Machine Learning PS: 2

Author: *LIUQiaoan 520030910220*

Course: *Autumn 2018, CS229: Machine Learning*
Date: *July 18, 2022*

You can see each problem at Problem 1, Problem 2, Problem 3, Problem 4, Problem 5, Problem 6.

Problem 1

(a) The most notable difference in training the logistic regression model on datasets A and B is: We need much less iterations when training on datasets A than B.

(b) After plot the datasets, we can see that datasets A can not be divided linearly, but datasets B can. We then come back to the code to see why.

In this logistic regression,

$$\nabla_\theta J(theta) = -\frac{1}{m}\sum_{i=1}^{m}\frac{y^{(i)}x^{(i)}}{1+\exp\left(y^{(i)}\theta^T x\right)}$$

we want to minimize

$$\begin{aligned} J(\theta) &= -\frac{1}{m}\sum_{i=1}^{m}\log p(y^{(i)}|x^{(i)};\theta) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\log\frac{1}{1+\exp\left(-y^{(i)}\theta^T x^{(i)}\right)} \\ &= \frac{1}{m}\sum_{i=1}^{m}\log 1+\exp\left(-y^{(i)}\theta^T x^{(i)}\right) \end{aligned}$$

Then we can use similar result of SVM. Since $y^{(i)}\theta^T x^{(i)})$ is functional margin and we can scale θ to increase functional margin without change the decision boundary when datasets can be seperated linearly. Then for datasets B, this code will repeat to increase θ and will not converge. However, for datasets A, since it's not linearly seperable, we can not infinitely increase θ.

(c)

i This can not lead to the convergence in datasets B, since this change can not prevent θ to be larger.

ii This can lead to the convergence of datasets B, since θ can only increase linearly over time, while the decaying of learning rate by $\frac{1}{t^2}$ will make $\alpha\nabla_\theta J(theta) < 1e-15$ after some iterations.

iii This can not lead to the convergence in datasets B, since the linear scaling can not prevent θ to be larger and then the functional margin to be larger.

iv This can lead to the convergence of datasets B, since this regularization term $\|\theta\|^2$ can prevent $\|\theta\|$ to be too big, thus it can converge after some iterations.

v This can lead to the convergence of datasets B, since add Gaussian noise can make the datasets not linearly seperable. (**After referring to the solution**)

(d) After read this article: https://www.zhihu.com/question/47746939. I get to know that loss is 0 only if functional margin is larger than or equal to 1, which is a more strict condition.

$$J(\theta) = \sum_{i=1}^{m} \max\{1 - y^{(i)}(w^T x^{(i)} + b), 0\}$$

Then if the functional margin is larger than 1, $\max\{1 - y^{(i)}(w^T x^{(i)} + b), 0\} = 0$ instead some negative value, so θ will not be too big. As a result, SVM using hinge loss is not vulnerable to datasets like B.

Problem 2

(a) This result is just come from the fact that logistic regression is included in exponential family.

$$l(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} h_\theta(x^{(i)}) + (1 - y^{(i)})(1 - h_\theta(x^{(i)})))$$

Then

$$\frac{\partial l(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} = 0 \iff \sum_{i=1}^{m} h_\theta(x^{(i)}) = \sum_{i=1}^{m} y^{(i)} \text{ (since } x_0^{(i)} = 1 \text{ for all } i \text{)}$$

Thus, for $(a, b) = (0, 1)$, we have $\{i \in I_{a,b}\} = S$,

$$\frac{\sum_{i \in I_{a,b}} h_\theta(x^{(i)})}{|\{i \in I_{a,b}\}|} = \frac{\sum_{i \in I_{a,b}} y^{(i)}}{|\{i \in I_{a,b}\}|}$$

(b) If we have a binary classification model that is perfectly calibrated, this condition does NOT necessarily imply that the model achieves perfect accuracy.

Proof. We use a contradiction. If the model achieves perfect accuracy, then for $(a, b) = (0.5, 1)$,

$$\frac{\sum_{i \in I_{a,b}} h_\theta(x^{(i)})}{|\{i \in I_{a,b}\}|} < 1$$
$$\frac{\sum_{i \in I_{a,b}} y^{(i)}}{|\{i \in I_{a,b}\}|} = 1$$

which means

$$\frac{\sum_{i\in I_{a,b}} h_\theta(x^{(i)})}{|\{i \in I_{a,b}\}|} \neq \frac{\sum_{i\in I_{a,b}} y^{(i)}}{|\{i \in I_{a,b}\}|}$$

So the model is not perfectly calibrated.

Also, the converse is not true.

