Stanford CS229 ps2 sol

Shuifan Wang

July 2025

1 Logistic Regression: Training stability

(a) This logistic regression training algorithm converges on set A, but not on set B.

```
==== Training model on data set A ====
Finished 10000 iterations
Finished 20000 iterations
Finished 30000 iterations
Converged in 30386 iterations

==== Training model on data set B ====
Finished 10000 iterations
Finished 20000 iterations
Finished 30000 iterations
Finished 40000 iterations
Finished 50000 iterations
Finished 60000 iterations
Finished 70000 iterations
Finished 80000 iterations
Finished 90000 iterations
Finished 100000 iterations
Finished 110000 iterations
...
```

(b) The scatter plot indicates that dataset B is separable, while dataset A is not. Examining the code, then you will find that the algorithm is to minimize

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

since the gradient is

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

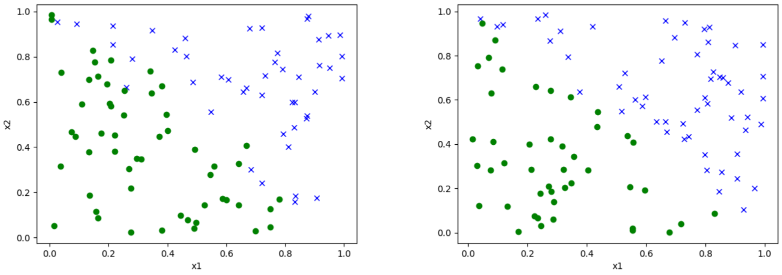

Figure 1: Problem 1(a)

If the dataset is separable, then we can always double θ to make the loss function smaller, while $y^{(i)}\theta^T x^{(i)} > 0$ still holds true. So θ can never converge.
(c)

i. No. It will make no difference as the gradient is multiplied by a scalar.

ii. Yes. The iteration will stop if the difference between θ is small enough due to a decreasing learning rate.

iii. No. It is the same way as scaling θ.

iv. Yes. A large regularization term will hinder θ from increasing.

v. Yes. In this way, the dataset may be inseparable.

(d) From the supplementary notes we know that the hinge loss is

$$l(\theta) = \max(0, 1 - y^{(i)}\theta^T x^{(i)})$$

for $(x^{(i)}, y^{(i)})$. If we multiply θ by a positive scalar, then $l(\theta)$ will be minimized to 0 and the algorithm stops.

2 Model Calibration

(a) Maximizing the likelihood gives

$$\frac{\partial l}{\partial \theta_j} = 0 \Rightarrow \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} = 0.$$

Since we have the intercept term $x_0^{(i)} = 1$,

$$\sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) = 0.$$

Therefore

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

$$= \frac{\sum_{i\in I_{a,b}} y^{(i)}}{|\{i\in I_{a,b}\}|} = \frac{\sum_{i\in I_{a,b}} \mathbb{I}\{y^{(i)}=1\}}{|\{i\in I_{a,b}\}|}.$$

(b) No. If $(a,b)=(0.6,0.8)$, then

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

while perfect accuracy indicates that

$$\frac{\sum_{i\in I_{a,b}} \mathbb{I}\{y^{(i)}=1\}}{|\{i\in I_{a,b}\}|} = 1.$$

The converse is also not true.
(c) With a L_2 regularization term,

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

(Since we are minimizing after introducing a regularization term, we take the negative value of logarithm.) If $\theta_0 = 0$, then the well-calibrated property still holds true, otherwise, not true.

3 Bayesian Interpretation of Regularization

(a) Since we have $p(\theta) = p(\theta|x)$,

$$p(\theta|x,y) = \frac{p(\theta,x,y)}{p(x,y)} = \frac{p(y|x,\theta)p(\theta|x)p(x)}{p(x,y)} = \frac{p(y|x,\theta)p(\theta)p(x)}{p(x,y)}.$$

For given $p(x,y)$ and $p(x)$,

$$\theta_{\text{MAP}} = \arg\max_\theta p(\theta|x,y) = \arg\max_\theta \frac{p(y|x,\theta)p(\theta)p(x)}{p(x,y)} = \arg\max_\theta p(y|x,\theta)p(\theta).$$

