Data Science: Process and Paradigms

Wagner Meira Jr., PhD

Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

March 20, 2023

Dimensions

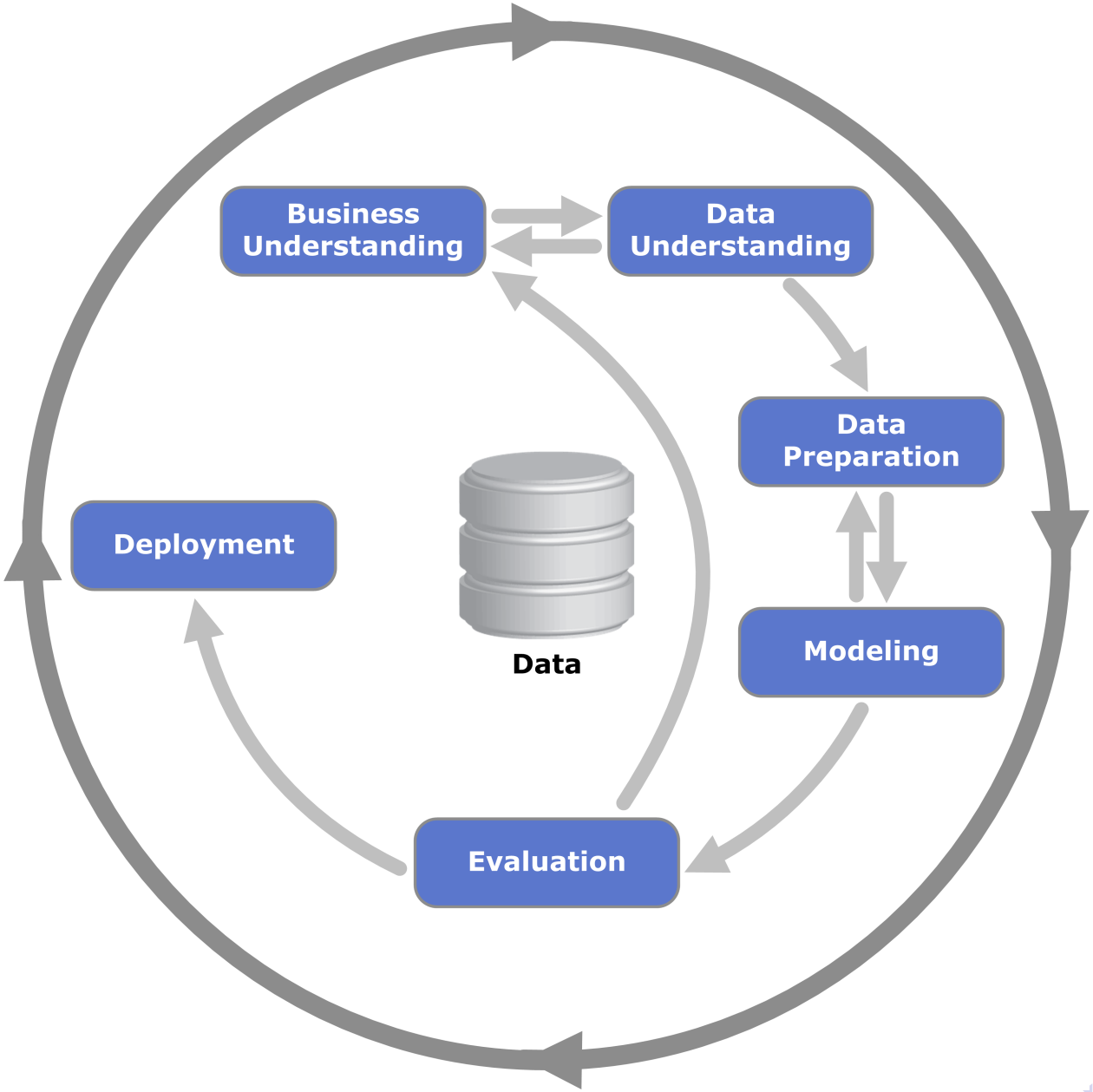
- Data
- Models and techniques
- Processes and Methodologies
- Technology
- People and organizations
- Ethical and societal issues

CRISP-DM

Cross-industry standard process for data mining

CRISP-DM is an open standard process model that describes common approaches used by data mining experts. It is the most widely-used analytics model.
CRISP-DM was conceived in 1996 and became a European Union project under the ESPRIT funding initiative in 1997. The project was led by five companies: Integral Solutions Ltd (ISL), Teradata, Daimler AG, NCR Corporation and OHRA, an insurance company.
Based on current research CRISP-DM is the most widely used form of data-mining model because of its various advantages which solved the existing problems in the data mining industries. Some of the drawbacks of this model is that it does not perform project management activities. The fact behind the success of CRISP-DM is that it is industry, tool, and application neutral.

CRISP-DM


Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Data

1.Business understanding

Background The Background provides a basic overview of the project context. This lists what area the project is working in, what problems have been identified and why data mining appears to provide a solution.

Business objectives and success criteria Business Objectives describe what the goals of the project are in business terms. For each objective, Business Success Criteria, i.e. explicit measures for determining whether or not the project succeeded in its objectives, should be provided. This section should also list objectives that were considered but rejected. The rationale of the selection of objectives should be given.

Inventory of resources The Inventory of Resources aims to identify personnel, data sources, technical facilities and other resources that may be useful in carrying out the project.

Requirements, assumptions and constraints This output lists general requirements about how the project is executed, type of project results, assumptions made about the nature of the problem and the data being used and constraints imposed on the project.

Risks and contingencies This output identifies problems that may occur in the project, describes the consequences and states what action can be taken to minimize the effect.

1.Business understanding

Terminology The Terminology allows people unfamiliar with the problems being addressed by the project to become more familiar with them.
Costs and benefits This describes the costs of the project and predicted business benefits if the project is successful (e.g. return on investment). Other less tangible benefits (e.g. customer satisfaction) should also be highlighted.
Data mining goals and success criteria The data mining goals state the results of the project that enable the achievement of the business objectives. As well as listing the probable data mining approaches, the success criteria for the results should also be listed in data mining terms.
Project plan This lists the stages to be executed in the project, together with duration, resources required, inputs, outputs and dependencies. Where possible it should make explicit the large-scale iterations in the data mining process, for example repetitions of the modeling and evaluation phases.
Initial assessment of tools and techniques This section gives an initial view of what tools and techniques are likely to be used and how. It describes the requirements for tools and techniques, lists available tools and techniques and matches them to requirements.

2.Data understanding

Initial data collection report This report describes how the different data sources identified in the inventory were captured and extracted.

- Background of data.
- List of data sources with broad area of required data covered by each.
- For each data source, method of acquisition or extraction.
- Problems encountered in data acquisition or extraction.

Data description report Each dataset acquired is described.

- Each data source described in detail.
- List of tables (may be only one) or other database objects.
- Description of each field including units, codes used, etc.

2.Data understanding

Data exploration report Describes the data exploration and its results.

- Background including broad goals of data exploration.
- For each area of exploration undertaken:
  - Expected regularities or patterns.
  - Method of detection.
  - Regularities or patterns found, expected and unexpected.
  - Any other surprises.
  - Conclusions for data transformation, data cleaning and any other pre-processing.
  - Conclusions related to data mining goals or business objectives.
  - Summary of conclusions.

Data quality report This report describes the completeness and accuracy of the data.

- Background including broad expectations about data quality.
- For each dataset:
  - Approach taken to assess data quality.
  - Results of data quality assessment.
  - Summary of data quality conclusions.

3.Data preparation

Dataset description report This provides a description of the dataset (after pre-processing) and the process by which it was produced.

- Background including broad goals and plan for pre-processing.
- Rationale for inclusion/exclusion of datasets.
- For each included dataset:
  - Description of the pre-processing, including the actions that were necessary to address any data quality issues.
  - Detailed description of the resultant dataset, table by table and field by field.
  - Rationale for inclusion/exclusion of attributes.
  - Discoveries made during pre-processing and any implications for further work.
  - Summary and conclusions.

4.Modeling

Modeling assumption This section defines explicitly any assumptions made about the data and any assumptions that are implicit in the modeling technique to be used.
Test design This section describes how the models are built, tested and evaluated.

- Background - outlines the modeling undertaken and its relation to the data mining goals.
- For each modeling task:
  - Broad description of the type of model and the training data to be used.
  - Explanation of how the model will be tested or assessed.
  - Description of any data required for testing.
  - Plan for production of test data if any.
  - Description of any planned examination of models by domain or data experts.
  - Summary of test plan.

4.Modeling

Model description This report describes the delivered models and overviews the process by which they were produced.

- Overview of models produced.
- For each model:
  - Type of model and relation to data mining goals.
  - Parameter settings used to produce the model.
  - Detailed description of the model and any special features.
  - For example:
    - For rule-based models, list the rules produced plus any assessment of per-rule or overall model accuracy and coverage.
    - For opaque models, list any technical information about the model (such as neural network topology) and any behavioral descriptions produced by the modeling process (such as accuracy or sensitivity).
    - Description of Model's behavior and interpretation.
  - Conclusions regarding patterns in the data (if any); sometimes the model will reveal important facts about the data without a separate assessment process (e.g. that the output or conclusion is duplicated in one of the inputs).
- Summary of conclusions.

Model assessment This section describes the results of testing the models according to the test design.

- Overview of assessment process and results including any deviations from the test plan.
- For each model:
  - Detailed assessment of model including measurements such as accuracy and interpretation of behavior.
  - Any comments on models by domain or data experts.
  - Summary assessment of model.
  - Insights into why a certain modeling technique and certain parameter settings led to good/bad results.
  - Summary assessment of complete model set.

5.Evaluation

Assessment of data mining results with respect to business success criteria This report compares the data mining results with the business objectives and the business success criteria.

- Review of Business Objectives and Business Success Criteria (which may have changed during and/or as a result of data mining).
- For each Business Success Criterion:
  - Detailed comparison between success criterion and data mining results.
  - Conclusions about achievability of success criterion and suitability of data mining process.
- Review of Project Success; has the project achieved the original Business Objectives?
- Are there new business objectives to be addressed later in the project or in new projects?
- Conclusions for future data mining projects.

Review of process This section assesses the effectiveness of the project and identifies any factors that may have been overlooked.

List of possible actions This section makes recommendations regarding the next steps in the project.

6.Deployment

Deployment plan This section specifies the deployment of the data mining results.

- Summary of deployable results (derived from Next Steps report).
- Description of deployment plan.

Monitoring and maintenance plan The monitoring and maintenance plan specifies how the deployed results are to be maintained.

- Overview of results deployment and indication of which results may require updating (and why).
- For each deployed result:
  - Description of how updating will be triggered (regular updates, trigger event, performance monitoring).
  - Description of how updating will be performed.
- Summary of the results updating process.

6.Deployment

Final report The final report is used to summarize the project and its results. Contents:

- Summary of Business Understanding: background, objectives and success criteria.
- Summary of data mining process.
- Summary of data mining results.
- Summary of results evaluation.
- Summary of deployment and maintenance plans.
- Cost/benefit analysis.
- Conclusions for the business.
- Conclusions for future data mining.

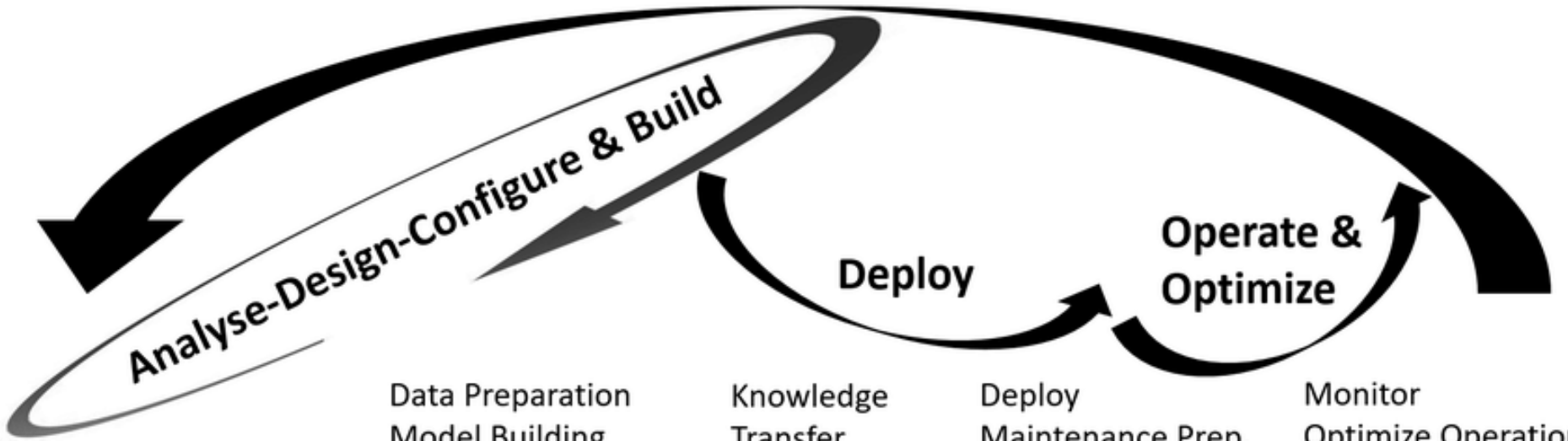
ASUM-DM

Analytics Solutions Unified Method for Data Mining/Predictive Analysis

The Analytics Solutions Unified Method (ASUM) is a step-by-step guide to conducting a complete implementation lifecycle for analytics solutions. It was created to accelerate your time to value and lower your risk by establishing consistent approaches and processes that increase your implementation efficiency. It contains structured steps, development activities, roles and responsibilities, templates and guidelines.