Proof. We use a contradiction. If the model is perfectly calibrated, then for $(a, b) = (0.5, 1)$,

$$\frac{\sum_{i\in I_{a,b}} y^{(i)}}{|\{i \in I_{a,b}\}|} = \frac{\sum_{i\in I_{a,b}} h_\theta(x^{(i)})}{|\{i \in I_{a,b}\}|} < 1$$

which means this model does not achieve perfect accuracy.

(c) If we use $\lambda\|\theta\|_2^2$ as a regularization term, the NLL is now:

$$l(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}h_\theta(x^{(i)}) + (1 - y^{(i)})(1 - h_\theta(x^{(i)}))) + \lambda\|\theta\|_2^2$$

If we take derivative, and since $x_0^{(i)} = 1$ for all i:

$$\frac{\partial l(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} + 2\lambda\theta_j = 0 \iff \sum_{i=1}^{m} h_\theta(x^{(i)}) + 2\lambda\theta_0 = \sum_{i=1}^{m} y^{(i)}$$

Thus, this model is not well-calibrated.

Problem 3

(a)

$$\begin{aligned}
p(\theta|x, y) &= \frac{p(x, y, \theta)}{p(x, y)} \\
&= \frac{p(y|x, \theta)p(x, \theta)}{p(x, y)} \\
&= \frac{p(y|x, \theta)p(\theta|x)p(x)}{p(y|x)p(x)} \\
&= \frac{p(y|x, \theta)p(\theta|x)}{p(y|x)} \\
&= \frac{p(y|x, \theta)p(\theta)}{p(y|x)}
\end{aligned}$$

Then

$$\begin{aligned}
\theta_{MAP} &= \arg\max_\theta \frac{p(y|x, \theta)p(\theta)}{p(y|x)} \\
&= \arg\max_\theta p(y|x, \theta)p(\theta)
\end{aligned}$$

(b) From sub-problem (a), we get that:

$$\begin{aligned}
\theta_{MAP} &= \arg\max_{\theta} p(y|x,\theta)p(\theta) \\
&= \arg\max_{\theta} p(y|x,\theta)\frac{1}{2\eta^2}\exp\left(-\frac{\|\theta\|_2^2}{2\eta^2}\right) \\
&= \arg\max_{\theta} \log p(y|x,\theta) - \frac{\|\theta\|_2^2}{2\eta^2} \quad \text{(take log function)} \\
&= \arg\min_{\theta} -\log p(y|x,\theta) + \frac{\|\theta\|_2^2}{2\eta^2}
\end{aligned}$$

Thus, $\lambda = \frac{1}{2\eta^2}$.

(c) For this specific instance, $p(y^{(i)}|x^{(i)},\theta) = \exp\left(\frac{\|y^{(i)}-\theta^T x\|^2}{2\sigma^2}\right)$,

$$\begin{aligned}
\theta_{MAP} &= \arg\min_{\theta} \left\{ -\sum_{i=1}^{m} \log\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|y^{(i)}-\theta^T x\|^2}{2\sigma^2}\right)\right] + \frac{1}{2\eta^2}\|\theta\|^2 \right\} \\
&= \arg\min_{\theta} \frac{1}{2\sigma^2}(\vec{y}-X\theta)^T(\vec{y}-X\theta) + \frac{1}{2\eta^2}\theta^T\theta
\end{aligned}$$

Let $l(\theta) \triangleq \frac{1}{2\sigma^2}(\vec{y}-X\theta)^T(\vec{y}-X\theta) + \frac{1}{2\eta^2}\theta^T\theta = \frac{1}{2\sigma^2}\vec{y}^T\vec{y} - \frac{1}{\sigma^2}\vec{y}^T X\theta + \frac{1}{2\sigma^2}\theta^T X^T X\theta + \frac{1}{2\eta^2}\theta^T\theta$, we take derivative of $l(\theta)$ w.r.t θ:

$$\nabla_\theta l(\theta) = -\frac{1}{\sigma^2}X^T\vec{y} + \frac{1}{\sigma^2}X^T X\theta + \frac{1}{\eta^2}\theta = 0 \iff \theta = (X^T X + \frac{\sigma^2}{\eta^2}I)^{-1}X^T\vec{y}$$

Thus,

$$\begin{aligned}
\theta_{MAP} &= \arg\min_{\theta} l(\theta) \\
&= (X^T X + \frac{\sigma^2}{\eta^2}I)^{-1}X^T\vec{y}
\end{aligned}$$

(**Note:** Must be careful when dealing with probability)

