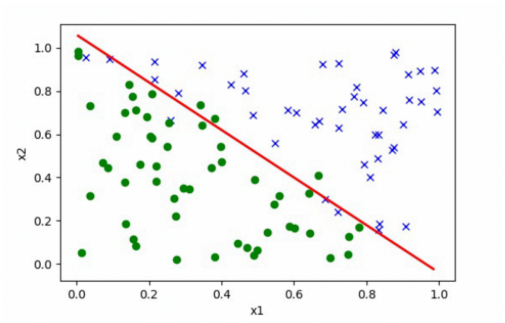
CS 229 Pset 2

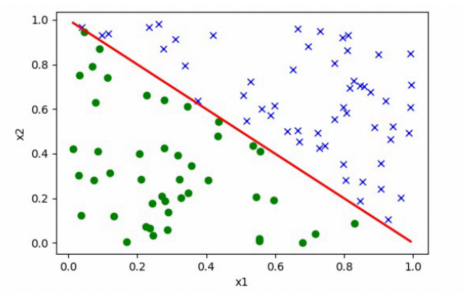
Question One.

(a) The model applied to dataset A converges, but the model on Dataset B does not converge.

(b)



Dataset A



Dataset B

Observe that Dataset B is linearly separable, so the logistic regression cannot choose an optimal θ since we can easily rescaling the weights of each θ to have infinitely many line seperating dataset B.

The loss function $L = -\log \frac{1}{1+\exp(-k\theta^T x)}$ as $k \to \infty$ will keep decreasing and so θ never converges.

In the other hand, rescaling θ on dataset A makes a different, and the Loss function is not monotonically decreasing. i.e. we can find a minimum value of θ.

(c) i. changing the learning rate cannot change the separability of the dataset, so B still cannot converge.

ii. changing the learning rate by scaling it cannot change the separability of the dataset, so B still cannot converge.

iii. Linear scaling can result in a linear transformation to the input data. However, this cannot change the separability of the dataset, so B still cannot converge.

iv. Adding a regularization is adding an additional penalty term in the error function. The weight parameter is smaller each time there is an update. Thereby, it tunes the loss function so that it may reach a minimum.

B may converge

v. The zero-mean Gaussian noise may change the data orientation of dataset B, which may make dataset initially inseperable.

B may converge.

(d) SVM is not vulnerable to dataset like B. The mechanism is dedicated to this scenario that SVM is finding a hyperplane that maximizes the margin. SVM is trying to calculate all the smallest margins of all possible separating hyperplane, and choose the largest from them. When the dataset is already separable, we reach the optimal case for SVM.

Question 2

(a)

Size of dictionary: 1722

(b) Naive Bayes had an accuracy of 0.956989247311828 on the testing set

(c) The top 5 indicative words for Naive Bayes are: ['urgent!', 'tone', 'prize', 'won', 'claim']

(d) The optimal SVM radius was 0.1
The SVM model had an accuracy of 0.9695340501792115 on the testing set

Question 3

(a) valid kernel

since K_1, K_2 are kernel, by the mercer theorem, K_1, K_2 are symmetric. So, K is symmetric since it is the sum of two symmetric matrices.

we want to show $K = K_1 + K_2$ is a valid kernel.

by adding matrices component-wise

$$\forall \vec{x},\ \vec{X}^T K \vec{X} = \vec{X}^T (K_1 + K_2) \vec{X}$$
$$= \vec{X}^T K_1 \vec{X} + \vec{X}^T K_2 \vec{X} \geq 0$$

given that K_1 and K_2 are valid kernel with properties (1) positive semi-definite (2) symmetric.

(b) not a kernel.

Consider a kernel function $K: \mathbb{R}^n \to \mathbb{R}$

and $K_1(x, z) = 3$, $K_2(x, z) = 5$.

if $K = K_1 - K_2$ is a valid kernel, it should be positive semi-definite

However, $K = K_1 - K_2 = 3 - 5 = -2 < 0$

it is contrary to our assumption

$K = K_1 - K_2$ is not a valid kernel.

(c) K is a kernel.

Consider $a > 0$

we know K_1 is symmetric by the mercer theorem. So, $aK_1 = K$ is also symmetric.

for all non-zero vector $x \in \mathbb{R}^n$,

$$x^T K x = x^T (aK_1) x$$
$$= a\, x^T K_1 x \geq 0$$

$\therefore x^T K x \geq 0$

(a) K is not a Kernel

consider $a > 0$

if $K_1(x, z) = 9$, then $K = -a K_1 = -9a < 0$

K is not a valid kernel.