ASUM-DM

Analytics Solutions Unified Method for Data Mining/Predictive Analysis



ASUM-DM

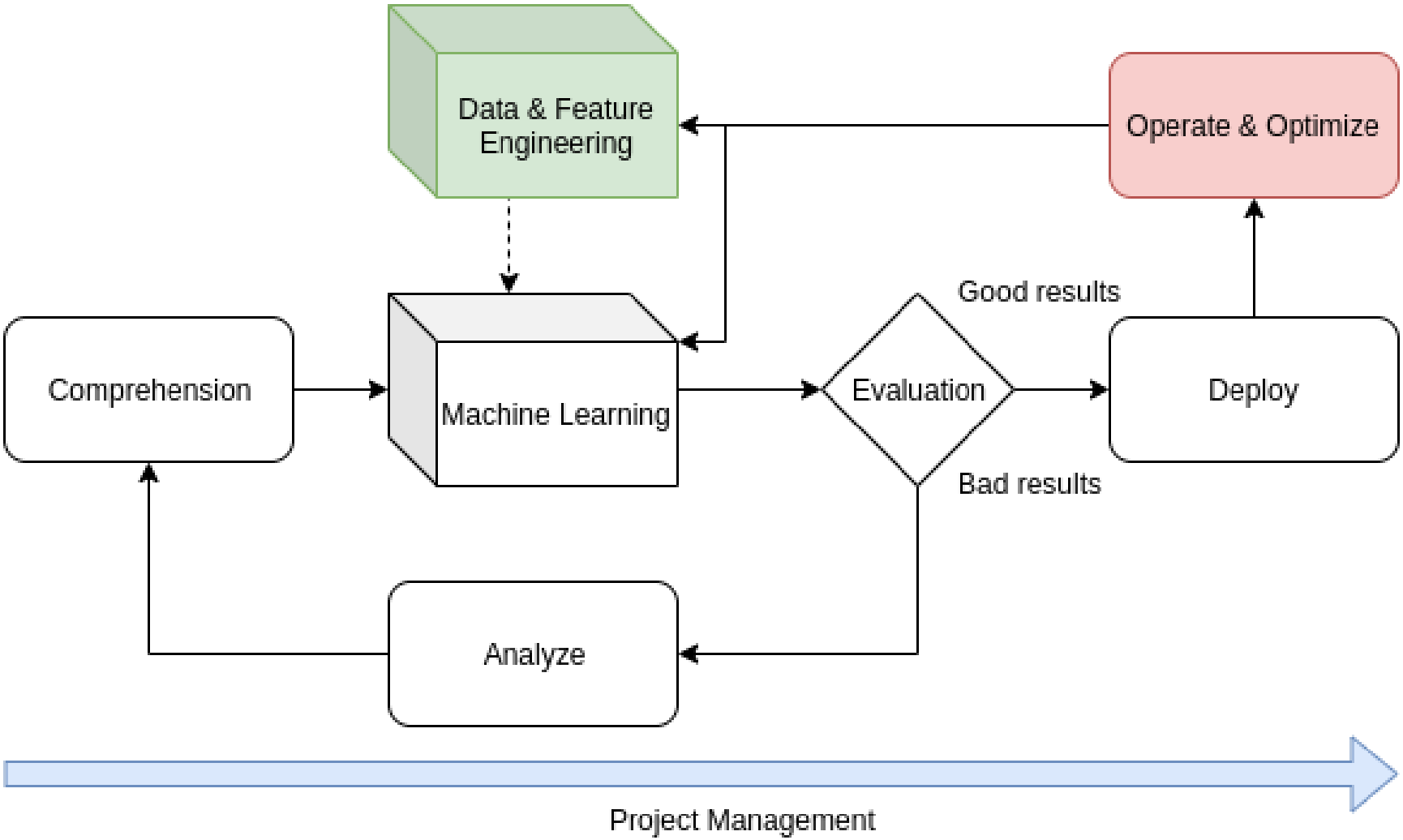
Analytics Solutions Unified Method for Data Mining/Predictive Analysis

- **Understand and analyze the problem:** define the current state of the organization's solution, and define the implementation objectives (increase the number of customers by individualizing suggestions?). It will then be necessary to define the prerequisites of the implementation (useful data and their means of obtaining, model performance, etc. . .).
- **Design:** define all the necessary components for implementation and then start building the model on the basis of pre-existing, transformed or simply synthesized data to start evaluating the different tracks and their feasibility in the form of prototypes.
- **Configure and Implement:** based on the prototypes created during the design phase, which is the most efficient and adapted to the problem? Then begins the implementation phase in real life situations. This iterative and incremental approach allows us to start adapting the existing software and hardware architecture to allow us to insert our solution. The model, almost functional, now requires only a few adjustments to get into production.

- **Deploy:** a roadmap is created. The model is then deployed and configured in a production environment and deadlines are set for the various precision measurements of the model, its re-training, and the frequency of maintenance of the used Hardware.
- **Maintain and Optimize:** the model is fully operational in production and begins to generate value (better medical diagnosis, increased revenue, better UX, etc. . .). The accuracy of the model and its parameters are regularly checked to stay cutting-edge.

ASUM-DM

Analytics Solutions Unified Method for Data Mining/Predictive Analysis



Data

- Tabular
  - categorical
  - numeric
- Text
- Graphs
- Sound
- Image
- Video

Data

Challenges

- Storage
- Accessing
- Engineering
  - Integration
  - Cleaning
  - Transformation
- Visualization

Data Mining

Concept

Automatic extraction of knowledge or patterns that are interesting (novel, useful, implicit, etc.) from large volumes of data.

Tasks

- Data engineering
- Characterization
- Prediction

Data Mining Models

Concept

A model aims to represent the nature or reality from a specific perspective. A model is an artificial construction where all extraneous details have been removed or abstracted, while keeping the key features necessary for analysis and understanding.

Task

Among all possible sets of entities, which ones are the most frequent? Or better, determine the sets of items that co-occur in a database more frequently than a given threshold.

Application Scenario

Market-basket problem: Given that a customer purchased items in set A, what are the most likely items to be purchased in the future?

Task

Given a similarity criterion, what is the entity partition that groups together the most similar entities?

Application Scenario

Customer segmentation: Partition a customer base into groups of similar customers, supporting different policies and strategies for each group.

Task

Given some knowledge about a domain, including classes or categories of entities, and a sample whose class is unknown, predict the class of the latter based on the existing knowledge.

Application Scenario

Credit scoring: A bank needs to decide whether it will loan money to a given person. It may use past experience with other persons who present a similar profile to decide whether or not it is worth giving the loan.

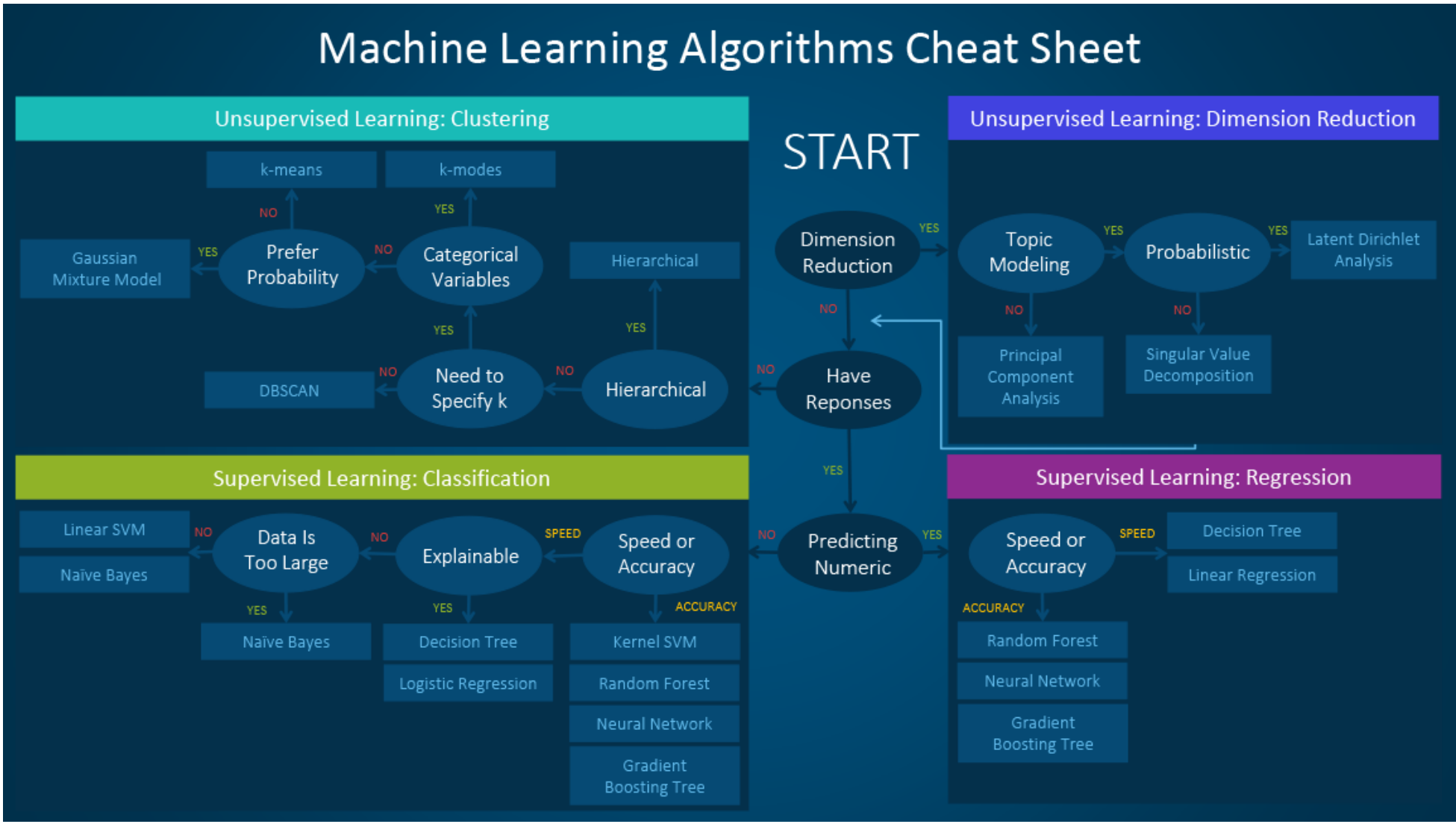
Task

Given some knowledge about a domain, including target numerical outcomes, and a sample whose outcome is unknown, predict the outcome of the latter based on the existing knowledge.

Application Scenario

Financial Forecasting: An investor needs to decide among several investment options (e.g., retail, stock, real state) and estimate their return in the future is key. She may use past experience in terms of relevant factors and models to perform the estimations.

Models and Algorithms



Data Mining

Paradigms

- **Combinatorial**
- Probabilistic
- Algebraic
- Graph-based
- Connectionist

Domain

Models partition (or select) entities based on their attributes and their combinations. Search space is discrete and finite, although potentially very large.

Task

Determine the best model according to a quality metric.

Strategies

- Pruning exhaustive search
- Heuristic approximation

- Frequent Itemset Mining
- k-Means
- DBScan
- Decision trees

Frequent Itemsets

Minimum support: $minsup = 3$

t	$\boldsymbol{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

Transaction Database

sup	itemsets
6	B
5	E, BE
4	$A, C, D, AB, AE, BC, BD, ABE$
3	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

Frequent Itemsets

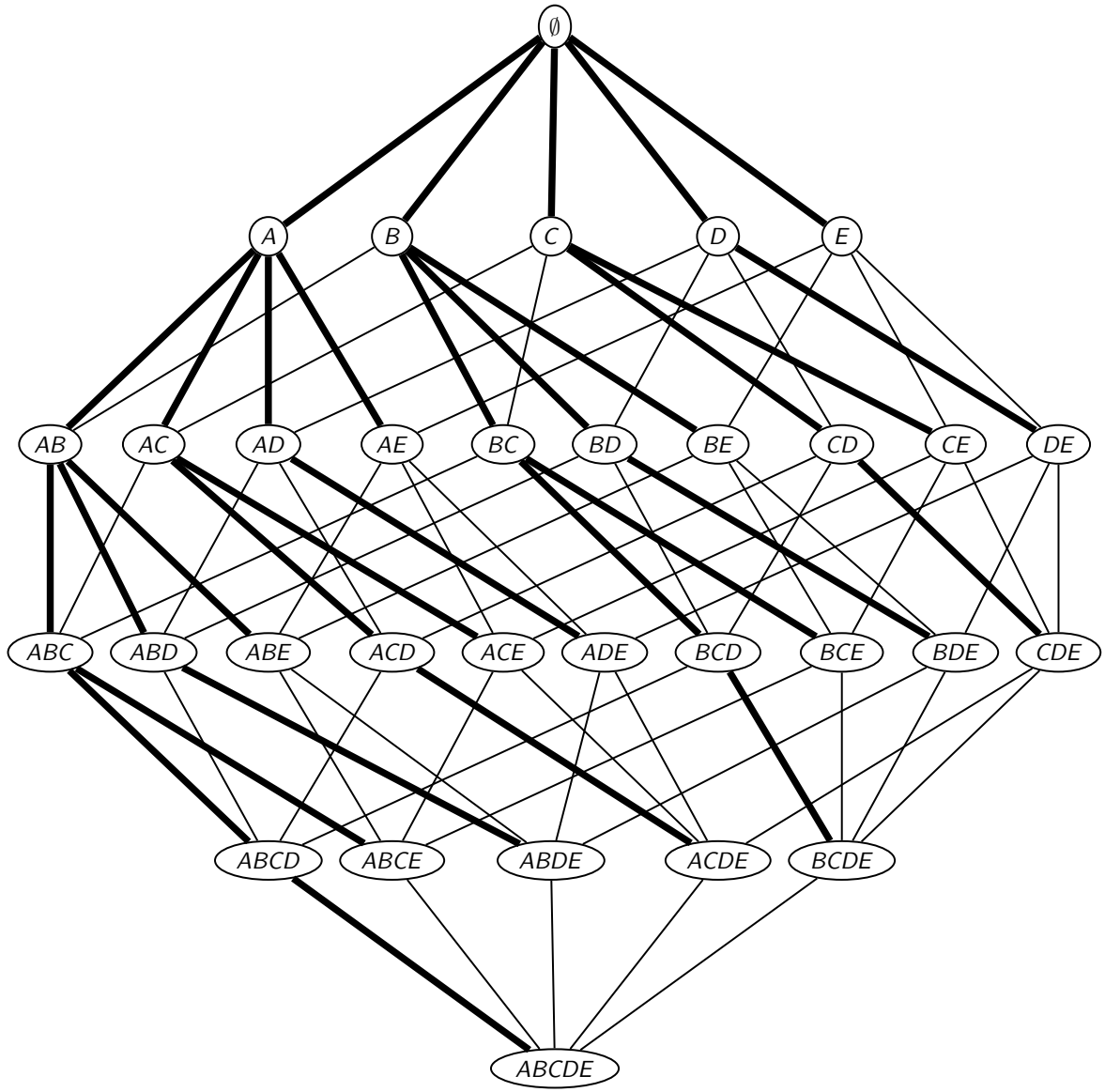
The 19 frequent itemsets shown in the table comprise the set $\mathcal{F}$. The sets of all frequent k-itemsets are

$$\mathcal{F}^{(1)} = \{A, B, C, D, E\}$$
$$\mathcal{F}^{(2)} = \{AB, AD, AE, BC, BD, BE, CE, DE\}$$
$$\mathcal{F}^{(3)} = \{ABD, ABE, ADE, BCE, BDE\}$$
$$\mathcal{F}^{(4)} = \{ABDE\}$$

Itemset lattice and prefix-based search tree

Itemset search space is a lattice where any two itemsets X and Y are connected by a link iff X is an *immediate subset* of Y, that is, $X \subseteq Y$ and $|X| = |Y| - 1$.

