آرین شکر - ۹۴۰۰۲۳۴۹۴ تکلیف پنجم شناسایی آماری الگو

3.1. perceptron cost function: $J(w) = \sum_{x \in Y} \delta_x w^T x$

where Y is the set of all misclassified input vectors. We know that $J(w)$ is piecewise linear because for values of w where the set Y is not affected we have:

$$J(w) = \sum_{x \in Y} \delta_x w^T x = w^T \overbrace{\Big(\sum_{x \in Y} \delta_x x\Big)}^{a} = w^T a = a^T w$$

linear

both a and w are vectors

But if we change the value of w smoothly at a certain threshold (w_t), the set Y of misclassified examples changes. New examples may enter the set while old examples may be exited. Let's call this new set Y' and $Y \cap Y' = A$ and $Y' - Y = B$ and $Y - Y' = C$. we have:

$$J_Y(w) = \sum_{x \in A} \delta_x w^T x + \sum_{x \in C} \delta_x w^T x$$

$$J_{Y'}(w) = \sum_{x \in A} \delta_x w^T x + \sum_{x \in B} \delta_x w^T x$$

Here's what we need to note here: when we approach w_t the original set Y is about to be altered into Y'. However, for every example that enters or leaves Y (i.e. sets B and C), the value of $w^T x$ must pass through zero because the sign of $w^T x$ for those examples must be flipped in order for them to enter or leave set Y. This change in sign occurs exactly at w_t therefore we have: (assumming we're increasing the value of w)

$$\lim_{w \to w_t^-} J(w) = \lim_{w \to w_t^-} J_Y(w) = \lim_{w \to w_t^-} \Big[\sum_{x \in A} \delta_x w^T x + \sum_{x \in C} \delta_x w^T x\Big]$$

$$= \sum_{x \in A} \delta_x w_t^T x + \cancelto{0}{\lim_{w \to w_t^-} \delta_x w^T x} = \lim_{w \to w_t^+} J(w)$$

with the same exact reasoning

$\Rightarrow J(w)$ is continuous

3.4. Reward and Punishment Perceptron Algorithm:

$$\begin{cases} w_{t+1} = w_t + \rho x_t, & \text{if } x_t \in \omega_1 \text{ and } w_t^T x_t \leq 0 \\ w_{t+1} = w_t - \rho x_t, & \text{if } x_t \in \omega_2 \text{ and } w_t^T x_t \geq 0 \\ w_{t+1} = w_t, & \text{o.w.} \end{cases}$$

The problem is not linearily separable without a bias so we will augment the input and weight vectors:

$$\omega_1 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right\}, \quad \omega_2 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}, \quad w_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Running the algorithm:

1. $w_0^T x_0 = 0, \ x_0 \in \omega_1 \rightarrow w_1 = w_0 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

2. $w_1^T x_1 = 1, \ x_1 \in \omega_1 \rightarrow w_2 = w_1$

3. $w_2^T x_2 = 1, \ x_2 \in \omega_2 \rightarrow w_3 = w_2 - \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$

4. $w_3^T x_3 = -1, \ x_3 \in \omega_2 \rightarrow w_4 = w_3$

5. $w_4^T x_4 = 0, \ x_4 \in \omega_1 \rightarrow w_5 = w_4 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

6. $w_5^T x_5 = 1, \ x_5 \in \omega_1 \rightarrow w_6 = w_5$

7. $w_6^T x_6 = 0, \ x_6 \in \omega_2 \rightarrow w_7 = w_6$

8. $w_7^T x_7 = 0, \ x_7 \in \omega_2 \rightarrow w_8 = w_7$

9. $w_8^T x_8 = 1, \ x_8 \in \omega_1 \rightarrow w_9 = w_8$

$\rightarrow$ all patterns classified correctly

$$\Rightarrow w_{final} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

3.5. Refer to the MATLAB script.

3.8. $J_{SSE} = \sum_{(x,y)\in S} (y - w^T x)^2$, $J_{MSE} = \frac{1}{|S|} \sum_{(x,y)\in S} (y - w^T x)^2$

