Knowledge Discovery & Data Mining

— Classification: Lazy learning —

Instructor: Yong Zhuang

yong.zhuang@gvsu.edu

Outline

- Lazy Learning vs. Eager Learning
- Instance-Based Methods
  - k-Nearest Neighbor
  - Case-Based Reasoning

Lazy vs. Eager Learning

Lazy Learning: Learns at prediction time. Retains entire dataset until a query is made.

- **Memory Usage:** High; must store all training data.
- **Computation:** Less time in training but more time in predicting
- **Examples:** Instance-based learning, …
- **Accuracy:** Utilizes a richer hypothesis space by forming an implicit global approximation through multiple local functions.
- Advantages:
  - Adaptable to changing data without retraining
  - Good for dynamic, frequently changing datasets
- Disadvantages:
  - High memory requirements
  - Can be slow if dataset is large

Eager Learning: Learns during training phase. Builds a model before seeing test instances.

- **Memory Usage:** Lower; only stores model parameters post-training.
- **Computation:** Intensive upfront training but faster at prediction time.
- **Examples:** Decision Trees, Naive Bayes, …
- **Accuracy:** Commits to a single hypothesis that covers the entire instance space, which may limit flexibility in some cases.
- Advantages:
  - Fast predictions due to precomputed model
  - Reduces storage requirements by using model parameters
- Disadvantages:
  - Requires retraining if data changes significantly
  - Computationally expensive to train, especially for complex models

