\
\pi\left[\exp\left(\sum_{i=1}^{d}[(\theta_{i,1}-\theta_{i,0})T_i(x_i) + A_i(\theta_{i,1}) - A_i(\theta_{i,0})]\right)+1\right] &= 1 \\
Sigmoid\left(\sum_{i=1}^{d}[(\theta_{i,0}-\theta_{i,1})T_i(x_i) + A_i(\theta_{i,0}) - A_i(\theta_{i,1})]\right) &= \pi
\end{aligned} \tag{2.3}$$

The decision boundary is given by the equation above.

(2):

Gaussian distribution:

$$P(X_i = x_i|Y=y) = \frac{1}{\sqrt{2\pi}\sigma_i} exp(-\frac{(x_i-\mu_{iy})^2}{2\sigma_i^2})$$

Like the question(1), calculate the posterior probability:

$$\begin{aligned}
P(Y=y|X_1=x_1,...,X_d=x_d) &= C\prod_{i=1}^{d} P(X_i=x_i|Y=y)P(Y) \\
&= C\frac{1}{(2\pi)^{\frac{d}{2}}\prod_{i=1}^{d}\sigma_i}\exp\left(-\sum_{i=1}^{d}\frac{(x_i-\mu_{iy})^2}{2\sigma_i^2}\right)\pi^y(1-\pi)^{1-y}
\end{aligned} \tag{2.4}$$

Let $P(Y=1|X=x) = P(Y=0|X=x)$ and solve this equation:

$$\begin{aligned}
ln(\pi) - \sum_{i=1}^{d}\frac{(x_i-\mu_{i,1})^2}{2\sigma_i^2} &= ln(1-\pi) - \sum_{i=1}^{d}\frac{(x_i-\mu_{i,0})^2}{2\sigma_i^2} \\
ln(\frac{\pi}{1-\pi}) &= \sum_{i=1}^{d}\frac{-2\mu_{i,1}x_i + \mu_{i,1}^2 + 2\mu_{i,0}x_i - \mu_{i,0}^2}{2\sigma_i^2} \\
ln(\frac{\pi}{1-\pi}) &= \sum_{i=1}^{d}\frac{2(\mu_{i,0}-\mu_{i,1})x_i + \mu_{i,1}^2 - \mu_{i,0}^2}{2\sigma_i^2} \\
\sum_{i=1}^{d}\frac{\mu_{i,0}-\mu_{i,1}}{\sigma_i^2}x_i + \sum_{i=1}^{d}\frac{\mu_{i,1}^2-\mu_{i,0}^2}{2\sigma_i^2} - ln(\frac{\pi}{1-\pi}) &= 0
\end{aligned} \tag{2.5}$$

The last equation shows the decision boundary. Because of it is a linear equation about x_i, it represents a hyperplane in the high-demension space when coordinates are x_i, which means the decision boundary is linear to feature X.

3 [70pts] Theoretical Analysis of k-means Algorithm

给定样本集$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, k-means聚类算法希望获得簇划分$\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$, 使得最小化欧式距离

$$J(\gamma, \mu_1, \ldots, \mu_k) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij} ||\mathbf{x}_i - \mu_j||^2 \tag{3.1}$$

其中, $\mu_1, \ldots, \mu_k$为k个簇的中心(means), $\gamma \in \mathbb{R}^{n \times k}$为指示矩阵(indicator matrix)定义如下：若$\mathbf{x}_i$属于第j个簇, 则$\gamma_{ij} = 1$, 否则为0.

则最经典的k-means聚类算法流程如算法2中所示(与课本中描述稍有差别, 但实际上是等价的)。

Algorithm 1: k-means Algorithm

Initialize $\mu_1, \ldots, \mu_k$.

repeat

  Step 1: Decide the class memberships of $\{\mathbf{x}_i\}_{i=1}^n$ by assigning each of them to its nearest cluster center.

$$\gamma_{ij} = \begin{cases} 1, & ||\mathbf{x}_i - \mu_j||^2 \leq ||\mathbf{x}_i - \mu_{j'}||^2, \forall j' \\ 0, & \text{otherwise} \end{cases}$$

  Step 2: For each $j \in \{1, \cdots, k\}$, recompute μ_j using the updated γ to be the center of mass of all points in C_j:

$$\mu_j = \frac{\sum_{i=1}^n \gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^n \gamma_{ij}}$$

until *the objective function J no longer changes*;

---

(1) **[10pts]** 试证明, 在算法2中, **Step 1**和**Step 2**都会使目标函数J的值降低.