Frequent itemsets can enumerated using either a BFS or DFS search on the *prefix tree*, where two itemsets X, Y are connected by a link iff X is an immediate subset and prefix of Y. This allows one to enumerate itemsets starting with an empty set, and adding one more item at a time.



The Apriori Algorithm

Apriori ($\boldsymbol{D}$, $\mathcal{I}$, $minsup$):

1 $\mathcal{F} \leftarrow \emptyset$

2 $\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ // Initial prefix tree with single items

3 **foreach** $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$

4 $k \leftarrow 1$ // k denotes the level

5 **while** $\mathcal{C}^{(k)} \neq \emptyset$ **do**

6 ComputeSupport ($\mathcal{C}^{(k)}, \boldsymbol{D}$)

7 **foreach** *leaf* $X \in \mathcal{C}^{(k)}$ **do**

8 **if** $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$

9 **else** remove X from $\mathcal{C}^{(k)}$

10 $\mathcal{C}^{(k+1)} \leftarrow$ ExtendPrefixTree ($\mathcal{C}^{(k)}$)

11 $k \leftarrow k + 1$

12 **return** $\mathcal{F}^{(k)}$

K-Means Algorithm

K-means $(\boldsymbol{D}, k, \epsilon)$**:**

$t = 0$

Randomly initialize k centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t \in \mathbb{R}^d$

repeat

  $t \leftarrow t + 1$

  $C_j \leftarrow \emptyset$ for all $j = 1, \cdots, k$

  `// Cluster Assignment Step`

  foreach $\boldsymbol{x}_j \in \boldsymbol{D}$ **do**

    $i^* \leftarrow \arg\min_i \left\{ \|\boldsymbol{x}_j - \boldsymbol{\mu}_i^t\|^2 \right\}$ `// Assign` $\boldsymbol{x}_j$ `to closest centroid`

    $C_{i^*} \leftarrow C_{i^*} \cup \{\boldsymbol{x}_j\}$

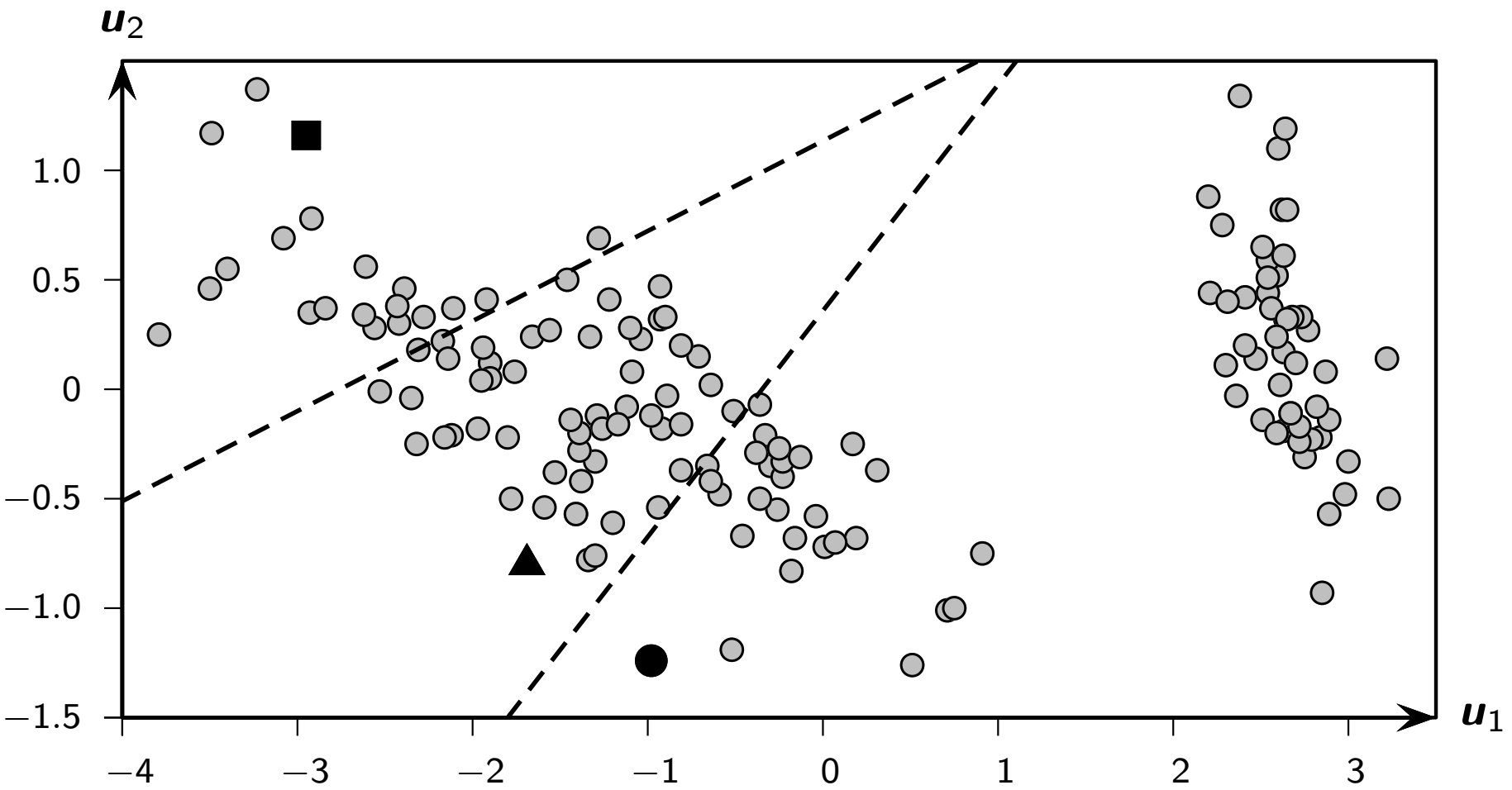
  `// Centroid Update Step`

  foreach $i = 1$ *to* k **do**

    $\boldsymbol{\mu}_i^t \leftarrow \frac{1}{|C_i|} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j$

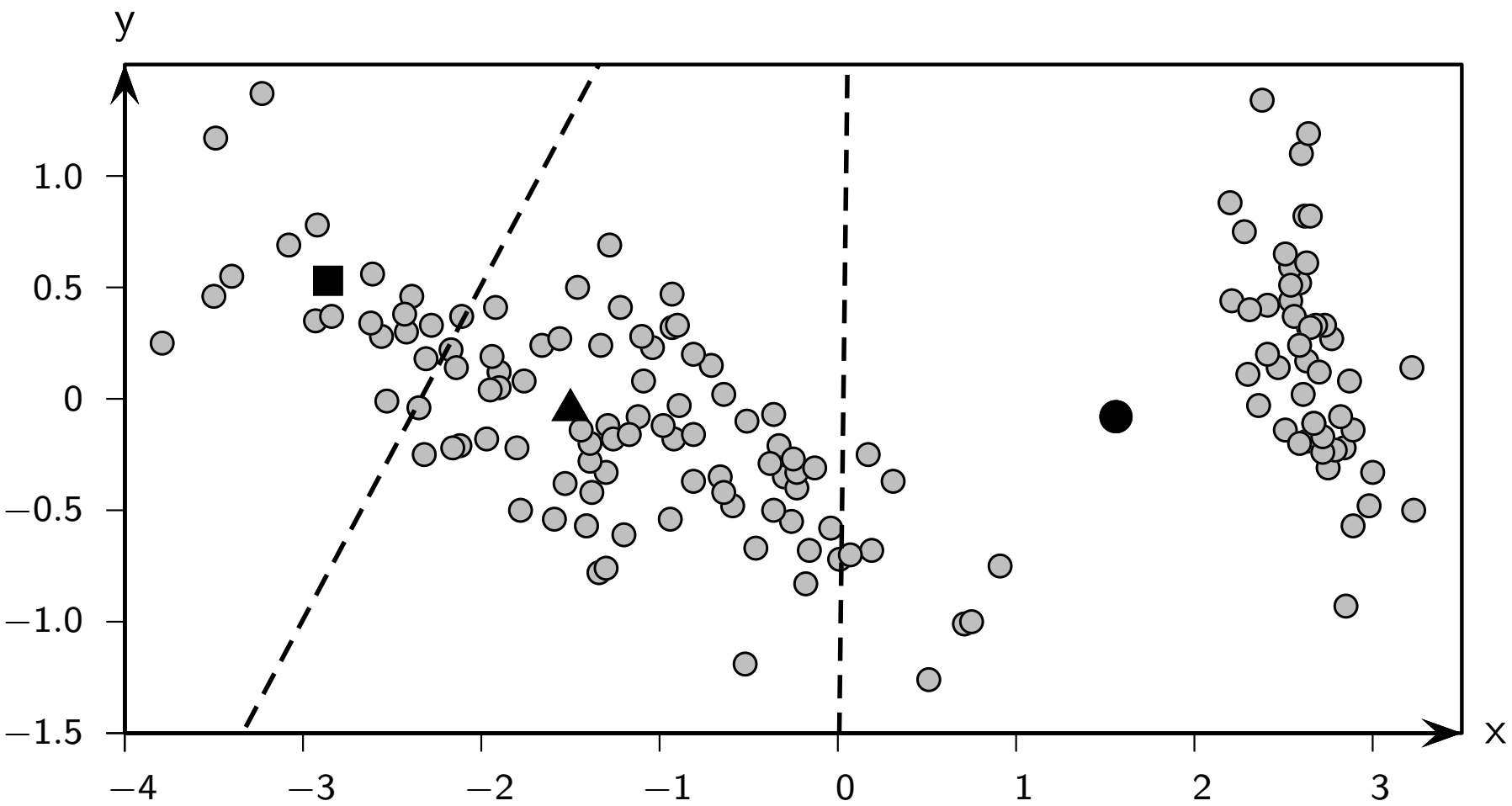
until $\sum_{i=1}^k \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$