Hence, $K = -aK_1$ is not a valid kernel when $a > 0$.

(e) K is a Kernel

$$K(x, z) = K_1(x, z) \cdot K_2(x, z)$$
$$= \left(\sum_{i=1}^{p} \phi_1^{(i)}(x)\, \phi_1^{(i)}(z) \right) \cdot \left(\sum_{i=1}^{p} \phi_2^{(i)}(x)\, \phi_2^{(i)}(z) \right)$$
$$= \sum_{i=1}^{p} \sum_{j=1}^{p} \left(\phi_1^{(i)}(x)\, \phi_2^{(j)}(x) \right) \cdot \left(\phi_1^{(i)}(z)\, \phi_2^{(j)}(z) \right)$$
$$= \sum_{i=1}^{p} \sum_{j=1}^{p} \phi_{ij}(x)\, \phi_{ij}(z)$$
$$= \left\langle \begin{bmatrix} \phi_{11}(x) \\ \vdots \\ \phi_{pp}(x) \end{bmatrix}, \begin{bmatrix} \phi_{11}(z) \\ \vdots \\ \phi_{pp}(z) \end{bmatrix} \right\rangle$$ is a dot product.

K is a valid kernel.

(f) K is a kernel.

consider the function $f: \mathbb{R} \to \mathbb{R}$ as a one-component vector function.

Then $K(x, z) = f(x)\, f(z)$ is the same as

$= \langle f(x), f(z) \rangle$

So K is a valid kernel.

(g) K is a valid kernel

$K(x, z) = K_3(\phi(x), \phi(z)) = \langle \phi_3(\phi(x)), \phi_3(\phi(z)) \rangle$

is a dot product

(h) K is a kernel

we can write $p(x)$ as $p(x) = a_0 + a_1 x + \cdots + a_k x^k$

such that $a_0, \cdots a_k > 0$

Then $K(x, z) = p(K_1(x, z))$

$= a_0 + a_1(K_1(x, z)) + \cdots + a_k(K_1(x, z))^k$

using the answer from part a, c and e

each term $a_i(K_1(x, z))^i$ is a valid kernel

So, K is a valid kernel

Question 4

(a) The parameter vector $\vec{\theta}$ is initialized as

$$\vec{0} = \sum_{j=1}^{i} \beta_j^{(0)} \phi(x^{(j)})$$

where $\beta_j^{(0)} = 0$ for all $j = 1, \cdots, i$

for each i, we add some multiple of $\phi(x^{(i)})$ to the term θ, so θ is always a linear combination of the training sample.

That $$\theta = \sum_{i=1}^{n} \beta_j^{(0)} \phi(x^{(j)})$$

Consider the new input $x^{(i+1)}$, the prediction expression is:

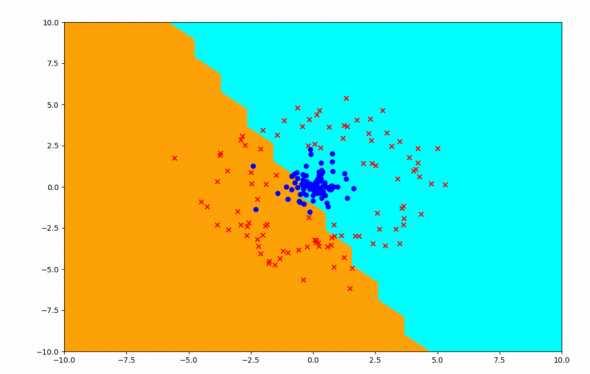
$$\begin{aligned} h_{\theta^{(i)}}(x^{(i+1)}) &= g(\theta^{(i)T} \phi(x^{(i+1)})) \\ &= \text{sign}(\theta^{(i)T} \phi(x^{(i+1)})) \\ &= \text{sign}\left(\sum_{j=1}^{i} \beta_j^{(0)} \phi(x^{(j)})^T \phi(x^{(i+1)})\right) \\ &= \text{sign}\left(\sum_{j=1}^{i} \beta_j^{(0)} K(\phi(x^{(j)}), \phi(x^{(i+1)}))\right) \end{aligned}$$