$$\frac{\partial J_{SSE}(w)}{\partial w} = -2 \sum_{(x,y)\in S} x(y - x^T \hat{w}) = 0 \Rightarrow \sum_{(x,y)\in S} x(y - x^T \hat{w}) = 0 \quad \text{I}$$

$$\frac{\partial J_{MSE}(w)}{\partial w} = \frac{-2}{|S|} \sum_{(x,y)\in S} x(y - x^T \hat{w}) = 0 \Rightarrow \sum_{(x,y)\in S} x(y - x^T \hat{w}) = 0 \quad \text{II}$$

Equations I and II are the same, which means that the optimal $\hat{w}$ obtained from either J_{SSE} or J_{MSE} will be similar.

3.10. If N is the number of samples and M is the number of classes, we can form matrices:

$$y = [y_1, y_2, \dots, y_m]^T \text{ and } W = [w_1, w_2, \dots, w_M]^T$$

The minimal J_{SSE} yields the optimal $\hat{w}$:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} \| y - w^T x \|^2 = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{M} (y_j^{(i)} - w_j^T x_j^{(i)})^2$$

However the summations in the last term are interchangable because their indices are independent. Therefore we have:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \sum_{j=1}^{M} \sum_{i=1}^{N} (y_j^{(i)} - w_j^T x_j^{(i)})^2$$

This means that instead of minimizing $\sum_{j=1}^{M} (y_j^{(i)} - w_j^T x_j^{(i)})^2$ for every sample $(x^{(i)}, y^{(i)})$ we can instead fix a class w_j and minimize the cost w.r.t. class over all input samples.

$\Rightarrow$ Instead of solving for w over N we can solve for w_j for every $j = 1, \dots, M$ over N

3.11. If x and y are jointly Gaussian the probability distribution is given by:

$$P(x,y) = \frac{1}{2\pi \sigma_x \sigma_y \sqrt{1-\alpha^2}} \exp\left\{ -\frac{1}{2(1-\alpha^2)} \left[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - 2\alpha \frac{(x-\mu_x)(x-\mu_y)}{\sigma_x \sigma_y} \right] \right\}$$

Additionally if x and y are jointly Gaussian the are also independently Gaussian by definition because:

$$x, y \text{ jointly gaussian} \longrightarrow ax+by \sim N(\mu, \sigma^2)$$

Setting $a=1$ and $b=0 \longrightarrow x \sim N(\mu_x, \sigma_x^2)$

Setting $b=1$ and $a=0 \longrightarrow y \sim N(\mu_y, \sigma_y^2)$

Also, using the probablity chain rule we know:

$$P(x,y) = P(x) \cdot P(y|x)$$

$$\Rightarrow P(y|x) = \frac{P(x,y)}{P(x)}, \text{ where } P(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left\{ -\frac{(x-\mu_x)^2}{2\sigma_x^2} \right\}$$

$$\Rightarrow P(y|x) = \underbrace{\frac{\sqrt{2\pi}\,\sigma_x}{2\pi \sigma_x \sigma_y \sqrt{1-\alpha^2}}}_{\frac{1}{\sigma_y\sqrt{2\pi(1-\alpha^2)}}} \exp\left\{ \underbrace{\frac{-(x-\mu_x)^2}{2\sigma_x^2(1-\alpha^2)} + \frac{(x-\mu_x)^2}{2\sigma_x^2}}_{\frac{-\alpha^2(x-\mu_x)^2}{2\sigma_x^2(1-\alpha^2)}} - \frac{1}{2(1-\alpha^2)} \left[\frac{(y-\mu_y)^2}{\sigma_y^2} - 2\alpha \frac{(x-\mu_x)(x-\mu_y)}{\sigma_x \sigma_y} \right] \right\}$$