K-means in 2D: Iris Principal Components



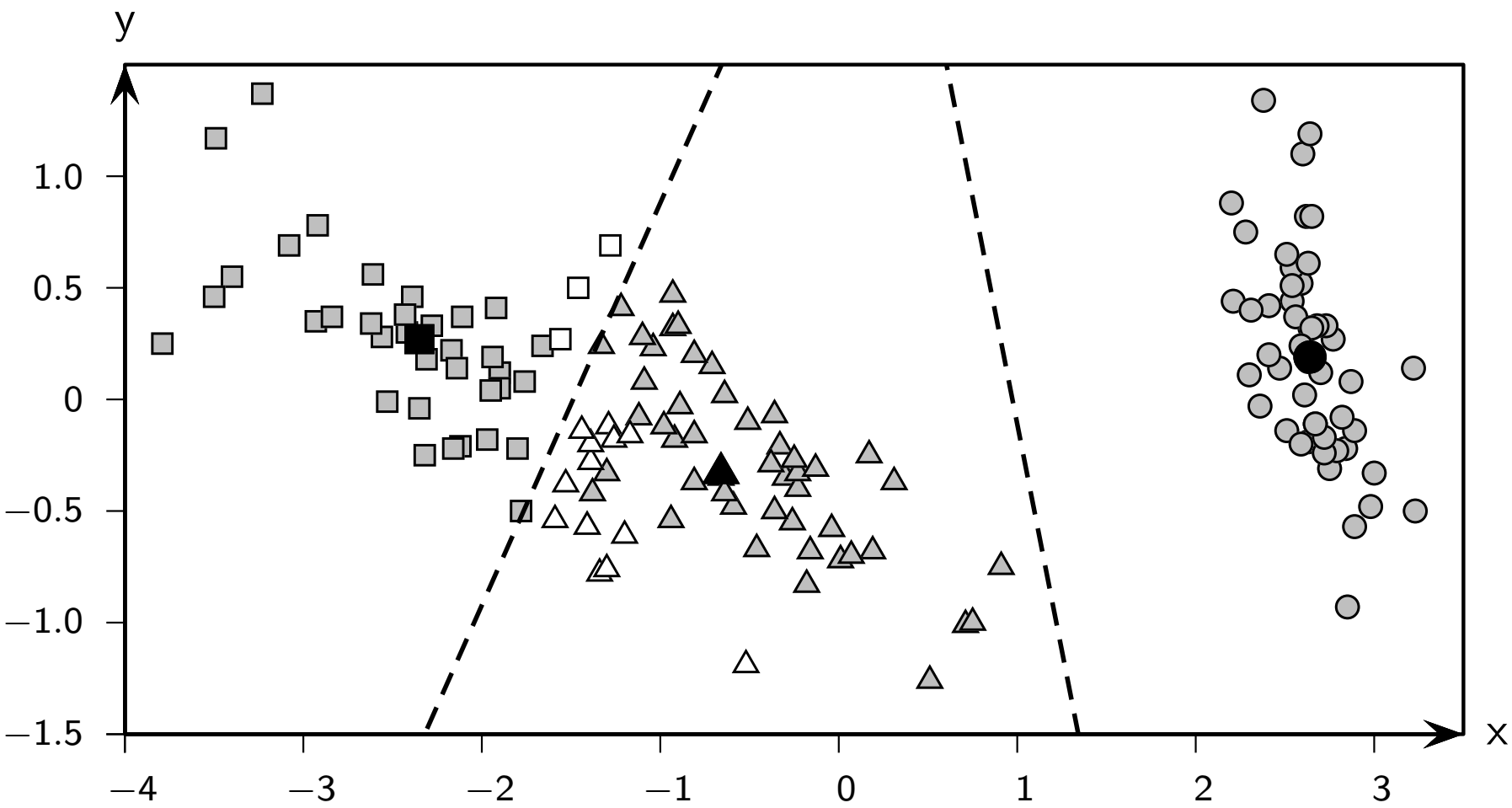
(a) Random initialization: $t = 0$

K-means in 2D: Iris Principal Components



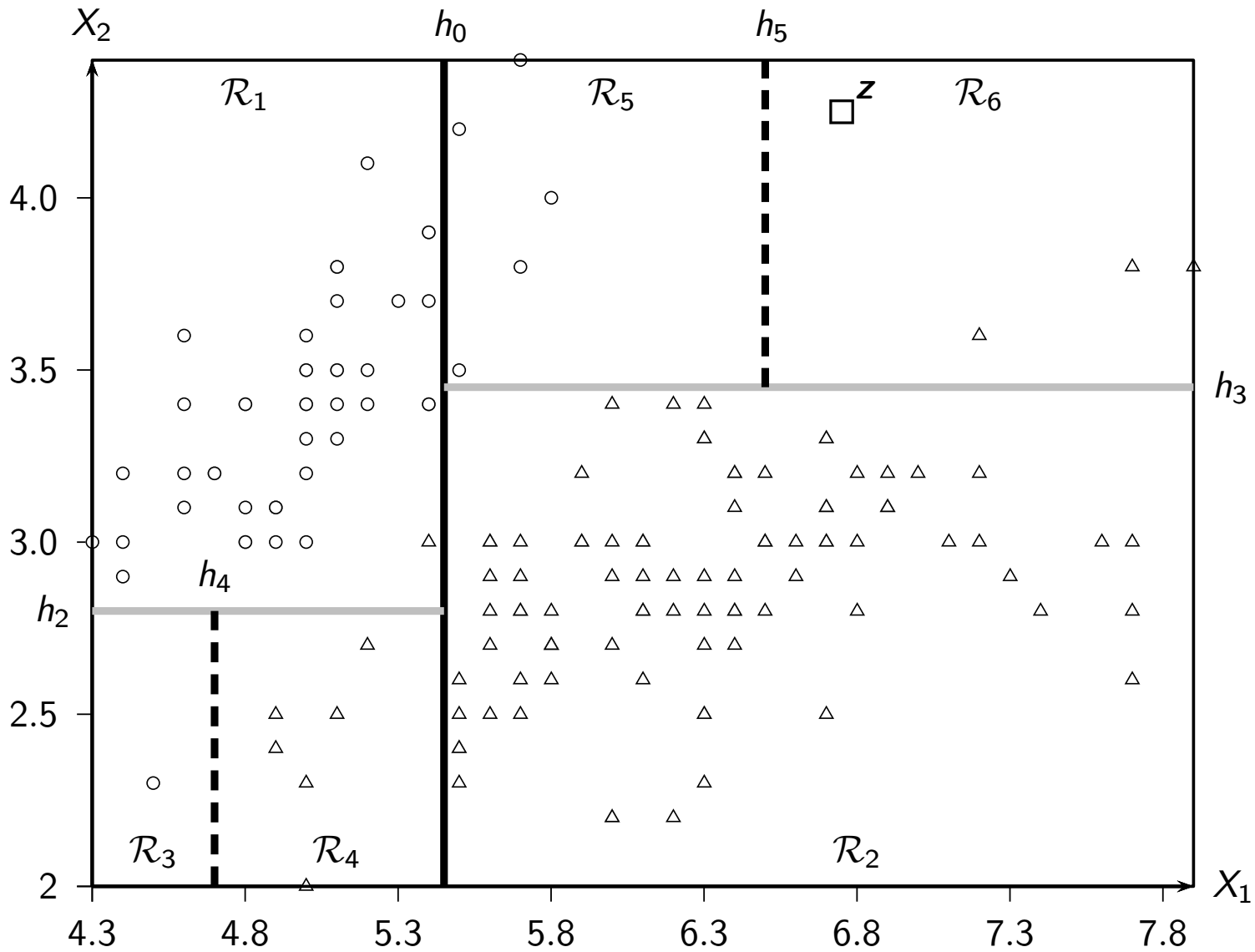
(b) Iteration: $t = 1$

K-means in 2D: Iris Principal Components



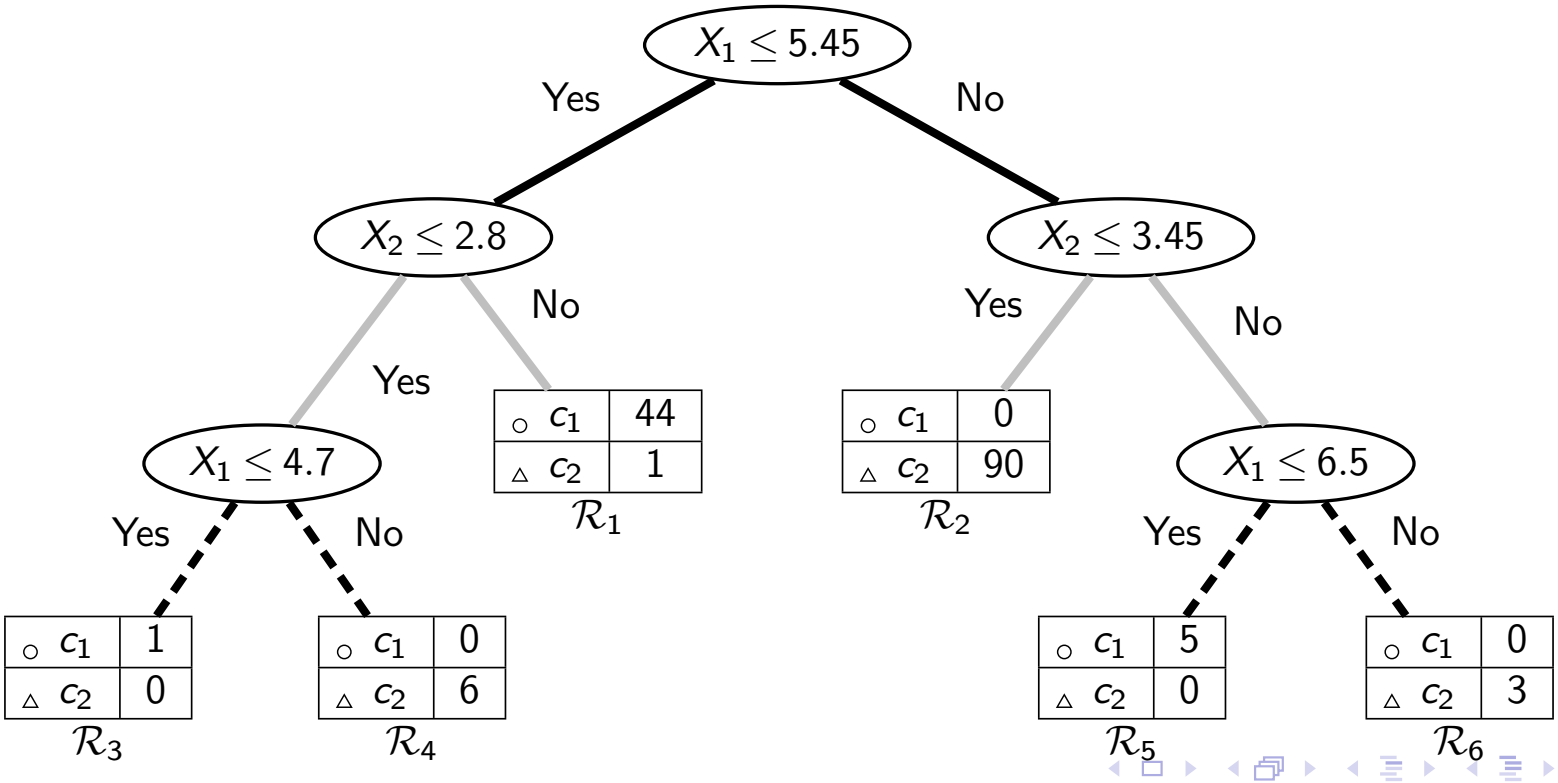
(c) Iteration: $t = 8$ (converged)

Decision Tree: Recursive Splits



Decision Tree

A decision tree consists of internal nodes that represent the decisions corresponding to the hyperplanes or split points (i.e., which half-space a given point lies in), and leaf nodes that represent regions or partitions of the data space, which are labeled with the majority class. A region is characterized by the subset of data points that lie in that region.



Data Mining

Paradigms

- Combinatorial
- **Probabilistic**
- Algebraic
- Graph-based
- Connectionist

Domain

Models are based on one or more probability density function(s) (PDF). Given a model and a dataset, search its parameter space, which may be continuous and/or discrete.

Task

Determine the best parameter models for a dataset, according to an optimization metric.

Strategies

- Direct
- Iterative

- Expectation-Maximization
- DenClue
- Naive Bayes

Expectation-Maximization Clustering Algorithm

Expectation-Maximization $(\boldsymbol{D}, k, \epsilon)$**:**

$t \leftarrow 0$

Randomly initialize $\boldsymbol{\mu}_1^t, \ldots, \boldsymbol{\mu}_k^t$

$\Sigma_i^t \leftarrow \boldsymbol{I}, \forall i = 1, \ldots, k$

repeat

  $t \leftarrow t + 1$

  for $i = 1, \ldots, k$ *and* $j = 1, \ldots, n$ **do**

    $w_{ij} \leftarrow \frac{f(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \Sigma_i) \cdot P(C_i)}{\sum_{a=1}^{k} f(\boldsymbol{x}_j | \boldsymbol{\mu}_a, \Sigma_a) \cdot P(C_a)}$ // posterior probability $P^t(C_i | \boldsymbol{x}_j)$

  for $i = 1, \ldots, k$ **do**

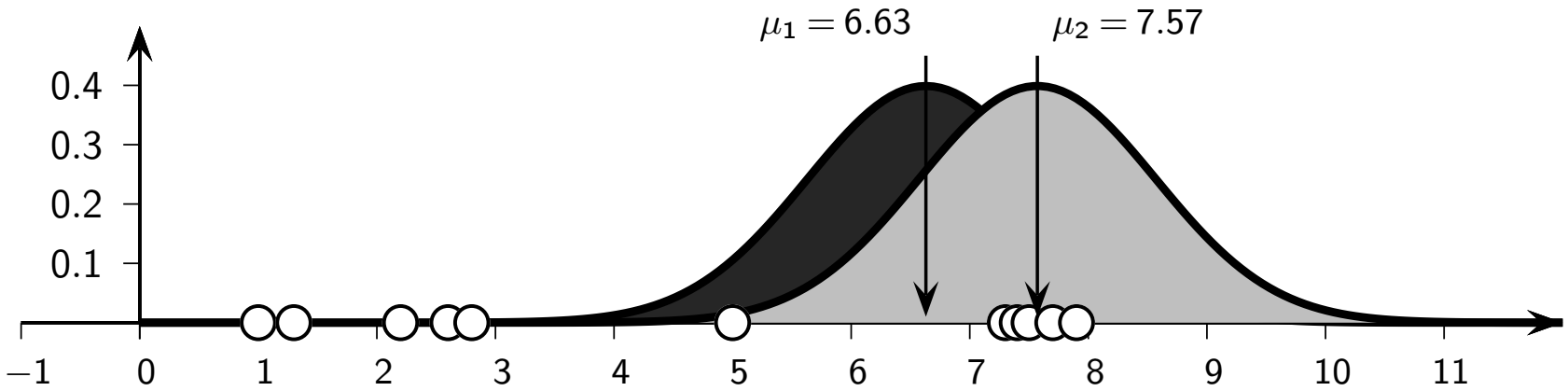
    $\boldsymbol{\mu}_i^t \leftarrow \frac{\sum_{j=1}^{n} w_{ij} \cdot \boldsymbol{x}_j}{\sum_{j=1}^{n} w_{ij}}$ // re-estimate mean

    $\Sigma_i^t \leftarrow \frac{\sum_{j=1}^{n} w_{ij} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} w_{ij}}$ // re-estimate covariance matrix

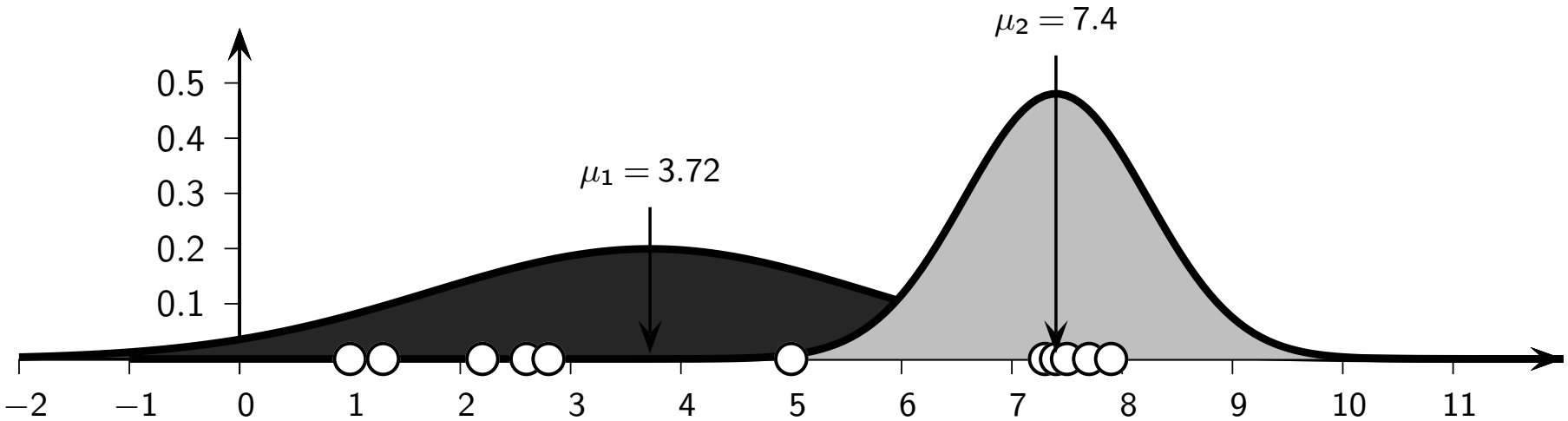
    $P^t(C_i) \leftarrow \frac{\sum_{j=1}^{n} w_{ij}}{n}$ // re-estimate priors

