BMEG3105 **Fall 2025**

Data analytics for personalized genomics and precision medicine
Lecture 9: Classification

October 3, 2025

Lecturer: Prof. LI Yu *Scribe: LI Haoyuan*

9.1 Recap from last lecture

9.1.1 Mahalanobis distance

Mahalanobis distance calculates distance considering the data distribution. The data distribution along a certain direction will affect the actual distance between two points:

- Sparse distribution: Even if the length between two points is the same, the actual distance will decrease.
- Dense distribution: Even if the length between two points is the same, the actual distance will increase.

Here is the formula to calculate the mahalanobis distance:

$$\text{mahalanobis}(p, q) = (p - q)^T \Sigma^{-1} (p - q)$$

where Σ is the covariance matrix.

9.1.2 Hierarchical clustering

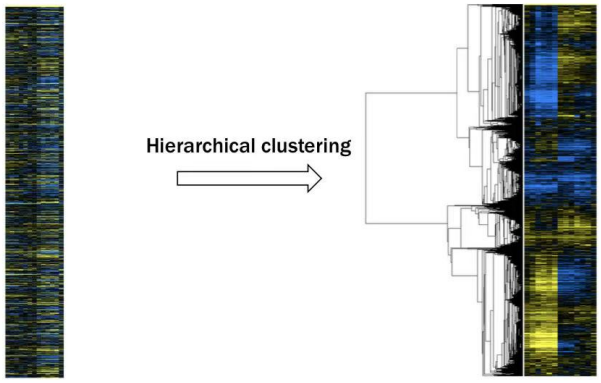


Figure 9.1: Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram (a tree-like diagram that records the sequences of merges).
- They may correspond to meaningful taxonomies, e.g., gene clusters, phylogeny reconstruction, animal kingdom classification, etc.

Steps of hierarchical clustering:

1. Compute the Similarity or Distance matrix.
2. Let each data point be a cluster.
3. Merge the two closest clusters.
4. Update the similarity or distance matrix.
5. Repeat steps 3&4 until only a single cluster remains.

Ways to update the distance matrix after merging: min, max, group average, distance between centroids, etc.

A running example is as follows:

Gene	**wt**	**mutant_1**	**mutant_2**	**mutant_3**
At4g35770	1.5	3	3	1.5
At1g30720	4	7.5	7.5	5
At4g27450	1.5	1	1	1.5
At2g34930	10	25	23	15
At2g05540	1	1	2	1

Table 9.1: Gene Expression Data Matrix

We use the Pearson correlation coefficient, which is represented by ρ, the formula for ρ is

$$\rho_{X,Y} = \text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

where:

- $\text{cov}(X,Y) = E[(X-\mu_X)(Y-\mu_Y)]$ is the covariance between X and Y
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y
- E denotes the expected value
- μ_X is the mean of X
- μ_Y is the mean of Y

Using Pearson's correlation coefficient, we have the following distance matrix:

	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770	1				
At1g30720	0.9733	1			
At4g27450	-1	-0.9733	1		
At2g34930	0.9493	0.9909	-0.9493	1	
At2g05540	0.5774	0.562	-0.5774	0.4528	1

Table 9.2: Distance Matrix

- Let each gene be a cluster.
- Then merge the two closest clusters, that is, At2g34930 and At1g30720.
- Update with minimum distance (largest correlation).

	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770					
At1g30720	0.9733				
At4g27450	-1	-0.9733 ->-0.9493			
At2g34930	0.9493 ->0.9733		-0.9493		
At2g05540	0.5774	0.562	-0.5774	0.4528 ->0.562	

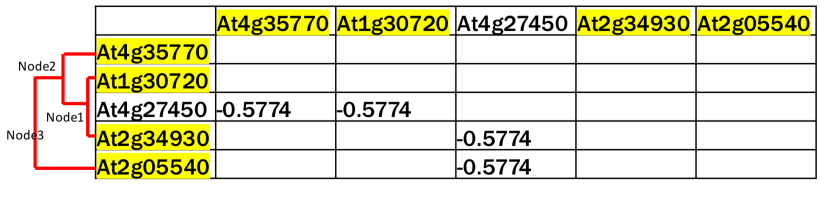
Table 9.3: Merge At2g34930 and At1g30720 and Update Distance Matrix

- Merge the two closest clusters, that is, At2g34930, At1g30720 and At4g35770.
- Update with minimum distance (largest correlation).