(2) **[10pts]** 试证明, 算法2会在有限步内停止。

(3) **[10pts]** 试证明, 目标函数J的最小值是关于k的非增函数, 其中k是聚类簇的数目。

(4) **[20pts]** 记$\hat{\mathbf{x}}$为n个样本的中心点, 定义如下变量,

total deviation	$T(X) = \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}\|^2 / n$
intra-cluster deviation	$W_j(X) = \sum_{i=1}^n \gamma_{ij} \|\mathbf{x}_i - \mu_j\|^2 / \sum_{i=1}^n \gamma_{ij}$
inter-cluster deviation	$B(X) = \sum_{j=1}^k \frac{\sum_{i=1}^n \gamma_{ij}}{n} \|\mu_j - \hat{\mathbf{x}}\|^2$

试探究以上三个变量之间有什么样的等式关系？基于此, 请证明, k-means聚类算法可以认为是在最小化intra-cluster deviation的加权平均, 同时近似最大化inter-cluster deviation.

(5) **[20pts]** 在公式(3.1)中, 我们使用ℓ_2-范数来度量距离(即欧式距离), 下面我们考虑使用ℓ_1-范数来度量距离

$$J'(\gamma, \mu_1, \ldots, \mu_k) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij} ||\mathbf{x}_i - \mu_j||_1 \tag{3.2}$$

- **[10pts]** 请仿效算法2(k-means-ℓ_2算法), 给出新的算法(命名为k-means-ℓ_1算法)以优化公式3.2中的目标函数J'.
- **[10pts]** 当样本集中存在少量异常点(outliers)时, 上述的k-means-ℓ_2和k-means-ℓ_1算法, 我们应该采用哪种算法？即, 哪个算法具有更好的鲁棒性？请说明理由。

Solution.

(1):

Step 1:

In cycle's step 1, each x_i will do another cycle to decide which μ_j it should belongs to by change the value of γ_{ij}. After the cycle of a sepcific x_i, the value of γ_{ij} may have two situation:

1) Change,the value of J keep the same

2) Not change, Because of $||\mathbf{x}_i - \mu_j||^2 \leq ||\mathbf{x}_i - \mu_{j'}||^2, \forall j'$, the new $||\mathbf{x}_i - \mu_j||^2$ will no more than before, which lead to value of J's decrease or the same according to its definition

So step 1 will lead to the decrease of J.

In cycle's step 2, each μ_j needs to update itself by γ_{ij}. After the cycle of a sepcific μ_j, the value of μ_j may have two situation:

1) μ_j won't change: loss function will also keep the same;

2) μ_j changes: $\mu_j = \frac{\sum_{i=1}^{n} \gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^{n} \gamma_{ij}}$;

Considering the partial derivative of loss function J:

$$\begin{aligned} \frac{\partial J}{\partial \mu_j} &= \frac{\partial \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij} ||\mathbf{x}_i - \mu_j||^2}{\partial \mu_j} \\ &= \sum_{i=1}^{n} \gamma_{ij} (\mathbf{x}_i - \mu_j) \\ &= \sum_{i=1}^{n} \gamma_{ij} \mathbf{x}_i - \sum_{i=1}^{n} \gamma_{ij} \mu_j \\ &= 0 \end{aligned} \tag{3.3}$$

Because of μ_j keep the same in the sum of i, so that the equation above can be rewritten as $\sum_{i=1}^{n} \gamma_{ij} \mathbf{x}_i = \mu_j \sum_{i=1}^{n} \gamma_{ij}$. Solve this equation and we can get the form in step 2:

$$\mu_j = \frac{\sum_{i=1}^{n} \gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^{n} \gamma_{ij}}$$

It means the value of loss function J has reach minimum by updating μ_j in this way. So the value of J will either keep the same or decrease.

In summary, both step 1 and step 2 will let J decrease or keep the same. If both step 1 and step 2 keep loss function J the same, the algorithm will stop. So we can be sure both step 1 and step 2 can decrease J in situation of algorithm still running.

(2):

Considering the situation that there are N samples and k clusters. So the total number of situations of γ_{ij} is k^N (according to counting principle, each x_i has k choices). We have proven that loss function J will decrease or keep the same in each step of cycle in question 1. So J has two situations in cycle:

1) J don't change: the algorithm will stop in this step.