until $\sum_{i=1}^{k} \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$

EM in One Dimension

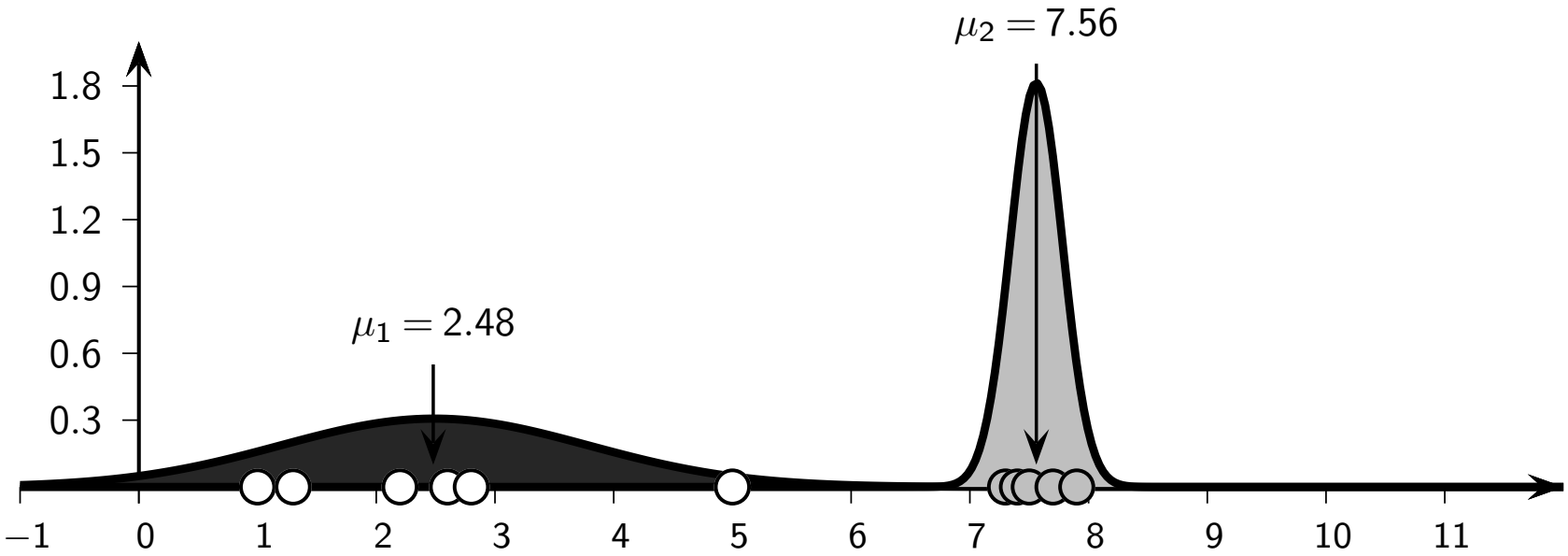


(a) Initialization: $t = 0$



(b) Iteration: $t = 1$

EM in One Dimension: Final Clusters



(c) Iteration: $t = 5$ (converged)

Bayes Classifier Algorithm

BayesClassifier ($\boldsymbol{D} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{n}$)**:**

for $i = 1, \ldots, k$ **do**

- $\boldsymbol{D}_i \leftarrow \{\boldsymbol{x}_j \mid y_j = c_i, j = 1, \ldots, n\}$ // class-specific subsets
- $n_i \leftarrow |\boldsymbol{D}_i|$ // cardinality
- $\hat{P}(c_i) \leftarrow n_i / n$ // prior probability
- $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in \boldsymbol{D}_i} \boldsymbol{x}_j$ // mean
- $\boldsymbol{Z}_i \leftarrow \boldsymbol{D}_i - 1_{n_i} \hat{\boldsymbol{\mu}}_i^T$ // centered data
- $\widehat{\boldsymbol{\Sigma}}_i \leftarrow \frac{1}{n_i} \boldsymbol{Z}_i^T \boldsymbol{Z}_i$ // covariance matrix

return $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i$ *for all* $i = 1, \ldots, k$

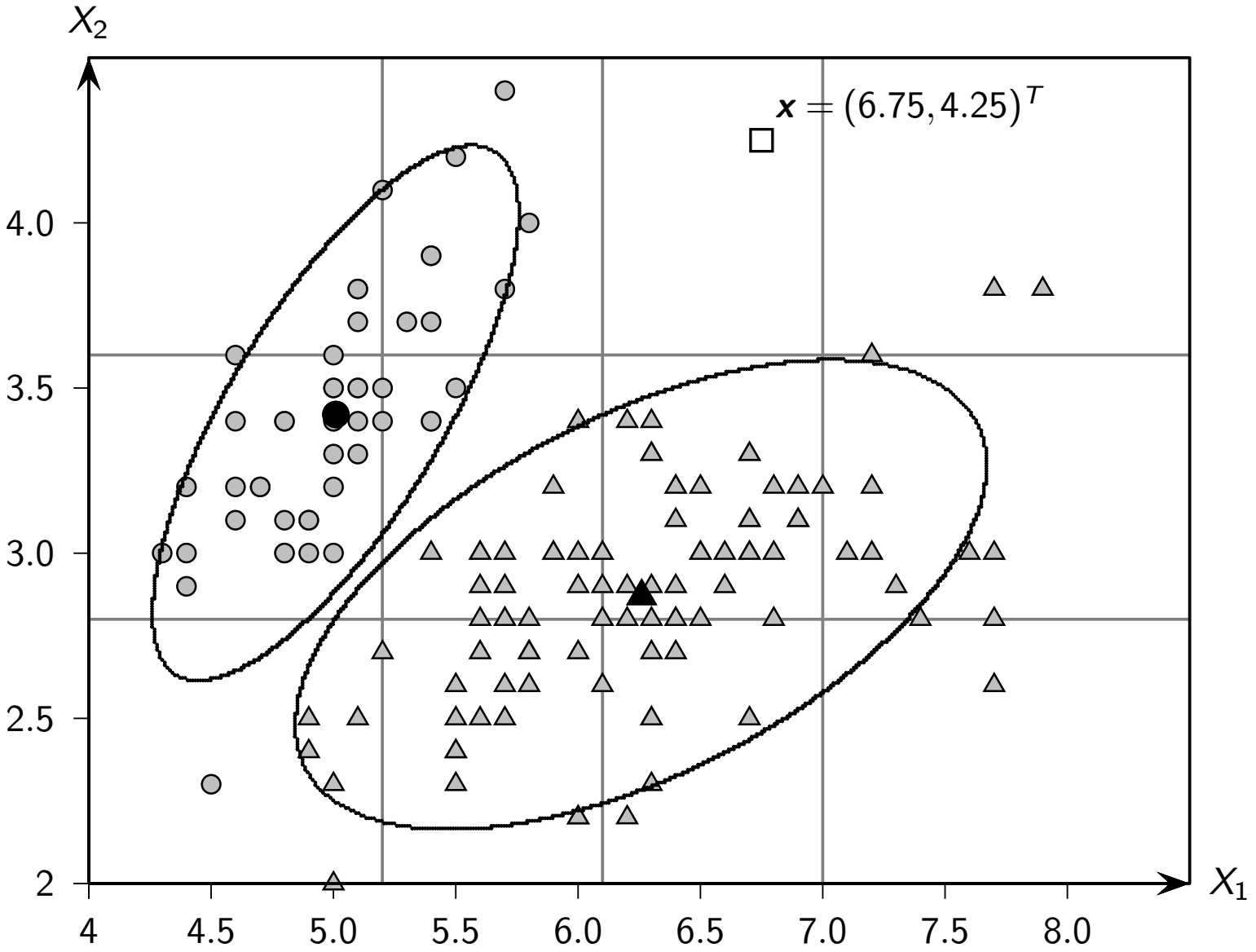
Testing ($\boldsymbol{x}$ **and** $\hat{P}(c_i)$, $\hat{\boldsymbol{\mu}}_i$, $\widehat{\boldsymbol{\Sigma}}_i$, **for all** $i \in [1, k]$)**:**

$\hat{y} \leftarrow \arg\max_{c_i} \left\{ f(\boldsymbol{x} | \hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i) \cdot P(c_i) \right\}$

return $\hat{y}$

Bayes Classifier: Iris Data

X_1:sepal length versus X_2:sepal width



Naive Bayes Algorithm

NaiveBayes ($\boldsymbol{D} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{n}$):

for $i = 1, \ldots, k$ **do**

  $\boldsymbol{D}_i \leftarrow \{\boldsymbol{x}_j^T \mid y_j = c_i, j = 1, \ldots, n\}$ // class-specific subsets

  $n_i \leftarrow |\boldsymbol{D}_i|$ // cardinality

  $\hat{P}(c_i) \leftarrow n_i / n$ // prior probability

  $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in \boldsymbol{D}_i} \boldsymbol{x}_j$ // mean

  $\overline{\boldsymbol{D}}_i = \boldsymbol{D}_i - 1 \cdot \hat{\boldsymbol{\mu}}_i^T$ // centered data for class c_i

  for $j = 1, .., d$ **do** // class-specific var for jth attribute

    $\hat{\sigma}_{ij}^2 \leftarrow \frac{1}{n_i} (\overline{X}_j^i)^T (\overline{X}_j^i)$ // variance

  $\hat{\boldsymbol{\sigma}}_i \leftarrow \left(\hat{\sigma}_{i1}^2, \ldots, \hat{\sigma}_{id}^2\right)^T$ // class-specific attribute variances

return $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i$ for all $i = 1, \ldots, k$

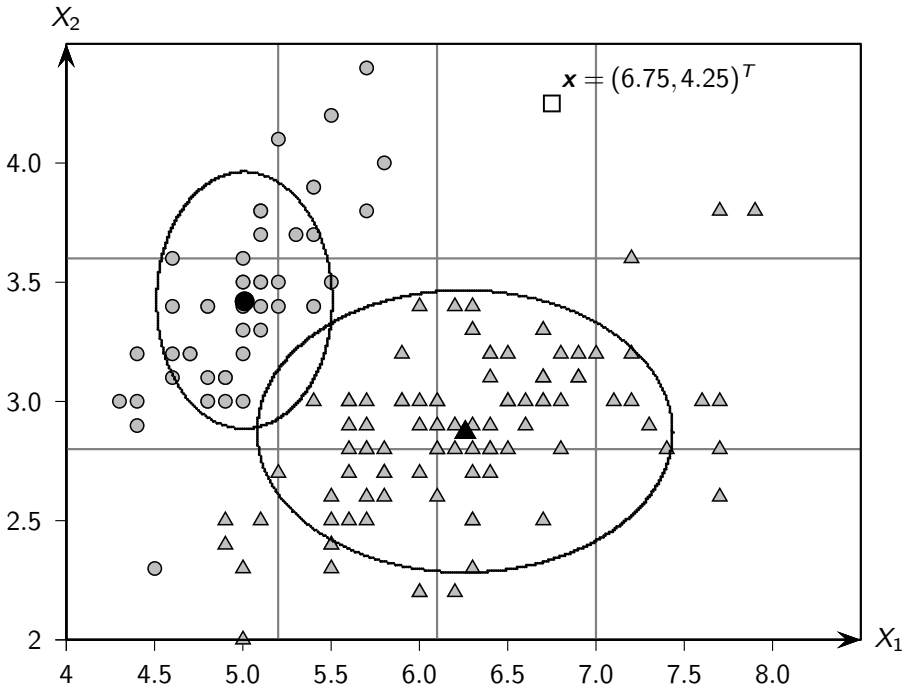
Testing ($\boldsymbol{x}$ and $\hat{P}(c_i)$, $\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\sigma}}_i$, for all $i \in [1, k]$):

$$\hat{y} \leftarrow \arg\max_{c_i} \left\{ \hat{P}(c_i) \prod_{j=1}^{d} f(x_j | \hat{\mu}_{ij}, \hat{\sigma}_{ij}^2) \right\}$$

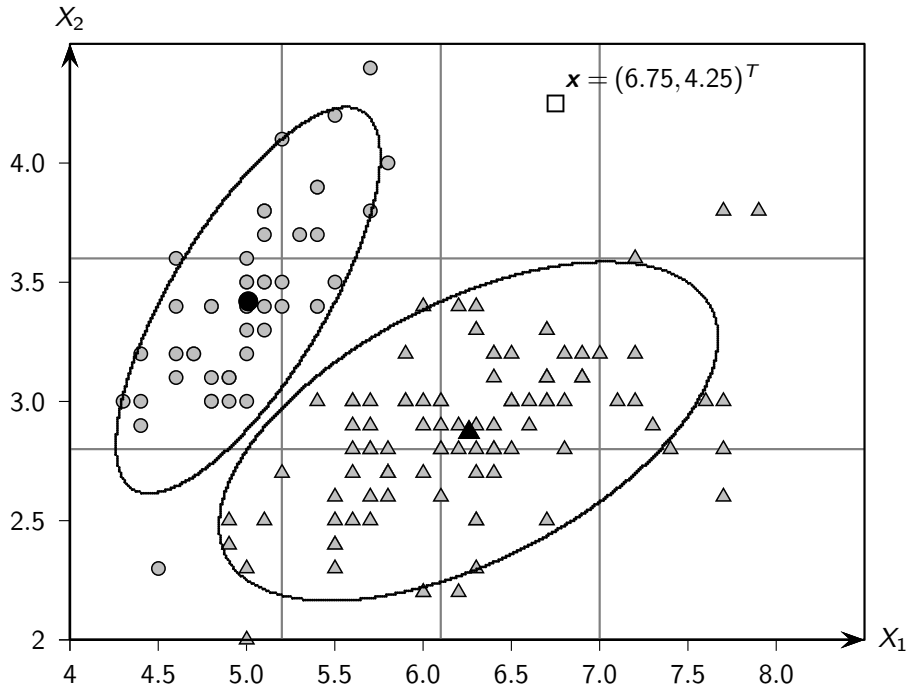
return $\hat{y}$

Naive Bayes versus Full Bayes Classifier

X_1:sepal length versus X_2:sepal width



(a) Naive Bayes



(b) Full Bayes

Data Mining

Paradigms

- Combinatorial
- Probabilistic
- **Algebraic**
- Graph-based
- Connectionist

Domain

Problem is modeled using linear algebra, enabling several existing algebraic models and algorithms.

Task

Determine the best models and their parameters, according to an optimization metric.

Strategies

- Direct
- Iterative

- Principal Component Analysis
- Support Vector Machines

SVM: Linear and Separable Case

Assume that the points are linearly separable, that is, there exists a separating hyperplane that perfectly classifies each point.