	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770					
At1g30720					
At4g27450	-1 ->-0.9493	-0.9493			
At2g34930			-0.9493		
At2g05540	0.5774	0.562 ->0.5774	-0.5774	0.562 ->0.5774	

Table 9.4: Merge At2g34930, At1g30720 and At4g35770 and Update Distance Matrix

- Merge last two clusters, that is, At2g34930, At1g30720, At4g35770, and At2g05540



	At4g35770	At1g30720	At4g27450	At2g34930	At2g05540
At4g35770					
At1g30720					
At4g27450	-0.5774	-0.5774			
At2g34930			-0.5774		
At2g05540			-0.5774		

Table 9.5: Merge At2g34930, At1g30720, At4g35770, and At2g05540

Hierarchical clustering in Python:

- Related link: `https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram`

9.2 Classification

9.2.1 Why Classification?

- Characteristics of each class
- Classify items
  - Better organization
  - To justify which class the new items belongs to
- Classify people
  - Patients: different treatment for different groups
    - Children, older
  - Customers:
    - To justify whether a person is within the targeting group

Why classification in biology?

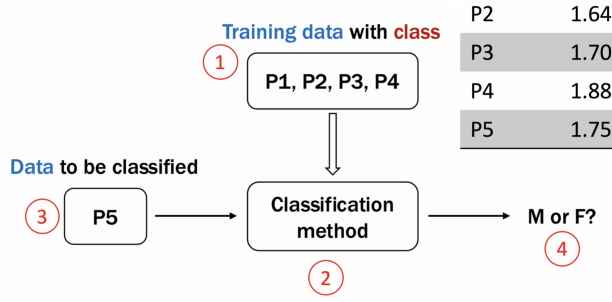
- Given a new gene expression profile, to justify it is normal or tumor

9.2.2 What is classification?

- Given a collection of records **(training set)**, each record contains a set of **attributes**. One of the attributes is the **class**.
- Find a method to **assign the class** of **previously unseen records** based on their other **attributes** and the **training set** as accurately as possible.

9.2.3 How to do classification?

1. Preparing training data with class
2. Use training data to construct a classification method
3. Apply the classification method onto new data
4. Obtain classification prediction



Person	Height(m)	Weight(kg)	Gender
P1	1.79	75	M
P2	1.64	54	F
P3	1.70	63	M
P4	1.88	78	M
P5	1.75	70	??

Figure 9.2: An Example to Explain the General Procedure of Classification

Most important things for classification method:

- Training data with class
- Classification method
- Data to be classified

9.3 K-nearest neighbor classification

KNN is a simple algorithm that **stores all available instances** and classifies new instances based on a **distance metric** to the available ones.

A metaphor for the basic idea: If it walks like a duck, quacks like a duck, then it is probably a duck.

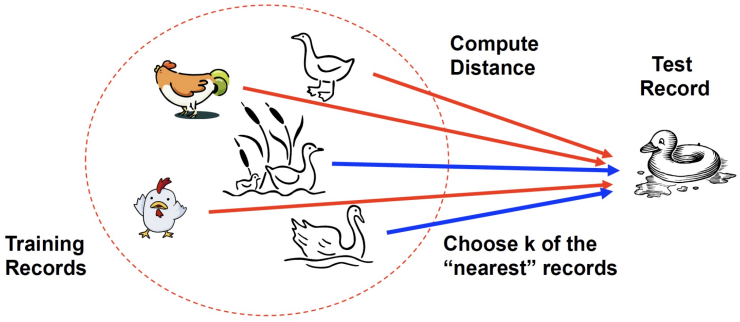


Figure 9.3: A Metaphor for the Basic Idea

9.3.1 Key Points of KNN

- Training process:
  - Store the available training instances
- Predicting process:
  - Find the K training instances that are closest to the query instance
  - Return the most frequent class label among those K instances
- Note: Data should be normalized.

