Feature Engineering

Wenting Tu

SHUFE, SIME

Machine Learning and Deep Lerning

Outline

What is Feature Engineering

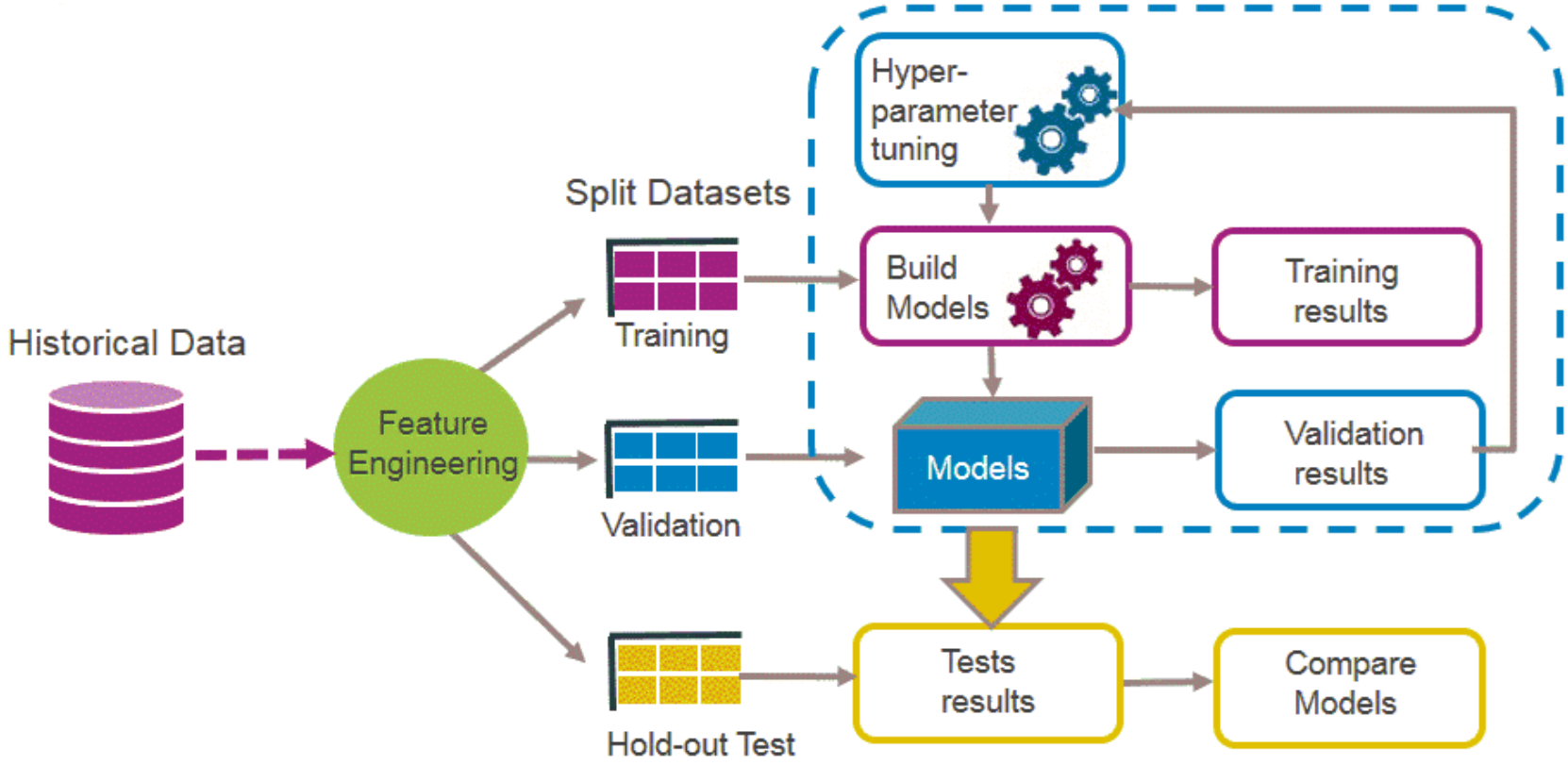
Feature Selection

Feature Extraction

Definition

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

- Wikipedia



Scope

- Create original features
  - Images → colours, textures, contours, ...
  - Signals → frequency, phase, samples, spectrum, ...
  - Time series → ticks, trends, self-similarities, ...
  - Biomed → dna sequence, genes, ...
  - Text → words, POS tags, grammatical dependencies, ...
- Feature Processing
  - Feature binarisation/discretization, ...
  - Feature normalisation, ...
  - Feature transformation (e.g., map features into high-dimensional space), ...
- Feature Selection and Extraction