(b)

$$\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_\theta \log p(y|x,\theta)p(\theta) \\
&= \arg\min_\theta(-\log p(y|x,\theta) - \log p(\theta)) \\
&= \arg\min_\theta(-\log p(y|x,\theta) - \log(\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\theta-\mu)^T\Sigma^{-1}(\theta-\mu)))) \\
&= \arg\min_\theta(-\log p(y|x,\theta) - \log(\frac{1}{\eta^n} \exp(-\frac{1}{2\eta^2}\theta^T\theta))) \\
&= \arg\min_\theta(-\log p(y|x,\theta) + \frac{1}{2\eta^2}||\theta||_2^2) \\
&= \arg\min_\theta(-\log p(y|x,\theta) + \lambda||\theta||_2^2).
\end{aligned}$$

So

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

(c)

$$\begin{aligned}
\theta_{\text{MAP}} &= \arg\min_\theta(-\log p(y|x,\theta) + \lambda||\theta||_2^2) \\
&= \arg\min_\theta(-\log p(\epsilon + \theta^T x|x,\theta) + \lambda||\theta||_2^2) \\
&= \arg\min_\theta(\sum_{i=1}^m(\frac{1}{2}\log(2\sigma^2) + \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) + \lambda||\theta||_2^2) \\
&= \arg\min_\theta(\frac{(Y-X\theta)^T(Y-X\theta)}{2\sigma^2} + \lambda\theta^T\theta) = \arg\min_\theta J(\theta).
\end{aligned}$$

$$\nabla_\theta J(\theta) = \nabla_\theta(\frac{\theta^T X^T X\theta - 2Y^T X\theta}{2\sigma^2} + \frac{\theta^T\theta}{2\eta^2}) = \frac{X^T X\theta - X^T Y}{\sigma^2} + \frac{\theta}{\eta^2} = 0.$$

Therefore,

$$\theta_{\text{MAP}} = (\eta^2 X^T X + \sigma^2 I)^{-1}(\eta^2 X^T Y) = (X^T X + \frac{\sigma^2}{\eta^2}I)^{-1}X^T Y.$$

$$(Y = \vec{y})$$

(d) Similarly,

$$
\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_\theta \log p(y|x,\theta)p(\theta) \\
&= \arg\min_\theta(-\log p(y|x,\theta) - \log p(\theta)) \\
&= \arg\min_\theta(-\log p(x^T\theta + \epsilon|x,\theta) - \log(\frac{1}{(2b)^n}\exp(-\sum_{i=1}^{n}\frac{|\theta_i|}{b}))) \\
&= \arg\min_\theta(\sum_{i=1}^{m}(\frac{1}{2}\log(2\sigma^2) + \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) + \sum_{i=1}^{n}(\log(2b) + \frac{|\theta_i|}{b})) \\
&= \arg\min_\theta(\frac{||X\theta - Y||_2^2}{2\sigma^2} + \frac{||\theta||_1}{b}) \\
&= \arg\min_\theta(||X\theta - Y||_2^2 + \frac{2\sigma^2||\theta||_1}{b}) = J(\theta).
\end{aligned}
$$

Therefore

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

4 Constructing kernels

(a) Yes. $K_1(x,z) + K_2(x,z)$ is symmetric and $y^T K y = y^T K_1 y + y^T K_2 y \geq 0$.
(b) No. Let $K_2 = 2K_1$, K is not semi-definite.
(c) It depends. As for $a > 0$, the answer is yes. (If $a = 0$, then K is useless.)
(d) It depends. As for $a < 0$, the answer is yes.
(e) Yes. $K_1(x,z)K_2(x,z)$ is symmetric and

$$
\begin{aligned}
y^T K y =& y^T K_1 K_2 y \\
=& \sum_i \sum_j y_i K_1(x^{(i)}, x^{(j)}) K_2(x^{(i)}, x^{(j)}) y_j \\
=& \sum_i \sum_j y_i \phi_1(x^{(i)})^T \phi_1(x^{(j)}) \phi_2(x^{(i)})^T \phi_2(x^{(j)}) y_j \\
=& \sum_i \sum_j y_i \phi_1(x^{(i)})^T \phi_1(x^{(j)}) \phi_2(x^{(i)})^T \phi_2(x^{(j)}) y_j \\
=& \sum_i \sum_j y_i (\sum_m \phi_{1m}(x^{(i)}) \phi_{1m}(x^{(j)}))(\sum_n \phi_{2n}(x^{(i)}) \phi_{2n}(x^{(j)})) y_j \\
=& \sum_m \sum_n \sum_i \sum_j y_i \phi_{1m}(x^{(i)}) \phi_{2n}(x^{(i)}) y_j \phi_{1m}(x^{(j)}) \phi_{2n}(x^{(j)}) \\
=& \sum_m \sum_n \sum_i (y_i \phi_{1m}(x^{(i)}) \phi_{2n}(x^{(i)}))^2 \geq 0.
\end{aligned}
$$

Therefore K is semi-definite.

