CS 229, Fall 2018

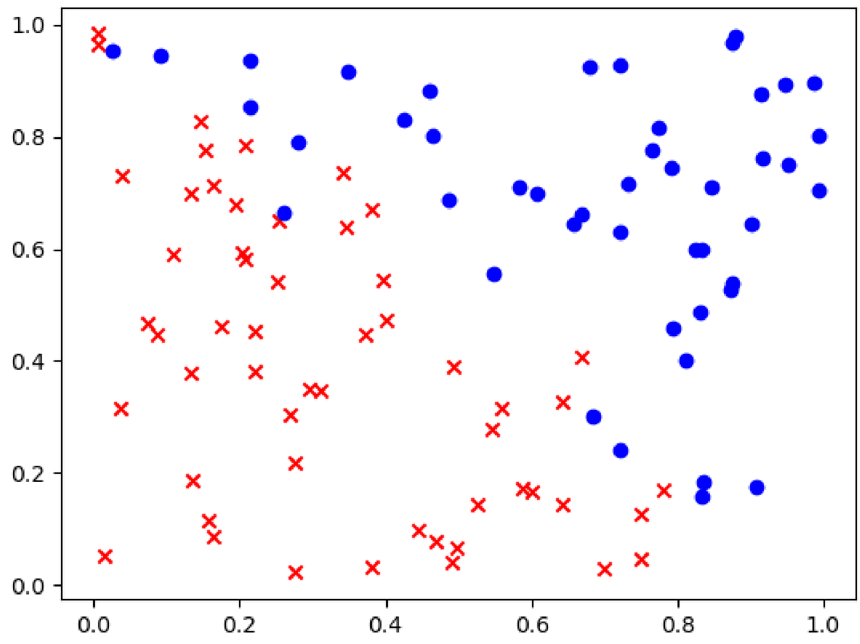
Problem Set #2 Solutions: Supervised Learning II

1.

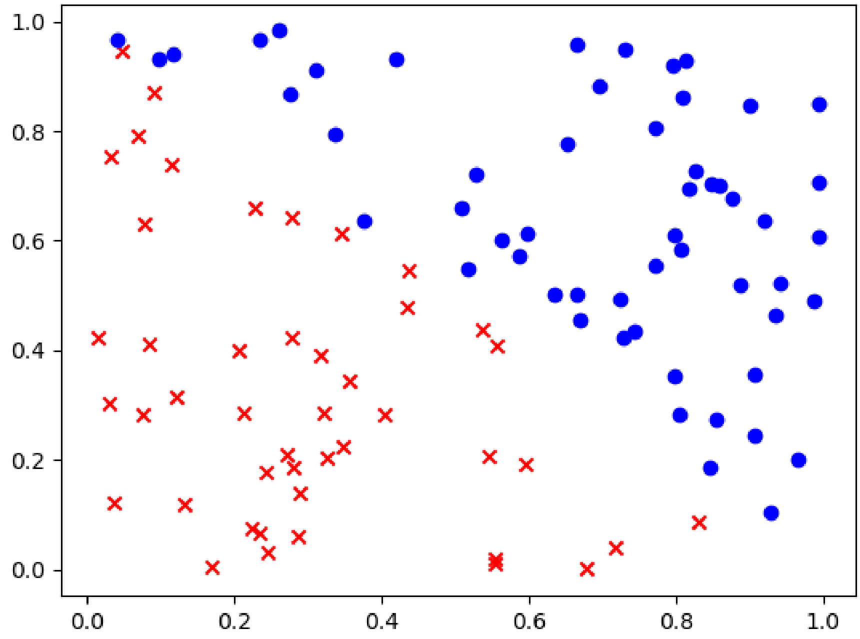
(a)

Logistic regression converged on dataset A, but didn't converge on dataset B.

(b)



Dataset A is not linearly separable



Dataset B is linearly separable

Recall that in SVM the functional margin $\hat{\gamma}$ is

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

Because there is no constraint on w (such as $||w||_2 = 1$), we can scale the w and b to increase the functional margin without changing the decision boundary.