Outline

What is Feature Engineering

Feature Selection

Feature Extraction

Feature Selection

- Goal

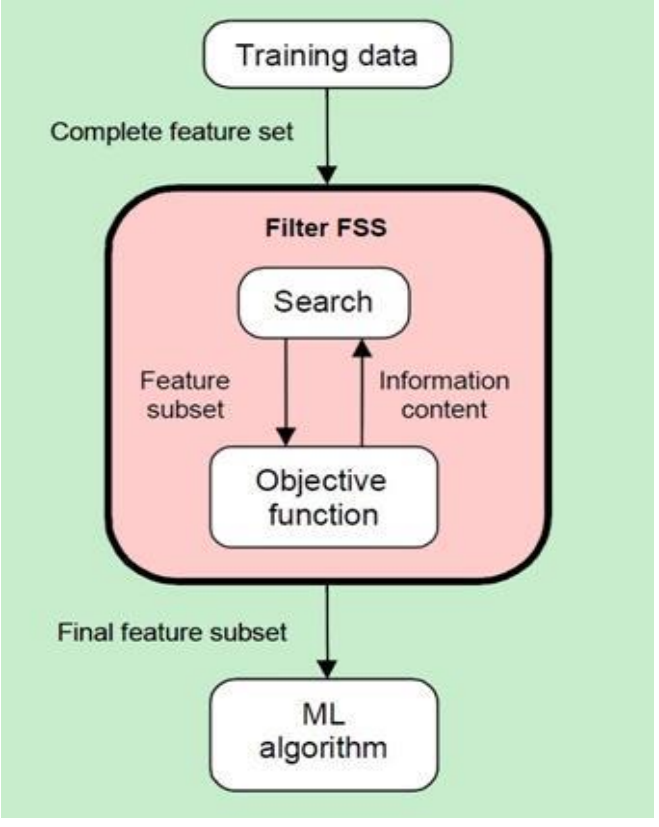
  Selecting the subset of all features without redundant or irrelevant feature

- Types
  - Filter approaches, ...
  - Wrapper approaches, ...
  - Embedded approaches, ...

Feature Selection

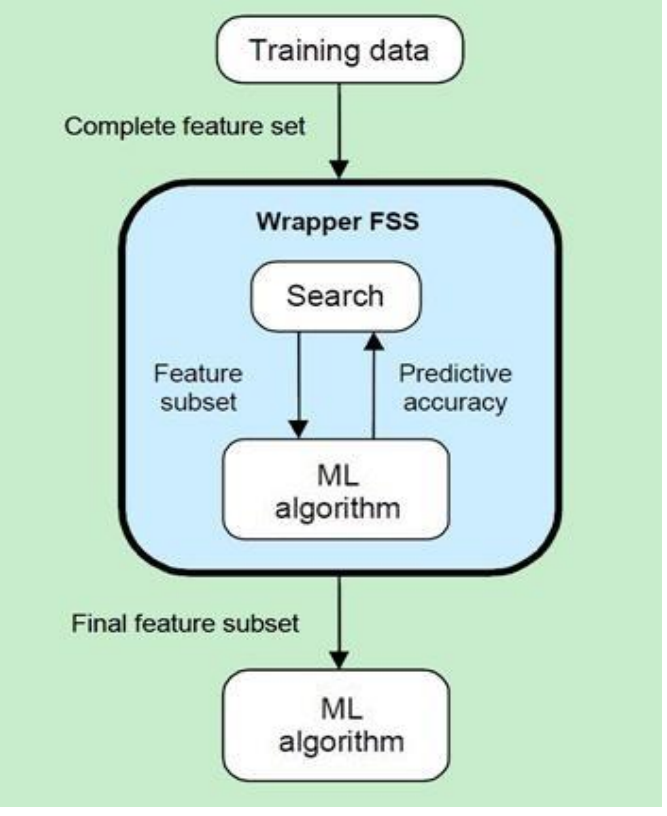
- Filter approaches
  - Select features by
    1 Computing the scores for each feature
    2 Ranking the features based on scores
    3 Select the top-k features or features with high scores
  - Example of feature score: information gain

$$IG(Y|X) = H(Y) - H(Y|X)$$



Feature Selection

- Wrapper approaches
  - Search through the space of all possible feature subsets
  - Each search subset is tried out with a learning algorithm
  - Typical steps:
    1 Initial subset selection
    2 Try a subset with a learner
    3 **Modify** the feature subset (e.g., forward selection or backward elimination)
    4 Rerun the learner
    5 Measure the difference
    6 GOTO 2

