习题二

Boardwell, Nanjing University

2017 年 7 月 10 日

1 [10pts] Lagrange Multiplier Methods

请通过拉格朗日乘子法(可参见教材附录B.1)证明《机器学习》教材中式(3.36)与式(3.37)等价。即下面公式(**??**)与(**??**)等价。

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} = 1 \end{aligned} \tag{1.1}$$

$$\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w} \tag{1.2}$$

Proof.
According to the definition of Lagrange Multiplier Methods, suppose that $f(x)$ is target function which need to be minimize and $g(x)$ is the restriction function, then we can solve the problem by minimize Lagrange Function $L(x,\lambda) = f(x) + \lambda g(x)$ without constraint.
In this problem,$L(x,\lambda) = -\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w} + \lambda(\mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} - 1)$.
Let $L(x,\lambda)$'s partial derivative of x and λ be 0, and we can solve out the equation when $L(x,\lambda)$ has been minimized:
$\frac{\partial_L}{\partial_\lambda} = \mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} - 1 = 0$
$\frac{\partial_L}{\partial_w} = \frac{\partial\lambda(\mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w}-1)}{\partial_w} - \frac{\partial\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w}}{\partial_w}$
$= -(\mathbf{S}_b + \mathbf{S}_b^{\mathrm{T}})\mathbf{w} + \lambda(\mathbf{S}_w + \mathbf{S}_w^{\mathrm{T}})\mathbf{w}$
(According to equation No.81 in The Matrix Cookbook 2.4.2)
$= -2\mathbf{S}_b\mathbf{w} + 2\lambda\mathbf{S}_w\mathbf{w} = 0$
(Because of both $\mathbf{S}_b$ and $\mathbf{S}_w$ are symmetric matrix)
By shifting item we can achieve that $\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w}$.

Considering the algebraic equivalence when we lagrange multiplier methods, we can ensure that the extremum of original problem is the same to the extremum of lagrange function, which can be regarded as their solution are equivalent,although lagrange function is a linear approximation to original problem and the problems they deal with may not be equivalent.

If the problem satisfy the condition $\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w}$,by the proof above we can deduce that it is the solution to minimize lagrange function,which represent the solution of original problem according to the definition of lagrange multiplier methods. □

2 [20pts] Multi-Class Logistic Regression

教材的章节3.3介绍了对数几率回归解决二分类问题的具体做法。假定现在的任务不再是二分类问题，而是多分类问题，其中$y \in \{1, 2 \ldots, K\}$。请将对数几率回归算法拓展到该多分类问题。

(1) **[10pts]** 给出该对率回归模型的“对数似然”(log-likelihood);

(2) **[10pts]** 计算出该“对数似然”的梯度。

提示1：假设该多分类问题满足如下$K-1$个对数几率，

$$\begin{aligned} \ln \frac{p(y=1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_1^{\mathrm{T}}\mathbf{x} + b_1 \\ \ln \frac{p(y=2|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_2^{\mathrm{T}}\mathbf{x} + b_2 \\ &\cdots \\ \ln \frac{p(y=K-1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x} + b_{K-1} \end{aligned}$$

提示2：定义指示函数$\mathbb{I}(\cdot)$，

$$\mathbb{I}(y=j) = \begin{cases} 1 & \text{若}y\text{等于}j \\ 0 & \text{若}y\text{不等于}j \end{cases}$$

Solution.

(1):In condition of $y=i$,the probability can be represent as:

$\ln \frac{p(y=i|\mathbf{x})}{p(y=K|\mathbf{x})} = \mathbf{w}_i^{\mathrm{T}}\mathbf{x} + b_i$, So that $\frac{p(y=i|\mathbf{x})}{p(y=K|\mathbf{x})} = \mathrm{e}^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}$

Therefore $p(y=i|\mathbf{x}) = \frac{\mathrm{e}^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}}{1+\sum_{s=1}^{K}\mathrm{e}^{\mathbf{w}_s^{\mathrm{T}}\mathbf{x}+b_s}}$

Note that $\beta_i = (w_i; b_i)$ for $i = 1, 2, ...K-1$,$\beta_i = (\mathbf{0}; 0)$ for $i = K$ and the matrix$\beta = (\beta_1, \beta_2, ..., \beta_K)$, the log-likelihood function can be write as:

$l(\beta) = \sum_{i=1}^{m} \ln p(y_i|\mathbf{x_i}; \beta_i)$

Rewrite the probability item:

