24-11-2016

Statistical Learning IV

1. Plug-in Estimators
2. Linear Discriminant Analysis (LDA)
3. Naive Bayes
4. Surrogate Losses
5. Log Loss
6. Logistic Regression

1. Plug-in Estimators

Bayes' rule gives: $p^*(y=k|x) = \frac{p^*(x|y=k)\,p^*(y=k)}{p^*(x)}$

Notation:

$f_k(x) := p^*(x|y=k)$

$\pi_k := p^*(y=k)$

Plug-in estimators:

- Estimate $p^*(y=k|x)$ by $\hat{p}(y=k|x)$ and predict with $\operatorname{argmax}_k \hat{p}(y=k|x)$

- By Bayes' rule, it is suffient to estimate f_k and π_k for all classes k:

$$\hat{f}(x) = \operatorname*{argmax}_k \frac{\hat{f}_k(x)\,\hat{\pi}_k}{\hat{p}^*(x)} = \operatorname*{argmax}_k \hat{f}_k(x)\cdot\hat{\pi}_k$$

N.B. Although we have used Bayes' rule, there is nothing inherently Bayesian about this approach. (Can estimate $\hat{f}_k$ by Bayesian or frequentist methods.)

5. Linear Discriminant Analysis (LDA)

Model: $f_k(x)$ is $N(\mu_k, \Sigma_k)$ is multivariate Gaussian (see fig. 4.5)

Take $\Sigma_k = \Sigma$ the same for all k.

Parameter estimates from T:

$\hat{\pi}_k = \frac{n_k}{N}$ where n_k is number of observations in class k.

$\hat{\mu}_k = \frac{1}{n_k} \sum_{i: y_i = k} x_i$ is mean of x_i in class k

$$\hat{\Sigma} = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i: y_i = k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

$$\begin{aligned}
\underset{k}{\operatorname{argmax}}\ \hat{f}_k(x) \cdot \hat{\pi}_k &= \underset{k}{\operatorname{argmax}}\ \log \hat{f}_k(x) + \log \hat{\pi}_k \\
&= \underset{k}{\operatorname{argmax}}\ \log\left(\frac{1}{(2\pi)^{p/2} |\hat{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\hat{\mu}_k)^T \hat{\Sigma}^{-1} (x-\hat{\mu}_k)} \right) + \log \hat{\pi}_k \\
&= \underset{k}{\operatorname{argmax}}\ -\frac{1}{2}(x-\hat{\mu}_k)^T \hat{\Sigma}^{-1}(x-\hat{\mu}_k) + \log \hat{\pi}_k \\
&= \underset{k}{\operatorname{argmax}}\ -\frac{1}{2} x^T \hat{\Sigma}^{-1} x + x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k
\end{aligned}$$

$\delta_k(x)$

linear in x, so decision boundary between any two classes also linear in x. Hence name LDA

Decision boundary: (between classes k_1 and k_2)

x s.t. $\delta_{k_1}(x) - \delta_{k_2}(x) = 0$

3. Naive Bayes

Model: features are independent given their class k

Remark: This implies that $\Sigma_k = \begin{pmatrix} \sigma_{k,1}^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_{k,p}^2 \end{pmatrix}$

is a diagonal matrix with p parameters, because (by assumption) all covariances are 0.

For each class k and feature vector x_i:

$$f_k(x_i) = \prod_{j=1}^{p} f_{kj}(x_{ij})$$

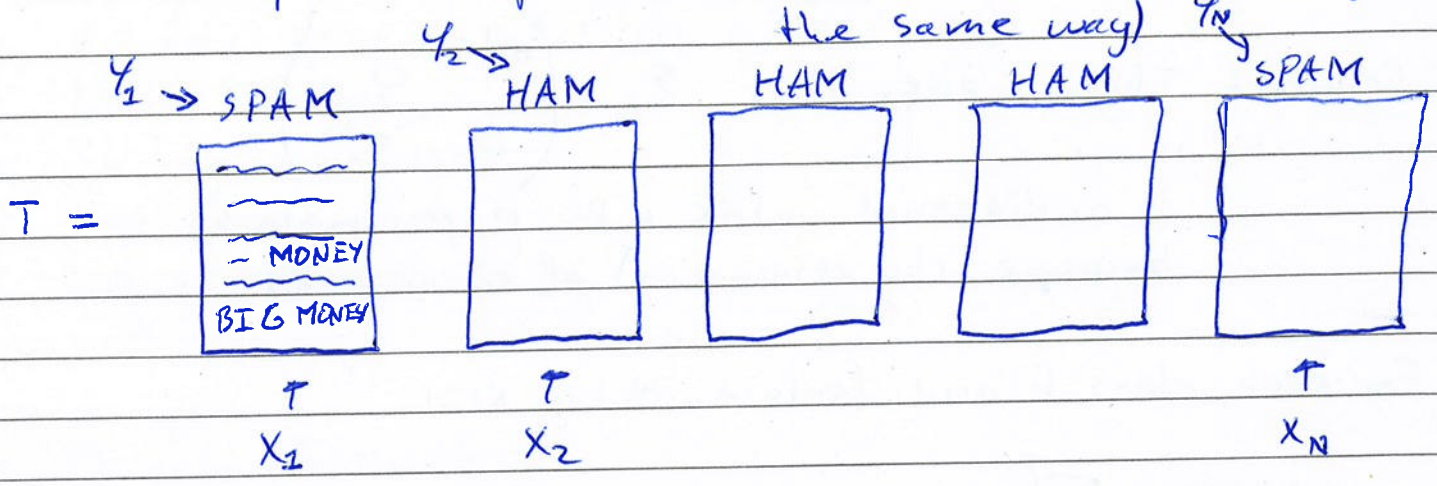
← independent features; very useful simplification if p is very large, because then we have no hope of learning the true distribution exactly anyway.

