Instance based classification:

The main idea behind instance-based classification is that similar instances have similar classification.

One of the simplest examples of instance based classification is K-nearest neighbor (K-NN) classifier. In K-NN an instance is assigned to the most common class among the instances similar to it.

Initialization, define K

↓

Compute distance (test instance, each training instance)

↓

Sort the distances

↓

Take K nearest neighbors

↓

Apply simple majority → Class

One of the main drawbacks of K-NN is the curse of dimensionality. As the feature space gets larger, the feature vectors become sparser and as a result distance between them increases. and distance can be dominated by irrelevant attributes. As a result, the neighborhood of a test instance doesn't contain instances that are 'similar' to it.

Linear classification:

It is a more powerful and systematic approach to classification. It has two major components:

(a) A score function that maps the raw data to class scores

(b) A loss function that measures the goodness of the scoring function in predicting the labels.

In linear classification, we use a linear scoring function

$$f(x^{(i)}, W, b) = Wx^{(i)} + b$$

where,

$$W = \begin{bmatrix} - w_1^T - \\ - w_2^T - \\ \vdots \\ - w_C^T - \end{bmatrix}$$

where C is the number of classes. The j^{th} entry of $f(x^{(i)}, W, b)$ is the confidence score of image $x^{(i)}$ belonging to class j. W and b are parameters of the scoring function.

Softmax classifier:

In softmax classifier we view the scores as unnormalized log probabilities for each class and use a cross-entropy loss of the form

$$L_i(\theta) = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_{y_j}}}\right)$$

where $f_{y_i} = w_i^T x^{(i)} + b_i$

$f_{y_j} = w_j^T x^{(i)} + b_j$

Then the loss for softmax is

$$L(\theta) = \frac{1}{m} \sum_{i=1}^{m} L_i(\theta)$$

Practice Problems:

1. We are given a test point X and it's K nearest neighbors $\{z_1, z_2, \dots, z_K\}$

(a) Let E_1 be the event that 1-NN classifier makes a mistake on X. So,

$$P(E_1)$$
$$= P(\text{label}(z_1) \neq \text{label}(X))$$
$$= 0.1$$

(b) Let E_3 be the event that 3-NN classifier makes a mistake on X

E_3 occurs when at least 2 of the 3 nearest neighbors of X have a label that is different from the label of X.

Z_1	Z_2	Z_3	Prob
D	D	D	$0.1 \times (0.2)^2$
D	D	S	$0.1 \times 0.2 \times 0.8$
D	S	D	$0.1 \times 0.8 \times 0.2$
S	D	D	$0.9 \times (0.2)^2$

$\therefore P(E_3) = 0.004 + 0.016 + 0.016 + 0.036$

$= 0.072$

(c) Since $P(E_3) < P(E_1)$ so 3-NN classifier is more robust than 1-NN classifier.

2

(a) From the plot we can see that $K=5$ and $K=11$ has the lowest mean validation error but $K=5$ has a smaller variance and hence better generalization and is more stable. $\therefore K^* = 5$

(b) As K increases the variance in the 5-fold CV error increases.

3

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - (x^{(i)} + \delta^{(i)})^T \theta\right)^2$$

a) Since $E[\cdot]$ is a linear operator,

so

$$E_{\delta \sim N}\left[\tilde{\mathcal{L}}(\theta)\right]$$

$$= \frac{1}{N} \sum_{i=1}^{N} E_{\delta \sim N}\left[\left(y^{(i)} - (x^{(i)} + \delta^{(i)})^T \theta\right)^2\right]$$

Hence, if we can compute the term then we are done. Let's compute the term.

