Missing values

Machine Learning

25.05.2018

Lecture plan

- Missing data
- Dealing with missing data

Lecture plan

- Missing data
- Dealing with missing data

Missing data

Missing data – some values are not stored in observation

Why it could happen?

- Measurement error
- Object provides no more information
- Data entry errors

Missing data

General fields for missing data:

- Economics
- Sociology
- Political science
- Medicine

Types of missing data

- Completely random – data unbiased, not connected with study, rare case
- Partly random – connected with study, but still random or not informative
- Not random – connected with study, consists some additional information

Lecture plan

- Missing data
- Dealing with missing data

Missing values

From the *Feature Engineering* lecture:

- Forget about the objects with missing values
- Forget about missing values (some algorithms can handle it)
- Try to fill them randomly
- Try to fill them in a clever way (MCMC sampling)
- Use special algorithms processing this type of uncertainty

Ignore samples

$$\begin{pmatrix} a & b & 0.0 & 3.0 & 5.0 & 3.0 \\ \cancel{c} & \cancel{\square} & \cancel{3.0} & \cancel{\square} & \cancel{0.0} & \cancel{1.0} \\ \cancel{c} & \cancel{a} & \cancel{4.0} & \cancel{7.0} & \cancel{\square} & \cancel{4.0} \\ a & b & 1.0 & 4.0 & 7.0 & 8.0 \\ \cancel{c} & \cancel{\square} & \cancel{1.0} & \cancel{0.0} & \cancel{6.0} & \cancel{2.0} \\ \cancel{a} & \cancel{\square} & \cancel{0.0} & \cancel{3.0} & \cancel{\square} & \cancel{5.0} \\ b & c & 0.0 & 6.0 & 0.0 & 7.0 \\ c & b & 5.0 & 2.0 & 2.0 & 5.0 \end{pmatrix} \rightarrow \begin{pmatrix} a & b & 0.0 & 3.0 & 5.0 & 3.0 \\ a & b & 1.0 & 4.0 & 7.0 & 8.0 \\ b & c & 0.0 & 6.0 & 0.0 & 7.0 \\ c & b & 5.0 & 2.0 & 2.0 & 5.0 \end{pmatrix}$$

Ignore values

$$
\begin{pmatrix}
a & b & 0.0 & 3.0 & 5.0 & 3.0 \\
c & \square & 3.0 & \square & 0.0 & 1.0 \\
c & a & 4.0 & 7.0 & \square & 4.0 \\
a & b & 1.0 & 4.0 & 7.0 & 8.0 \\
c & \square & 1.0 & 0.0 & 6.0 & 2.0 \\
a & \square & 0.0 & 3.0 & \square & 5.0 \\
b & c & 0.0 & 6.0 & 0.0 & 7.0 \\
c & b & 5.0 & 2.0 & 2.0 & 5.0
\end{pmatrix}
\longrightarrow
\begin{pmatrix}
a & 0.0 & 3.0 \\
c & 3.0 & 1.0 \\
c & 4.0 & 4.0 \\
a & 1.0 & 8.0 \\
c & 1.0 & 2.0 \\
a & 0.0 & 5.0 \\
b & 0.0 & 7.0 \\
c & 5.0 & 5.0
\end{pmatrix}
$$

Ignore by model

$$\bar{x} = (c, \square, 3.0, \square, 0.0, 1.0)$$

$$\bar{y} = (c, a, 4.0, 7.0, \square, 4.0)$$

$$\mu(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^{n} \frac{\mu(x_i, y_i)^2}{6}}$$

$$\Downarrow$$

$$\mu(\bar{x}, \bar{y}) = \sqrt{\frac{\mu_1(c, c)^2 + \mu_3(3.0, 4.0)^2 + \mu_6(1.0, 4.0)^2}{3}}$$

Ignore by model

$$\begin{pmatrix} a & b & 0.0 & 3.0 & 5.0 & 3.0 \\ c & \square & 3.0 & \square & 0.0 & 1.0 \\ c & a & 4.0 & 7.0 & \square & 4.0 \\ a & b & 1.0 & 4.0 & 7.0 & 8.0 \\ c & \square & 1.0 & 0.0 & 6.0 & 2.0 \\ a & \square & 0.0 & 3.0 & \square & 5.0 \\ b & c & 0.0 & 6.0 & 0.0 & 7.0 \\ c & b & 5.0 & 2.0 & 2.0 & 5.0 \end{pmatrix} \rightarrow \begin{pmatrix} a & b & 0.0 & 3.0 & 5.0 & 3.0 \\ c & b & 3.0 & 3.6 & 0.0 & 1.0 \\ c & a & 4.0 & 7.0 & 3.3 & 4.0 \\ a & b & 1.0 & 4.0 & 7.0 & 8.0 \\ c & b & 1.0 & 0.0 & 6.0 & 2.0 \\ a & b & 0.0 & 3.0 & 3.3 & 5.0 \\ b & c & 0.0 & 6.0 & 0.0 & 7.0 \\ c & b & 5.0 & 2.0 & 2.0 & 5.0 \end{pmatrix}$$

