1. Logistic Regression: Training Stability.

(a) A converged quickly while B can not converge.

(b) Dataset B is linea seperable while A is not.

check the code in "cal_grad"
we can write out the loss fenction

$$\nabla_\theta J(\theta) = -\frac{1}{m}\sum_i^m \frac{y^{(i)}x^{(i)}}{1+e^{-y^{(i)}x^T\theta}} \qquad \theta := \theta - lr\cdot\nabla_\theta\theta.$$

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

The dataset B is linear separable, so we can update θ that satisfy $y^{(i)}x^T\theta > 0$ for all cases.
In order to minimize $J(\theta)$, we need parameter θ close to positive infinite.
$\|\nabla_\theta J(\theta)\|$ increases during this process, so $|\theta_{old} - \theta_{new}|$ will gets longer thus it can't converge.

How ever for unseparable cases, as θ increase, $J(\theta)$ for examples in the orther side of classification boundary increases as well. $\nabla J(\theta)$ decreases which leads $|\theta_{old} - \theta_{new}|$ converges within a small value.

(c) (i) No. learning rate is a fixed constant and can not affect $\nabla_\theta J(\theta)$.
(ii) Yes. There will be some iteration that $lr\cdot\nabla_\theta J(\theta) < 10^{-15}$
(iii) No. linear scale will not change the seperability of data,
(iv) Yes. This constrains θ can not be upclate to infinite.
(v.) Yes. This allows data be unseperable linearly.

(d) hinge loss: $J(\vec{y}) = \sum_m^n \max(0, 1-y\cdot\vec{y})$, $\vec{y} = \vec{w}x+b$
when dataset is linear separable, $y\vec{y} > 0$ for all cases.
we can increase parameter w to make $y\vec{y} > 1$.
thus $J(\vec{y})$ can be minimize to 0.

2. Model Caliberation.

(a) the log likelihood for logistic regression is

$$l(\theta) = \log\prod_{i=1}^m p(y|x;\theta) = \sum_{i=1}^m y_i \log p(y_i=1|x;\theta) + (1-y_i)\log p(y_i=0|x;\theta)$$
$$= \sum_{i=1}^m y_i \log h_\theta(x^{(i)}) + (1-y_i)\log(1-h_\theta(x^{(i)}))$$

$$\frac{\partial l(\theta)}{\partial\theta_j} = \sum_{i=1}^m y_i\frac{1}{h_\theta(x)}\cdot h_\theta'(x) + \frac{(1-y_i)}{1-h_\theta(x^{(i)})}(-h_\theta'(x))$$
$$= \sum_{i=1}^m y_i\frac{1}{h_\theta(x)}h_\theta(x)(1-h_\theta(x))x_j - \frac{(1-y_i)}{1-h_\theta(x^{(i)})}h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))x_j$$
$$= \sum_{i=1}^m y_i(1-h_\theta(x^{(i)})) - (1-y_i)h_\theta(x^{(i)}) = \sum_{i=1}^m (y_i - h_\theta(x^{(i)}))x_j.$$

set $j=0$. because of intercept term $x_0=1$.

$$\sum_{i=1}^{m} y_i = \sum_{i=1}^{m} h_\theta(x^{(i)}),$$

$y_i = \mathbb{1}\{y^{(i)}=1\}$. $h_\theta(x^{(i)}) = P(y^{(i)}=1 \mid x^{(i)};\theta)$. $|\{i \in [a,b]\}| = |\{x^{(i)}, y^{(i)}\}| = m$.

$$\frac{\sum_{i\in[a,b]} P(y^{(i)}=1 \mid x^{(i)};\theta)}{|\{i \in [a,b]\}|} = \frac{\sum_{i\in[a,b]} \mathbb{1}\{y^{(i)}=1\}}{|\{i \in [a,b]\}|} \text{ holds true.}$$

(b) As a binary classification problem,

$$0.5 < p(y^{(i)}=1 \mid x;\theta) < 1.$$

so, for $I \in I_{(0.5,1)}$ when model is perfectly calibrated.

$$\frac{\sum_{i\in[0.5,1]} p(y^{(i)}=1 \mid x^{(i)};\theta)}{|\{i \in [0.5,1]\}|} = \frac{\sum_{i\in[0.5,1]} \mathbb{1}\{y^{(i)}=1\}}{|\{i \in [0.5,1]\}|} < \frac{\sum_{i\in[0.5,1]} \mathbb{1}\{y^{(i)}=1\}}{|\{i \in [0,1]\}|} = \text{precision}.$$

from above we can see the model doesn't achieve it's perfect accuracy. The converse is not necessarily true as well.

(c) Applying L_2 regularization to the loss function

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

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

set $j=0$, $x_j=1$.

$$\sum_{i}^{m} y^{(i)} = \sum_{i=1}^{m} h_\theta(x^{(i)}) - \lambda\theta_0.$$ so the model will not be well-calibrated, it can achieve the best precision.

