5318 – 复习课 – 上

K-NN

1R

PRISM

Linear Regression

Naïve Bayes

Ensemble method

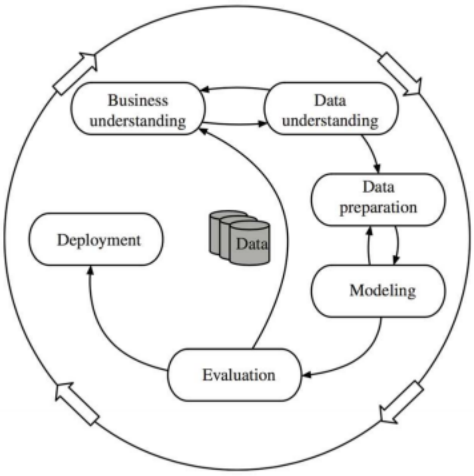
Supervised learning vs Unsupervised learning

训练时有没有用到 label

Two types of supervised learning

- Classification: the variable to be predicted is categorical (i.e. its values belong to a pre-specified, finite set of possibilities)
- Regression: the variable to be predicted is numeric

CRISP-DM - cross-industry process for data mining



Business understanding – 理解市场目标(objectives)和需求(requirements)

Data understanding – 分析 initial dataset

Data preparation – 数据预处理

Modelling – 建造模型

Evaluation – 评估模型好坏

Deployment – 落实到软件系统（不属于 5318 的内容）

数据预处理

Data cleaning – reduce noise (ml1b – 搜 noise), replace missing value

Data pre-processing – Data aggregation, Dimensionality reduction, feature extraction, feature selection, convert attributes type, normalization

Simple Matching Coefficient (SMC)

SMC = (f11+f00)/(f01+f10+f11+f00)

Correlation

Co-var(x, y) = Co-rel(x, y) * std(x) * std(y)

[-1, 1]

K - Nearest Neighbors – 分类 – lazy method

Why normalization for knn?

Used to avoid the dominance of attributes with large values over attributes with small values when calculating distance

KNN 优点

Often very accurate

Easy to understand

KNN 缺点

Slow for big datasets

Not effective for high-dimensional data

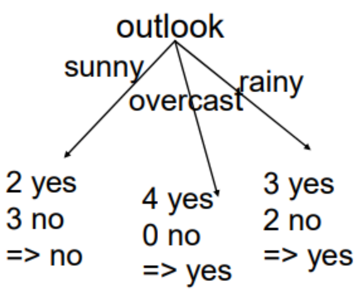
Sensitive to the value of k

1R – one rule (单层 decision tree)

If one attribute satisfies condition:
than the result can be decided.

How to determine the class label for the leaves?

Take the majority class

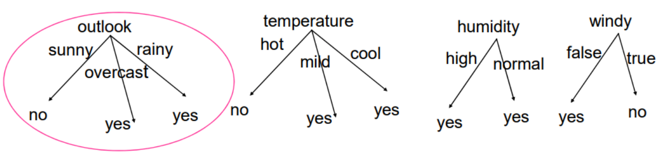


outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

How to select the best rule (attribute)?

The one with the smallest error rate (i.e. with the highest accuracy) on training data

- 1R algorithm
  - Generate a rule (decision stump) for each attribute
  - Evaluate each rule on the training data and calculate the number of errors
  - Choose the one with the smallest number of errors



Errors on training data: 4 5 4 5

rule No	attribute	attribute values & counts	rules	errors	total errors
1	outlook	sunny: 2 yes, 3 no overcast: 4 yes, 0 no rainy: 3 yes, 2 no	sunny -> no overcast -> yes rainy -> yes	2/5 0/4 2/5	4/14
2	temp.	hot: 2 yes, 2 no* mild: 4 yes, 2 no cool: 2 yes, 1 no	hot -> no mild -> yes cool -> yes	2/4 2/6 1/4	5/14
3	humidity	high: 4 yes, 3 no normal: 6 yes, 1 no	high -> yes normal -> yes	3/7 1/7	4/14
5	windy	true: 3 yes, 3 no* false: 6 yes, 2 no	true -> no false -> yes	3/6 2/8	5/14