$p(y_i|\mathbf{x_i}; \beta_i) = \sum_{j=1}^{K} \mathbb{I}(y=j) \ln p_j(\mathbf{x_i}; \beta)$

$= \sum_{j=1}^{K} \mathbb{I}(y=j)(\beta_j^{\mathrm{T}}\mathbf{x_i} - \ln(1 + \sum_{s=1}^{K} \mathrm{e}^{\beta_s^{\mathrm{T}}\mathbf{x_i}}))$

So the log-likelihood function:$l(\beta) = \sum_{i=1}^{m}\sum_{j=1}^{K} \mathbb{I}(y=j)(\beta_j^{\mathrm{T}}\mathbf{x_i} - \ln(1+\sum_{s=1}^{K} \mathrm{e}^{\beta_s^{\mathrm{T}}\mathbf{x_i}}))$

Note that $a = (\mathbb{I}(y=1), \mathbb{I}(y=2), ..., \mathbb{I}(y=K))$.

Take the matrix β as the independent variable,so the log-likelihood function can also be written as:

$l(\beta) = \sum_{i=1}^{m}(a\beta^{\mathrm{T}}\mathbf{x_i} - \ln(1 + \sum_{s=1}^{K} \mathrm{e}^{\beta_s^{\mathrm{T}}\mathbf{x_i}}))$

And the $l(\beta)$ used to be iterated by Newton Methods will be added a minus:

$l(\beta) = \sum_{i=1}^{m}(-a\beta^{\mathrm{T}}\mathbf{x_i} + \ln(1 + \sum_{s=1}^{K} \mathrm{e}^{\beta_s^{\mathrm{T}}\mathbf{x_i}}))$

(2):Now that we have the expression of $l(\beta)$,take partial derivative of each β_i will lead to the gradient:
$[\nabla l(\beta)]_j = \frac{\partial l(\beta)}{\partial_{\beta_j}} = \sum_{i=1}^{m}(-\sum_{s=1}^{K}\mathbb{I}(y_i = s)\frac{\partial \beta_s^{\mathrm{T}}\mathbf{x_i}}{\partial_{\beta_j}} + \frac{\partial \ln(1+\sum_{s=1}^{K}\mathbf{e}^{\beta_s^{\mathrm{T}}\mathbf{x_i}})}{\partial_{\beta_j}})$
Because of $\frac{\partial \beta_j^{\mathrm{T}}\mathbf{x_i}}{\partial_{\beta_j}} = \mathbf{x_i}$ and $\frac{\partial \beta_s^{\mathrm{T}}\mathbf{x_i}}{\partial_{\beta_j}} = 0$(when s is not equal to j).
So $[\nabla l(\beta)]_j = \sum_{i=1}^{m}(-\sum_{s=1}^{K}\mathbb{I}(y_i = s)\mathbb{I}(s = j)\mathbf{x_i} + \frac{\mathbf{x_i}\mathbf{e}^{\beta_j^{\mathrm{T}}\mathbf{x_i}}}{1+\sum_{s=1}^{K}\mathbf{e}^{\beta_s^{\mathrm{T}}\mathbf{x_i}}})$
$= \sum_{i=1}^{m}(-\sum_{s=1}^{K}\mathbb{I}(y_i = s)\mathbb{I}(s = j)\mathbf{x_i} + \mathbf{x_i}p(y = j|\mathbf{x}))$
And the final answer is $[\nabla l(\beta)] = ([\nabla l(\beta)]_1, [\nabla l(\beta)]_2, ..., [\nabla l(\beta)]_K)$,which is a row vector and the j^{th} item of it $[\nabla l(\beta)]_j$ is a column vector.

3 [35pts] Logistic Regression in Practice

对数几率回归(Logistic Regression, 简称LR)是实际应用中非常常用的分类学习算法。

(1) **[30pts]** 请编程实现二分类的LR, 要求采用牛顿法进行优化求解, 其更新公式可参考《机器学习》教材公式(3.29)。详细编程题指南请参见链接：`http://lamda.nju.edu.cn/ml2017/PS2/ML2_programming.html`