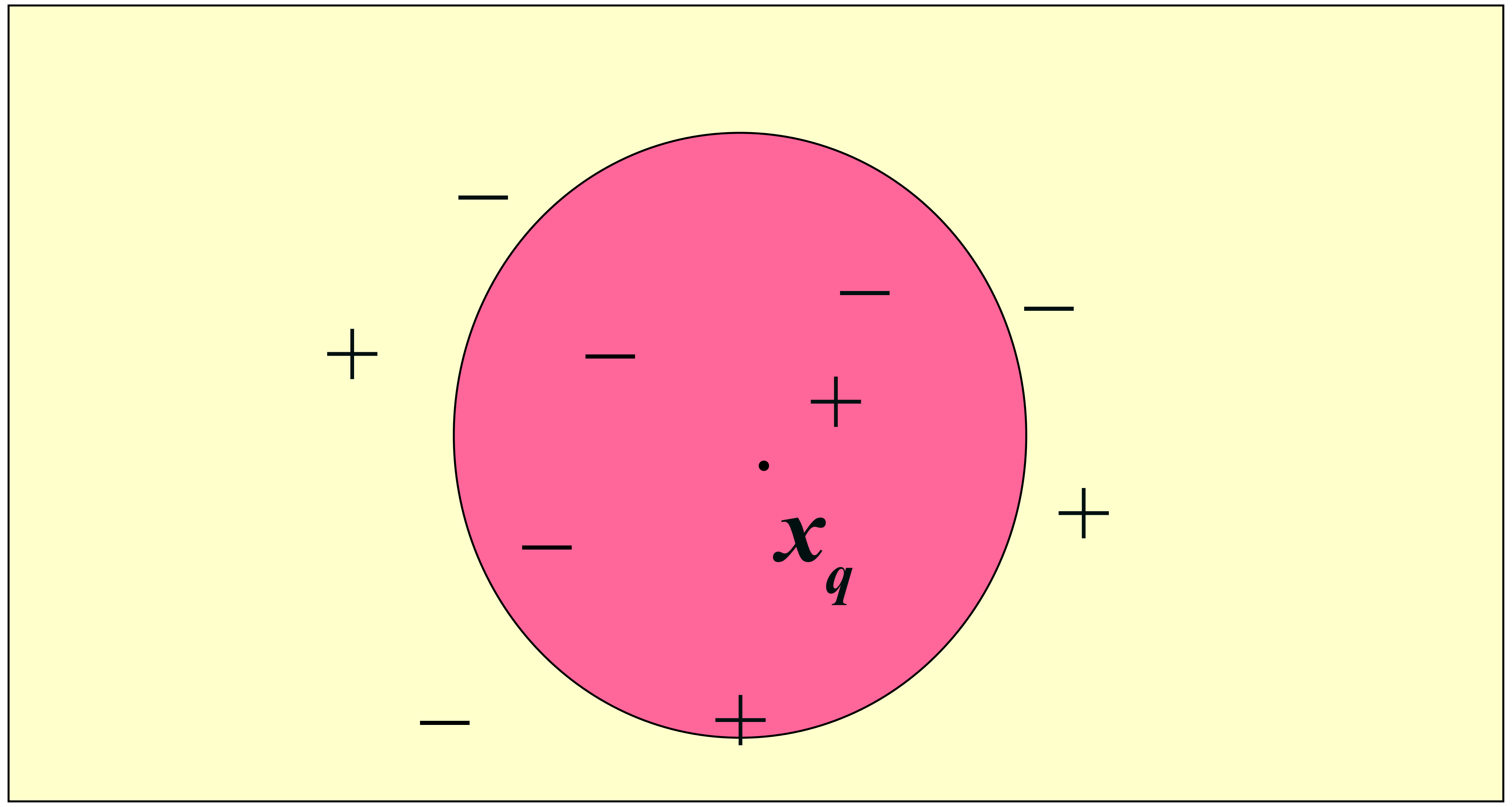
Lazy Learner: Instance-Based Methods

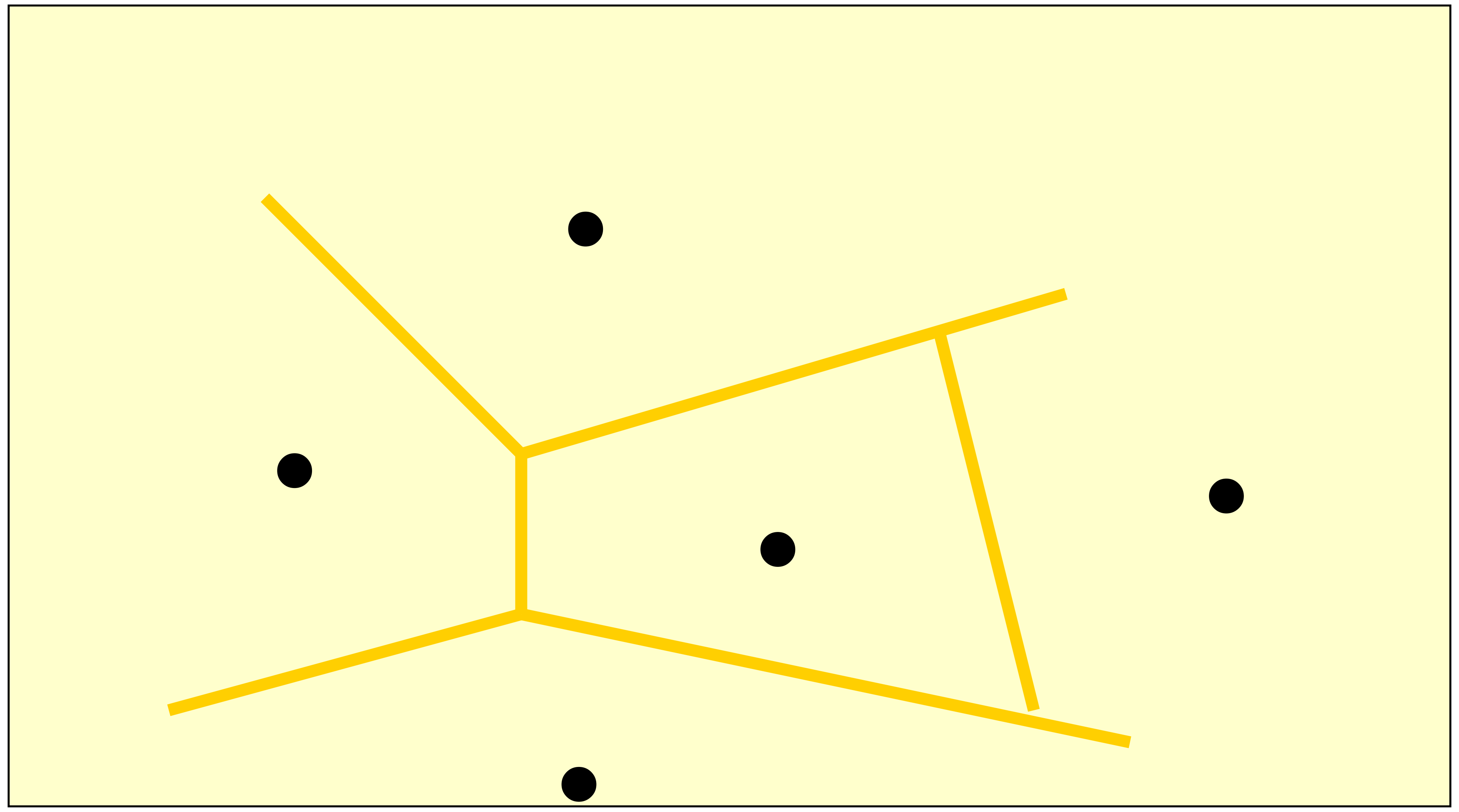
Instance-based learning: stores training examples and postpones processing until a new instance needs classification (referred to as "lazy evaluation").

