Lecture 4 Feb 4

Classification

- Classify a sample $\vec{x}$ to a finite y.
- Use the data ($\vec{x}$'s) features to determine model
- Find a model to form a "decision boundary" for a sample

Classification by Regression

We are trying to split the data into it's classes where as prediction regression tries to fit the model.

Use the same model: $\hat{y}_i = f(x) = \vec{x}^T \vec{w}$

Note: using the LS error function does not give us the best classification model since it calculates the distance of each residual

Zero/One Loss Function (Better Loss Func)

Zero for right value, one for wrong values.

We are counting the number of wrong values.

Count # of -1 in $\text{sgn}(y f(x))$

Logistic Regression

Assuming a 2-class universe

We can get the probability of a sample x as,

$$p(x) = p(x, C_1) + p(x, C_2) = p(x|C_1)p(C_1) + p(x|C_2)p(C_2)$$

Thus

$$p(C_1|x) = \frac{p(x|C_1) \cdot p(C_1)}{p(x)} \qquad \text{(Posteriori)}$$

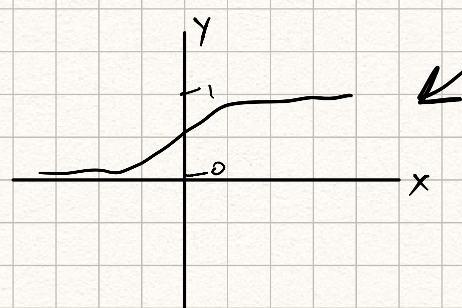
$$= \frac{p(x|C_1) \cdot p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)}$$

$$= \frac{1}{1 + \frac{p(x|C_2)p(C_2)}{p(x|C_1)p(C_1)}}$$

$$= \frac{1}{1 + e^{-a(x)}} = g(a(x))$$

where $\underline{\underline{a(x)}} = \ln\left(\frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}\right)$ is the ratio of probabilities of C_1, C_2

and $g(a)$ is called the sigmoid function also used in ANN.



So when $a(x) = 0$, $g(a) = \frac{1}{2}$ and we use $g(a) = \frac{1}{2}$ as the boundary point for binary classification.

Normal Generated Classes.

Now under the assumption that the classes are generated normally that is $C_i \sim N(\vec{\mu}, \Sigma)$

$a(x)$ becomes a linear function

$$a(x) = \vec{w}^T\vec{x} + b \quad \text{or} \quad \vec{w}^T\vec{x}$$

and thus $g(a(x))$ becomes

$$g(\vec{w}^T\vec{x}) = \frac{1}{1+e^{-\vec{w}^T\vec{x}}} = p(c_1|\vec{x})$$

Thus given an arbitrary regression line $w^T\vec{x}$, and sample point $\vec{x}^0$, the model predicts $\vec{x}^0$'s class based on which side of the line it's on.

$$\text{sgn}(\vec{w}^T\vec{x}^0) \in \{-1, 1\}$$

$$g(\vec{w}^T\vec{x}) \in [0,1] \text{ based on }, \; g(-1)\to 0, \; g(1) \to 1$$

Prediction the Model

Since there is no reason to assume one line is more likely, we assume a uniform $\vec{w}$; $p(w) = 1$.

Thus A Posteriori $\propto$ MLE.

$$\begin{aligned}
ML: p(\vec{x},\vec{y}\,|\,\vec{w}) &\propto p(\vec{y}\,|\,\vec{w},\vec{x}) \\
&= \prod p(y_i\,|\,\vec{w}, x_i) \\
&= \prod p(c_i|x_i)^{y_i} \cdot (1-p(c_i|x_i))^{(1-y_i)}
\end{aligned}$$

And $L(\vec{w}) = -\sum y_i \ln p(c_i|x_i) + (1-y_i)\ln(1-p(c_i|x_i))$

Let $p_i = p(c_i|x_i)$

$$\frac{dL}{d\vec{w}} = -\sum \frac{y_i}{p_i}\cdot\frac{d}{d\vec{w}}p_i + \frac{1-y_i}{1-p_i}\,\frac{d}{d\vec{w}}(1-p_i)$$

Since $p_i = p(c_i|x_i) = g(\vec{w}^T\vec{x}_i) = g(a(x_i))$

and $\frac{dg}{da} = g(a)\cdot(1-g(a))$

we have $\frac{dp_i}{d\vec{w}} = \frac{dg}{da} \cdot \frac{da}{dw}$

$$= -\sum y_i (1-p_i)\vec{x}_i - (1-y_i) p_i \vec{x}_i$$

$$= -\sum (y_i - p_i) \vec{x}_i$$

Problem with log-classification is that there's no closed form solution. But $\frac{dL}{d\vec{w}}$ is convex thus there's a minimum.

Regularization of Logistic Regression

We can regularize the $\frac{dL}{dw}$ function by adding $\frac{w}{\sigma^2}$ or by adding $\frac{w^T w}{2\sigma^2}$ to L.

This is equivalent to assuming a normal prior where,

$$p(\vec{w}) = G(w; 0, \sigma^2) \sim N(0, \sigma^2)$$

We regularize since a big $\vec{w}$ makes $g(a(x))$ be have like a step function which can make estimation unstable.