Now,

$$\left(y^{(i)} - (x^{(i)} + \delta^{(i)})^T\theta\right)^2$$

$$= \left[\left(y^{(i)} - x^{(i)T}\theta\right) - \delta^{(i)T}\theta\right]^2$$

$$= \left(y^{(i)} - x^{(i)T}\theta\right)^2 - 2\left(y^{(i)} - x^{(i)T}\theta\right)\left(\delta^{(i)T}\theta\right) + \left(\delta^{(i)T}\theta\right)^2$$

Since $E[\cdot]$ is a linear operator, so

$$= E_{\delta\sim N}[\quad] - E_{\delta\sim N}[\quad] + E_{\delta\sim N}[\quad]$$

Now let's compute the above 3 quantities:

Since [highlight] has no δ dependence, so

$$E_{\delta\sim N}[\;] = (y^{(i)} - x^{(i)T}\theta)^2$$

Now,

$$E_{\delta\sim N}\left[-2(y^{(i)} - x^{(i)T}\theta)(\delta^{(i)T}\theta)\right]$$

$$= -2(y^{(i)} - x^{(i)T}\theta)\, E_{\delta\sim N}\left[\delta^{(i)T}\theta\right]$$

From problem statement, $E[\delta^{(i)}] = 0 \in \mathbb{R}^d$

Hence,

$$E_{\delta\sim N}\left[-2(y^{(i)} - x^{(i)T}\theta)(\delta^{(i)T}\theta)\right]$$

$$= 0$$

Lastly,

$$E_{\delta\sim N}\left[\left(\delta^{(i)T}\theta\right)^2\right]$$

$$= E_{\delta\sim N}\left[\theta^T \delta^{(i)} \delta^{(i)T} \theta\right]$$

$$= \theta^T E_{\delta\sim N}\left[\delta^{(i)} \delta^{(i)T}\right] \theta$$

From the hint we know,

$$E_{\delta\sim N}\left[\delta\delta^T\right] = \sigma^2 I$$

Hence,

$$E_{\delta\sim N}\left[\left(\delta^{(i)T}\theta\right)^2\right]$$

$$= \sigma^2 \theta^T\theta = \sigma^2 \|\theta\|_2^2$$

Putting it all together,

$$= \left(y^{(i)} - x^{(i)T}\theta \right)^2 + \sigma^2 \|\theta\|_2^2$$

$\therefore$

$$E_{\delta \sim N}\left[\tilde{L}(\theta) \right]$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - x^{(i)T}\theta \right)^2 + \sigma^2 \|\theta\|_2^2$$

$$= L(\theta) + R$$

where $R = \sigma^2 \|\theta\|_2^2$

b) As the regularization strength $\sigma \to 0$, then we have no regularization and hence the model might overfit the data.

c) As the regularization strength $\sigma \to \infty$, then the objective of the cost function is to minimize the L-2 norm of parameters θ and hence $\theta \to 0$ and the model will underfit the data.

4

(i) $i = j$:

By chain rule,

$$\frac{\partial r_i}{\partial w_i} = \frac{\partial z_i}{\partial w_i} \frac{\partial r_i}{\partial z_i}$$

Now,

$$\frac{\partial z_i}{\partial w_i} = X$$

Recall quotient rule from calculus,

$$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}$$

Then using the quotient rule,
we get

$$\frac{\partial}{\partial z_i}\left[\frac{e^{z_i}}{\sum_p e^{z_p}}\right] = \frac{e^{z_i}}{\sum_p e^{z_p}} - \left[\frac{e^{z_i}}{\sum_p e^{z_p}}\right]^2$$

$$= r_i(1-r_i)$$

Hence,

$$\frac{\partial r_i}{\partial w_i} = r_i(1-r_i)\,x$$

Again by chain rule,

$$\nabla_{w_i} c(r_i) = \frac{\partial r_i}{\partial w_i} \quad \nabla_{r_i} c(r_i)$$

Now,

$$\nabla_{r_i} c(r_i) = -\frac{1}{r_i}$$

Putting it all together,

$$\nabla_{w_i} c(r_i) = (r_i - 1)X$$

ii) $\underline{i \neq j:}$

By chain rule

$$\frac{\partial r_j}{\partial w_i} = \frac{\partial z_i}{\partial w_i} \frac{\partial r_j}{\partial z_i}$$

