BMEG 3105 Fall 2022

Data analytics for personalized genomics and precision medicine
Dimension reduction

Lecturer: Yu LI (李煜) from CSE

Liyu95.com, liyu@cse.cuhk.edu.hk

Wednesday, 12 October 2022

- **Expected outcomes:**

1. Dimension reduction
2. Further arrangement
3. Neural networks

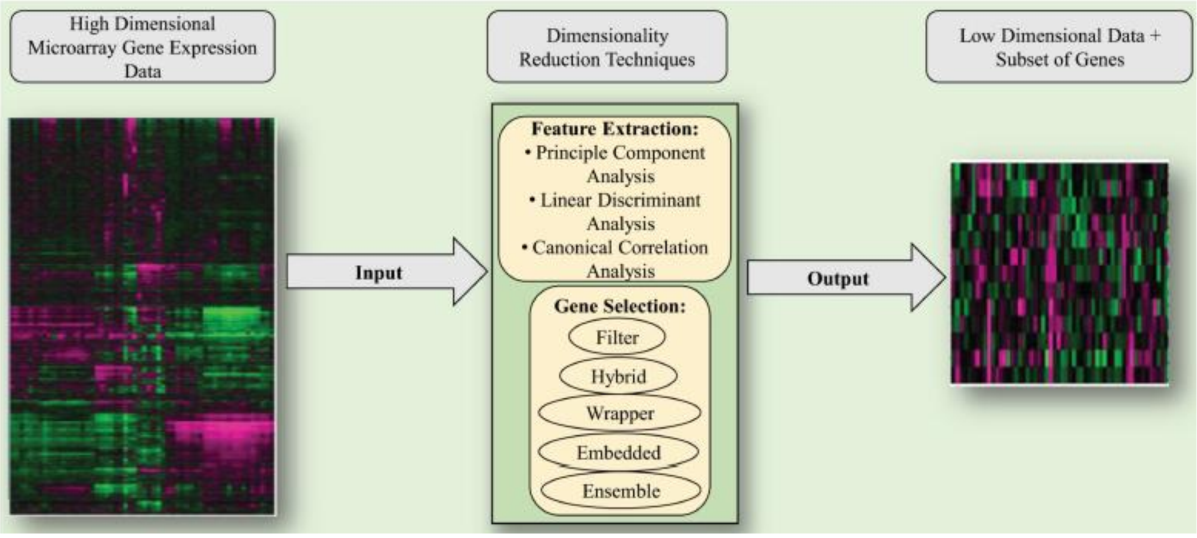
1. Dimension reduction:

Reasons to use feature selection and dimension reduction

a. Biodata can be huge, noise, unrelated, and duplicated

- Irrelevant genes and highly correlated genes cannot be included in data analysis
- Pathways are formed when we combine some genes together into one value which is much useful
- Use feature selection and dimension reduction to solve the above problems

b. Flow diagram of performing the feature selection and dimension reduction


High Dimensional Microarray Gene Expression Data
Input
Dimensionality Reduction Techniques
Feature Extraction:
• Principle Component Analysis
• Linear Discriminant Analysis
• Canonical Correlation Analysis
Gene Selection:
Filter
Hybrid
Wrapper
Embedded
Ensemble
Output
Low Dimensional Data + Subset of Genes

c. Benefits of feature selection and dimension reduction

- Data compression: efficient storage & retrieval
- Improve prediction performance: remove unrelated performance
- Understand the prediction results: to know what genes are related to cancer prediction
- Facilitate data visualization: understand the distance between cells visually

Feature selection

a. Steps

- Choose the best subset genes from all the genes
- Feature ranking
- Feature subset selection: Filter and Wrapper

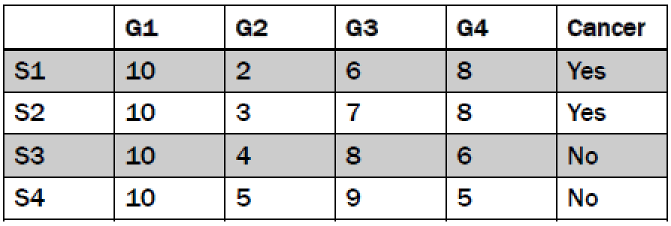
b. Best subset

i. Filter

- Classification performance is not involved in the selection loop
- Variance thresholds: Features with a higher variance contain more useful information e.g., Age, Height
- Information gain: Features should be different

ii. Wrapper

- Using the classification performance to guide selection
- Computational expensive
- Recursive feature elimination
- Sequential feature selection