2) J still changes: The total number of J's situation is finite. J will definately change its situation unidirectionally (decreasing). So this procedure won't continue all the time and turn to situation 1.

According to the analysis above, J will decrease all the way and doesn't change in finite steps, which means the algorithm ends.

(3):

Considering the situation that there are k clusters and J has already reached the minimum point in this condition.

Then add a new $k+1_{th}$ cluster into them. For each sample x_i, do a cycle to check the γ_{ij} whether change or not. There are two following situation:

1) for j from 1 to $k+1$ run a cycle, if each γ_{ij} doesn't change: J will keep the same

2) for j from 1 to $k+1$ run a cycle, if γ_{ij} changes: Because of $||\mathbf{x}_i - \mu_{j'}||^2 \leq ||\mathbf{x}_i - \mu_j||^2$ ($\mu_{j'}$ means the new cluster, μ_j means the old cluster), according to the definition of J, J will also decrease. So that:

$$J^{(k+1)} \leq J^{(k)}_{min}$$

Continually run this algorithm, according to the analysis before, J will still decreases or keeps the same. So that:

$$J^{(k+1)}_{min} \leq J^{(k+1)} \leq J^{(k)}_{min}$$

And we have proven that J_{min} is a non-increasing function of k.

(4):

Because of $\sum_{j=1}^{k} \gamma_{ij} = 1, \forall i = 1, 2, ..., n$, the total deviation, can be rewritten as:

$$T(X) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij} \|\mathbf{x}_i - \hat{\mathbf{x}}\|^2 / n$$

Exchange the summation order of i and j, add a item μ_j into it and unfold $T(x)$:

$$
\begin{aligned}
T(x) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\|\mathbf{x}_i-\hat{\mathbf{x}}\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}\|\mathbf{x}_i-\mu_j+\mu_j-\hat{\mathbf{x}}\|^2 \\
&= \frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n}\gamma_{ij}\left[\|\mathbf{x}_i-\mu_j\|^2+\|\mu_j-\hat{\mathbf{x}}\|^2+2(\mathbf{x}_i-\mu_j)^T(\mu_j-\hat{\mathbf{x}})\right] \\
&= \frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n}\left[\gamma_{ij}\|\mathbf{x}_i-\mu_j\|^2+\gamma_{ij}\|\mu_j-\hat{\mathbf{x}}\|^2+2\gamma_{ij}(\mathbf{x}_i-\mu_j)^T(\mu_j-\hat{\mathbf{x}})\right] \\
&= \frac{1}{n}\sum_{j=1}^{k}[W_j(X)\sum_{i=1}^{n}\gamma_{ij}]+B(X)+\frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n}2\gamma_{ij}(\mathbf{x}_i-\mu_j)^T(\mu_j-\hat{\mathbf{x}})
\end{aligned}
\tag{3.4}
$$

The equation above means the $T(X)$ can be represented as a combination of k $W_j(X)$s' weighted average, $B(x)$ and a cross item. The k $W_j(X)$s' weighted average is also the loss function J to be minimized in the algorithm. So we can prove that $k-means$ algorithm minimize the weighted average of intra-cluster deviation. Because of $T(x)$ keeps the same once the data is given, the $T(x)$ can be regarded as a constant in the algorithm. So we can obtain:

$$
T(X)-J=B(X)+\frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n}2\gamma_{ij}(\mathbf{x}_i-\mu_j)^T(\mu_j-\hat{\mathbf{x}})
$$

The equation means this algorithm also maximizes the right item of this equation, so considering the cross item we can think $k-means$ approximately maximize $B(X)$, which is inter-cluster deviation.