The update rule at the $(i+1)$ time

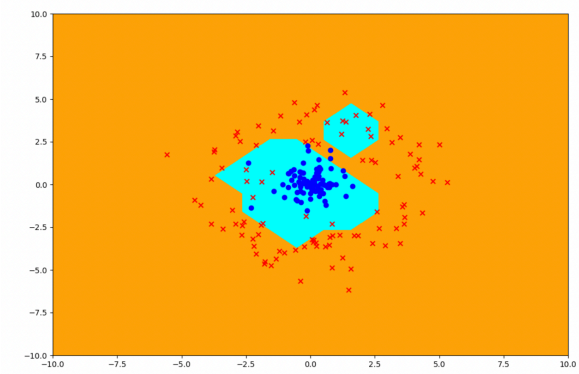
$$\begin{aligned} \theta^{(i+1)} &= \theta^{(i)} + \alpha(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)}))\phi(x^{(i+1)}) \\ &= \sum_{j=1}^{i} \beta_j^{(0)} \phi(x^{(j)}) + \alpha(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)}))\phi(x^{(i+1)}) \\ &= \sum_{j=1}^{i} \beta_j^{(0)} \phi(x^{(j)}) + \alpha\left(y^{(i+1)} - g\left(\sum_{j=1}^{i} \beta_j^{(0)} K(x^{(j)}, x^{(i+1)})\right)\right)\phi(x^{(i+1)}) \\ &= \sum_{j=1}^{i} \beta_j^{(0)} \phi(x^{(j)}) + \alpha\left(y^{(i+1)} - \text{sign}\left(\sum_{j=1}^{i} \beta_j^{(0)} K(x^{(j)}, x^{(i+1)})\right)\right)\phi(x^{(i+1)}) \end{aligned}$$

$$\beta_{i+1} = \alpha(y^{(i+1)} - h_{\theta^{(i)}}(x^{(i+1)})) = \alpha\left(y^{(i+1)} - \text{sign}\left(\sum_{j=1}^{i} \beta_j^{(0)} K(x^{(j)}, x^{(i+1)})\right)\right)$$

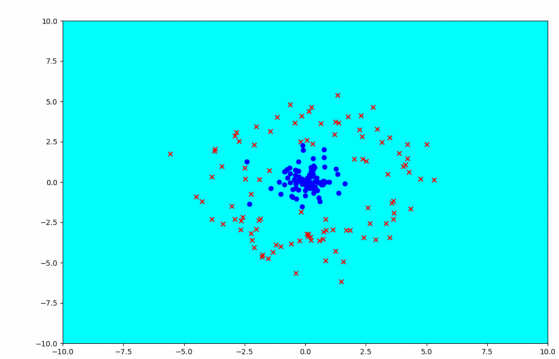
(b)



Dot product kernel



RBF kernel



not-a-psd kernel.

(C) The dot-product kernel can only classify data basd on a linear decision boundary, which cannot fit this non-linear dataset. The not-a-psd kernel does not answer the requirement of this questions, i.e. whether x = z is irrelevent to this classification problem, so this method fails. The RBF kernel has a really high-dimensional feature map so it is capable to learn the non-linear feature of this dataset better than the other methods.

Question 5

(a) For an input example $x^{(i)}$ of the class l such that l is the index $\in \{1, \dots, K\}$ where $y^{(i)} = [0, \dots, 1, \dots, 0]^T$ contains 1 at its l-th position.

we write the loss function:

$$CE(y^{(i)}, \hat{y}^{(i)}) = -\sum_{k=1}^{K} y_k^{(i)} \log \hat{y}_k^{(i)}$$

$$= -\log \hat{y}_l^{(i)}$$

$$= -\log \left(\frac{\exp(z_l^{(i)})}{\sum_{k=1}^{K} \exp(z_k^{(i)})} \right)$$

$$= \log \left(\sum_{k=1}^{K} \exp(z_k^{(i)}) \right) - z_l^{(i)}$$

Take $j = l$, there is

$$\frac{\partial CE(y^{(i)}, \hat{y}^{(i)})}{\partial z_j^{(i)}} = \frac{\exp(z_j^{(i)})}{\sum_{k=1}^{K} \exp(z_k^{(i)})} - 1 = \hat{y}_l^{(i)} - y_l^{(i)}$$