	G1	G2	G3	G4	Cancer
S1	10	2	6	8	Yes
S2	10	3	7	8	Yes
S3	10	4	8	6	No
S4	10	5	9	5	No

Ways to choose wrapper

1. No feature

2. Find the first best feature using cross-fold validation

3. Add the second feature using crossfold Validation

4. …

5. Until the new feature does not improve the performance

Feature extraction

a. Steps

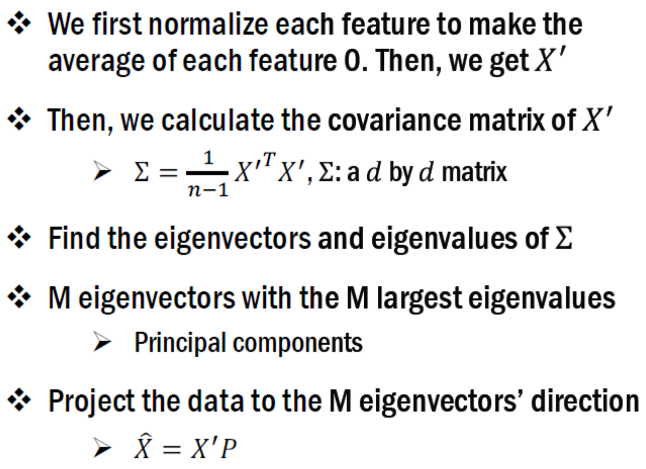
- Extract new features by linear or non-linear combination of the original features e.g., New feature = Gene 1 + Gene 2
- New features may not have physical interpretation/meaning (usually for non-linear)
- PCA, SVD, Isomap, LLE, CCA, et. al.

b. Principal components analysis (PCA)

i. Concepts

- A two-dimensional scatter of points that show a high degree of correlation

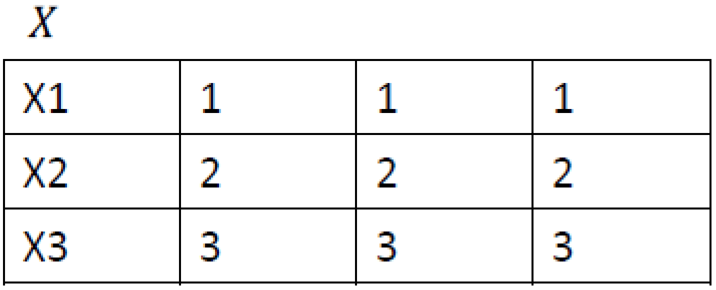
ii. Steps



- ❖ We first normalize each feature to make the average of each feature 0. Then, we get X'
- ❖ Then, we calculate the covariance matrix of X'
  - ➢ $\Sigma = \frac{1}{n-1}X'^{T}X'$, Σ: a d by d matrix
- ❖ Find the eigenvectors and eigenvalues of Σ
- ❖ M eigenvectors with the M largest eigenvalues
  - ➢ Principal components
- ❖ Project the data to the M eigenvectors' direction
  - ➢ $\hat{X} = X'P$

Example:

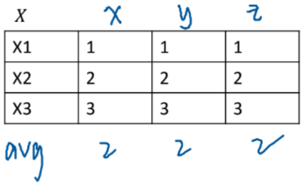
1. Original matrix



X

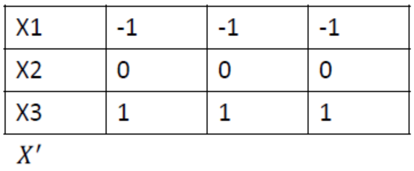
X1	1	1	1
X2	2	2	2
X3	3	3	3

2. Normalization



X	x	y	z
X1	1	1	1
X2	2	2	2
X3	3	3	3
avg	2	2	2

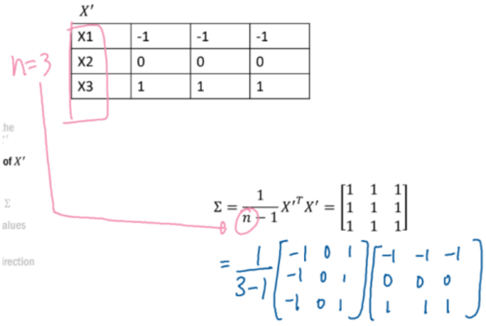
step2	x	y	z
X_1	1-2=-1	1-2=-1	1-2=-1
X_2	2-2=0	2-2=0	2-2=0
X_3	3-2=1	3-2=1	3-2=1



X1	-1	-1	-1
X2	0	0	0
X3	1	1	1

X'

3. Calculate the covariance



X'

X1	-1	-1	-1
X2	0	0	0
X3	1	1	1

n=3

$\Sigma = \frac{1}{n-1} X'^T X' = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$= \frac{1}{3-1} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

$$\Sigma = \frac{1}{n-1} X'^T X' = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4. Calculate eigenvalue and eigenvector