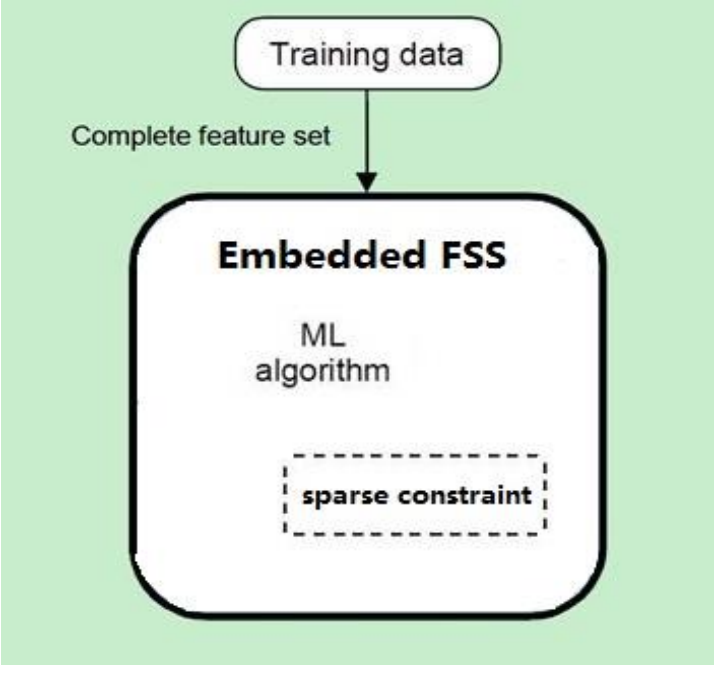


Feature Selection

- Embedded approaches

  Similar to the wrapper approach in the sense that the features are specifically selected for a certain learning algorithm, but in this approach, the features are selected during the learning process.