3. Bayesian Interpretation of Regularization

MAP: maximum a posteriori estimation $\quad \theta_{MAP} = \arg\max_\theta p(\theta \mid x, y)$

MLE: maximum likelihood estimation $\quad \theta_{MLE} = \arg\max_\theta p(y \mid x, \theta)$

(a) $$p(\theta \mid x, y) = \frac{p(x, y, \theta)}{p(x, y)} = \frac{p(y \mid x, \theta)\, p(x, \theta)}{p(x, y)} = \frac{p(y \mid x, \theta)\, p(\theta \mid x)\, p(x)}{p(x, y)}$$

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

Assume that $p(\theta \mid x) = p(\theta)$

$$\therefore \arg\max_\theta p(\theta \mid x, y) = \arg\max_\theta \frac{p(y \mid x, \theta)\, p(\theta)\, p(x)}{p(x, y)} = \arg\max_\theta p(y \mid x, \theta) p(\theta).$$

(b) negative log likelihood loss function is that:

$$J(\theta) = -\log p(y \mid x, \theta) + \lambda\|\theta_2\|^2$$

MLE estimation is

$$\theta_{MLE} = \arg\min_\theta -\log p(y \mid x, \theta) + \lambda\|\theta_2\|^2.$$

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

$$= \arg\max_\theta \log(p(y \mid x, \theta) p(\theta))$$

Remark:

Different regularization stands for different assumptions for θ.

$$= \arg\max_\theta \log p(y|x,\theta) + \log p(\theta)$$
$$= \arg\min_\theta -\log p(y|x,\theta) - \log p(\theta).$$

$\theta \sim \mathcal{N}(0, \eta^2 I)$

$$p(\theta) = \frac{1}{(2\pi)^{n/2}\eta^n} \exp\left\{-\frac{1}{2\eta^2}\theta^\top\theta\right\} = \frac{1}{(2\pi)^{\frac{n}{2}}\eta^n} \exp\left\{-\frac{1}{2\eta^2}\|\theta\|_2^2\right\}$$

$$\log p(\theta) = -\frac{1}{2\eta^2}\|\theta\|_2^2 - \frac{n}{2}\log 2\pi - n\log\eta.$$

$$\therefore \theta_{MAP} = \arg\min_\theta \frac{1}{2\eta^2}\|\theta\|_2^2 + \frac{n}{2}\log 2\pi + n\log\eta - \log p(y|x,\theta)$$

$$= \arg\min_\theta \frac{1}{2\eta^2}\|\theta\|_2^2 - \log p(y|x,\theta)$$

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

(c) For a specific instance, $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$y^{(i)} = \theta^\top x^{(i)} + \epsilon$

$y^{(i)} \sim \mathcal{N}(\theta^\top x^{(i)}, \sigma^2)$

$$\theta_{MAP} = \arg\min_\theta \frac{1}{2\eta^2}\|\theta\|_2^2 - \log p(y|x,\theta)$$
$$= \arg\min_\theta \frac{1}{2\eta^2}\|\theta\|_2^2 - \sum_{i=1}^m \log p(y^{(i)}|x^{(i)},\theta)$$
$$= \arg\min_\theta \frac{1}{2\eta^2}\|\theta\|_2^2 - \sum_{i=1}^m \log \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(y^{(i)} - \theta^\top x^{(i)})^2\right\}$$
$$= \arg\min_\theta \frac{1}{2\eta^2}\|\theta\|_2^2 + \sum_{i=1}^m \frac{1}{2\sigma^2}(y^{(i)} - \theta^\top x^{(i)})^2$$
$$= \arg\min_\theta \frac{1}{2\sigma^2}(\vec{y} - X\theta)^\top(\vec{y} - X\theta) + \frac{1}{2\eta^2}\theta^\top\theta$$

$$J(\theta) = \frac{1}{2\sigma^2}(\vec{y} - X\theta)^\top(\vec{y} - X\theta) + \frac{1}{2\eta^2}\theta^\top\theta.$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{\sigma^2}\sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})(-x_j^{(i)}) + \frac{1}{\eta^2}\theta.$$

$$\nabla_\theta J(\theta) = \frac{1}{\sigma^2} X^\top(X\theta - \vec{y}) + \frac{1}{\eta^2}\theta.$$

Set $\nabla_\theta J(\theta) = 0$, we have $\theta = \left(X^\top X + \frac{\sigma^2}{\eta^2} I\right)^{-1} X^\top \vec{y}$

(d) $\theta \sim \mathcal{L}(0, b)$

$$p(\theta) = \frac{1}{2b}\exp\left(-\frac{|\theta|}{b}\right) \qquad y|x,\theta \sim \mathcal{N}(\theta^\top x, \sigma^2)$$

$$\theta_{MAP} = \arg\max_\theta p(\theta|x,y) = \arg\max_\theta p(y|x,\theta)p(\theta)$$
$$= \arg\max_\theta \log p(y|x,\theta) + \log p(\theta)$$
$$= \arg\max_\theta \sum_{i=1}^m \log \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(y^{(i)} - \theta^\top x^{(i)})^2\right\} + \log\frac{1}{2b}\exp\left\{-\frac{|\theta|}{b}\right\}$$
$$= \arg\max_\theta -\sum_{i=1}^m \frac{1}{2\sigma^2}(y^{(i)} - \theta^\top x^{(i)})^2 - \frac{|\theta|}{b}$$
$$= \arg\min_\theta \sum_{i=1}^m \frac{1}{2\sigma^2}(y^{(i)} - \theta^\top x^{(i)})^2 + \frac{|\theta|}{b}$$
$$= \arg\min_\theta \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 + \frac{2\sigma^2}{b}|\theta|$$
$$= \arg\min_\theta J(\theta). \qquad \gamma = \frac{2\sigma^2}{b}$$