(f) Yes. $f(x)f(z) = f(z)f(x)$ is symmetric and

$$\begin{aligned} y^T K y =& \sum_i \sum_j y_i f(x^{(i)}) f(x^{(j)}) y_j \\ =& (\sum_i y_i f(x^{(i)}))^2 \geq 0. \end{aligned}$$

Therefore K is semi-definite.

(g) Yes. $K(x,z) = K_3(\phi(x), \phi(z)) = K_3(\phi(z), \phi(x)) = K(z,x)$ is symmetric and

$$\begin{aligned} y^T K y =& \sum_i \sum_j y_i \xi(\phi(x^{(i)})) \xi(\phi(x^{(j)})) y_j \\ =& (\sum_i y_i \xi(\phi(x^{(i)})))^2 \geq 0. \end{aligned}$$

Therefore K is semi-definite. Regardless of the input.

(h) Yes. We can derive it using (a) and (e).

5 Kernelizing the Perceptron

(a)

i. $\theta^{(0)} = 0$. Since we have $\theta^{(i+1)} := \theta^{(i)} + \alpha(y^{(i+1)} - h_{\theta^{(i)}}(\phi(x^{(i+1)})))\phi(x^{(i+1)})$,

$$\theta^{(i)} = \alpha \sum_{j=1}^{i} (y^{(j)} - h_{\theta^{(j-1)}}(\phi(x^{(j)})))\phi(x^{(j)}), i = 1, 2 \ldots$$

$h_{\theta^{(j-1)}}(\phi(x^{(j)})) = 0$ or 1, so $\theta^{(i)}$ is a linear combination of $\phi(x^{(i)})$.

ii.

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

iii.

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

and $h_{\theta^{(i)}}(\phi(x^{(i+1)}))$ can be computed via ii. Denote $\alpha(y^{(j)} - h_{\theta^{(j-1)}}(\phi(x^{(j)})))$ by β_j, then the update rule should be

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

(b) (c) We can see from figure that dot kernel fails, but radial basis function kernel succeeds in classifying the points. This is because the dot kernel has no feature mapping, and therefore it retains the original linearity.

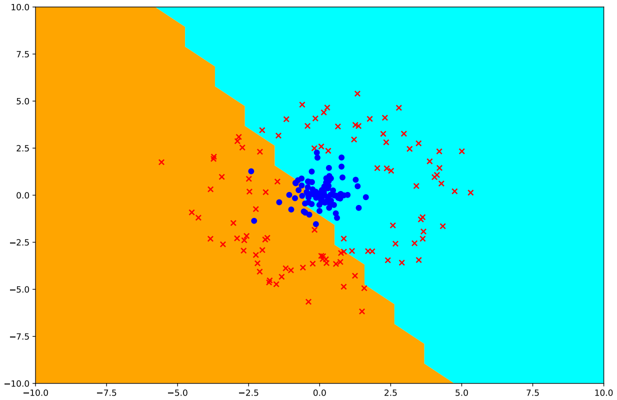

Figure 2: Problem 5 dot

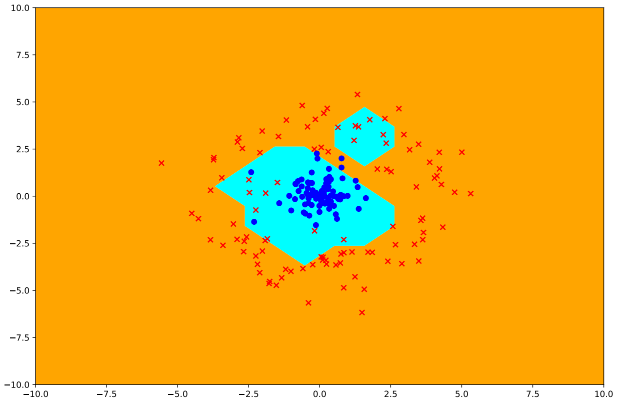

Figure 3: Problem 5 rbf

6 Spam classification

(b) We supplement here the derivation.