By quotient rule,

$$\frac{\partial}{\partial z_i}\left[\frac{e^{z_j}}{\sum_p e^{z_p}}\right] = \frac{-e^{z_j}}{\sum_p e^{z_p}} \cdot \frac{e^{z_i}}{\sum_p e^{z_p}}$$

$$= -r_i r_j$$

Hence,

$$\frac{\partial r_j}{\partial w_i} = -r_i r_j X$$

Again by chain rule,

$$\nabla_{w_i} c(r_j) = \frac{\partial r_j}{\partial w_i} \nabla_{r_j} c(r_j)$$

Now,

$$\nabla_{r_j} c(r_j) = -\frac{1}{r_j}$$

Hence,

$$\nabla_{w_i} c(r_j) = r_i X$$

5

$$\mathcal{L}(w,b) = \frac{1}{K}\sum_{i=1}^{K} \text{hinge}_{y^{(i)}}(x^{(i)}) + \lambda \|w\|_1$$

Since gradient is a linear operator,

so

$$\nabla_W \mathcal{L}(w,b) = \frac{1}{K}\sum_{i=1}^{K} \nabla_w \text{hinge}_{y^{(i)}}(x^{(i)})$$

$$+ \lambda \nabla_W \|W\|_1$$

Let's compute the gradients highlighted

in and .

$$\text{hinge}_{y^{(i)}}(x^{(i)}) = \max(0, 1 - y^{(i)}(w^T x^{(i)} + b))$$

The above function is a piecewise linear function and can be represented as

$$\text{hinge}_{y^{(i)}}(x^{(i)}) = \begin{cases} 0, & y^{(i)}(w^T x^{(i)} + b) > 1 \\ 1 - y^{(i)}(w^T x^{(i)} + b), & \text{otherwise} \end{cases}$$

If $y^{(i)}(w^T x^{(i)} + b) > 1$, then

$$\nabla_W \,\text{hinge}_{y^{(i)}}(x^{(i)}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^d$$

If $y^{(i)}(w^T x^{(i)} + b) < 1$, then

$\nabla_w \text{hinge}_{y^{(i)}}(x^{(i)}) = -y^{(i)} x^{(i)} \in \mathbb{R}^d$

Putting it all together,

$$\nabla_w \text{hinge}_{y^{(i)}}(x^{(i)}) = \mathbb{I}_{\{y^{(i)}(w^T x^{(i)} + b) < 1\}} \odot (-y^{(i)} x^{(i)})$$

where,

$\mathbb{I}_{\{y^{(i)}(w^T x^{(i)} + b) < 1\}}$ is a vector of all ones if $y^{(i)}(w^T x^{(i)} + b) < 1$

or a vector of all zeros if $y^{(i)}(w^T x^{(i)} + b) > 1$

1:

$$\|w\|_1 = \sum_{i=1}^{d} |w_i|$$

Hence,

$$\nabla_w \|w\|_1 = \begin{bmatrix} \nabla_{w_1} |w_1| \\ \nabla_{w_2} |w_2| \\ \vdots \\ \nabla_{w_d} |w_d| \end{bmatrix}$$

Now,

$$|w_i| = \begin{cases} w_i, & w_i > 0 \\ -w_i, & w_i < 0 \end{cases}$$

for $i = 1, \dots d$.

So,

$$\nabla_{w_i} |w_i| = \begin{cases} 1, & w_i > 0 \\ -1, & w_i < 0 \end{cases}$$

The above can be written more compactly using the $\mathrm{sgn}(\cdot)$ operator:

$$\mathrm{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

$$\nabla_{w_i} |w_i| = \mathrm{sgn}(w_i)$$

Hence,

$$\nabla_w \|w\|_1 = \operatorname{sgn}(w) \in \mathbb{R}^d$$

where $\operatorname{sgn}(\cdot)$ operates on each element of w.

Putting everything together,

$$\nabla_w \mathcal{L}(w, b)$$

$$= \frac{1}{K} \sum_{i=1}^{K} \mathbb{I}\{y^{(i)}(w^T x^{(i)} + b) < 1\} \odot (-y^{(i)} x^{(i)})$$

$$+ \lambda \operatorname{sgn}(w)$$