4. Constructing Kernels.

(a) Yes.
K_1, K_2 are symmetric. So K_1+K_2 is symmetric as well.
K_1, K_2 are PSD.
$z^T K z = z^T (K_1+K_2) z = z^T K_1 z + z^T K_2 z \geq 0$. So K is PSD as well.

(b) No.
counter example, $K_2 = -2K_1$
$z^T K z = z^T (K_1 - 2K_1) z = -z^T K_1 z$, NSD.

(c) Yes. α is a positive real number.
$z^T K z = \alpha z^T K_1 z \geq 0$.

(d) No.
$z^T K z = -\alpha z^T K_1 z \leq 0$. NSD.

(e) Yes.
$$z^T K z = \sum_i \sum_j z_i K_{ij} z_j$$
$$= \sum_i \sum_j z_i K_1(x^{(i)}, x^{(j)}) K_2(x^{(i)}, x^{(j)}) z_j$$
$$= \sum_i \sum_j z_i \phi_1(x^{(i)})^T \phi_1(x^{(j)}) \phi_2(x^{(i)})^T \phi_2(x^{(j)}) z_j$$
$$= \sum_i \sum_j z_i \left(\sum_a \phi_{1a}(x^{(i)}) \phi_{1a}(x^{(j)})\right)\left(\sum_b \phi_{2b}(x^{(i)}) \phi_{2b}(x^{(j)})\right) z_j$$
$$= \sum_a \sum_b \sum_i \left(\phi_{1a}(x^{(i)}) \phi_{2b}(x^{(i)}) z_i\right)^2 \geq 0$$

(f) Yes
$$z^T K z = \sum_i \sum_j z_i K_{ij} z_j$$
$$= \sum_i \sum_j z_i f(x^{(i)}) f(x^{(j)}) z_j$$
$$= \left(\sum_i z_i f(x^{(i)})\right)^2 \geq 0.$$

(g) Yes
$$z^T K z = \sum_i \sum_j z_i K_{ij} z_j$$
$$= \sum_i \sum_j z_i K_3(\phi(x^{(i)}), \phi(x^{(j)})) z_j$$
$$= \sum_i \sum_j z_i \phi_3(\phi(x^{(i)}))^T \phi_3(\phi(x^{(j)})) z_j$$
$$= \sum_i \sum_j z_i \sum_a \phi_{3a}(\phi(x^{(i)})) \phi_{3a}(\phi(x^{(j)})) z_j$$
$$= \sum_a \sum_i \left(\phi_{3a}(\phi(x^{(i)})) z_i\right)^2 \geq 0.$$

(h) Yes.
$p(x) = a_0 + a_1 x + \cdots + a_n x^n$, $a_1 \cdots a_n \in \mathbb{R}^+$
$K(x,z) = p(K_1(x,z)) = a_0 + a_1 K_1(x,z) + a_2 K_2(x,z)^2 + \cdots + a_n K_n(x,z)^n$.
As I proved before, ① $K(x,z) = K_1(x,z) K_2(x,z)$ is a valid kernel.
generalize this conclusion to multinomial case:
$K(x,z) = (K_1(x,z))^n$ is a valid kernel.

② $k(x,z) = \alpha k_1(x,z)$. $\alpha \in \mathbb{R}^+$ is a valid kernel.

③ $k(x,z) = k_1(x,z) + k_2(x,z)$ is a valid kernel

So $k(x,z) = p(k_1(x,z))$ is a valid kernel.

5. kernelizing the Perceptron.

(a) Apply kernel trick to perceptron to make it work in high dimensional space (eg. ∞)

(i) Recall the update rule:

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

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

(ii) $h_{\theta^{(i)}}(x^{(i+1)}) = g(\theta^{(i)T}\phi(x^{(i+1)}))$

$$= g\left(\sum_{j=0}^{i} \beta_j \phi^T(x^{(j)})\phi(x^{(i+1)})\right)$$

$$= \text{sign}\left(\sum_{j=0}^{i} \beta_j \langle \phi(x^{(j)}), \phi(x^{(i+1)})\rangle\right)$$

$$= \text{sign}\left(\sum_{j=0}^{i} \beta_j K(x^{(j)}, \phi(x^{(i+1)}))\right)$$

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

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

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

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

(c) Dot kernel performs badly. For the dot kernel, $\phi(x) = x$. So it doesn't map the data to higher dimension, the data is still not linear seperable.

6. Spam Classification