* - random choice

Final rule - rule 1:
if outlook=sunny then play=no
elseif outlook=overcast then play=yes
elseif outlook=rainy then play=yes

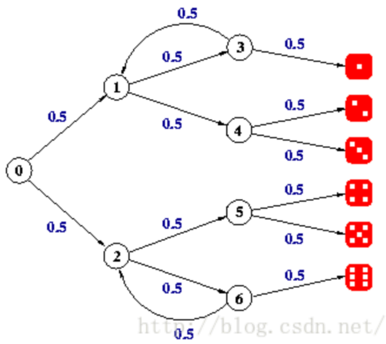
outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

PRISM

Generate a rule by adding tests that maximize the rule’ s accuracy

PRISM—*probabilistic model checker概率模型检测器*

骰子模型 The dieexample



http://www.prismmodelchecker.org/tutorial/die.php *与马尔科夫链有关*

age	spectacle prescription	astigmatism	tear production rate	recommended lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

p/t (accuracy)

- 9 possible tests for the 4 attributes based on num. attribute values (3+2+2+2):

  age = young 2/8

  age=young in 8 ex. and in 2 of them class=hard

  age= pre-presbyoptic 1/8

  age = presbyoptic 1/8

  spectacle prescription = myope 3/12

  spectacle prescription = hypermetrope 1/12

  astigmatism = no 0/12

  astigmatism = yes 4/12

  tear production rate = reduced 0/12

  tear production rate = normal 4/12

- Best test (highest accuracy): astigmatism = yes

- Further refinement by adding tests:

  if astigmatism = yes and ? then recommendation = hard

- Possible tests:

  age = young 2/4

  age= pre-presbyoptic 1/4

  age = presbyoptic 1/4

  spectacle prescription = myope 3/6

  spectacle prescription = hypermetrope 1/6

  tear production rate = reduced 0/6

  tear production rate = normal 4/6

- Best test: tear production rate = normal

- Further refinement:

```
if astigmatism = yes & tear production = normal and ? then
recommendation = hard
```

- Possible tests

```
age = young                                2/2
age= pre-presbyoptic                       1/2
age = presbyoptic                          1/2
spectacle prescription = myope             3/3
spectacle prescription = hypermetrope 1/3
```

- Best test: tie between the 1st and 4th; choose the one with the greater coverage (4th)
- New rule:

```
if astigmatism = yes & tear production = normal & spectacle
prescription = myope then recommendation = hard
```

Linear Regression

& linear regression – regression

& logistic regression – classification

Loss function – SSE (主要的)

$$SSE = \sum_i (y_i - \hat{y}_i)^2$$

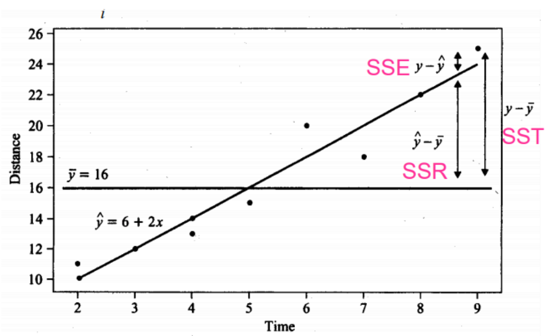
[residual]

SST – sum of total errors

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

SSR – sum of squared regression errors

$$SSR = \sum_{i}^{n} (\hat{y}_i - \bar{y})^2$$



R2 - measures the goodness of fit of the regression line found by the least squares method

$$R^2 = \frac{SSR}{SST}$$

[0,1] – higher, better

另外两种

Mean Squared Error (MSE): $MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

True or False?