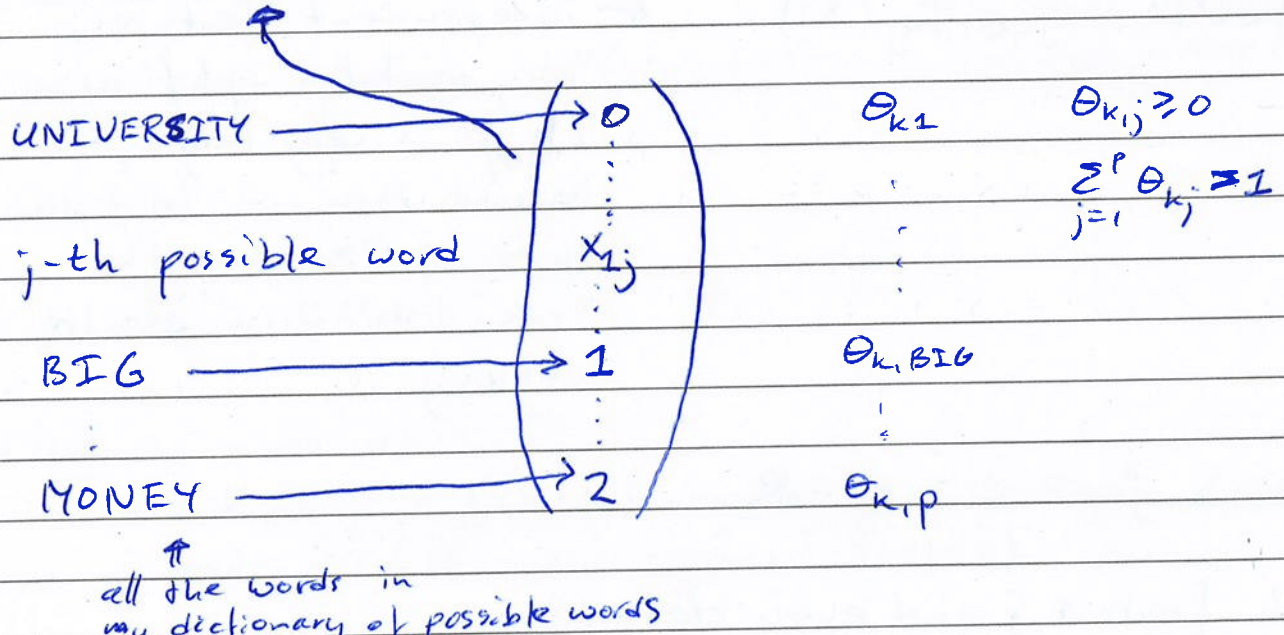
Continuous features: $x_{ij} \in \mathbb{R}$

For each feature j and every class k:
estimate f_{kj} by fitting a univariate Gaussian (similar to LDA).

Discrete features:

Consider spam classification. (other text classification goes the same way)





(1) Forget position of each word in text, and represent each e-mail by counts of how many times words occur.

(2) Use separate multinomial model per class k:

$f_k(x_i) = \prod_{j=1}^{p} \theta_{kj}^{n_{ij}}$, where n_{ij} is the number of times the j-th word occurs in the i-th e-mail.

$\hat{\pi}_k = \frac{\#\text{e-mails in class } k}{N}$

$\hat{\theta}_{k,MONEY} = \frac{\#\text{MONEY in e-mails from class } k + \text{small num}}{\#\text{words in all e-mails from class } k + p \cdot \text{small num}}$

Without adding the "small numbers": if a spam e-mail contains one word that we have never seen in a spam e-mail before, then $f_{SPAM}(x_i) = 0$, and the e-mail will be classified as HAM. So a single word can throw off classification.

Too simple/naive?

- Only need $\hat{P}(y=k_1|x) > \hat{P}(y=k_2|x)$ whenever $P^*(y=k_1|x) > P^*(y=k_2|x)$
- Possible even if $\hat{P}$ is poor estimate of P^*!