The goal of SVMs is to choose the canonical hyperplane, h^*, that yields the maximum margin among all possible separating hyperplanes

$$h^* = \arg\max_{\boldsymbol{w},b} \left\{ \frac{1}{\|\boldsymbol{w}\|} \right\}$$

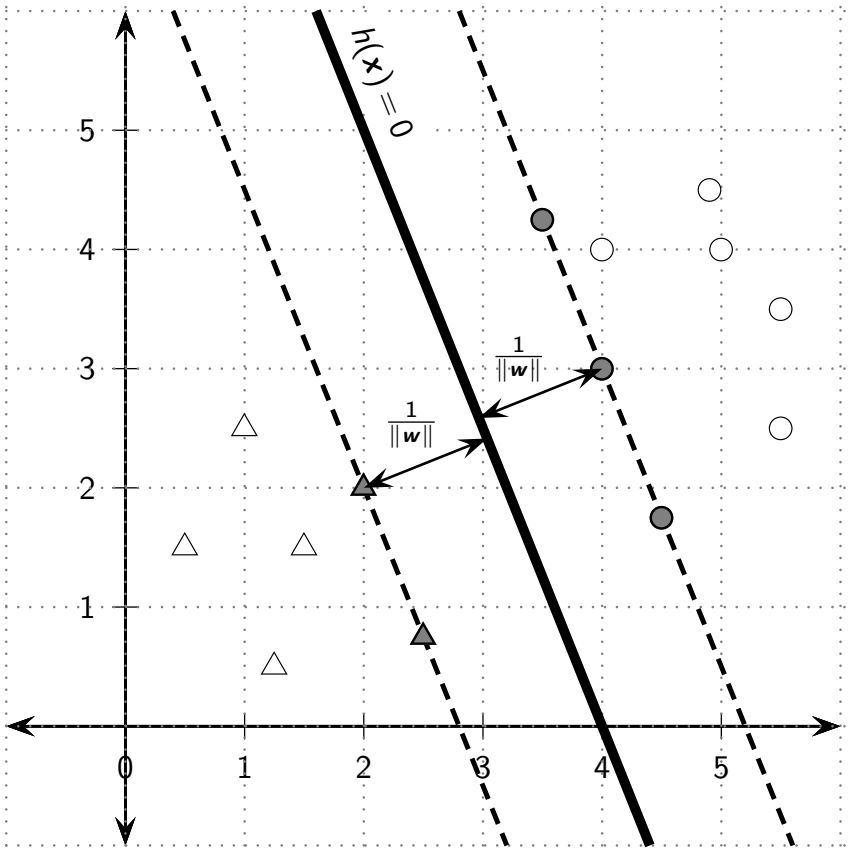
We can obtain an equivalent minimization formulation:

$$\textbf{Objective Function:} \quad \min_{\boldsymbol{w},b} \left\{ \frac{\|\boldsymbol{w}\|^2}{2} \right\}$$

$$\textbf{Linear Constraints:} \quad y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) \geq 1, \ \forall \boldsymbol{x}_i \in \boldsymbol{D}$$

Separating Hyperplane: Margin and Support Vectors

Shaded points are support vectors



$$h(\boldsymbol{x}) = \begin{pmatrix} 5 \\ 2 \end{pmatrix}^T \boldsymbol{x} - 20 = 0$$

Given $\boldsymbol{x}^* = (2,2)^T$, $y^* = -1$.

$$s = \frac{1}{y^* h(\boldsymbol{x}^*)} = \frac{1}{-1\left(\begin{pmatrix} 5 \\ 2 \end{pmatrix}^T \begin{pmatrix} 2 \\ 2 \end{pmatrix} - 20\right)} = \frac{1}{6}$$

$$\boldsymbol{w} = \frac{1}{6}\begin{pmatrix} 5 \\ 2 \end{pmatrix} = \begin{pmatrix} 5/6 \\ 2/6 \end{pmatrix} \qquad b = \frac{-20}{6}$$

$$h(\boldsymbol{x}) = \begin{pmatrix} 5/6 \\ 2/6 \end{pmatrix}^T \boldsymbol{x} - 20/6 = \begin{pmatrix} 0.833 \\ 0.333 \end{pmatrix}^T \boldsymbol{x} - 3.33$$

$$\delta^* = \frac{y^*\, h(\boldsymbol{x}^*)}{\|\boldsymbol{w}\|} = \frac{1}{\sqrt{\left(\frac{5}{6}\right)^2 + \left(\frac{2}{6}\right)^2}} = \frac{6}{\sqrt{29}} = 1.114$$

SVM: Soft Margin or Linearly Non-separable Case

In the nonseparable case, also called the *soft margin* the SVM objective function is

$$\textbf{Objective Function:} \quad \min_{\boldsymbol{w},b,\xi_i} \left\{ \frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{n} (\xi_i)^k \right\}$$

$$\textbf{Linear Constraints:} \quad y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) \geq 1 - \xi_i, \ \forall \boldsymbol{x}_i \in \boldsymbol{D}$$

$$\xi_i \geq 0 \ \forall \boldsymbol{x}_i \in \boldsymbol{D}$$

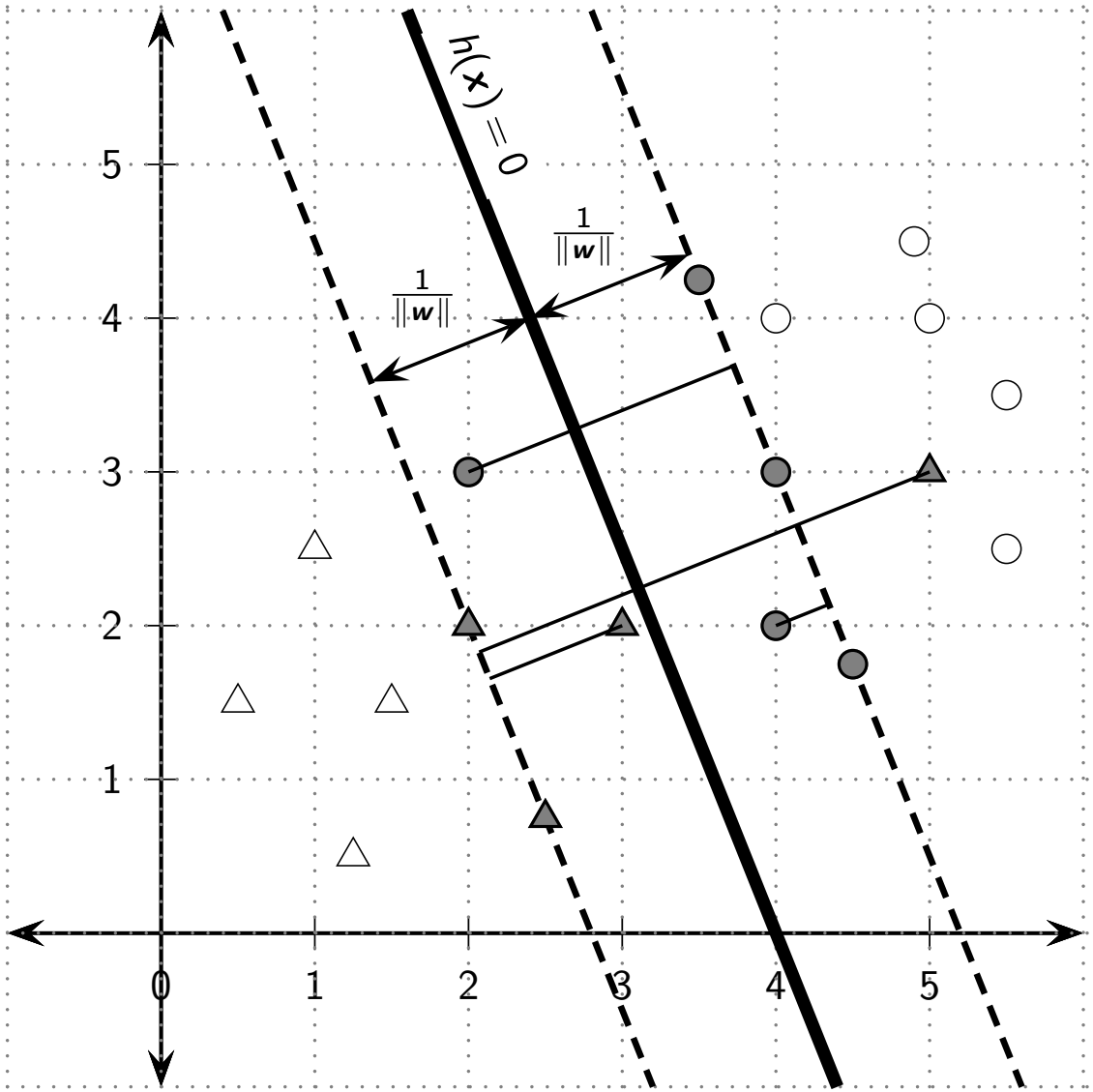
where C and k are constants that incorporate the cost of misclassification.

The term $\sum_{i=1}^{n} (\xi_i)^k$ gives the *loss*, that is, an estimate of the deviation from the separable case.

The scalar C is a *regularization constant* that controls the trade-off between maximizing the margin or minimizing the loss. For example, if $C \to 0$, then the loss component essentially disappears, and the objective defaults to maximizing the margin. On the other hand, if $C \to \infty$, then the margin ceases to have much effect, and the objective function tries to minimize the loss.

Soft Margin Hyperplane

Shaded points are the support vectors



Nonlinear SVMs: Kernel Trick

To apply the kernel trick for nonlinear SVM classification, we have to show that all operations require only the kernel function:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$$

Applying ϕ to each point, we can obtain the new dataset in the feature space $\boldsymbol{D}_\phi = \{\phi(\boldsymbol{x}_i), y_i\}_{i=1}^n$.

The SVM objective function in feature space is given as

$$\textbf{Objective Function:} \quad \min_{\boldsymbol{w}, b, \xi_i} \left\{ \frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{n} (\xi_i)^k \right\}$$

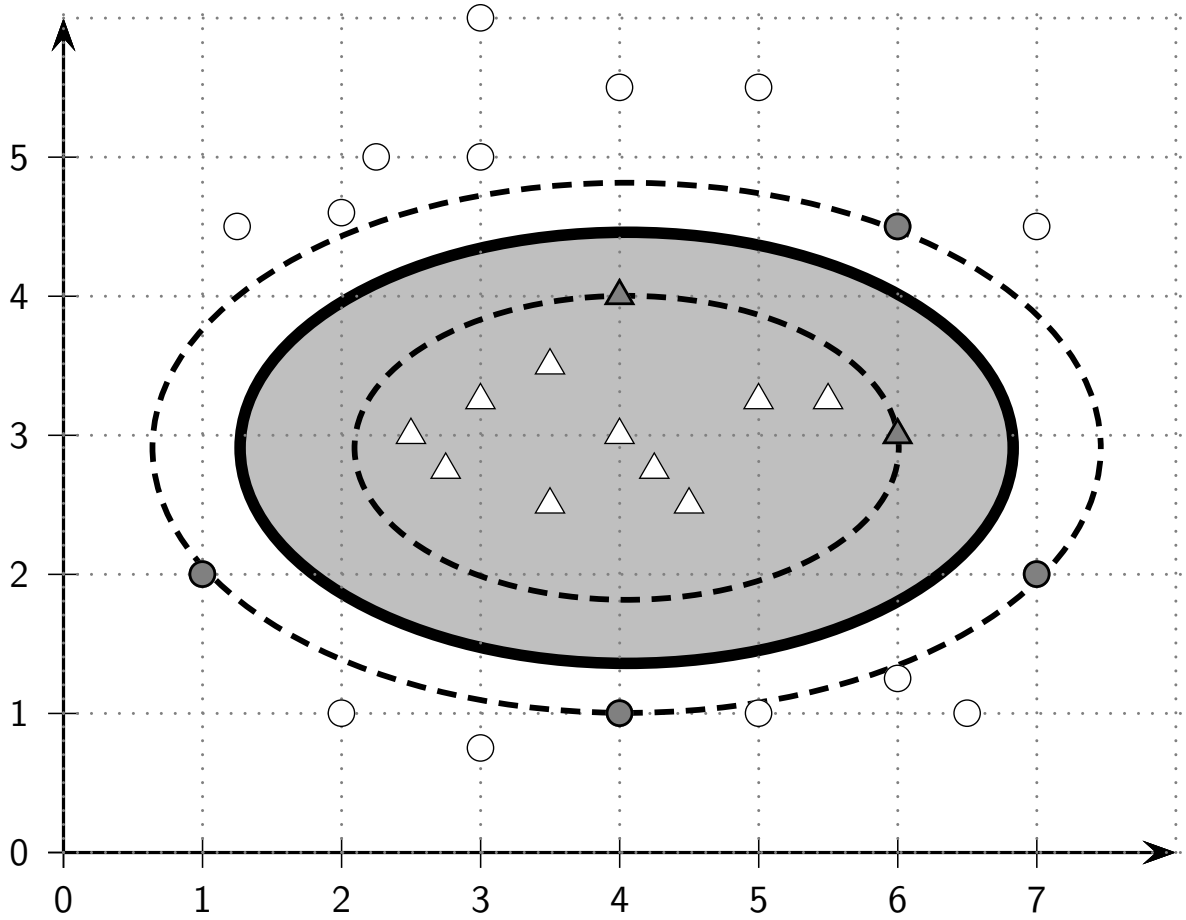
$$\textbf{Linear Constraints:} \quad y_i \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i, \text{and } \xi_i \geq 0, \ \forall \boldsymbol{x}_i \in \boldsymbol{D}$$

where $\boldsymbol{w}$ is the weight vector, b is the bias, and ξ_i are the slack variables, all in feature space.

Nonlinear SVM

There is no linear classifier that can discriminate between the points. However, there exists a perfect quadratic classifier that can separate the two classes.

$$\phi(\boldsymbol{x}) = (\sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^T$$



Data Mining

Paradigms

- Combinatorial
- Probabilistic
- Algebraic
- **Graph-based**
- Connectionist

Domain

Input data is modeled as a graph, enabling not just richer representations but also several existing models and algorithms.

Task

Determine the best representation and technique, according to an optimization metric.