1) The regression line minimizes the sum of the residuals False

No, the sum of squared residuals

2) If all residuals are 0, SST=SSR True

If the residuals are 0 =>SSE will be 0; SST=SSR+SSE => SST=SSR

3) If the value of the correlation coefficient is negative, this indicates that the 2 variables are negatively correlated True

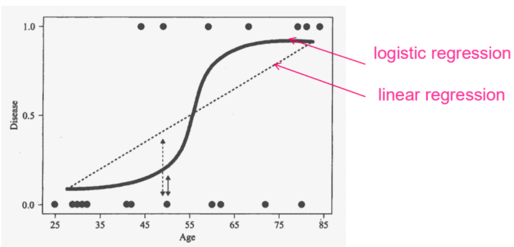
4) The value of the correlation coefficient can be calculated given the value of R^2 False

$r = \pm\sqrt{R^2}$

5) SSR represents an overall measure of the prediction error on the training set by using the regression line False

No, this is R^2

Logistic regression



$$p = \frac{e^{b_0+b_1x}}{1+e^{b_0+b_1x}}$$

It uses the **maximum likelihood method** to find the parameters b0 and b1 - the curve that best fits the data

$$\ln(odds) = b_0 + b_1 x \qquad => \qquad odds = e^{(b_0 + b_1 x_1)}$$

Compare:

- Logistic regression: $\ln(odds) = b_0 + b_1 x$
- Linear regression: $\hat{y} = \mathrm{b}_0 + \mathrm{b}_1 x$

Over-fitting (必考)

Overfitting:

- Small error on the training set but high error on test set (new examples)
- The classifier has memorized the training examples but has not learned to generalize to new examples!

It occurs when

- we fit a model too closely to the particularities of the training set – the resulting model is too specific, works well on the training data but doesn't work well on new data

Reasons –

Noise in the training data

Training set is too small

Model is too complex

How to deal with overfitting?

加正则项，增加 training set，降低模型复杂度

Underfitting

The model is too simple and doesn' t capture all important aspects of the data

Ridge regression = regularization + LR

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}_{\text{MSE}} + \alpha \underbrace{\sum_{i=1}^{n} w_i^2}_{\text{regularization term}}$$

Goal: high accuracy on training data (low MSE)

low complexity model – w close to 0

the regression coefficient w is chosen so that they not only fit well the training data (as in LR) but also satisfy an additional constraint - the magnitude of the coefficients is as small as possible, i.e. close to 0

Why?

Each feature will have little effect on the outcome

Small slope of the regression line

Lasso regression = L1 norm + LR

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}_{\text{MSE}} + \alpha \underbrace{\sum_{i=1}^{n} ||w_i||}_{\text{regularization term (L1 norm)}}$$

Goal: high accuracy on training data (low MSE)

low complexity model

Overfitting and regularization

- Overfitting - high accuracy on training data but low accuracy on test data (low generalization)
- High model complexity -> low generalization
- Regularization is a method to avoid overfitting – it makes the model more restrictive (less complex)
- Ridge and Lasso regression are regularized linear regression models

Probabilistic methods: Naïve Bayes

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

(重要假设 – feature 之间 independent !!!)

Similarly we calculate the other conditional probabilities:

$$P(yes \mid E) = \frac{P(E_1 \mid yes)P(E_2 \mid yes)P(E_3 \mid yes)P(E_4 \mid yes)P(yes)}{P(E)}$$

P(E1|yes)=P(outlook=sunny|yes)=?/9=2/9

P(E2|yes)=P(temp=cool|yes)=3/9

P(E3|yes)=P(humidity=high|yes)=3/9

P(E4|yes)=P(windy=true|yes)=3/9

P(yes)=?

- the prior probability of play=yes - the probability of play=yes without E, i.e. without knowing anything about the particular day
- calculated from the "play" column = 9/14

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

The "zero-frequency" problem

What if an attribute value does not occur with every class value?