$$\Rightarrow P(y|x) = \frac{1}{\sigma_y\sqrt{2\pi(1-\alpha^2)}} \exp\left\{ -\frac{1}{2(1-\alpha^2)} \underbrace{\left[\frac{\alpha^2(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - 2\alpha \frac{(x-\mu_x)(x-\mu_y)}{\sigma_x \sigma_y} \right]}_{\left(\frac{\alpha(x-\mu_x)}{\sigma_x} - \frac{(y-\mu_y)}{\sigma_y}\right)^2} \right\}$$

$$E[y|x] = \int y P(y|x)\,dy = \frac{1}{\sigma_y\sqrt{2\pi(1-\alpha^2)}}\int y \times \exp\left\{-\frac{\left(\frac{\alpha(x-\mu_x)}{\sigma_x} - \frac{y-\mu_y}{\sigma_y}\right)^2}{2\sigma_y^2(1-\alpha^2)}\right\}dy$$

$= \ldots ?$

3.13. MSE attempts to minimize a cost function $J(w)$.

$J(w) = E[(f(x;w) - y)^2]$ → the expectation of squared errors of the trained function w.r.t. targets (y).

$$E[(f(x;w)-y)^2] = P(x,\omega_1)(f(x;w)-y_1)^2 + P(x,\omega_2)(f(x;w)-y)^2 \quad \text{(I)}$$

$p(x,\omega_i) = P(\omega_i|x)\cdot P(x)$; $y_1 = +1\ (\omega_1)$, $y_2 = -1\ (\omega_2)$ $\Rightarrow$ (I) $= (f(x;w)-1)^2 P(\omega_1|x) + (f(x;w)+1)^2 p(\omega_2|x)$

$$= f(x;w)^2 - 2f(x;w)[P(\omega_1|x) - P(\omega_2|x)] + \overbrace{P(\omega_1|x) + P(\omega_2|x)}^{1}$$

Adding and subtracting $[p(\omega_1|x) - p(\omega_2|x)]^2$ we have:

$$f(x;w)^2 - 2f(x;w)[P(\omega_1|x) - P(\omega_2|x)] + [P(\omega_1|x) - P(\omega_2|x)]^2 - [P(\omega_1|x) - P(\omega_2|x)]^2 + 1$$

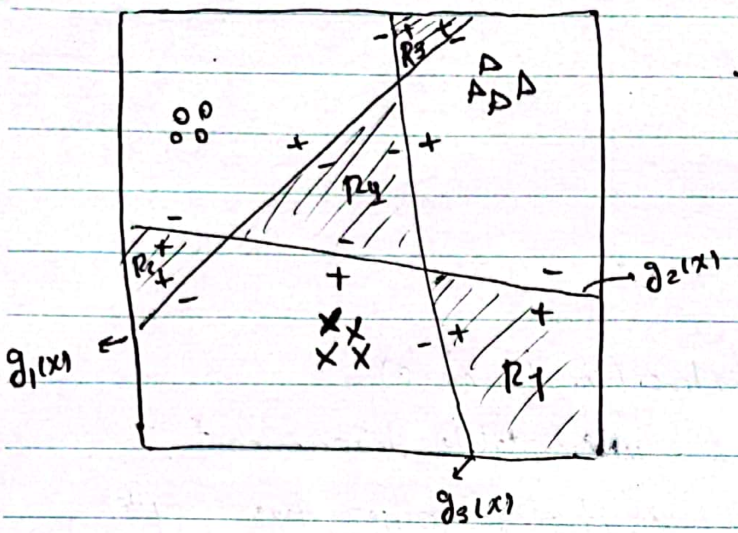
$$= \Big(f(x;w) - \overbrace{[P(\omega_1|x) - P(\omega_2|x)]}^{g(x),\ \text{the optimal Bayes decision surface.}}\Big)^2 - \underline{[P(\omega_1|x) - P(\omega_2|x)]^2 + 1}$$

this part is not affected by the parameters of f, namely w therefore we ignore it in the minimization task.

$$\Rightarrow \hat{w}_{MSE} = \arg\min_w E[(f(x;w)-y)^2] = \arg\min_w E[(f(x;w) - [g(x)])^2]$$

minimizing $J(w)$ is equivalent to approximating $g(x)$ in MSE optimal sense.