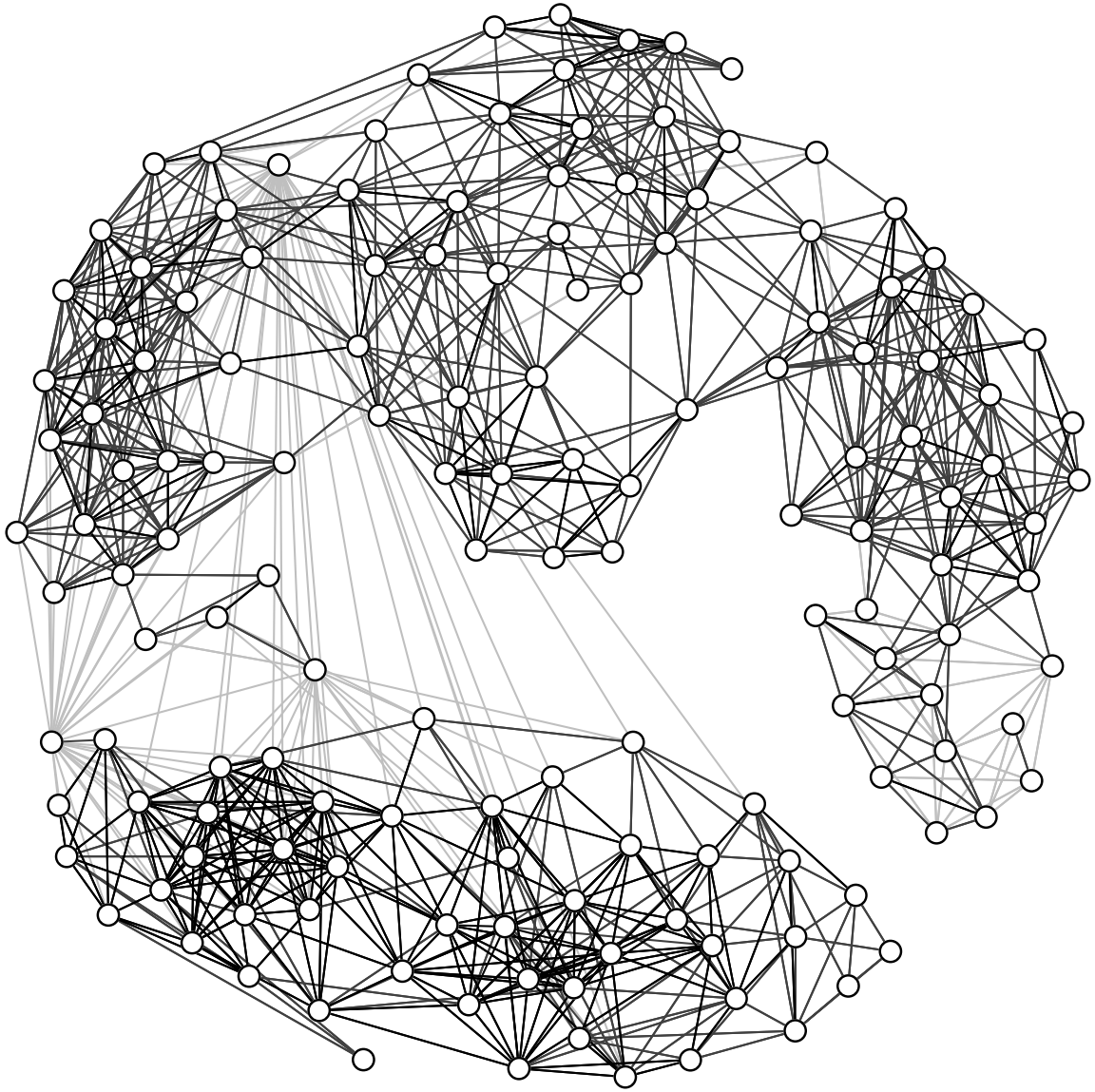
Challenge

How can we handle the larger complexity and numerosity induced by graphs?

- Frequent Subgraph Mining
- Spectral Clustering

Iris Similarity Graph: Mutual Nearest Neighbors

$|V| = n = 150$, $|E| = m = 1730$



Spectral Clustering Algorithm

Spectral Clustering $(\boldsymbol{D}, k)$**:**

1 Compute the similarity matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$

2 **if** *ratio cut* **then** $\boldsymbol{B} \leftarrow \boldsymbol{L}$

3 **else if** *normalized cut* **then** $\boldsymbol{B} \leftarrow \boldsymbol{L}^s$ or $\boldsymbol{L}^a$

4 Solve $\boldsymbol{B}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$ for $i = n, \ldots, n-k+1$, where $\lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_{n-k+1}$

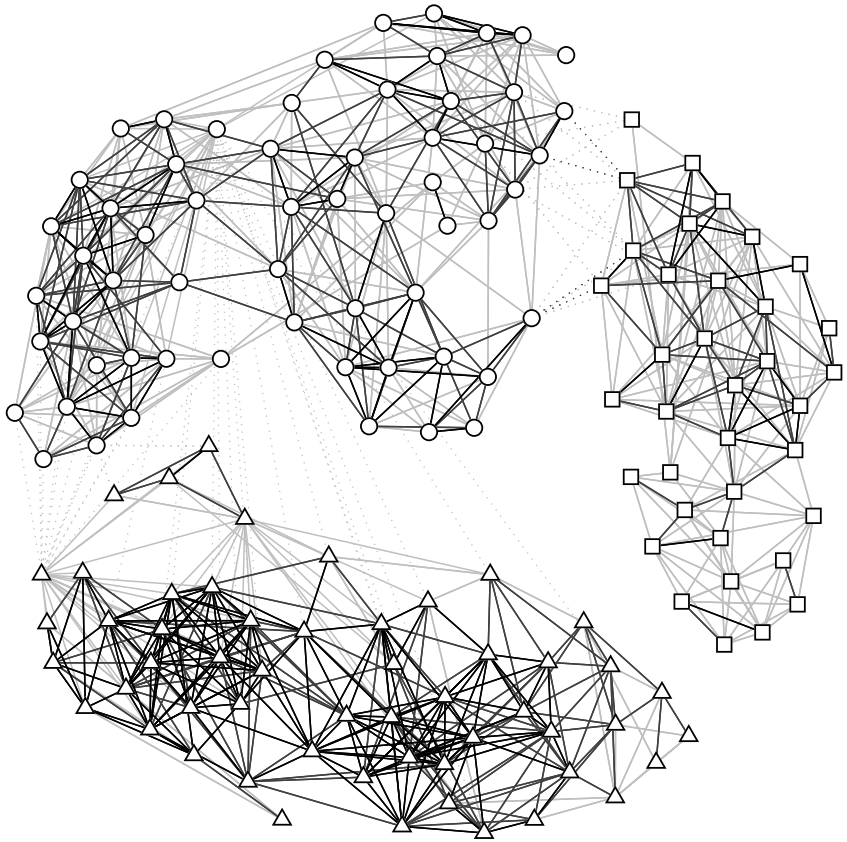
5 $\boldsymbol{U} \leftarrow \begin{pmatrix} \boldsymbol{u}_n & \boldsymbol{u}_{n-1} & \cdots & \boldsymbol{u}_{n-k+1} \end{pmatrix}$

6 $\boldsymbol{Y} \leftarrow$ normalize rows of $\boldsymbol{U}$

7 $\mathcal{C} \leftarrow \{C_1, \ldots, C_k\}$ via K-means on $\boldsymbol{Y}$

Normalized Cut on Iris Graph

$k = 3$, normalized asymmetric Laplacian



	setosa	virginica	versicolor
C_1 (triangle)	50	0	4
C_2 (square)	0	36	0
C_3 (circle)	0	14	46

Data Mining

Paradigms

- Combinatorial
- Probabilistic
- Algebraic
- Graph-based
- **Connectionist**

Domain

Model is built based on a large number of simple functions organized as a topology and weighted according to the training data.

Task

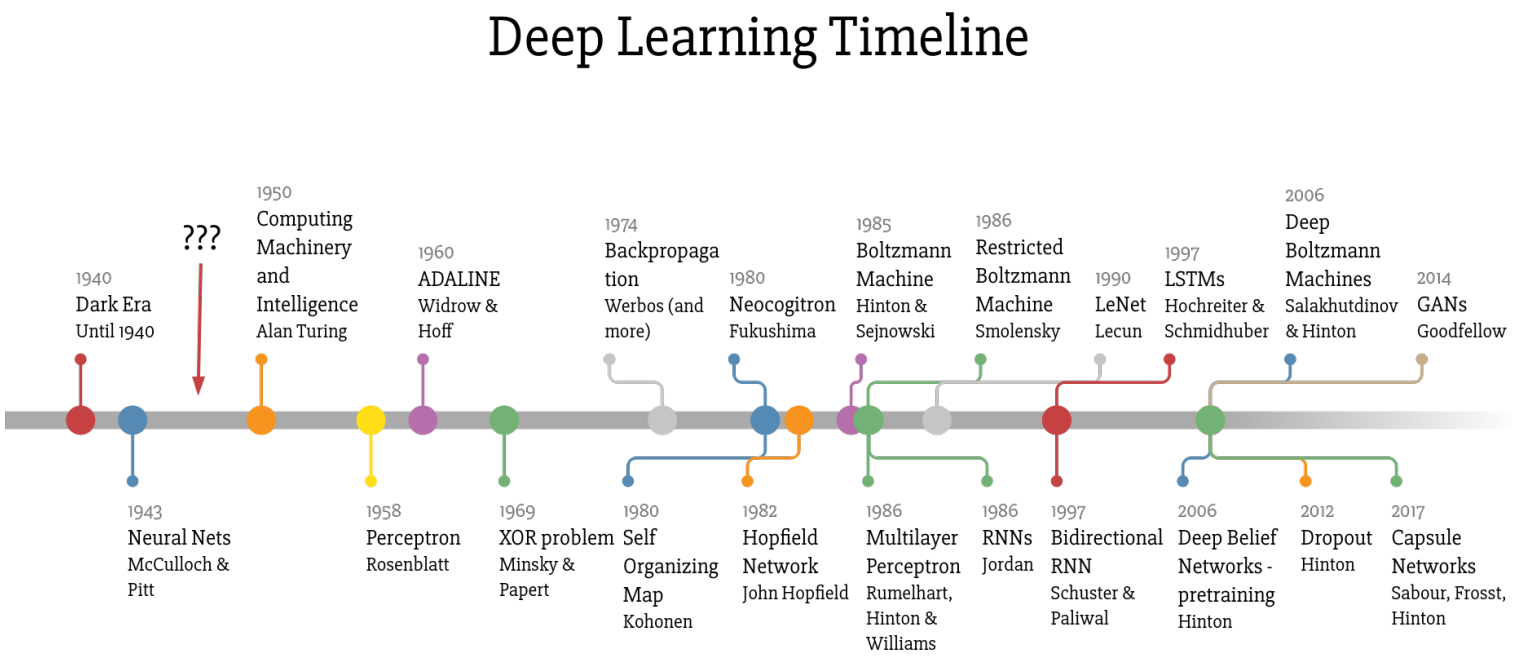
Determine the topology, the activation function, the thresholds and the weights, among other hyperparameters.

Challenges

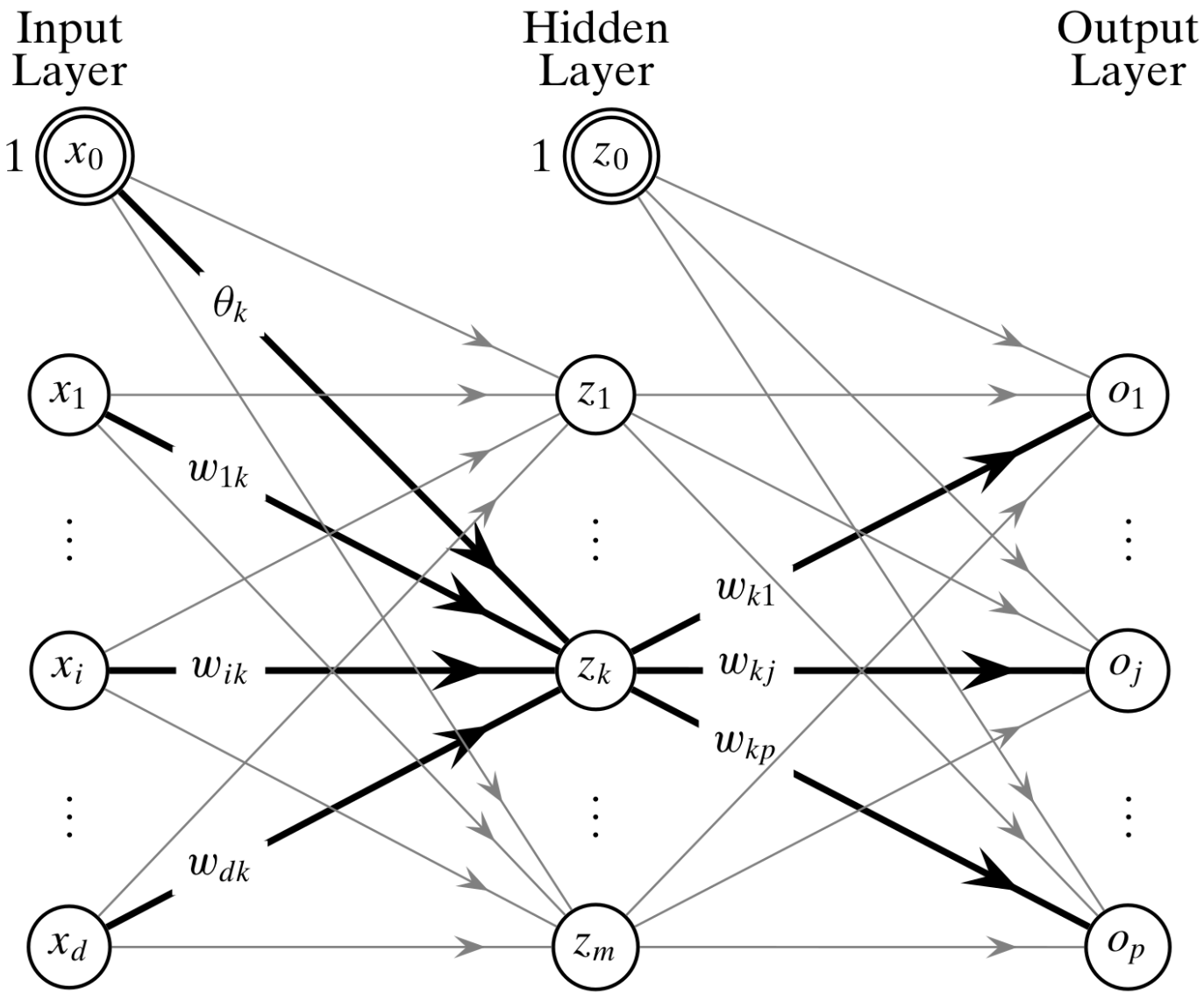
- Design and calibration
- Availability of enough training data
- Interpretability of the results

- Multilayer perceptrons
- Deep multilayer perceptrons
- Recurrent neural networks
- Convolutional networks