  e.g., LASSO, decision trees



Outline

What is Feature Engineering

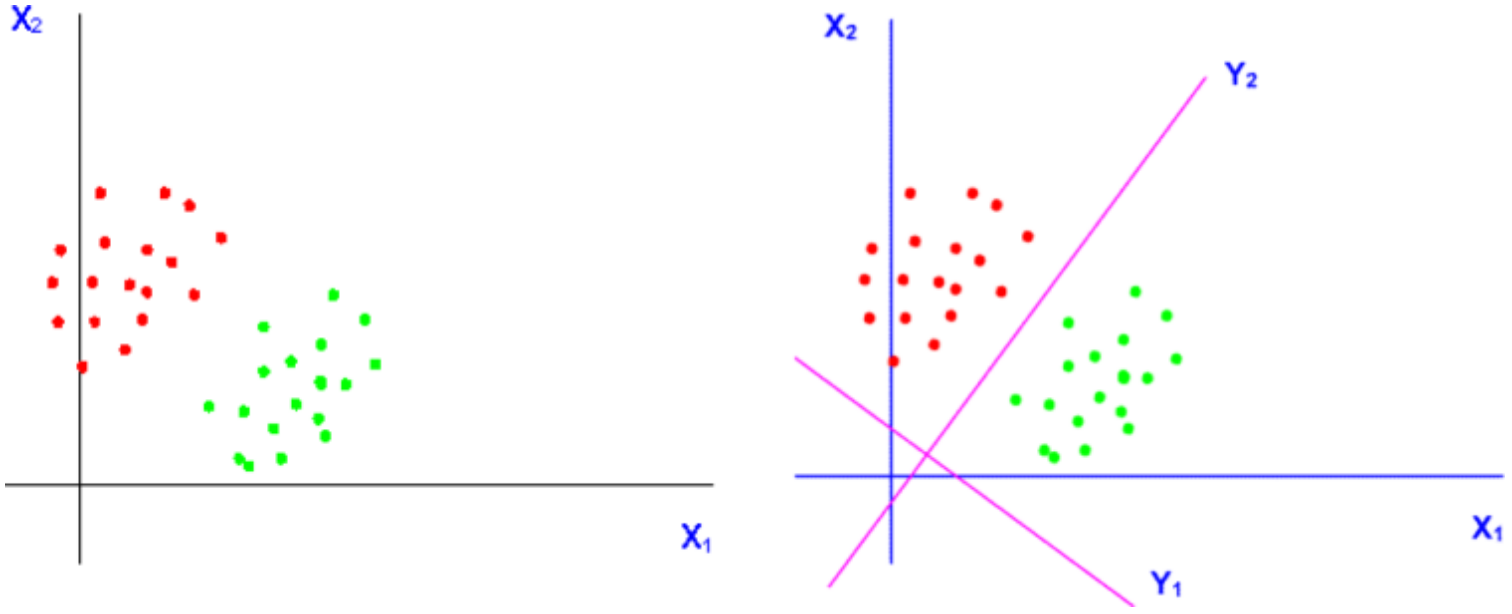
Feature Selection

Feature Extraction

Feature Extraction

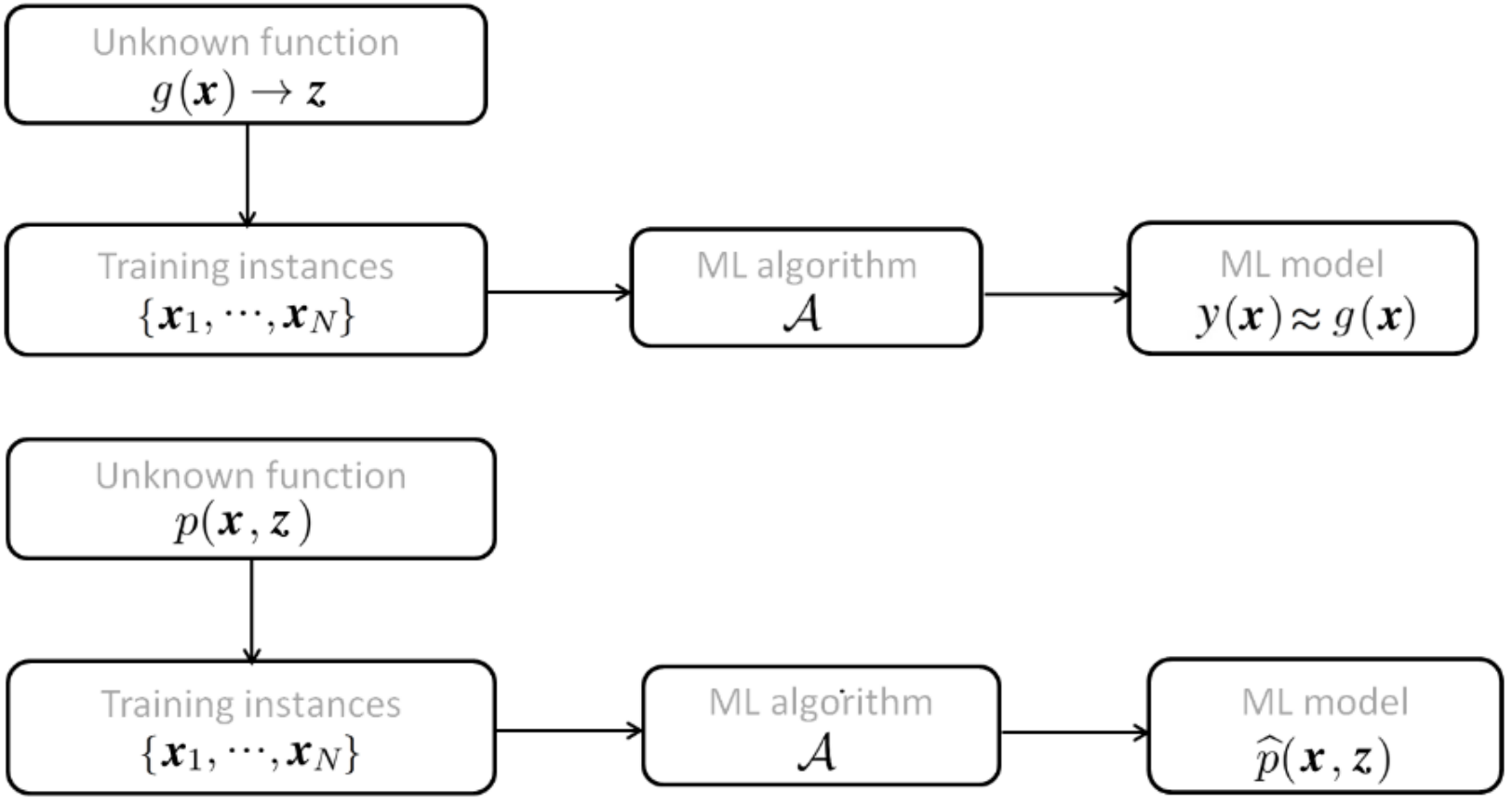
- Goal

  Feature extraction aims to transform the input data into a set of features which can very well represent the input data.



Feature Extraction

- Unsupervised feature extraction



Principal Component Analysis (PCA)

- Maximum variance formulation

$$z_1 = y(\boldsymbol{x}, \boldsymbol{u}_1) = \boldsymbol{u}_1^T \boldsymbol{x} \quad \boldsymbol{u}_1^* = \arg\max_{\boldsymbol{u}_1} \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 \text{ s.t. } \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$$

$$\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{x}_n - \bar{\boldsymbol{x}}) (\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T, \ \bar{\boldsymbol{x}} = \frac{1}{N} \sum_{n=1} \boldsymbol{x}_n$$

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 = \frac{1}{N} \sum_{n=1}^{N} \left\{ \boldsymbol{u}_1^T \boldsymbol{x}_n - \boldsymbol{u}_1^T \bar{\boldsymbol{x}} \right\}^2$$

Principal Component Analysis (PCA)