(5):

$$
J'(\gamma,\mu_1,\ldots,\mu_k)=\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}||\mathbf{x}_i-\mu_j||_1
$$

So we can solve minimum of J' by taking partial derivation: (Note that $\mu_j^{(s)}$ and $\mathbf{x}_i^{(s)}$ mean the sth item of vector)

$$
\begin{aligned}
\frac{\partial J'}{\partial\mu_j^{(s)}} &= \frac{\partial\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}||\mathbf{x}_i^{(s)}-\mu_j^{(s)}||_1}{\partial\mu_j^{(s)}} \\
&= \sum_{i=1}^{n}\gamma_{ij}Sign(\mathbf{x}_i^{(s)}-\mu_j^{(s)})=0
\end{aligned}
\tag{3.5}
$$

The $Sign(x)$ function in equation is :

$$
Sign(x)=\begin{cases}1, & x>0\\ -1, & x<0\end{cases}
\tag{3.6}
$$

The sum of $Sign(\mathbf{x}_i^{(s)}-\mu_j^{(s)}$ have to be zero only when half $\mathbf{x}_i^{(s)}$ are less than $\mu_j^{(s)}$ and half $\mathbf{x}_i^{(s)}$ are larger than $\mu_j^{(s)}$. So we should find the median item to update vector μ_j's each item separately. And we have the algorithm of k-means-ℓ_1:

Algorithm 2: k-means-ℓ_1 Algorithm

Initialize $\mu_1, \ldots, \mu_k$.

repeat

  Step 1: Decide the class memberships of $\{\mathbf{x}_i\}_{i=1}^n$ by assigning each of them to its nearest cluster center in Manhattan distance.

$$\gamma_{ij} = \begin{cases} 1, & ||\mathbf{x}_i - \mu_j||_1 \leq ||\mathbf{x}_i - \mu_{j'}||_1, \forall j' \\ 0, & \text{otherwise} \end{cases}$$

  Step 2: For each $j \in \{1, \cdots, k\}$, recompute μ_j's each item $\mu_j^{(s)}$ using the updated γ to be the median position of all $x_i^{(s)}$ which belongs to cluster of μ_j (for each $\gamma_{ij} = 1$):

  for s in range(d):
    for j in range(k):
      y=[ ]
      for i in range(n):
        if (γ_{ij} == 1):
          y.append($x_i^{(s)}$)
      y.sort()
      if y.shape%2==0:
        $\mu_j^{(s)} = \frac{y[\frac{y.shape}{2}]+y[\frac{y.shape}{2}-1]}{2}$
      else:
        $\mu_j^{(s)} = y[\frac{y.shape-1}{2}]$

until *the objective function J no longer changes*;

If the data set has some outliers, we should use k-means-ℓ_1 algorithm.
In fact, the k-means-ℓ_1 algorithm let μ_j's each component be the median of each component of those x_i's belongs to it. It means even if there are some outliers, it will sort them and take the median value which is only related the order of them. A few outliers won't affect the median value if the values near median is still correct. So it is more robust.
However, the k-means-ℓ_2 algorithm take the average Euclidean distance to each x_i. If there are some outliers, the average may be affected a lot by them, which means k-means-ℓ_2 algorithm isn't robust.

4 [50pts] Kernel, Optimization and Learning

给定样本集$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$, $\mathcal{F} = \{\Phi_1 \cdots, \boldsymbol{\Phi}_d\}$为非线性映射族。考虑如下的优化问题

$$\min_{\mathbf{w}, \mu \in \Delta_q} \quad \frac{1}{2}\sum_{k=1}^{d} \frac{1}{\mu_k}\|\mathbf{w}_k\|_2^2 + C\sum_{i=1}^{m} \max\left\{0, 1 - y_i\left(\sum_{k=1}^{d} \mathbf{w}_k \cdot \boldsymbol{\Phi}_k(\mathbf{x}_i)\right)\right\} \tag{4.1}$$

其中, $\Delta_q = \{\boldsymbol{\mu} | \mu_k \geq 0, k = 1, \cdots, d; \|\boldsymbol{\mu}\|_q = 1\}$.

(1) **[40pts]** 请证明, 下面的问题4.2是优化问题4.1的对偶问题。

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & 2\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1} - \left\| \begin{matrix} \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{K}_1\mathbf{Y}\boldsymbol{\alpha} \\ \vdots \\ \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{K}_d\mathbf{Y}\boldsymbol{\alpha} \end{matrix} \right\|_p \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \end{aligned} \tag{4.2}$$

其中, p和q满足共轭关系, 即$\frac{1}{p} + \frac{1}{q} = 1$. 同时, $\mathbf{Y} = \mathrm{diag}([y_1, \cdots, y_m])$, $\mathbf{K}_k$是由$\boldsymbol{\Phi}_k$定义的核函数(kernel).