In this problem, the labels y are {-1, +1} instead of {0, 1}. So the loss function $J(\theta)$ is

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

Notice there is $y^{(i)} \theta^T x^{(i)}$ in the expression above, this has a similar property like the functional margin.

When the dataset is linearly separable, $y^{(i)} \theta^T x^{(i)} > 0$ for all training examples. So we can scale θ to make $J(\theta)$ smaller (close to 0). However, when the dataset is not linearly separable, $y^{(i)} \theta^T x^{(i)}$ could be greater or smaller than 0. So we can't arbitrarily scale θ to reduce $J(\theta)$.

(c)

i.

No, using a different learning rate will not help to reduce the value of θ.

ii.

Yes, using learning rate decay (e.g. by factor $1/t^2$) will make $\alpha \nabla_\theta J(\theta) \leq 10^{-15}$ in a few iterations.

iii.

No, as you can see in the pictures the input feature are already scaled.

iv.

Yes, adding L_2 regularization will help reduce the value of θ.

v.

Yes, adding noise could make the dataset becomes not linearly separable.

But how to control the scale of noise to avoid losing accuracy?

(d)

SVM use hinge loss is not vulnerable to linearly separable dataset like B.

Here is the hinge loss

$$J(\hat{y}) = \max(0, 1 - y \cdot \hat{y}), \text{ where } \hat{y} = w^T x + b$$

Assume that the dataset is linearly separable, so $y \cdot \hat{y} > 0$.

When we increase w and b to make $|\hat{y}| \geq 1$, then $J(\hat{y}) = 0$.

2.

(a)

The log likelihood is

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

After training, the gradients are equal to 0

$$\frac{\partial \ell(\theta)}{\partial \theta_j} = \sum_{i=1}^{m} (y^{(i)} - h(x^{(i)})) x_j^{(i)} = 0$$

Set $j = 0$. Because $x_0^{(i)} = 1$, so

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

$$\sum_{i=1}^{m} h(x^{(i)}) = \sum_{i=1}^{m} y^{(i)}$$

$$h(x^{(i)}) = P(y^{(i)} = 1 | x^{(i)}; \theta), \ y^{(i)} = \mathbb{I}\{y^{(i)} = 1\}$$

$$\sum_{i=1}^{m} P(y^{(i)} = 1 | x^{(i)}; \theta) = \sum_{i=1}^{m} \mathbb{I}\{y^{(i)} = 1\}$$

When $(a, b) = (0, 1)$, $I_{a,b} = \{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ and $|\{i \in I_{a,b}\}| = m$

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

(b)

The model is perfectly calibrated doesn't necessarily imply that the model achieves perfect accuracy.

The converse is also not necessarily true.

Assume that $(a, b) = (0.5, 1)$.

When the model achieves perfect accuracy, the predictions are all correct, i.e.

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

For all $i \in I_{a,b}$

$$0.5 < P(y^{(i)} = 1|x^{(i)}; \theta) < 1$$

So

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

However, when the model is perfectly calibrated, the following property always hold

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

So model is perfectly calibrated doesn't mean model achieves perfect accuracy. The converse neither.

(c)

When adding L_2 regularization, θ is not the maximum likelihood parameter learned after training.

Furthermore, the loss function is

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

After training, the gradients are equal to 0

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

Set $j = 0$. Because $x_0^{(i)} = 1$, so

$$\sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) + \lambda\theta_0 = 0$$

$$\sum_{i=1}^{m} h(x^{(i)}) + \lambda\theta_0 = \sum_{i=1}^{m} y^{(i)}$$

$$\sum_{i=1}^{m} P(y^{(i)} = 1|x^{(i)}; \theta) + \lambda\theta_0 = \sum_{i=1}^{m} \mathbb{I}\{y^{(i)} = 1\}$$

So the model will not be well-calibrated.

3.