- Typical approaches
  - *k*-nearest neighbor approach
    - Classifies based on the k closest examples in the training set.
    - Instances represented as points in a Euclidean space.
  - Locally weighted regression
    - Builds a local approximation around the target instance.
    - Useful in non-linear cases where global approximations are insufficient.
  - Case-based reasoning
    - Uses symbolic representations and knowledge-based inference
    - Classifies by drawing parallels to similar past cases rather than relying solely on numeric distances.

The k-Nearest Neighbor Algorithm

- All instances correspond to points in the n-D space
- The nearest neighbor are defined in terms of Euclidean distance, dist($\mathbf{X}_1$, $\mathbf{X}_2$)
- Target function could be discrete- or real- valued
- For discrete-valued, *k*-NN returns the most common value among the *k* training examples nearest to x_q
- Vonoroi diagram: the decision surface induced by 1-NN for a typical set of training examples





Discussion on the k-NN Algorithm

k-NN for Real-valued Predictions

The **k-NN (k-Nearest Neighbors)** algorithm can also be utilized for real-valued predictions for an unknown tuple. Instead of returning a class label, it returns the mean of the attribute values of its k nearest neighbors.

Distance-weighted Nearest Neighbor Algorithm

Instead of treating all neighbors equally, the distance-weighted version of the k-NN algorithm weights the contribution of each neighbor based on their distance to the query point, x_q:

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

- Closer neighbors have a greater influence on the prediction.
- The weight typically decreases as the distance increases.

Advantages and Challenges

- **Robustness**: k-NN is robust to noisy data since it averages the values of k-nearest neighbors.
- **Curse of Dimensionality**: With many attributes, the distances between neighbors could be dominated by irrelevant attributes. This makes the algorithm less effective in high-dimensional spaces.
  - **Solution**: One way to overcome this is through axes stretching or the elimination of the least relevant attributes.
  - axes stretching: Scaling the axes in the feature space, certain dimensions or attributes are given more importance (or less), thus allowing the distances in those dimensions to have a greater impact on the overall distance calculation.

Case-Based Reasoning (CBR)

Medical History:

- Smoking: Former smoker
- Pre-existing Lung Condition: None
- Recent Hospitalization: No

- **CBR**: Uses a database of problem solutions to solve new problems
- Store symbolic description (tuples or cases) — not points in a Euclidean space(KNN)
- Applications: Customer-service (product-related diagnosis), legal ruling
- Methodology
  - Instances represented by rich symbolic descriptions
  - Identical Match: On receiving a new case, the system checks for an identical existing case. If found, the solution for that case is provided.
  - Similar Cases: If no identical case exists, it searches for similar cases. These can be seen as "neighbors" to the new case.
  - Solution Combination: Tries to merge solutions of neighboring cases for the new case. If conflicts arise, backtracking and leveraging background knowledge may be needed.
- Challenges
  - Determining a useful similarity metric. Methods to combine solutions. Selecting important features for indexing cases. Developing efficient indexing techniques. Striking a balance between accuracy and efficiency, especially as the number of stored cases grows.

Summary

- Lazy Learning vs. Eager Learning
- Instance-Based Methods
  - k-Nearest Neighbor
  - Case-Based Reasoning