[eigenvalue] [eigenvector]

$$\Sigma * V = \lambda * V$$

$$|\Sigma - \lambda I| = 0$$

$\lambda_1 = 3$ $V_1 = \begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix}$ $\lambda_{2,3} = 0$ $V_{2,3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$$\begin{vmatrix} 1-\lambda & 1 & 1 \\ 1 & 1-\lambda & 1 \\ 1 & 1 & 1-\lambda \end{vmatrix} = 0$$

$$(1-\lambda)^3 + 1 + 1 - (1-\lambda) \\ - (1-\lambda) - (1-\lambda) = 0$$

$$\lambda = 3 \; or \; \lambda = 0$$

5. Project

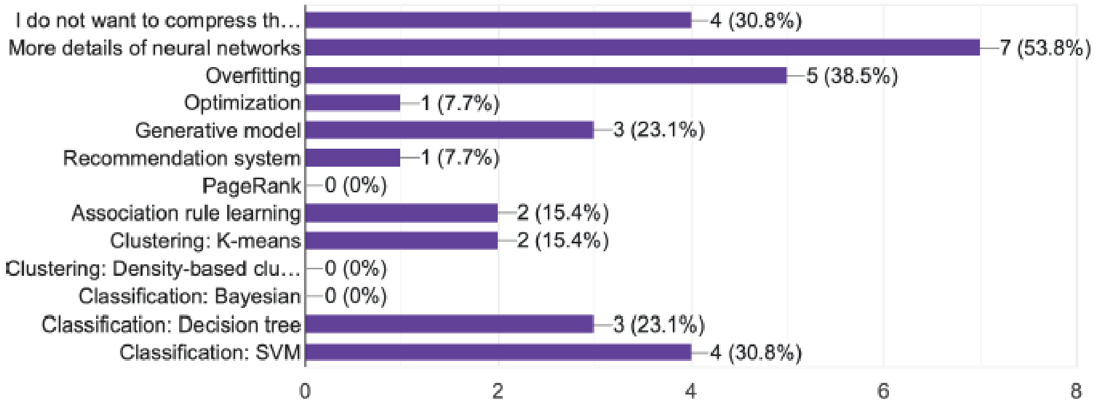
$$P = \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \end{bmatrix}$$

$$\hat{X} = X'P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \end{bmatrix} = \begin{bmatrix} -\sqrt{3} & 0 \\ 0 & 0 \\ \sqrt{3} & 0 \end{bmatrix}$$

X1	$-\sqrt{3}$	0
X2	0	0
X3	$\sqrt{3}$	0

$\hat{X}$

2. Further arrangement:



3. Neural networks:

Logistic regression (LR)

a. Steps

i. Logistic function

$$\frac{1}{1+e^{-(w_h H+w_w W+w_0)}} \geq 0.5$$

ii. Training

To get w_h and w_w, and w_0

iii. Testing

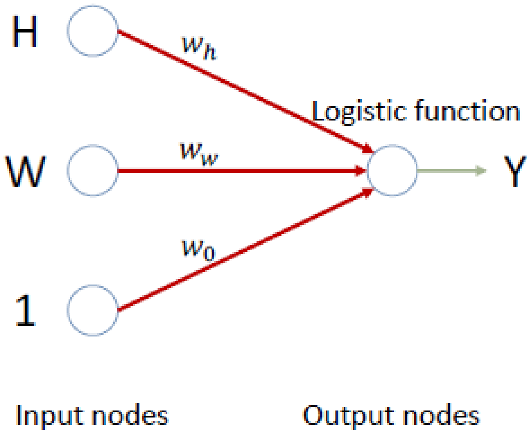
- run the formula

b. Problems as classification

- the relationship among different variables within the image may be much more complicated than simple linear combination
- The model capacity is not enough
- Underfitting

c. LR as a neural network

i. Flow diagram



$$y_{output} = \frac{1}{1+e^{-(w_h H+w_w W+w_0)}}$$

ii. Problems

- The relation between the output and input may be nonlinear
- The relation between the output and input can be very complex