(2) **[5pts]** 请简要谈谈你对本次编程实践的感想(如过程中遇到哪些障碍以及如何解决, 对编程实践作业的建议与意见等)。

Solution. (2):
Feelings:
1.Read the guide carefully before you start to program for yourself and learn to know useful library functions instead of fearing to read English documents.I don't know the sklearn library functions at the begining and finish the 10 fold cross validation by myself, but later because of lack of knowledge of the number of new dataset i have to rewrite the cross validation by sklearn.
2.Develop a good program habit.Don't give the variable a name which is difficult to understand and spend a lot of time to debug.
3.Learn to use vectorization processing on arrays instead of cycling,espacially in python and matlab.Use matrix caculation instead of multiplying them by cycling.
Difficulties:
1.Don't know how to control the end of iteration when using Newton Iteration Method because type float64 is too easy to overflow.
What I used in the last is give it a relatively small but effective times to iteration.(I observe a phenomenon that when the iteration times over 5, the accuracy start to decline.)
2.Fear of overflow in type float64.
3.The program itself is not that hard but I use too much time on it and don't know why.(It is still relatively hard for a non-CS student like me if don't know to use library functions).
Suggestion: Please introduce more library functions and documents to us because I used to realize it by myself if don't know there is a library function.(However the real difficulty in

this homework for me is input formulas in LaTeX...I want to escape for many times).

4 [35pts] Linear Regression with Regularization Term

给定数据集$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$, 其中$\mathbf{x}_i = (x_{i1}; x_{i2}; \cdots; x_{id}) \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, 当我们采用线性回归模型求解时, 实际上是在求解下述优化问题：

$$\hat{\mathbf{w}}^*_{\mathbf{LS}} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2, \tag{4.1}$$

其中, $\mathbf{y} = [y_1, \cdots, y_m]^{\mathrm{T}} \in \mathbb{R}^m, \mathbf{X} = [\mathbf{x}_1^{\mathrm{T}}; \mathbf{x}_2^{\mathrm{T}}; \cdots; \mathbf{x}_m^{\mathrm{T}}] \in \mathbb{R}^{m \times d}$, 下面的问题中, 为简化求解过程, 我们暂不考虑线性回归中的截距(intercept)。

在实际问题中, 我们常常不会直接利用线性回归对数据进行拟合, 这是因为当样本特征很多, 而样本数相对较少时, 直接线性回归很容易陷入过拟合。为缓解过拟合问题, 常对公式(**??**)引入正则化项, 通常形式如下：

$$\hat{\mathbf{w}}^*_{\mathbf{reg}} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\Omega(\mathbf{w}), \tag{4.2}$$

其中, $\lambda > 0$为正则化参数, $\Omega(\mathbf{w})$是正则化项, 根据模型偏好选择不同的Ω。

下面, 假设样本特征矩阵$\mathbf{X}$满足列正交性质, 即$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{I}$, 其中$\mathbf{I} \in \mathbb{R}^{d \times d}$是单位矩阵, 请回答下面的问题(需要给出详细的求解过程)：

(1) **[5pts]** 考虑线性回归问题, 即对应于公式(**??**), 请给出最优解$\hat{\mathbf{w}}^*_{\mathbf{LS}}$的闭式解表达式;

(2) **[10pts]** 考虑岭回归(ridge regression)问题, 即对应于公式(**??**)中$\Omega(\mathbf{w}) = \|\mathbf{w}\|_2^2 = \sum_{i=1}^d w_i^2$时, 请给出最优解$\hat{\mathbf{w}}^*_{\mathbf{Ridge}}$的闭式解表达式;

(3) **[10pts]** 考虑LASSO问题, 即对应于公式(**??**)中$\Omega(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_{i=1}^d |w_i|$时, 请给出最优解$\hat{\mathbf{w}}^*_{\mathbf{LASSO}}$的闭式解表达式;

(4) **[10pts]** 考虑ℓ_0-范数正则化问题,

$$\hat{\mathbf{w}}^*_{\ell_0} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_0, \tag{4.3}$$

其中, $\|\mathbf{w}\|_0 = \sum_{i=1}^d \mathbb{I}[w_i \neq 0]$,即$\|\mathbf{w}\|_0$表示$\mathbf{w}$中非零项的个数。通常来说, 上述问题是NP-Hard问题, 且是非凸问题, 很难进行有效地优化得到最优解。实际上, 问题(3)中的LASSO可以视为是近些年研究者求解ℓ_0-范数正则化的凸松弛问题。

但当假设样本特征矩阵$\mathbf{X}$满足列正交性质, 即$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{I}$时, ℓ_0-范数正则化问题存在闭式解。请给出最优解$\hat{\mathbf{w}}^*_{\ell_0}$的闭式解表达式, 并简要说明若去除列正交性质假设后, 为什么问题会变得非常困难？

Solution.