- Maximum variance formulation

$$z_1 = y(\boldsymbol{x}, \boldsymbol{u}_1) = \boldsymbol{u}_1^T \boldsymbol{x} \quad \boldsymbol{u}_1^* = \arg\max_{\boldsymbol{u}_1} \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 \text{ s.t. } \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$$

$$\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{x}_n - \bar{\boldsymbol{x}}) (\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T, \ \bar{\boldsymbol{x}} = \frac{1}{N} \sum_{n=1} \boldsymbol{x}_n$$

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 = \frac{1}{N} \sum_{n=1}^{N} \left\{ \boldsymbol{u}_1^T \boldsymbol{x}_n - \boldsymbol{u}_1^T \bar{\boldsymbol{x}} \right\}^2$$

$$\boldsymbol{u}_1^* = \arg\max_{\boldsymbol{u}_1} \left\{ \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 + \lambda_1 \left(1 - \boldsymbol{u}_1^T \boldsymbol{u}_1 \right) \right\}$$

$$\boldsymbol{S} \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1$$

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 = \lambda_1$$

$\boldsymbol{u}_1^*$ = the eigenvector of the data covariance matrix $\boldsymbol{S}$ having the largest eigenvalue λ_1

Principal Component Analysis (PCA)

- Maximum variance formulation

$$y(\boldsymbol{x}, \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M\}) = \left(\boldsymbol{u}_1^T\boldsymbol{x}, \cdots, \boldsymbol{u}_M^T\boldsymbol{x}\right)^T$$

$$\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M\}^*$$

= the M eigenvectors of the data covariance matrix $\boldsymbol{S}$ corresponding to the M largest eigenvalues $\lambda_1, \ldots, \lambda_M$

(easily shown using proof by induction)

Principal Component Analysis (PCA)

- Minimum-error formulation

$\{\boldsymbol{u}_i\}_{i=1}^{D}$ is a complete orthonormal set of D-dimensional basis vectors

$$\boldsymbol{x}_n = \sum_{i=1}^{D} \alpha_{ni}\boldsymbol{u}_i \quad \alpha_{nj} = \boldsymbol{x}_n^T\boldsymbol{u}_j \quad \boldsymbol{x}_n = \sum_{i=1}^{D} \left(\boldsymbol{x}_n^T\boldsymbol{u}_i\right)\boldsymbol{u}_i$$

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i + \sum_{i=M+1}^{D} b_i\boldsymbol{u}_i \quad \{x_{n1},\ldots,x_{nD}\} \longrightarrow \{z_{n1},\ldots,z_{nM}\}$$

$$z_{nj}^* = \arg\min_{z_{ni}} J, \quad J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

$$z_{nj}^* = \boldsymbol{x}_n^T\boldsymbol{u}_j, b_j^* = \bar{\boldsymbol{x}}^T\boldsymbol{u}_j \longrightarrow \boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} \left\{\left(\boldsymbol{x}_n - \bar{\boldsymbol{x}}\right)^T\boldsymbol{u}_i\right\}\boldsymbol{u}_i$$

Principal Component Analysis (PCA)

- Minimum-error formulation

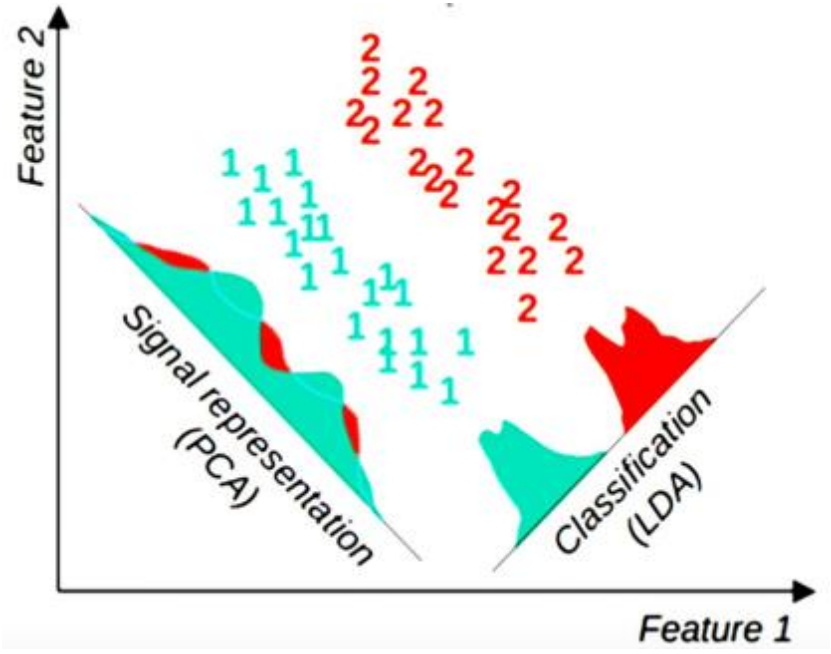
$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} \left(\boldsymbol{x}_n^T \boldsymbol{u}_i - \bar{\boldsymbol{x}}^T \boldsymbol{u}_i \right)^2 = \sum_{i=M+1}^{D} \boldsymbol{u}_i^T \boldsymbol{S} \boldsymbol{u}_i$$

$$\{\boldsymbol{u}_{M+1}, \ldots, \boldsymbol{u}_D\}^* = \arg\min J$$

$$\{\boldsymbol{u}_{M+1}, \ldots, \boldsymbol{u}_D\}^* = \text{the } D - M \text{ eigenvectors of the data covariance matrix } \boldsymbol{S} \text{ corresponding to the } D - M \text{ smallest eigencalues}$$