9.3.2 What Should We Determine When Using KNN?

- A distance metric
- The number of neighbors to look at, that is, K
- A weighing function (optional)

9.3.3 Ways to Choose K

- In practice, using a value of K somewhere between **5 and 10** gives good results for **the most low-dimensional data sets**
- A good K can also be chosen by using **cross-validation**
  - To assess how the results will generalize to an independent data set. (Further discuss in lecture 11)

9.3.4 Factors Affecting the Results of the KNN Algorithm for a Specific Testing Data Point

- The choice of the distance measurement
- The choice of K
- Data normalization
- Training data size
- Training data class

9.3.5 The standard procedure of KNN

Suppose we have chosen the distance metric and K

- Normalization
- Compute distances
- Identify the K most similar data
- Take their class out and find the mode class

9.3.6 A running example of KNN

Suppose we have chosen the Euclidean distance and K=2

Person	Height(m)	Weight(kg)	Gender
P1	1.79	75	M
P2	1.64	54	F
P3	1.70	63	M
P4	1.88	78	M
P5	1.75	70	??

Table 9.6: Origin Data Matrix of the Running Example

- Min-max normalization

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

Table 9.7: Normalized Data Matrix

- Compute Euclidean distance and identify the 2 most similar data

Person	P5	Gender
P1	0.267	M
P2	0.809	F
P3	0.358	M
P4	0.636	M
P5	0	??

Table 9.8: Euclidean Distances Between P5 and Each Data and Highlight the 2 Most Similar Data

- Take their class out and find the mode class
  - Since the genders in 2 most similar data are both M, the prediction of the gender of P5 is **M**.

KNN in Python:

- Related link: `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

9.4 Clustering VS Classification

9.4.1 Comparing Clustering and Classification

Clustering

- Data to be clustered
- Similarity measurement
- Clustering algorithm (the executive procedure)

Classification

- Training data with class
- Classification method
- Data to be classified

	Clustering	Classification
Goal	Find similarity (clusters) in the data	Assign class to the new data
Data	Data without class	Training data with class and testing data without class
Classes	Unknown number of classes	Known number of classes
Output	The cluster index for each point	The class assignment of the testing data
Algorithm	One phase	Two phases (training and application)

Table 9.9: Comparing Clustering and Classification

9.4.2 Unsupervised learning and supervised learning

Unsupervised learning

- Machine learning algorithms to analyze and cluster unlabeled data
- Example: clustering and dimension reduction

Supervised learning

- Machine learning algorithms to classify and predict outcomes, trained on labeled data
- Example: classification and regression