(a)

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

Assume that $p(\theta) = p(\theta|x)$, then

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

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

(b)

$$\begin{aligned}\theta_{\text{MAP}} &= \arg\max_\theta p(y|x,\theta)p(\theta) \\ &= \arg\max_\theta \log\Big(p(y|x,\theta)p(\theta)\Big) \\ &= \arg\max_\theta \log p(y|x,\theta) + \log p(\theta) \\ &= \arg\min_\theta -\log p(y|x,\theta) - \log p(\theta)\end{aligned}$$

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

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

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

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

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

(c)

$$\epsilon^{(i)} \sim \mathcal{N}(0,\sigma^2)$$

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

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

$$p(y^{(i)}|x^{(i)},\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{1}{2\sigma^2}(y^{(i)} - \theta^T x^{(i)})^2\}$$

$$\begin{aligned}p(\vec{y}|X,\theta) &= \prod_{i=1}^m p(y^{(i)}|x^{(i)},\theta) \\ &= \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{1}{2\sigma^2}(y^{(i)} - \theta^T x^{(i)})^2\} \\ &= \frac{1}{(2\pi)^{m/2}\sigma^m}\exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^m (y^{(i)} - \theta^T x^{(i)})^2\} \\ &= \frac{1}{(2\pi)^{m/2}\sigma^m}\exp\{-\frac{1}{2\sigma^2}||X\theta - \vec{y}||_2^2\}\end{aligned}$$

$$\log p(\vec{y}|X,\theta) = -\frac{m}{2}\log(2\pi) - m\log\sigma - \frac{1}{2\sigma^2}||X\theta - \vec{y}||_2^2$$

$$\begin{aligned}\theta_{\text{MAP}} &= \arg\min_\theta - \log p(y|x,\theta) + \frac{1}{2\eta^2}||\theta||_2^2 \\ &= \arg\min_\theta \frac{1}{2\sigma^2}||X\theta - \vec{y}||_2^2 + \frac{1}{2\eta^2}||\theta||_2^2\end{aligned}$$

$$J(\theta) = \frac{1}{2\sigma^2}||X\theta - \vec{y}||_2^2 + \frac{1}{2\eta^2}||\theta||_2^2$$

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

$$\theta_{\text{MAP}} = \arg\min_\theta J(\theta) = (X^TX + \frac{\sigma^2}{\eta^2}I)^{-1}X^T\vec{y}$$

(d)

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

$$p(\theta) = \frac{1}{(2b)^n}\exp\{-\frac{1}{b}||\theta||_1\}$$

$$\log p(\theta) = -n\log(2b) - \frac{1}{b}||\theta||_1$$

$$\begin{aligned}\theta_{\text{MAP}} &= \arg\min_\theta \frac{1}{2\sigma^2}||X\theta - \vec{y}||_2^2 - \log p(\theta) \\ &= \arg\min_\theta \frac{1}{2\sigma^2}||X\theta - \vec{y}||_2^2 + \frac{1}{b}||\theta||_1\end{aligned}$$

$$J(\theta) = ||X\theta - \vec{y}||_2^2 + \gamma||\theta||_1$$

$$\theta_{\text{MAP}} = \arg\min_\theta J(\theta)$$

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

4.

(a)

Yes, K_1 and K_2 are both PSD, so $K_1 + K_2$ is PSD.

$$z^TKz = z^T(K_1 + K_2)z = z^TK_1z + z^TK_2z \geq 0$$

(b)

No, although K_1 and K_2 are both PSD, $K_1 - K_2$ may not be PSD.

For example, $K_2 = 2K_1$

$$z^TKz = z^T(K_1 - K_2)z = z^T(K_1 - 2K_1)z = -z^TK_1z \leq 0$$

(c)

Yes, K_1 is PSD, so $aK_1(a \in \mathbb{R}^+)$ is PSD.

$$z^T K z = z^T a K_1 z = a \cdot z^T K_1 z \geq 0$$

(d)

No, K_1 is PSD, so $-aK_1(a \in \mathbb{R}^+)$ is not PSD.

$$z^T K z = z^T (-aK_1) z = -a \cdot z^T K_1 z \leq 0$$

(e)

Yes, $K_1 K_2$ is PSD.

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

(f)

Yes, K is PSD.