4. Surrogate Losses

Classification: find $\hat{f}$ with small expected 0/1-loss

$$EPE(\hat{f}) = \mathbb{E}_{x,y}[L(y, \hat{f}(x))]$$

0 if $\hat{f}(x)$ predicts y correctly
1 if $\hat{f}(x)$ predicts y incorrectly

Empirical Risk Minimization:

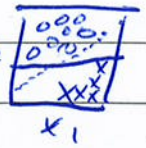
$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i))$$

Suppose $\mathcal{F}$ is set of all linear functions: $\mathcal{F} = \{f_\beta(x) = x^T\beta \mid \beta \in \mathbb{R}^p\}$

$y \in \{-1, +1\}$ $L(y, f(x)) = \begin{cases} 0 & \text{if } \operatorname{sign}(f(x)) = y \\ 1 & \text{if } \operatorname{sign}(f(x)) \neq y \end{cases}$

Difficulties:

- ERM solution not unique for linearly separable data:



- Computational problems for non-separable data.

Surrogate/proxy loss L':

Care about loss L, but train classifier with loss L'.

Want: L large $\Leftrightarrow$ L' large

L small $\Leftrightarrow$ L' small

Usually: surrogate L' convex in parameters, so can optimize efficiently

Two most important surrogates:

logistic loss (today), hinge loss (used in support vector machines (in 3 weeks))

For 0/1-loss, predict correctly iff

$$\text{sign}(x^T\hat{\beta}) = y \Leftrightarrow yx^T\hat{\beta} > 0$$

Most surrogate losses are therefore decreasing as a function of $yx^T\hat{\beta}$.

(see slide)

Exception: squared error N.B: $y^2 = 1$

$$(y - x^T\hat{\beta})^2 = y^2(y - x^T\hat{\beta})^2 = (y^2 - yx^T\hat{\beta})^2 = (1 - yx^T\hat{\beta})^2$$

increases for $yx^T\hat{\beta} > 1$!

This is undesirable!

5. Log(arithmic) Loss

* My predictions are <u>probability distributions</u> P for y

* $L(y, P) = -\log P(y)$
  - ↳ small if I gave large probability to y
  - ↳ large if I gave small probability to y

Maximum Likelihood = ERM for log loss:

$$\text{ML: } \arg\max_\beta \prod_{i=1}^N p_\beta(y_i|x_i) = \arg\min_\beta \sum_{i=1}^N -\log p_\beta(y_i|x_i)$$
$$= \arg\min_\beta \sum_{i=1}^N L(y_i, p_\beta(\cdot|x_i))$$

Bayes optimal predictor for log loss is true distribution:

$$\arg\min_p \mathbb{E}_y[L(y, p)] = p^*(\cdot|x)$$

6. Logistic Regression (for two classes for simplicity)

(How can we use the power of linear regression for classification?)

Bad idea: $\hat{p}(y=1|x) = x^T\hat{\beta}$

E.g. y_i is opening the window during i-th lecture
x_{i3} is avg temperature in the room in °C.
$\hat{\beta}_3 = \frac{1}{5}$
Then $+10°C \Rightarrow x^T\hat{\beta} + 2 \Rightarrow \hat{p}(y=1|x) + 2 > 1$! ($10 \cdot \frac{1}{5}$)

Probabilities do <u>not</u> behave linearly.

Better: $\log P_\beta(Y=1|x) = x^T\beta \iff P_\beta(y=1|x) = e^{x^T\beta}$

$x^T\hat{\beta}+2 \Rightarrow P_{\hat{\beta}}(y=1|x)$ increases by <u>factor</u> e^2,
but can still go outside $[0,1]$

Logistic Regression:

$$\log \frac{P_\beta(Y=1|x)}{P_\beta(Y=-1|x)} = x^T\beta \iff \begin{cases} P_\beta(y=1|x) = \frac{e^{x^T\beta}}{1+e^{x^T\beta}} = \frac{1}{1+e^{-x^T\beta}} \\ P_\beta(y=-1|x) = \frac{1}{1+e^{x^T\beta}} \end{cases}$$

(log odds is linear)

$$P_\beta(y|x) = \frac{1}{1+e^{-yx^T\beta}}$$

decision boundary:

$$P_{\hat{\beta}}(Y=1|x) > \tfrac{1}{2} \iff x^T\hat{\beta} > 0 \Rightarrow \quad x \text{ s.t } x^T\hat{\beta} = 0$$

is linear

logistic loss = log loss for logistic regression:

$$L(y,x^T\beta) = -\log P_\beta(y|x) = \log(1+e^{-yx^T\beta})$$

Train with maximum likelihood = ERM for logistic loss

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{N} \log(1+e^{-y_i x_i^T\beta})$$

No formula for minimum, but can find it by convex optimization

<u>Important extensions:</u>

$$\text{minimize} \sum_{i=1}^{N} \log(1+e^{-y_i x_i^T\beta}) + \lambda \text{pen}(\beta)$$

L_2-penalty: $\text{pen}(\beta) = \sum_{j=2}^{p} \beta_j^2$ (like ridge regression)

L_1-penalty: $\text{pen}(\beta) = \sum_{j=2}^{p} |\beta_j|$ (like lasso)

Considerations for penalization similar as for squared error.