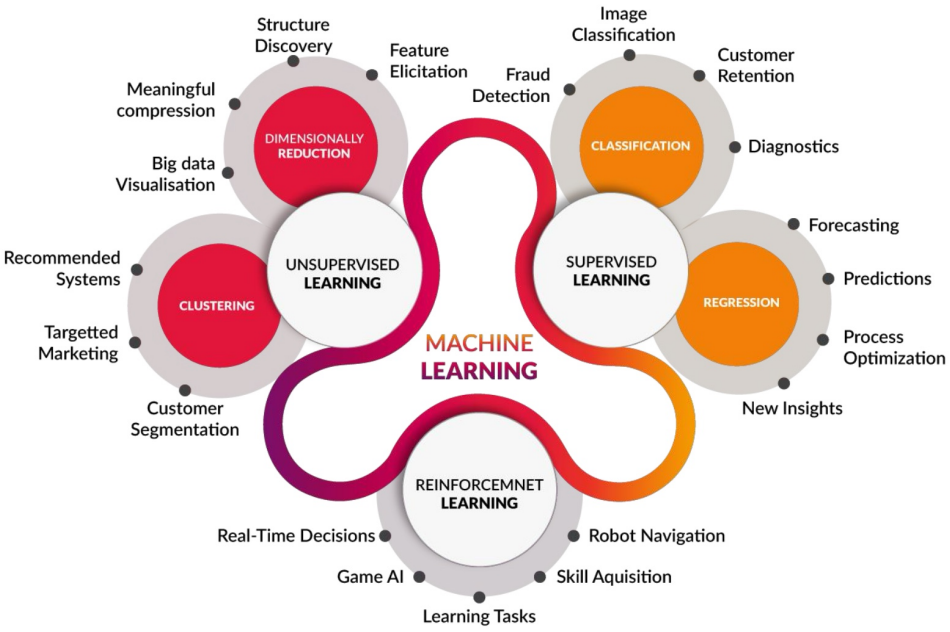


Figure 9.4: Areas of Machine Learning

9.5 Supplement

9.5.1 Potential project-3

Data pre-processing for the gene expression matrix

- Data collecting and merging (if needed)
- Exploration
- Visualization
- Data cleaning
- Perform classification

9.5.2 Resource and uncovered topics

- Introduction to data mining: Chapter 4.1, 4.2 & 5.2
- Problem of KNN (Next lecture)
- Logistic regression and neural network (Next lecture)
- Decision tree/SVM/Bayesian...
- Model overfitting
- Cross-validation (Lecture 11)

9.5.3 The goal of this course

- Understand what is happening under the package and black box
- Understand what we are doing
- Thus, use those packages and tools correctly

9.6 Cutting-edge Method of Image Classification (Optional)

Note: This section was not covered in lecture. With the instructor's permission, scribe included it in the notes because it is highly relevant to classification and presents novel, interesting and efficient methods.

9.6.1 Vision Transformer (Dosovitskiy et al., 2020)

Main point:

- The image is divided into multiple patches. These patches are transformed into a sequence, processed into input embeddings, and then handled using an almost standard Transformer architecture for images. This achieves a unification of NLP and CV and significantly advances the development of multimodal AI.

Method:

- The image is divided into patches. These patches are formed into a sequence and each is passed through a linear projection layer to obtain feature vectors (the patch embedding process). Similar to Transformers in NLP, position embeddings are used to provide positional information for the patches. After embedding, the data passes through a standard Transformer encoder. Following BERT, the output of a special [class] token is used as the final output. This final output is fed into a common MLP head for classification.

9.6.2 CLIP (Radford et al., 2021)

Main point:

- CLIP is a learning paradigm that uses natural language as supervisory signal, inspired by advancements in NLP (such as the GPT series learning from raw text through autoregressive or masked language modeling). CLIP emphasizes learning perception from natural language.
- The advantages of natural language include:
  - **Scalability**: Text data is easy to collect and does not require specific formatting.
  - **Flexibility**: Language can describe unlimited visual concepts, supporting *zero-shot transfer* – the ability to handle new tasks without training on specific datasets.
- **Key Innovation**: Instead of predicting specific words in the text, CLIP predicts whether the entire text matches an image (*contrastive objective*). This simplifies the task, avoids the complexity of text diversity (such as variations in descriptions and comments), and improves efficiency.

Method:

- **Input:** A batch of N (image, text) pairs.
- **Encoding:** An image encoder extracts image features $\mathbf{I}_f$, and a text encoder extracts text features $\mathbf{T}_f$.
- **Projection:** Linear projection into a shared d_e-dimensional embedding space, followed by L2 normalization to obtain $\mathbf{I}_e$ and $\mathbf{T}_e$.
- **Similarity Calculation:** Compute an $N \times N$ cosine similarity matrix, scaled by a learnable temperature parameter τ (initialized to 0.07 to avoid training instability).
- **Loss Function:** A symmetric cross-entropy loss that maximizes the similarity of the correct pairs and minimizes the similarity of the $(N^2 - N)$ incorrect pairs.
- **Data Augmentation:** Only random square cropping is used, without complex transformations.

References

Dosovitskiy, Alexey et al. (2020). “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *CoRR* abs/2010.11929. arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

Radford, Alec et al. (2021). “Learning Transferable Visual Models From Natural Language Supervision”. In: *CoRR* abs/2103.00020. arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020.