iii. Solutions to solve the problems

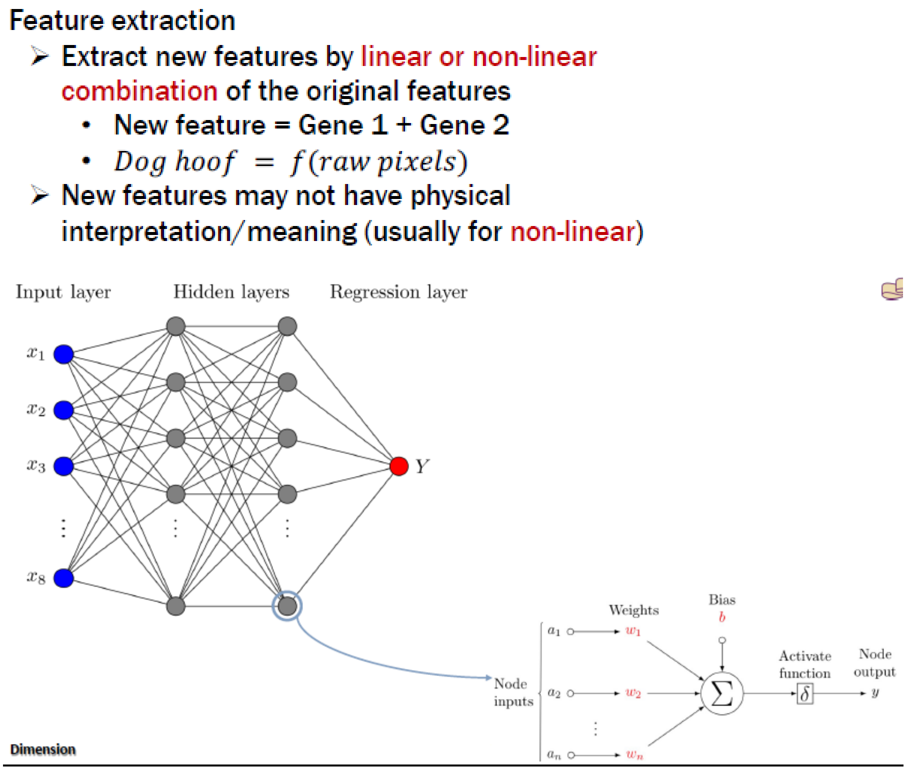
- Increase the number of nodes
- Increase the number of layers
- Add non-linear function
- Change LR to deep neural networks

d. LR as deep neural networks

i. Concepts

- Fully connected layers
- A general function approximator
- We can approximate any function (relation) if we have enough nodes and layers
- Universal approximation theorem
- The function is much more complicated, and the number of parameters is very large
- We may use it resolve complex problems with a huge amount of data

ii. Visualization of the internal nodes

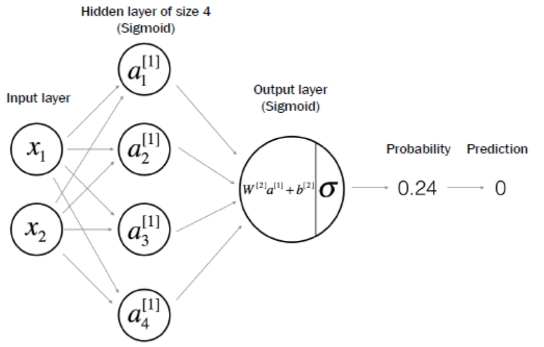


Feature extraction

- ➢ Extract new features by linear or non-linear combination of the original features
  - New feature = Gene 1 + Gene 2
  - $Dog\ hoof\ =\ f(raw\ pixels)$
- ➢ New features may not have physical interpretation/meaning (usually for non-linear)

iii. Hidden layer

- Calculation of each internal node



$$a_1 = \frac{1}{1+e^{-(w_{11}*x_1+w_{21}*x_2+b_1)}}$$

$$a_2 = \frac{1}{1+e^{-(w_{12}*x_1+w_{22}*x_2+b_2)}}$$

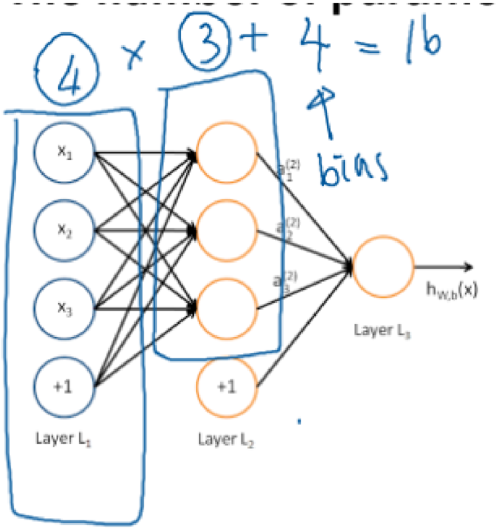
$$a_3 = \frac{1}{1+e^{-(w_{13}*x_1+w_{23}*x_2+b_3)}}$$

$$a_4 = \frac{1}{1+e^{-(w_{14}*x_1+w_{24}*x_2+b_4)}}$$

$$Y = \frac{1}{1+e^{-(w_1*a_1+w_2*a_2+w_3*a_3+w_4*a_4+b)}}$$

- Number of hidden layers

  Product of the numbers of each layer + numbers of bias



Parameters: 4*3+4 = 16

e. FS and DR in Python

i. Tools: Scikit-learn