$f : \mathbb{R}^n \mapsto \mathbb{R}$ is a real-valued function, then

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

(g)

Yes, $K_3(\phi(x), \phi(z))$ is a valid kernel, no matter what the inputs are.

(h)

Yes, $p(K_1)$ is a valid kernel.

$p(x)$ is a polynomial function with coefficients $c_k > 0$, $k = 0, 1, \ldots, n$

$$p(x) = \sum_{k=0}^{n} c_k x^k$$

$$K(x, z) = p(K_1(x, z)) = \sum_{k=0}^{n} c_k \left(K_1(x, z) \right)^k$$

From (e) we know $K(x,z)=K_1(x,z)K_2(x,z)$ is a valid kernel, so $K(x,z)=\left(K_1(x,z)\right)^k$ is valid.

From (a) and (c), we know $K(x,z)=K_1(x,z)+K_2(x,z)$ and $K(x,z)=aK_1(x,z), a\in\mathbb{R}^+$ are both valid.

So $K(x,z)=\sum_{k=0}^{n}c_k\left(K_1(x,z)\right)^k$ is a valid kernel.

5.

(a)

i.

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

ii.

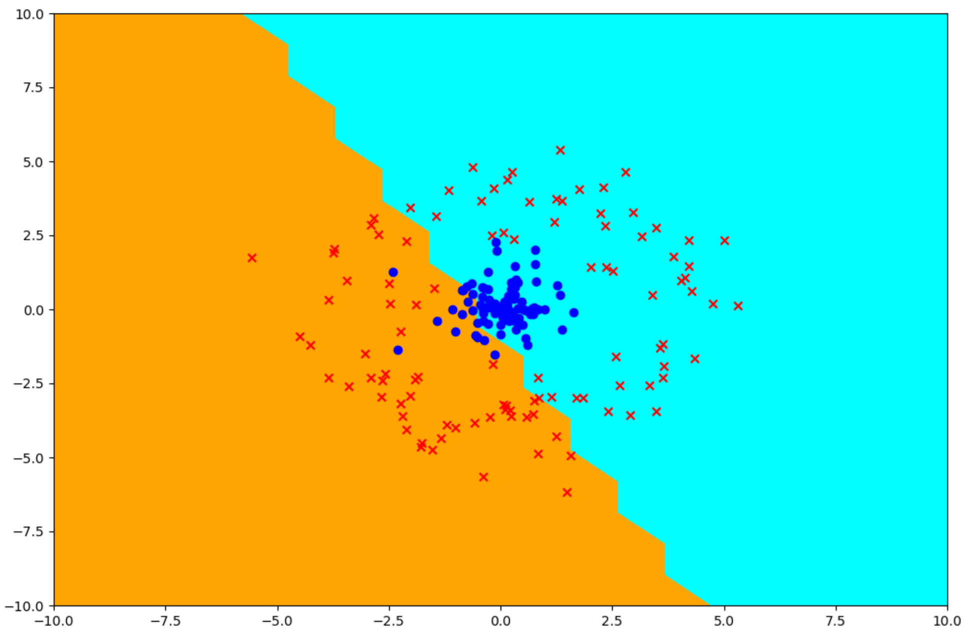
$$\begin{aligned}
h_{\theta^{(i)}}\left(\phi(x^{(i+1)})\right)&=g\left({\theta^{(i)}}^T\phi(x^{(i+1)})\right)\\
&=\operatorname{sign}\left({\theta^{(i)}}^T\phi(x^{(i+1)})\right)\\
&=\operatorname{sign}\left(\sum_{j=1}^{i}\beta_j\phi(x^{(j)})^T\phi(x^{(i+1)})\right)\\
&=\operatorname{sign}\left(\sum_{j=1}^{i}\beta_j\left\langle\phi(x^{(j)}),\phi(x^{(i+1)})\right\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

iii.