consider the case $j \neq l$, there is

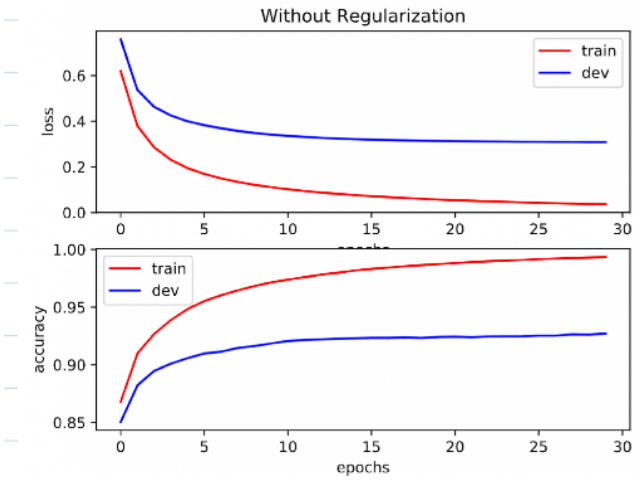
$$\frac{\partial CE(y^{(i)}, \hat{y}^{(i)})}{\partial z_j^{(i)}} = \frac{\exp(z_j^{(i)})}{\sum_{k=1}^{K} \exp(z_k^{(i)})} = \hat{y}_j^{(i)} - 0 = \hat{y}_j^{(i)} - y_j^{(i)}$$

[as $y_j^{(i)} = 0$]

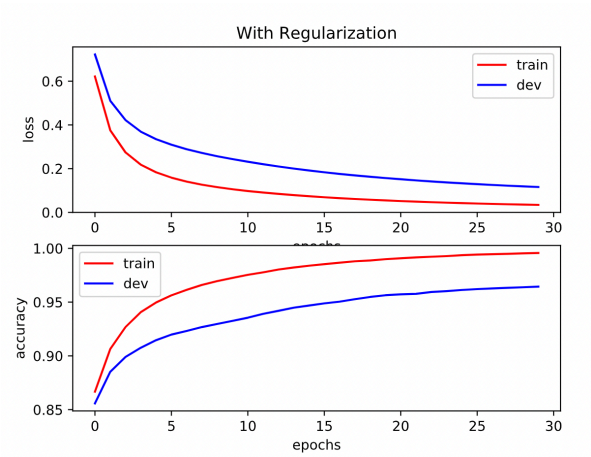
For cases above, we know that $\forall j$,

we show that $\nabla_{z^{(i)}} CE(y^{(i)}, \hat{y}^{(i)}) = \hat{y}^{(i)} - y^{(i)}$

(b)



(c)



Compare

without regularization, the loss for the dev set plateaus and does not decrease further, while the loss on the dev set with regularization continues to decrease, which is what we want to see. Similarly, without regularization, the accuracy also plateaus on the dev set, while with regularization, it continues to increase as the epochs reach 30.

(d) For model without regularization, the accuracy is 0.93200
For model with regularization, the accuracy is 0.96530
Because regularization avoids over-fitting by tuning the training model, it is sensible to see better performance in the regularization model..

Question 6.

(a) suppose $p(\theta) = p(\theta|x)$

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

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

$$= \frac{1}{p(y|x)} \arg\max_{\theta} p(y|x,\theta)\, p(\theta|x)$$

$$= \arg\max_{\theta} p(y|x,\theta)\, p(\theta)$$

(b) Given that $\theta_{map} = \arg\max_{\theta} p(y|x,\theta)\, p(\theta)$

$$\theta_{map} = \arg\max_{\theta} \log\left[p(y|x,\theta)\, p(\theta)\right]$$

$$= \arg\min_{\theta} -\log\left[p(y|x,\theta)\, p(\theta)\right]$$

$$= \arg\min_{\theta} -\log p(y|x,\theta) - \log\left(\frac{1}{(2\pi)^{\frac{k}{2}} \|\eta^2 I\|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\theta^T (\eta^2 I)^{-1}\theta\right)\right)$$

$$= \arg\min_{\theta} -\log p(y|x,\theta) + \left(\frac{1}{2}\theta^T (\eta^2 I)^{-1}\theta\right)$$

$$= \arg\min_{\theta} -\log p(y|x,\theta) + \frac{\|\theta\|_2^2}{2\eta^2}$$

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

(c) since $\vec{\varepsilon} \sim N(0, \sigma^2)$; $\vec{y} = X\theta + \vec{\varepsilon}$ we can infer that $\vec{y} \mid X \sim N(X\theta, \sigma^2 I)$

$$\theta_{map} = \arg\min_{\theta} -\log P(y|x,\theta) + \left(\frac{1}{2\sigma^2}\|\theta\|_2^2\right)$$