(1):$\hat{\mathbf{w}}^*_{\mathbf{LS}} = argmin\frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}))$

Let $E_{\hat{\mathbf{w}}_{\mathbf{LS}}} = \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})$,take partial devirative on $\hat{\mathbf{w}}_{\mathbf{LS}}$ and we can get:

$\frac{\partial_{E_{\hat{\mathbf{w}}_{\mathbf{LS}}}}}{\partial_{\hat{\mathbf{w}}_{\mathbf{LS}}}} = \mathbf{X}^{\mathbf{T}}(\mathbf{X}\hat{\mathbf{w}}_{\mathbf{LS}} - \mathbf{y}) = 0$

So $\hat{\mathbf{w}}_{\mathbf{LS}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y} = \mathbf{X}^{\mathbf{T}}\mathbf{y}$(because of $\mathbf{X}^{\mathbf{T}}\mathbf{X} = \mathbf{I}$)

(2): Let $E_{\hat{\mathbf{w}}} = \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}) + \lambda\mathbf{w}^{\mathbf{T}}\mathbf{w}$, take partial devirative on $\hat{\mathbf{w}}$:

$\frac{\partial_{E_{\hat{\mathbf{w}}}}}{\partial_{\hat{\mathbf{w}}}} = \mathbf{X^T}(\mathbf{X}\hat{\mathbf{w}} - \mathbf{y}) + 2\lambda\mathbf{w} = 0$
$\mathbf{X^T y} = (\mathbf{X^T X} + 2\lambda\mathbf{I})\mathbf{w} = (1 + 2\lambda)\mathbf{w}$
$\hat{\mathbf{w}}^*_{\mathbf{Ridge}} = \frac{1}{1+2\lambda}\mathbf{X^T y}$
(3): Let $E_{\hat{\mathbf{w}}} = \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}) + \lambda \sum |\mathbf{w}_i|$.
In order to caculate the value of $\frac{\partial|\mathbf{w}_i|}{\partial \mathbf{w}_j}$($|w_i|$ can't get derivative when $w_i = 0$), define the function :

$$sign(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases}$$

So that the result of $E_{\hat{\mathbf{w}}}$'s partial derivative on $\hat{\mathbf{w}}$ is:
$\frac{\partial_{E_{\hat{\mathbf{w}}}}}{\partial_{\hat{\mathbf{w}}}} = \mathbf{X^T}(\mathbf{X}\hat{\mathbf{w}} - \mathbf{y}) + \lambda A$ In which $A = (sign(w_1); sign(w_2); ...; sign(w_d))$
So that the solution is : $\hat{\mathbf{w}}^*_{\mathbf{LASSO}} = \mathbf{X^T y} - \lambda A$
(4):The only solution to l_0-norm optimization is equal to l_1-norm question,if there is a optimal solution to this question.(that's why many researchers turn to research l_1-norm question to substitute the origin question)
When $\mathbf{X^T X} = \mathbf{I}$, there is a optimization solution to l_0-norm question which is:
$\hat{\mathbf{w}}^* = \mathbf{X^T y} - \lambda A$ (the same as solution to l_1-norm question)
Let $E_{\hat{\mathbf{w}}} = \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}) + \lambda \sum_{i=1}^{d} \mathbb{I}[w_i \neq 0]$
Take partial derivative on $E_{\hat{\mathbf{w}}}$ like before.
Because of $\mathbb{I}[w_i \neq 0]$ can't get derivative when $w_i = 0$.(As a physics student, I learned a Dirac's δ function which is a generalized functionin our course about Fourier Transform and δ function is the generalized derivative of Heaviside function.But the left derivative of $\mathbb{I}[w_i \neq 0]$ is $-\delta(0)$ and the right derivative of $\mathbb{I}[w_i \neq 0]$ is $\delta(0)$ so that its derivative is not exist)
So $\frac{\partial \mathbb{I}[w_i \neq 0]}{\partial w_i} = 0 (w_i \neq 0)$
However, if can't get the derivative of $\mathbb{I}[w_i \neq 0]$ in $w_i = 0$,it is difficult to solve $\hat{\mathbf{w}}^*_{\ell_0}$ and that's why l_0 norm optimization is difficult to solve.