$$\begin{aligned}\ell(\phi_y, \phi_{k|y=1}, \phi_{k|y=0}) &= \sum_{i=1}^{m} \log p(x^{(i)}, y^{(i)};\ \phi_y, \phi_{k|y=1}, \phi_{k|y=0}) \\ &= \sum_{i=1}^{m} \log(p(x^{(i)} \mid y^{(i)};\ \phi_{k|y=1}, \phi_{k|y=0})\ p(y^{(i)};\ \phi_y)),\end{aligned}$$

where $x^{(i)} \mid y^{(i)} = 1; \phi_{k|y=1} \sim \mathrm{Mult}(n^{(i)}, \phi_{k|y=1})$, $\phi_{k|y=1}$ is the probability that word k appears given $y = 1$. Similar for $\phi_{k|y=0}$. Therefore

$$p(x^{(i)} \mid y^{(i)} = 1) = \frac{n^{(i)}!}{\prod_k x_k^{(i)}!} \prod_{k=1}^{n} \phi_{k|y=1}^{x_k^{(i)}},$$

where $n^{(i)}$ denotes the whole words count in message i. Therefore, to maximize ℓ w.r.t $\phi_{k|y=1}$ is equivalent to maximizing

$$\sum_{i=1}^{m} 1\{y^{(i)} = 1\} \sum_{k=1}^{n} x_k^{(i)} \log \phi_{k|y=1},$$

subject to the constraints $\phi_{k|y=1} \geq 0$ and $\sum_{k=1}^{n} \phi_{k|y=1} = 1$, where $x_k^{(i)}$ is the number of times that word k appears in the i-th message. By introducing a Lagrange multiplier $\lambda \in \mathbb{R}$ corresponding to the second constraint, we have

$$\mathcal{L}(\lambda, \phi_{k|y=1}) = \sum_{i=1}^{m} 1\{y^{(i)} = 1\} \sum_{i=1}^{n} x_k^{(i)} \log \phi_{k|y=1} - \lambda\big(\sum_{k=1}^{n} \phi_{k|y=1} - 1\big).$$

By solving the Lagrange multiplier problem and applying Laplace smoothing,

$$\phi_{k|y=1} := \frac{1 + \sum_{i=1}^{m} 1\{y^{(i)} = 1\} x_k^{(i)}}{n + \sum_{i=1}^{m} 1\{y^{(i)} = 1\} \sum_{j=1}^{n} x_j^{(i)}}.$$

Similarly,

$$\phi_{k|y=0} := \frac{1 + \sum_{i=1}^{m} 1\{y^{(i)} = 0\} x_k^{(i)}}{n + \sum_{i=1}^{m} 1\{y^{(i)} = 0\} \sum_{j=1}^{n} x_j^{(i)}}$$

and ϕ_y is still $(1/m) \sum_{i=1}^{m} 1\{y^{(i)} = 1\}$.

To make a prediction, instead of computing the probabilities explicitly, we compare the logarithm of $p(y = 1 \mid x)/p(y = 0 \mid x)$ with 0. (Copied from the solution.) (Next question offers some hints.)

$$\begin{aligned}\log \frac{p(y=1 \mid x)}{p(y=0 \mid x)} &= \log \frac{p(x \mid y=1)\ p(y=1)}{p(x \mid y=0)\ p(y=0)}\\ &= \log \frac{\left(\prod_{k=1}^{n} p(x_k \mid y=1)\right)p(y=1)}{\left(\prod_{k=1}^{n} p(x_k \mid y=0)\right)p(y=0)}\\ &= \sum_{k=1}^{n} x_k(\log \phi_{k|y=1} - \log \phi_{k|y=0}) + \log \frac{\phi_y}{1-\phi_y}.\end{aligned}$$

Naive Bayes had an accuracy of 0.978494623655914 on the testing set.
(c) The top 5 indicative words for Naive Bayes are: `['claim', 'won', 'prize', 'tone', 'urgent!']`.
(d) The optimal SVM radius was 0.1. The SVM model had an accuracy of 0.9695340501792115 on the testing set.