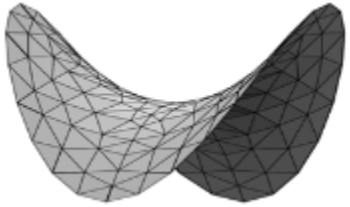
Linear Discriminant Analysis (LDA)

- Both Linear Discriminant Analysis (LDA) and PCA are linear transformation techniques that are commonly used for dimensionality reduction.
- In contrast to PCA, LDA is "supervised" and computes the directions ("linear discriminants") that will represent the axes that that maximize the separation between multiple classes.

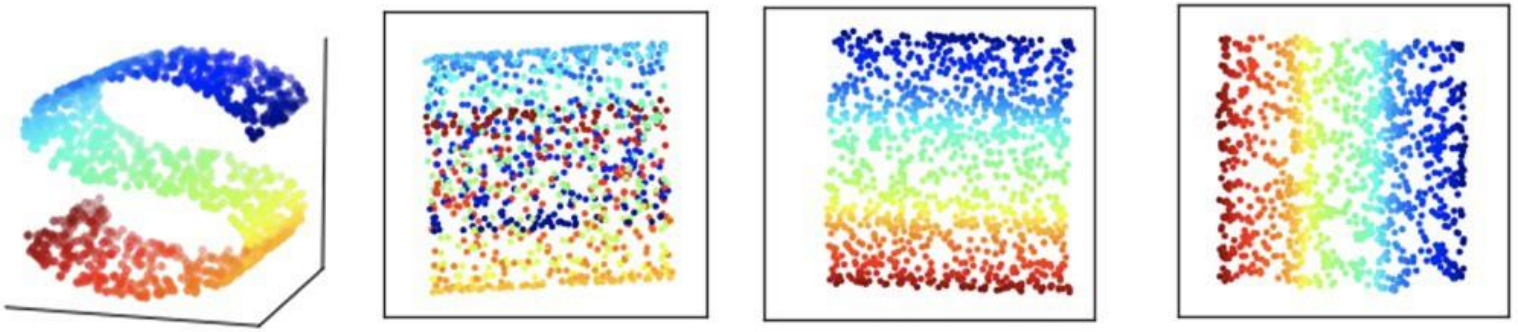


Manifold Learning

- Manifold learning aims to uncovers the intrinsic dimensionality



E.g.
Isomap infers a distance matrix using distances along the manifold
LLE finds a mapping to preserve local linear relationships between neighbors



Text Feature Extraction

- Word Space Model

$$D = \{d_1, d_2, \cdots, d_n\} \xrightarrow{W=\{w_1, w_2, \cdots, w_m\}} X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

  - Ilustration

Doc1: Text mining is to identify useful information.

Doc2: Useful information is mined from text.

Doc3: Apple is delicious.

	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

Text Feature Extraction

- Word Space Model

$$D = \{d_1, d_2, \cdots, d_n\} \xrightarrow{W=\{w_1, w_2, \cdots, w_m\}} X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

- Topic Space Model

$$W = \{w_1, w_2, \cdots, w_m\}$$
$$\downarrow$$
$$D = \{d_1, d_2, \cdots, d_n\} \xrightarrow{T = \{t_1, t_2 \cdots t_k\}} Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{bmatrix}$$

Topic Space Model

- Latent Semantic Analysis (LSA)
  - Illustration

1. The Neatest Little Guide to Stock Market Investing
2. Investing For Dummies, 4th Edition
3. The Little Book of Common Sense Investing: The Only Way to Guarantee Your Fair Share of Stock Market Returns
4. The Little Book of Value Investing
5. Value Investing: From Graham to Buffett and Beyond
6. Rich Dad's Guide to Investing: What the Rich Invest in, That the Poor and the Middle Class Do Not!
7. Investing in Real Estate, 5th Edition
8. Stock Investing For Dummies
9. Rich Dad's Advisors: The ABC's of Real Estate Investing: The Secrets of Finding Hidden Profits Most Investors Miss

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

U_k

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

Singular Value Decomposition (SVD)