E.g. suppose that the training data was different:

outlook=sunny had always occurred together with play=no, i.e.

outlook=sunny had never occurred together with play=yes

Then: P(outlook=sunny|yes)=0

$$P(yes \mid E) = \frac{\underbrace{P(E_1 \mid yes)}_{=0}P(E_2 \mid yes)P(E_3 \mid yes)P(E_4 \mid yes)P(yes)}{P(E)}$$

=> P(yes|E)=0, regardless of the other probability values

This means that the prediction for new examples with outlook=sunny will always be play=no, completely ignoring the values of the other attributes

Remedy: add 1 to the nominator and *m* to the denominator (*m* - number of attribute values = 3 for outlook)

This is called Laplace correction or smoothing

- it ensures that the probabilities will never be 0

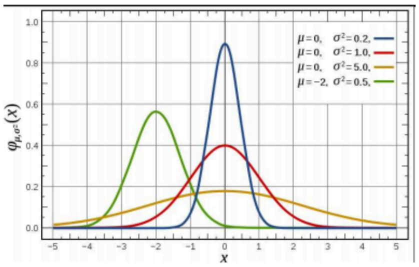
There is a generalization of the Laplace correction called *m*-estimate

Gaussian Bayes

Probability density function for a *normal* distribution with mean μ and standard deviation σ:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability density function is not exactly the probability but it is closely related



Reminder about μ and σ :

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n-1}}$$

$$f(temperature = 66 \mid yes) = \frac{1}{6.2\sqrt{2\pi}} e^{-\frac{(66-73)^2}{2 * 6.2^2}} = 0.034$$

μ for temp. for play=yes

σ for temp. for play=yes

- Similarly: $f(humidity = 90 \mid yes) = 0.0221$

$$P(yes \mid E) = \frac{\frac{2}{9}\, 0.034\ 0.0221 \frac{3}{9} \frac{9}{14}}{P(E)} = \frac{0.000036}{P(E)}$$

$$P(no \mid E) = \frac{\frac{3}{5}\, 0.0291\ 0.038 \frac{3}{5} \frac{5}{14}}{P(E)} = \frac{0.000136}{P(E)}$$

- P(no|E) >P(yes|E) => Naïve Bayes predicts play=no

Evaluation

Holdout method – training and testing (acc)

N fold Cross-validation

Leave-one-out cross-validation

Set the number of folds to the number of training examples

(for n training examples, build classifier n times)

优点：

Makes the best use of data - the greatest possible amount of data is used for training

Deterministic procedure – no random sampling is involved - the same result will be obtained every time

缺点：

High computational cost, especially for large datasets

Confuse matrix

examples	# assigned to class yes	# assigned to class no
# from class yes	true positives (tp)	false negatives (fn)
# from class no	false positives (fp)	true negatives (tn)

$$P = \frac{tp}{tp + fp} \qquad R = \frac{tp}{tp + fn} \qquad F1 = \frac{2PR}{P + R}$$

Precision **recall** **F1-score**

Ensemble method –

combination of multiple classifiers

Bagging

Bagging is also called bootstrap aggregation

(取出有放回)

Dataset with 10 examples:

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

Boosting （the most used ensemble method）

Idea: Make the classifiers complement each other
How: The next classifier should be created using examples that were difficult for the previous classifiers

(AdaBoost Boosting and Gradient Boosting)

AdaBoost - weighed training set
(每个训练 sample 有一个权重) – 权重代表 - 被正确分类的难度
权重越高，下一次被选取的可能性就越高

Gradient Boosting - adds a new model that minimizes the error of the previous model

Bagging vs Boosting

Similarities

- Use voting (for classification) and averaging (for prediction) to combine the outputs of the individual learners
- Combine classifiers of the same type, typically trees – e.g. decision stumps or decision trees

Differences

- Creating base classifiers:
  - Bagging – separately
  - Boosting – iteratively – the new ones are encouraged to become experts for the misclassified examples by the previous base learners (complementary expertise)
- Combination method
  - Bagging – equal weighs to all base learners
  - Boosting – different weights - based on performance on training data