$$= \arg\min_{\theta} -\log\left(\frac{1}{(2\pi)^{\frac{n}{2}}\|\sigma^2 I\|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\vec{y}-X\theta)^T(\sigma^2 I)^{-1}(\vec{y}-X\theta) + \left(\frac{1}{2\sigma^2}\|\theta\|_2^2\right)\right)\right)$$

$$= \arg\min_{\theta} -\log\left(\frac{1}{(2\pi)^{\frac{n}{2}}\|\sigma^2 I\|^{\frac{1}{2}}}\right) + \left(-\frac{1}{2}(\vec{y}-X\theta)^T(\sigma^2 I)^{-1}(\vec{y}-X\theta) + \left(\frac{1}{2\sigma^2}\|\theta\|_2^2\right)\right)$$

$$= \arg\min_{\theta} \left(-\frac{1}{2}(\vec{y}-X\theta)^T(\sigma^2 I)^{-1}(\vec{y}-X\theta)\right) + \left(\frac{1}{2\sigma^2}\|\theta\|_2^2\right)$$

$$= \arg\min_{\theta} \frac{\|\vec{y}-X\theta\|_2^2}{2\sigma^2} + \frac{\|\theta\|_2^2}{2\eta^2}$$

Let $J(\theta) = \dfrac{\|X\theta - \vec{y}\|_2^2}{2\sigma^2} + \dfrac{\|\theta\|_2^2}{2\eta^2}$

$$= (X\theta - \vec{y})^T(X\theta - y) + \frac{\|\theta\|_2^2}{2\eta^2}$$

Calculate $\nabla_\theta J(\theta) = 2(X^TX\theta - 2X^T\vec{y}) + 2\dfrac{\sigma^2}{\eta^2}\theta$

$$= 2\left(X^TX\theta + \frac{\sigma^2}{\eta^2}I\right)\theta - 2X^T\vec{y} = 0.$$

$$\Rightarrow \theta_{map} = \left(X^TX + \frac{\sigma^2}{\eta^2}I\right)^{-1}X^T\vec{y}$$

(4) Suppose using the linear regression model $\vec{y} = \theta^T X + \vec{\varepsilon}$

that $\vec{\varepsilon} \sim N(0, \sigma^2)$ and $\theta_i \sim L(0, b)$

we can show that $\vec{y} \mid X, \theta$ is normally distributed.

$$f_{\vec{Y}|X}(y|x) = \frac{f_{\vec{Y}X}(y,x)}{f_X(x)}$$

$$f_{\vec{Y}X}(y,x) = f_{\vec{y}X}(\theta^T x + \varepsilon, x) = f_{\varepsilon X}(y - \theta^T x, x)$$

The independence of ε and X implies that

$$f_{\varepsilon X}(y - \theta^T x, x) = f_\varepsilon(y - \theta^T x) f_X(x)$$

Since $\varepsilon \sim N(0, \sigma^2)$

$$f_{\vec{Y}|X}(y|x) = f_\varepsilon(y - \theta^T x) = \frac{1}{2\sigma\sqrt{(\pi)}} \exp\left(\frac{1}{2}\left(\frac{y - \theta^T x}{\sigma}\right)^2\right)$$

This implies that $\vec{Y}|X \sim N(\theta^T X, \sigma^2)$

and $p(\theta_i) = \frac{1}{2b} \exp\left(-\frac{|\theta_i|}{b}\right)$

Deduce $\theta_{MAP} = \arg\min_\theta -\log(p(\theta)\, p(\vec{y}|X, \theta))$

$$= \arg\min_\theta -\log\left[\frac{1}{(2\pi)^{\frac{n}{2}} \|\sigma^2 I\|^{\frac{1}{2}}} \exp\left(\frac{\|\vec{y} - \theta^T X\|_2^2}{2\sigma^2}\right)\right]$$

$$- \sum_{i=1}^{d} \log\left(\frac{1}{2b} \exp\left(-\frac{|\theta_i|}{b}\right)\right)$$

$$= \arg\min_\theta \frac{\|\vec{y} - \theta^T X\|_2^2}{2\sigma^2} + \sum_{i=1}^{d} \frac{|\theta_i|}{b}$$

$$= \arg\min_\theta \|\vec{y} - \theta^T X\|_2^2 + \frac{2\sigma^2}{b} \|\theta\|_1$$

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