WEKA weka.filters.unsupervised.attribute.
ReplaceMissingValues
Scikit-learn sklearn.preprocessing.Imputer

Recommendation system

$$\begin{array}{cc} & \text{Users} \\ \text{Items} & \begin{pmatrix} 4 & \square & 1 & 2 & \square & 5 \\ 1 & 3 & \square & \square & 3 & 3 \\ \square & 1 & \square & 3 & \square & \square \\ 3 & 4 & 5 & \square & 5 & \square \\ \square & 2 & 3 & \square & 4 & 2 \\ 4 & \square & \square & 2 & \square & \square \\ \square & 3 & 2 & 4 & 2 & 4 \\ 5 & \square & 1 & \square & \square & 1 \end{pmatrix} \end{array}$$

Recommendation via classification

$$\begin{pmatrix} 4 & \square & 1 & 2 & \square & 5 \\ 1 & 3 & \square & \square & 3 & 3 \\ \square & 1 & \square & 3 & \square & \square \\ 3 & 4 & 5 & \square & 5 & \square \\ \square & 2 & 3 & \square & 4 & 2 \\ 4 & \square & \square & 2 & \square & \square \\ \square & 3 & 2 & 4 & 2 & 4 \\ 5 & \square & 1 & \square & \square & 1 \end{pmatrix} \longrightarrow \left(\begin{array}{ccccc|c} 4 & \square & 2 & \square & 5 & 1 \\ 3 & 4 & \square & 5 & \square & 5 \\ \square & 2 & \square & 4 & 2 & 3 \\ \square & 3 & 4 & 2 & 4 & 2 \\ 5 & \square & \square & \square & 1 & 1 \\ \hline 1 & 3 & \square & 3 & 3 & \square \\ \square & 1 & 3 & \square & \square & \square \\ 4 & \square & 2 & \square & \square & \square \end{array}\right)$$

Imputation

Imputation – a process of replacement missing values with substituted values.

- Recommended to use several types of imputation, with at least 20 to 100 of replacements for each

Netflix prize

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

Low-rank

$$R^{[n\times m]} \approx Q^{[n\times k]} \cdot P^{[k\times m]}, k \ll m, n$$

$$R = Q \times P$$

R – Item-user matrix
Q – Item-feature matrix
P – feature-user matrix

Matrix factorization

For all *i,u* where $r_{i,u}$ not missing

$$e_{i,u}^{t} = r_{i,u} - \sum_{f=1}^{k} q_{i,f}^{t} \cdot p_{f,u}^{t}$$

$$p_{f,u}^{t+1} = p_{f,u}^{t} + \gamma^{t} \cdot \sum_{i} \left(e_{i,u}^{t} \cdot q_{i,f}^{t}\right)$$

$$q_{i,f}^{t+1} = q_{i,f}^{t} + \gamma^{t} \cdot \sum_{u} \left(e_{i,u}^{t} \cdot p_{f,u}^{t}\right)$$

EM algorithm

- The expectation E-step Given a set of parameter estimates, such as a mean vector and covariance matrix for a multivariate normal distribution, the E-step calculates the conditional expectation of the complete-data log likelihood given the observed data and the parameter estimates.
- The maximization M-step Given a complete-data log likelihood, the M-step finds the parameter estimates to maximize the complete-data log likelihood from the E-step.

The imputation I-step

Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. That is, if you denote the variables with missing values for observation *i* by $Y_{i(mis)}$ and the variables with observed values by $Y_{i(obs)}$, then the I-step draws values for $Y_{i(mis)}$ from a conditional distribution $Y_{i(mis)}$ for given $Y_{i(obs)}$.

The posterior P-step

Given a complete sample, the P-step simulates the posterior population mean vector and covariance matrix. These new estimates are then used in the next I-step. Without prior information about the parameters, a noninformative prior is used. You can also use other informative priors. For example, a prior information about the covariance matrix can help to stabilize the inference about the mean vector for a near singular covariance matrix.