(d) We use the result of sub-problem (a):

$$
\begin{aligned}
\theta_{MAP} &= \arg\max_{\theta} p(y|x,\theta)p(\theta) \\
&= \arg\max_{\theta} p(y|x,\theta)\frac{1}{2b}\exp(-\frac{|\theta|}{b}) \\
&= \arg\max_{\theta} \log p(y|x,\theta) - \frac{|\theta|}{b} \quad \text{(take log function)} \\
&= \arg\min_{\theta} -\log p(y|x,\theta) + \frac{|\theta|}{b} \\
&= \arg\min_{\theta} \left\{ -\sum_{i=1}^{m} \log\left[\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{\|y^{(i)}-\theta^T x\|^2}{2\sigma^2})\right] + \frac{1}{2\eta^2}\|\theta\|^2 \right\} \\
&= \arg\min_{\theta} \frac{1}{2\sigma^2}\|\vec{y}-X\theta\|_2^2 + \frac{|\theta|}{b} \\
&= \arg\min_{\theta} \frac{1}{2\sigma^2}(\|\vec{y}-X\theta\|_2^2 + \frac{2\sigma}{b}|\theta|) \\
&= \arg\min_{\theta} \frac{1}{2\sigma^2}J(\theta)
\end{aligned}
$$

where $\gamma = \frac{2\sigma}{b}$.

Conclusion: For this problem, we have two remarks:

Remark: Linear regression with L_2 regularization is also commonly called *Ridge regression*, and when L_1 regularization is employed, is commonly called *Lasso regression*. These regularizations can be applied to any Generalized Linear models just as above (by replacing $\log p(y|x;)$ with the appropriate family likelihood). Regularization techniques of the above type are also called *weight decay*, and *shrinkage*. The Gaussian and Laplace priors encourage the parameter values to be closer to their mean (i.e., zero), which results in the shrinkage effect.

Remark: Lasso regression (i.e L_1 regularization) is known to result in sparse parameters, where most of the parameter values are zero, with only some of them non-zero.

Then we can also see that adding Gaussian and Laplace priors to MAP has the same effect as adding a regularization term for MLE.

Problem 4

Suppose $z \in \mathbb{R}^n$. (a) Yes.

- Symmetric. $K(x^{(i)}, x^{(j)}) = K_1(x^{(i)}, x^{(j)}) + K_2(x^{(i)}, x^{(j)}) = K_1(x^{(j)}, x^{(i)}) + K_2(x^{(j)}, x^{(i)}) = K(x^{(j)}, x^{(i)})$.

- PSD.

$$z^T K(x^{(i)}, x^{(j)})z = z^T(K_1(x^{(i)}, x^{(j)}) + K_2(x^{(i)}, x^{(j)}))z = z^T K_1(x^{(i)}, x^{(j)})z + z^T K_2(x^{(i)}, x^{(j)})z \geq 0$$

.

(b) No. Suppose $K_1(x,z) = I, K_2(x,z) = 2I \in \mathbb{R}^n \times \mathbb{R}^n$, then $K(x,z) = -I \in \mathbb{R}^n \times \mathbb{R}^n$, which is not PSD.

(c) Yes.

- Symmetric. $K(x^{(i)}, x^{(j)}) = aK_1(x^{(i)}, x^{(j)}) = aK_1(x^{(j)}, x^{()i}) = K(x^{(j)}, x^{(i)})$.
- PSD. $z^T K(x^{(i)}, x^{(j)})z = az^T K_1(x^{(i)}, x^{(j)})z \geq 0$.

(d) No, suppose $K_1(x, z) = I \in \mathbb{R}^n \times \mathbb{R}^n$, then $K(x, z) = -aI \in \mathbb{R}^n \times \mathbb{R}^n$, which is not PSD.

(e) Yes.

- Symmetric.
$$K(x^{(i)}, x^{(j)}) = K_1(x^{(i)}, x^{(j)})K_2(x^{(i)}, x^{(j)}) = K_1(x^{(j)}, x^{(i)})K_2(x^{(j)}, x^{(i)}) = K(x^{(j)}, x^{(i)})$$
.
- PSD.
$$\begin{aligned}
z^T K(x^{(i)}, x^{(j)})z &= \sum_i \sum_j z^T K_1(x^{(i)}, x^{(j)})K_2(x^{(i)}, x^{(j)})z \\
&= \sum_i \sum_j z_i \phi_1^T(x^{(i)})\phi_1(x^{(j)})\phi_2^T(x^{(i)})\phi_2(x^{(j)})z_j \\
&= \sum_i \sum_j z_i \sum_k \phi_1(x^{(i)})_k \phi_1(x^{(j)})_k \sum_l \phi_2(x^{(i)})_l \phi_2(x^{(j)})_l z_j \\
&= \sum_k \sum_l \sum_i (z_i \phi_1(x^{(i)})_k \phi_2(x^{(i)})_l) \sum_j (z_j \phi_1(x^{(j)})_k \phi_2(x^{(j)})_l) \\
&= \sum_k \sum_l (\sum_i (z_i \phi_1(x^{(i)})_k \phi_2(x^{(i)})_l))^2 \\
&\geq 0
\end{aligned}$$

(f) Yes.