Σ_k

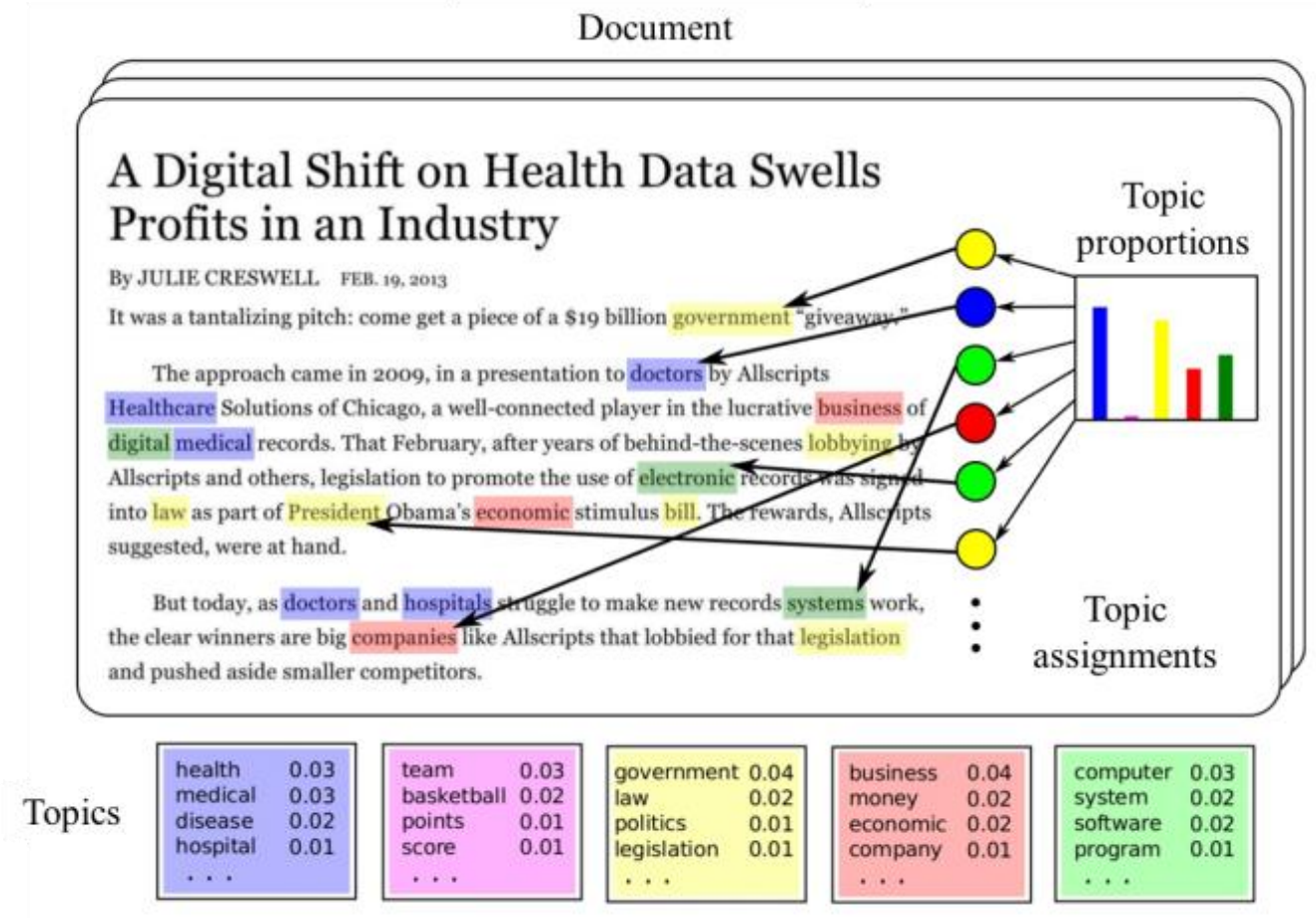
3.91	0	0
0	2.61	0
0	0	2.00

V_k^T

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

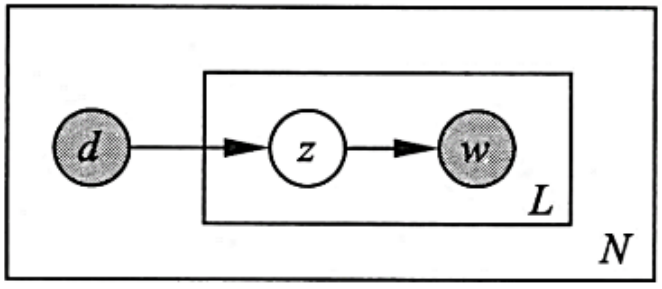
Probabilistic Topic Models

- Topic as a probabilistic distribution over words.
- Document as a mixture of topics.