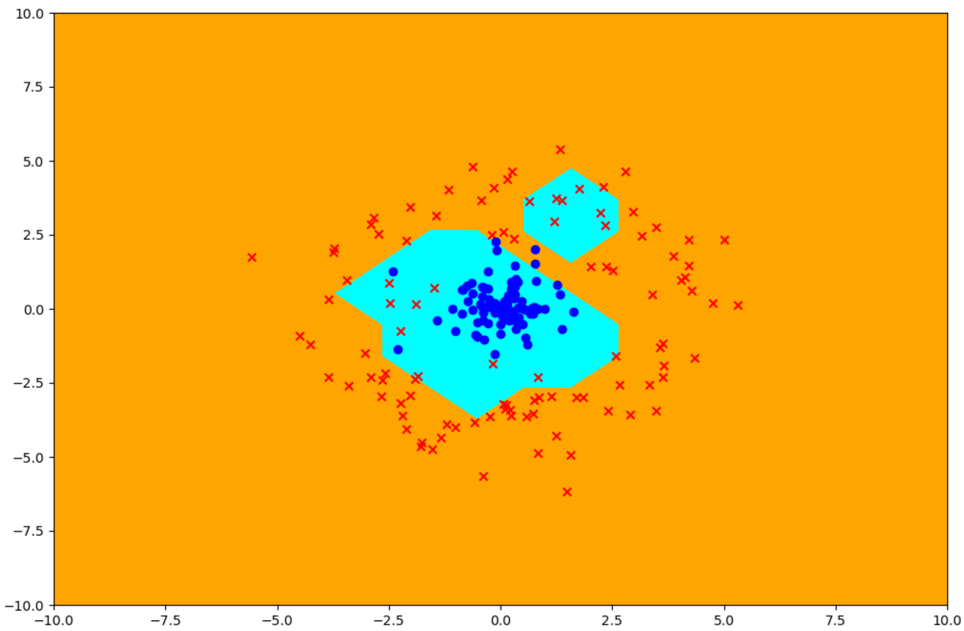
$$\begin{aligned}
\theta^{(i+1)}:&=\theta^{(i)}+\alpha\left(y^{(i+1)}-h_{\theta^{(i)}}\left(\phi(x^{(i+1)})\right)\right)\phi(x^{(i+1)})\\
&=\sum_{j=1}^{i}\beta_j\phi(x^{(j)})+\alpha\left(y^{(i+1)}-\operatorname{sign}\left(\sum_{j=1}^{i}\beta_jK\left(x^{(j)},x^{(i+1)}\right)\right)\right)\phi(x^{(i+1)})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

$$\beta_{i+1}=\alpha\left(y^{(i+1)}-\operatorname{sign}\left(\sum_{j=1}^{i}\beta_jK\left(x^{(j)},x^{(i+1)}\right)\right)\right)$$

(c)



dot kernel



rbf kernel

Dot kernel performs poorly than rbf kernel. Because dot kernel doesn't do feature mapping $\phi(x) = x$, and the dataset is not linearly separable.

6.

(b)

$$p(y=1|x) = \frac{\prod_{j=1}^{d} p(x_j|y=1)p(y=1)}{\prod_{j=1}^{d} p(x_j|y=1)p(y=1) + \prod_{j=1}^{d} p(x_j|y=0)p(y=0)}$$
$$= \frac{1}{1 + \frac{\prod_{j=1}^{d} p(x_j|y=0)p(y=0)}{\prod_{j=1}^{d} p(x_j|y=1)p(y=1)}}$$

$$p(y=1|x) > 0.5$$
$$\Updownarrow$$
$$\prod_{j=1}^{d} p(x_j|y=1)p(y=1) > \prod_{j=1}^{d} p(x_j|y=0)p(y=0)$$
$$\Updownarrow$$
$$\log\left(\prod_{j=1}^{d} p(x_j|y=1)p(y=1)\right) > \log\left(\prod_{j=1}^{d} p(x_j|y=0)p(y=0)\right)$$
$$\Updownarrow$$
$$\sum_{j=1}^{d} \log p(x_j|y=1) + \log p(y=1) > \sum_{j=1}^{d} \log p(x_j|y=0) + \log p(y=0)$$

(c)

5 most indicative tokens: 'claim', 'won', 'prize', 'tone', 'urgent!'

(d)

optimal radius: 0.1