- Symmetric. $K(x, z) = f(x)f(z) = f(z)f(x) = K(z, x)$.
- PSD. $z^T K(x^{(i)}, x^{(j)})z = \sum_i \sum_j z_i K_{ij} z_j = \sum_i \sum_j f(x^{(i)})f(x^{(j)})z_j = (\sum_i z_i f(x^{(i)}))^2 \geq 0$.

(g) Yes.

- Symmetric. $K(x, z) = K_3(\phi(x), \phi(z)) = K_3(\phi(z), \phi(x)) = K(z, x)$.
- PSD. Since K_3 is a valid kernel, $z^T K_3 z \geq 0$.

(h) Yes. This sub-problem can directly be proved from sub-problem (a), (c) and (e). From (e), we can get that any positive integer order of a kernel is still a kernel, and from (c) we can get that positive coefficients times a kernel is still a kernel, then from (a), adding each order together is a kernel. So $p(K_1(x, z))$ is a kernel.

Problem 5

(a) i. From the updating rule:

$$\theta^{(i+1)} := \theta^{(i)} + \alpha(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)}))\phi(x^{(i+1)})$$

we can get that $\theta^{(i)}$ is a linear combination of $\phi(x^{(j)}), j = 1, 2, \ldots, i$. We let the coefficients to be β_j, then $\theta^{(i)} = \sum_{j=1}^{i} \beta_j \phi(x^{(j)})$. Thus, $\theta^{(0)}$ can be expressed as $\sum_{j=1}^{0} \beta_j \phi(x^{(j)}) = \vec{0}$.

ii.

$$\begin{aligned} h_{\theta^{(i)}}(x^{(i+1)}) &= g({\theta^{(i)}}^T \phi(x^{(i+1)})) \\ &= sign(\sum_{j=1}^{i} \beta_j \phi(x^{(j)})^T \phi(x^{(i+1)})) \\ &= sign(\sum_{j=1}^{i} \beta_j K(x^{(j)}, x^{(i+1)})) \end{aligned}$$

iii. Plug the result of i. and ii. into the update rule:

$$\begin{aligned} \theta^{(i+1)} &:= \theta^{(i)} + \alpha(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)}))\phi(x^{(i+1)}) \\ &= \sum_{j=1}^{i} \beta_j \phi(x^{(j)}) + \alpha(y^{(i+1)} - sign(\sum_{j=1}^{i} \beta_j K(x^{(j)}, x^{(i+1)})))\phi(x^{(i+1)}) \end{aligned}$$

Then $\beta_{i+1} = \alpha(y^{(i+1)} - sign(\sum_{j=1}^{i} \beta_j K(x^{(j)}, x^{(i+1)})))$.

(c) The dot product kernel performs badly. Since it only uses linear features, and can only have a linear decision boundary. While the radial basis function kernel have infinite-dimension features, thus it can predict pretty well.

Problem 6

(b) When inplement `predict_from_naive_bayes_model`, we use the formula:

$$\begin{aligned} p(y=1|x) &= \frac{(\prod_{j=1}^{n} p(x_j|y=1)_j^x)p(y=1)}{(\prod_{j=1}^{n} p(x_j|y=1)_j^x)p(y=1) + (\prod_{j=1}^{n} p(x_j|y=0)_j^x)p(y=0)} \\ &= \frac{1}{1 + \frac{(\prod_{j=1}^{n} p(x_j|y=0)^{x_j})p(y=0)}{(\prod_{j=1}^{n} p(x_j|y=1)^{x_j})p(y=1)}} \\ &= \frac{1}{1 + \exp\left\{\sum_{j=1}^{n} x_j \log p(x_j|y=0) - \sum_{j=1}^{n} x_j \log p(x_j|y=1) + \log\frac{1-\phi}{\phi}\right\}} \end{aligned}$$

Then we can use $\sum_{j=1}^{n} x_j \log p(x_j|y=0) - \sum_{j=1}^{n} x_j \log p(x_j|y=1) + \log\frac{1-\phi}{\phi}$ as an indicator.