Lec 3

Thursday, September 5, 2019 10:59

Recap: Supervised learning

$$X_1, Y_1, \ldots, X_n, Y_n \xrightarrow{\text{Algo}} \boxed{f} \to \hat{Y} \overset{?}{\approx} Y$$

$$\uparrow$$

$$X$$

kNN as a prob classifier

$$\hat{\mathbb{P}}(Y=g|X) = \frac{1}{K} \sum_{j=1}^{K} \mathbb{I}[Y_{i_j} = g]$$

For $G = \{0,1\}$

$$\hat{Y} = \underset{g \in \{0,1\}}{\operatorname{argmax}} \hat{\mathbb{P}}(Y=g|X) = \mathbb{I}[\hat{\mathbb{P}}(Y=1|X) > \tfrac{1}{2}]$$

let's use $\mathbb{I}[\hat{\mathbb{P}}(Y=1|X) > \theta]$ instead

this changes the classification rates.

Classification rates

$\#TP$ = no. of test examples w/ $Y=1$ $\hat{Y}=1$

$\#TN, \#FN, \#FP$ def similarly

Confusion matrix

$\hat{Y} \backslash Y$	1	0
1	$\#TP$	$\#FP$
0	$\#FN$	$\#TN$

Rates:

$$TPR = \frac{\#TP}{\#P} = \frac{\#TP}{\#TP + \#FN}$$

$$TNR = \frac{\#TN}{\ } = \frac{\#TN}{\ }$$

$$TPR = \frac{\#P}{} = \frac{}{\#TP + \#FN}$$

$$TNR = \frac{\#TN}{\#N} = \frac{\#TN}{\#TN + \#FP}$$

$FNR = 1 - TPR \qquad FPR = 1 - TNR$

$$Acc = 1 - (\text{Avg 0-1 loss}) = \frac{\#TP + \#TN}{\#TP + \#TN + \#FP + \#FN}$$

Recall = TPR

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP}$$

$$\text{F-score} = \frac{2}{\frac{1}{\text{prec}} + \frac{1}{\text{recall}}}$$

Receiver Operating Characteristic (ROC) curve

The ROC curve is the plot of FPR (x-axis) vs TPR (y-axis) as we vary the classification threshold, θ.

ROC tells us how good a prob classification is, regardless of the specific threshold.

Can quantify how good an ROC curve looks in terms of its Area Under the curve (AUC)

$AUC = \frac{1}{2} \to$ corresponds to

$AUC = \frac{1}{2} \to$ corresponds to random guessing

$AUC = 1 \to$ corresponds to perfect prediction

Fun & useful fact:

AUC exactly equal to the prob that given two examples, one pos & one neg, that our score sorts them in the right order.

Regression

Recall regression is the setting when Y can be real-valued

$$Y \in G = \mathbb{R}$$

kNN as a regression method

$$\hat{Y} = \frac{1}{K} \sum_{j=1}^{K} Y_{i_j}$$

Lazy vs Eager training

kNN is an example of "lazy" training or "memory-based"

- wait until we get a query X in order to compute $f(X)$

" "

in order to compute $f(x)$"

"Eager" training tries to build our predictor f ahead of time & summarize it into a simple model.

The Linear Model

Posit a model parametrized by $\beta \in \mathbb{R}^{p+1}$

$$f_\beta(x) = \beta_0 + \sum_{i=1}^{p} \beta_j \cdot x_j$$
$$= \underbrace{\beta_0}_{\text{intercept/bias}} + \underbrace{\beta_{1:p}^T}_{\text{coefficients/slopes/weights}} x$$

To make things simple, we'll just assume that one of the features in X is the constant feature: $X_{i1} = 1 \ \forall i$

Then we write $\beta \in \mathbb{R}^p$

$$f_\beta(x) = \beta^T x$$

The question is now how to choose f_β in this model class — i.e. how to choose β

Least Squares

To evaluate the fit of any f to regression data we'll use the squared error loss:

$$\ell(y, \hat{y}) = (y - \hat{y})^2$$

This leads to the avg squared residuals (empirical) risk:

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^n \ell(Y_i, f(X_i))$$

$$= \frac{1}{n}\sum_{i=1}^n (\underbrace{Y_i - f(X_i)}_{\text{residual}})^2$$

The linear model defines a particular class of f's:

$$f \in \{f_\beta : \beta \in \mathbb{R}^p\}$$

Seek the f in this class that does the best on the training data.

$$\hat{R}_n(f_\beta) = \frac{1}{n}\sum_{i=1}^n (Y_i - f_\beta(X_i))^2$$

$$= \frac{1}{n}\sum_{i=1}^n (Y_i - \beta^T X_i)^2$$

$$= \frac{1}{n}\| \mathbb{Y} - \mathbb{X}\beta \|_2^2$$

Where we define $\mathbb{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \in \mathbb{R}^n$

design matrix $\to$ $\mathbb{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \in \mathbb{R}^{n \times p}$

$$\mathbb{X}\beta = \begin{pmatrix} x_1^T\beta \\ \vdots \\ x_n^T\beta \end{pmatrix} = \begin{pmatrix} f_\beta(x_1) \\ \vdots \\ f_\beta(x_n) \end{pmatrix} \in \mathbb{R}^n$$

We want $\min_{\beta \in \mathbb{R}} \hat{R}_n(f_\beta)$

Call the optimizer $\hat{\beta}$

diff'able & convex (cup-like) fn so min occurs at critical pts

$$\frac{\partial \hat{R}_n(f_\beta)}{\partial \beta_j} = \frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial \beta_j}(\beta^T x_i - y_i)^2$$

$$= \frac{1}{n}\sum_i 2(\beta^T x_i - y_i)\, x_{ij}$$

$$= \frac{2}{n}\sum_i (\beta^T x_i - y_i)\, \mathbb{X}_{ij}$$

$$\nabla_\beta \hat{R}_n(f_\beta) = \begin{pmatrix} \partial \hat{R}_n(f_\beta)/\partial\beta_1 \\ \vdots \\ \partial \hat{R}_n(f_\beta)/\partial\beta_p \end{pmatrix} = \frac{2}{n}\mathbb{X}^T(\mathbb{X}\beta - \mathbb{Y})$$

Want to solve for β in

$$0 = \nabla_\beta \hat{R}_n(f_\beta) = \frac{2}{n}\mathbb{X}^T(\mathbb{X}\beta - \mathbb{Y})$$