(2) **[10pts]** 考虑在优化问题4.2中, 当$p = 1$时, 试化简该问题。

Solution. (1):
Like the solution to SVM of soft margin in the textbook (Page 130), introduce the slack variable $xi_i \geq 0$. Rewrite the optimization problem as:

$$\begin{aligned} \min_{\mathbf{w}, \boldsymbol{\mu} \in \Delta_q, \boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{k=1}^{d} \frac{1}{\mu_k}\|\mathbf{w}_k\|_2^2 + C\sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & \xi_i \geq 0 \\ & y_i\left(\sum_{k=1}^{d} \mathbf{w}_k \Phi_k(\mathbf{x}_i)\right) \geq 1 - \xi_i \\ & \|\boldsymbol{\mu}\|_q = 1 \\ & \mu_k \geq 0 \end{aligned} \tag{4.3}$$

So we can introduce the Lagrange multiplier $\boldsymbol{\alpha}$,$\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and η to construct the Lagrange function:

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \eta) =& \frac{1}{2}\sum_{k=1}^{d} \frac{1}{\mu_k}\|\mathbf{w}_k\|_2^2 + C\sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i\left(1 - \xi_i - y_i(\sum_{k=1}^{d} \mathbf{w}_k \cdot \boldsymbol{\Phi}_k(\mathbf{x}_i))\right) \\ & - \sum_{i=1}^{m} \beta_i\xi_i - \sum_{i=1}^{m} \gamma_i\mu_i + \eta(\|\boldsymbol{\mu}\|_q - 1) \end{aligned} \tag{4.4}$$

Take partial derivative of Lagrange function to $\mathbf{w}_k$, μ_k and ξ_i and let them equal to zero:

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}_k} &= \frac{\mathbf{w}_k}{\mu_k} - \sum_{i=1}^{m} \alpha_i y_i \Phi_k(\mathbf{x}_i) = 0 \\
\frac{\partial L}{\partial \mu_k} &= -\frac{1}{2}\frac{\|\mathbf{w}_k\|_2^2}{\mu_k^2} - \gamma_k + \eta(\frac{\mu_k}{\|\boldsymbol{\mu}\|_q})^{q-1} = 0 \\
\frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \beta_i = 0
\end{aligned} \tag{4.5}$$

So we can obtain the equation relationship between those variables:

$$\begin{aligned}
\frac{\mathbf{w}_k}{\mu_k} &= \sum_{i=1}^{m} \alpha_i y_i \Phi_k(\mathbf{x}_i) \\
\eta(\frac{\mu_k}{\|\boldsymbol{\mu}\|_q})^{q-1} &= \frac{1}{2}\frac{\|\mathbf{w}_k\|_2^2}{\mu_k^2} + \gamma_k \\
C &= \alpha_i + \beta_i
\end{aligned} \tag{4.6}$$

Bring them back to the Lagrange function, erase $\mathbf{w}_k$, μ_k, ξ_i, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and η. Meanwhile notice that

$$C - \alpha_i = \beta_i \geq 0$$

so that we can obtain the dual problem:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\left[\sum_{k=1}^{d}(\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \Phi_k(\mathbf{x}_i)\Phi_k(\mathbf{x}_j))^p\right]^{\frac{1}{p}} \\
\text{s.t.} \quad & 0 \leq \alpha_i \leq C, \forall i
\end{aligned} \tag{4.7}$$

It can be simply recorded as the dual problem in equation 4.2:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & 2\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1} - \left\| \begin{matrix} \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{K}_1\mathbf{Y}\boldsymbol{\alpha} \\ \vdots \\ \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{K}_d\mathbf{Y}\boldsymbol{\alpha} \end{matrix} \right\|_p \\
\text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}
\end{aligned} \tag{4.8}$$

(2):
Unfold the $1 - norm$ form (sum of absolute values):

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & 2\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1} - \sum_{k=1}^{d} |\alpha^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{K}_k\mathbf{Y}\boldsymbol{\alpha}| \\
\text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}
\end{aligned} \tag{4.9}$$

Because of the Kernel matrix are always semi-definite matrix, so their absolute values are themselves. So the form can be simplified as:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^{\mathrm{T}}\left(2 \cdot \mathbf{1} - \sum_{k=1}^{d} Y^{\mathrm{T}}\mathbf{K}_k\mathbf{Y}\boldsymbol{\alpha}\right) \\
\text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}
\end{aligned} \tag{4.10}$$