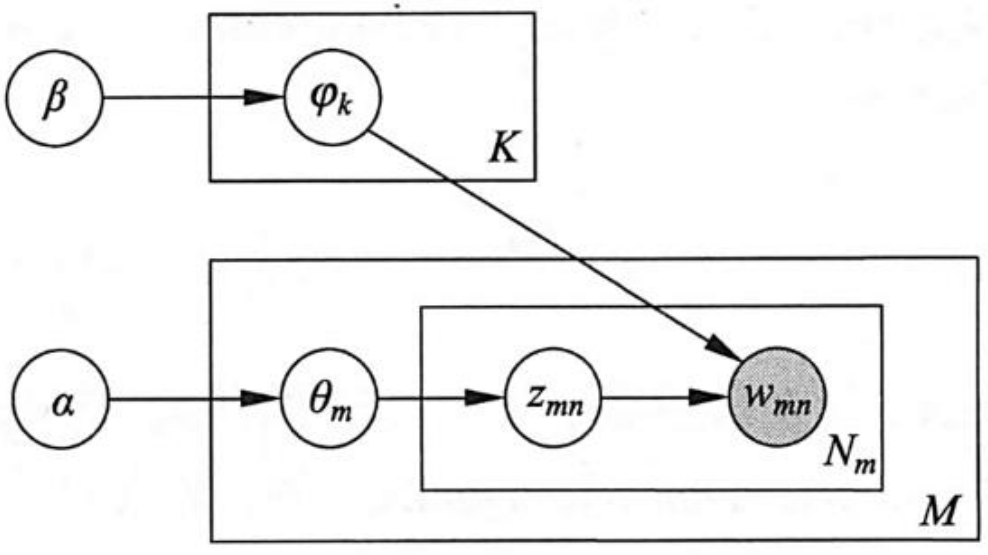


Probabilistic Topic Models

- Probabilistic Latent Semantic Analysis (PLSA)



- Latent Dirichlet Allocation (LDA)

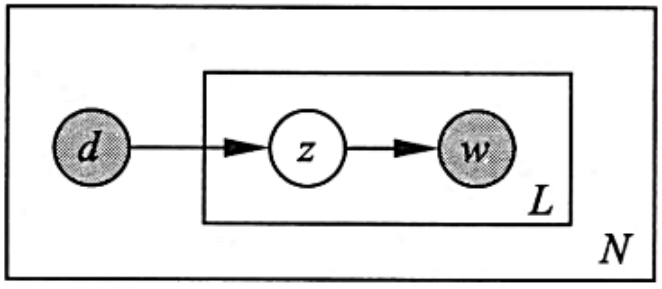


References:

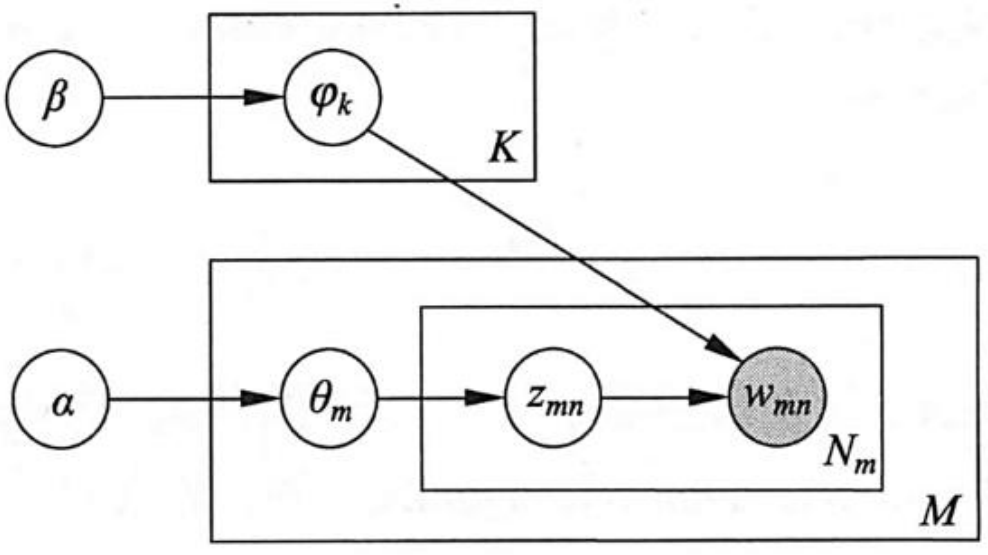
《统计学习方法（第2版）》(chapters of *潜在语义分析* & *概率潜在语义分析* & *潜在狄利克雷分析*)

Probabilistic Topic Models

- Probabilistic Latent Semantic Analysis (PLSA)



- Latent Dirichlet Allocation (LDA)



Thanks

Some images and slides are from the internet.
If related to copyright, please contact me.

tu.wenting@mail.shufe.edu.cn