3.15. Take the example below in a bivariate space:



Legend: $o \rightarrow \omega_1$

$\Delta \rightarrow \omega_2$

$X \rightarrow \omega_2$

as we can see with this configuration regions R_1, R_2, R_3 result in positive $g(x)$ for more than one class and do not contain any training data. while R_4 results in negative $g(x)$ for all classes.

3.16. If we write the KKT conditions for the problem stated in example 3.5 we have:

$$\begin{aligned}
&\lambda_1(w_1+w_2+w_0-1)=0\\
&\lambda_2(w_1-w_2+w_0-1)=0\\
&\lambda_3(w_1-w_2+w_0-1)=0\\
&\lambda_4(w_1+w_2-w_0-1)=0
\end{aligned}
\xrightarrow[\text{through the origin "} w_0=0 \text{"}]{\text{restricting to lines passing}}
\begin{cases}
\lambda_1(w_1+w_2-1)=0\\
\lambda_2(w_1-w_2-1)=0\\
\lambda_3(w_1-w_2-1)=0\\
\lambda_4(w_1+w_2-1)=0
\end{cases}$$

By removing w_0 we have effective reduced the number of Constraints to two as λ_1, λ_4 and λ_2, λ_3 can be squashed together.

Therefore the new KKT conditions are:

$$\begin{cases}
\lambda_1'(w_1+w_2-1)=0\\
\lambda_2'(w_1-w_2-1)=0
\end{cases} \quad \text{I}
\qquad
\begin{aligned}
&\frac{\partial L}{\partial w_1}=0 \Rightarrow w_1=\lambda_1'+\lambda_2'\\
&\frac{\partial L}{\partial w_2}=0 \Rightarrow w_2=\lambda_1'-\lambda_2'
\end{aligned} \quad \text{II}$$

Substitution of (II) in (I) results in:

$$\begin{cases} \lambda_1'(2\lambda_1'-1)=0 \\ \lambda_2'(2\lambda_2'-1)=0 \end{cases}$$

Now we will consider 4 cases:

Case #1: both λ_1' and λ_2' are inactive ($\lambda_1'=0$; $\lambda_2'=0$)

$\Rightarrow w_1=0, w_2=0 \longrightarrow g(x)=0 \longrightarrow$ unacceptable discriminant as it misclassifies two samples while we're using hard-SVM.

Case #2 and #3: either one of λ_1' or λ_2' is active ($\lambda_1'=0, \lambda_2'\neq 0$ or $\lambda_1'\neq 0, \lambda_2'=0$)

Both cases are similar so w.l.o.g let's assume $\lambda_1'=0, \lambda_2'\neq 0$

$\rightarrow (2\lambda_2'-1)=0 \rightarrow \lambda_2'=\frac{1}{2} \longrightarrow w_1=\frac{1}{2}, w_2=-\frac{1}{2}$

$\Rightarrow g(x)=\frac{1}{2}x_1-\frac{1}{2}x_2 \longrightarrow$ unacceptable discriminant since it misclassifies one sample

(for the other case we get $g(x)=-\frac{1}{2}x_1+\frac{1}{2}x_2$ which is unacceptable for the same reason)

Case #4: Both λ_1' and λ_2' are active ($\lambda_1'\neq 0$ and $\lambda_2'\neq 0$)

$$\rightarrow \begin{cases} 2\lambda_1'-1=0 \\ 2\lambda_2'-1=0 \end{cases} \rightarrow \begin{cases} \lambda_1=\frac{1}{2} \\ \lambda_2=\frac{1}{2} \end{cases} \rightarrow \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}+\frac{1}{2} \\ \frac{1}{2}-\frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$\rightarrow g(x)=x_1 \longrightarrow$ which is also the boundary achieved in example 3.5 and classifies perfectly.