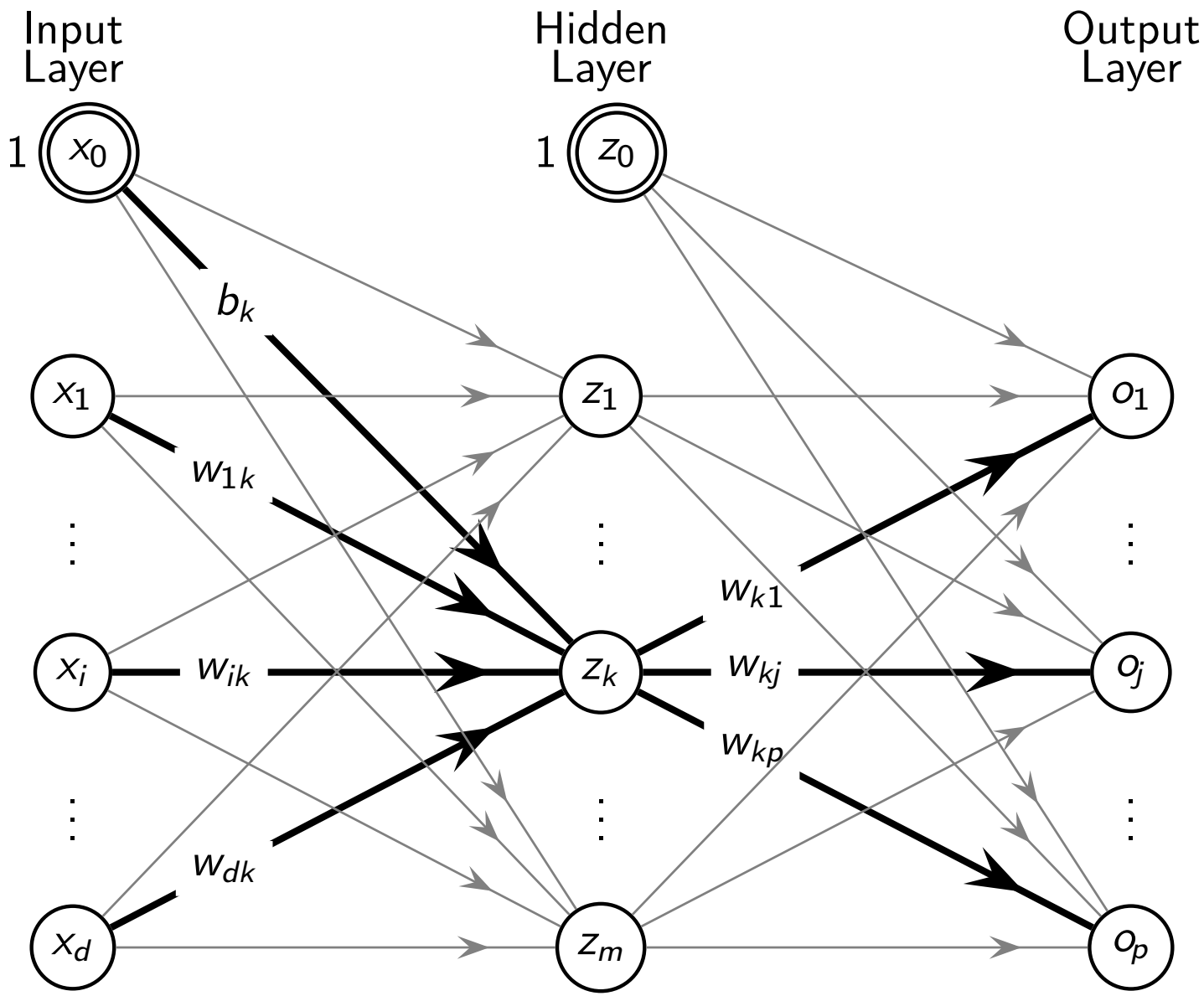
Connectionist Models



Connectionist Models



Multilayer Perceptron: One Hidden Layer



MLP Training: Stochastic Gradient Descent

The MLP training takes multiple iterations over the input points. For each input $\boldsymbol{x}_i$, the MLP computes the output vector $\boldsymbol{o}_i$ via the feed-forward step. In the backpropagation phase, we compute the error gradient vector $\boldsymbol{\delta}_o$ with respect to the net at output neurons, followed by $\boldsymbol{\delta}_h$ for hidden neurons.

In the stochastic gradient descent step, we compute the error gradients with respect to the weights and biases, which are used to update the weight matrices and bias vectors.

MLP-Training ($\boldsymbol{D}, m, \eta,$ `maxiter`):

`// Initialize bias vectors`

1 $\boldsymbol{b}_h \leftarrow$ random m-dimensional vector with small values

2 $\boldsymbol{b}_o \leftarrow$ random p-dimensional vector with small values

`// Initialize weight matrices`

3 $\boldsymbol{W}_h \leftarrow$ random $d \times m$ matrix with small values

4 $\boldsymbol{W}_o \leftarrow$ random $m \times p$ matrix with small values

5 $t \leftarrow 0$ `// iteration counter`

MLP Training: Stochastic Gradient Descent

repeat

  foreach $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \boldsymbol{D}$ *in random order* **do**

    // Feed-forward phase

    $\boldsymbol{z}_i \leftarrow f\left(\boldsymbol{b}_h + \boldsymbol{W}_h^T \boldsymbol{x}_i\right)$

    $\boldsymbol{o}_i \leftarrow f\left(\boldsymbol{b}_o + \boldsymbol{W}_o^T \boldsymbol{z}_i\right)$

    // Backpropagation phase: net gradients

    $\boldsymbol{\delta}_o \leftarrow \boldsymbol{o}_i \odot (1 - \boldsymbol{o}_i) \odot (\boldsymbol{o}_i - \boldsymbol{y}_i)$

    $\boldsymbol{\delta}_h \leftarrow \boldsymbol{z}_i \odot (1 - \boldsymbol{z}_i) \odot (\boldsymbol{W}_o \cdot \boldsymbol{\delta}_o)$

    // Gradient descent for bias vectors

    $\nabla_{\boldsymbol{b}_o} \leftarrow \boldsymbol{\delta}_o; \quad \boldsymbol{b}_o \leftarrow \boldsymbol{b}_o - \eta \cdot \nabla_{\boldsymbol{b}_o}$

    $\nabla_{\boldsymbol{b}_h} \leftarrow \boldsymbol{\delta}_h; \quad \boldsymbol{b}_h \leftarrow \boldsymbol{b}_h - \eta \cdot \nabla_{\boldsymbol{b}_h}$

    // Gradient descent for weight matrices

    $\nabla_{\boldsymbol{W}_o} \leftarrow \boldsymbol{z}_i \cdot \boldsymbol{\delta}_o^T; \quad \boldsymbol{W}_o \leftarrow \boldsymbol{W}_o - \eta \cdot \nabla_{\boldsymbol{W}_o}$

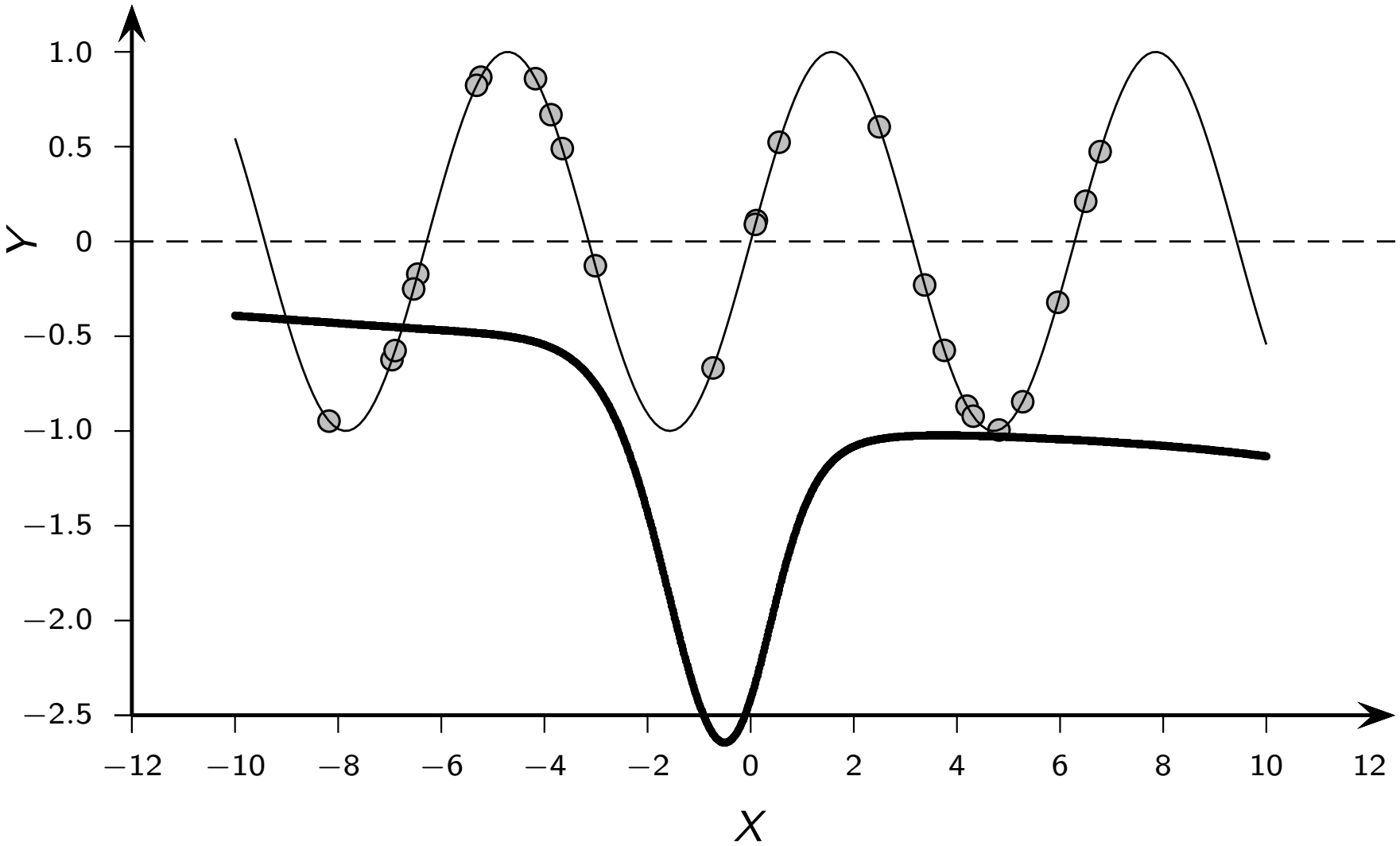
    $\nabla_{\boldsymbol{W}_h} \leftarrow \boldsymbol{x}_i \cdot \boldsymbol{\delta}_h^T; \quad \boldsymbol{W}_h \leftarrow \boldsymbol{W}_h - \eta \cdot \nabla_{\boldsymbol{W}_h}$

  $t \leftarrow t + 1$

until $t \geq$ maxiter

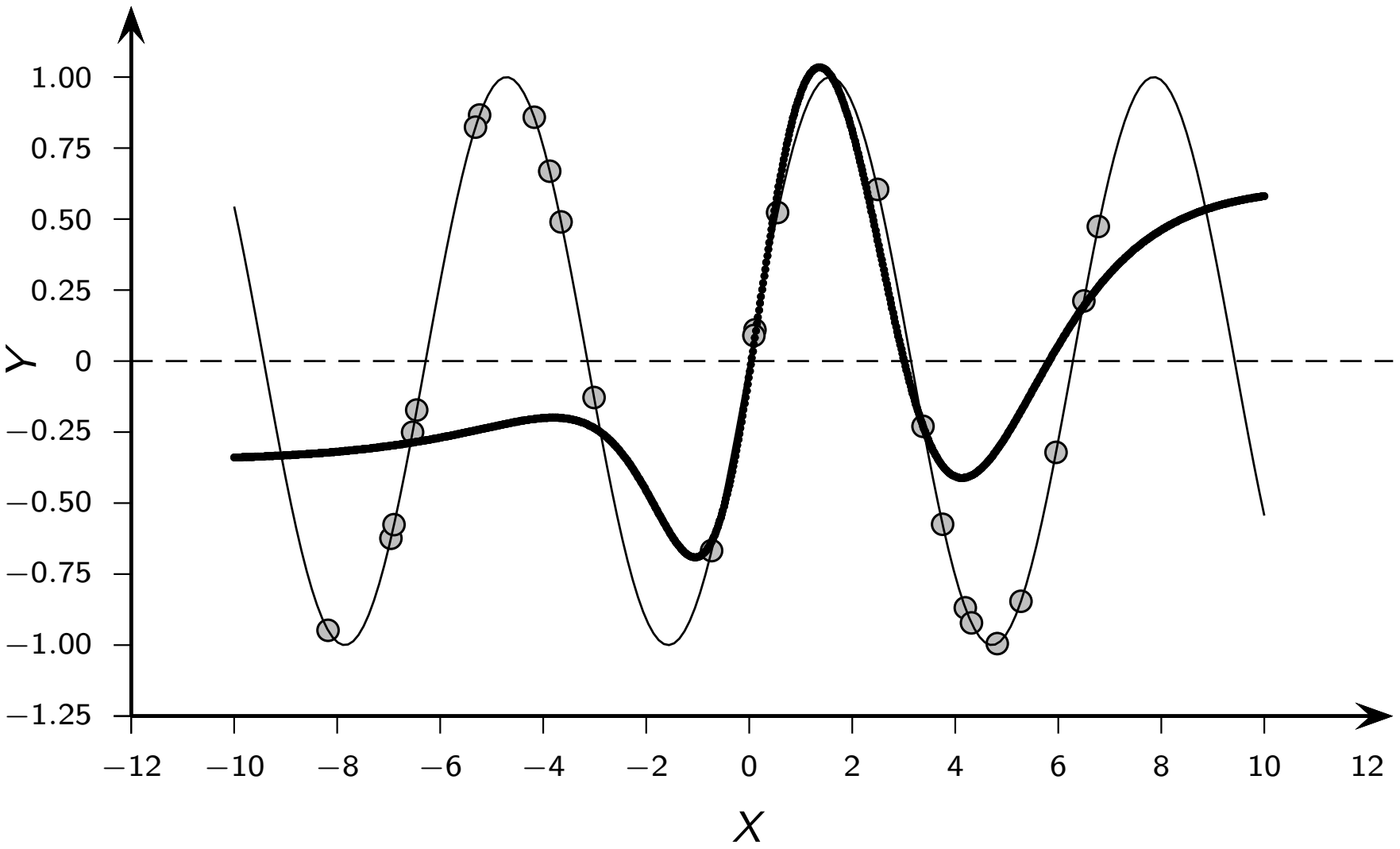
MLP for sine curve

$t = 1$



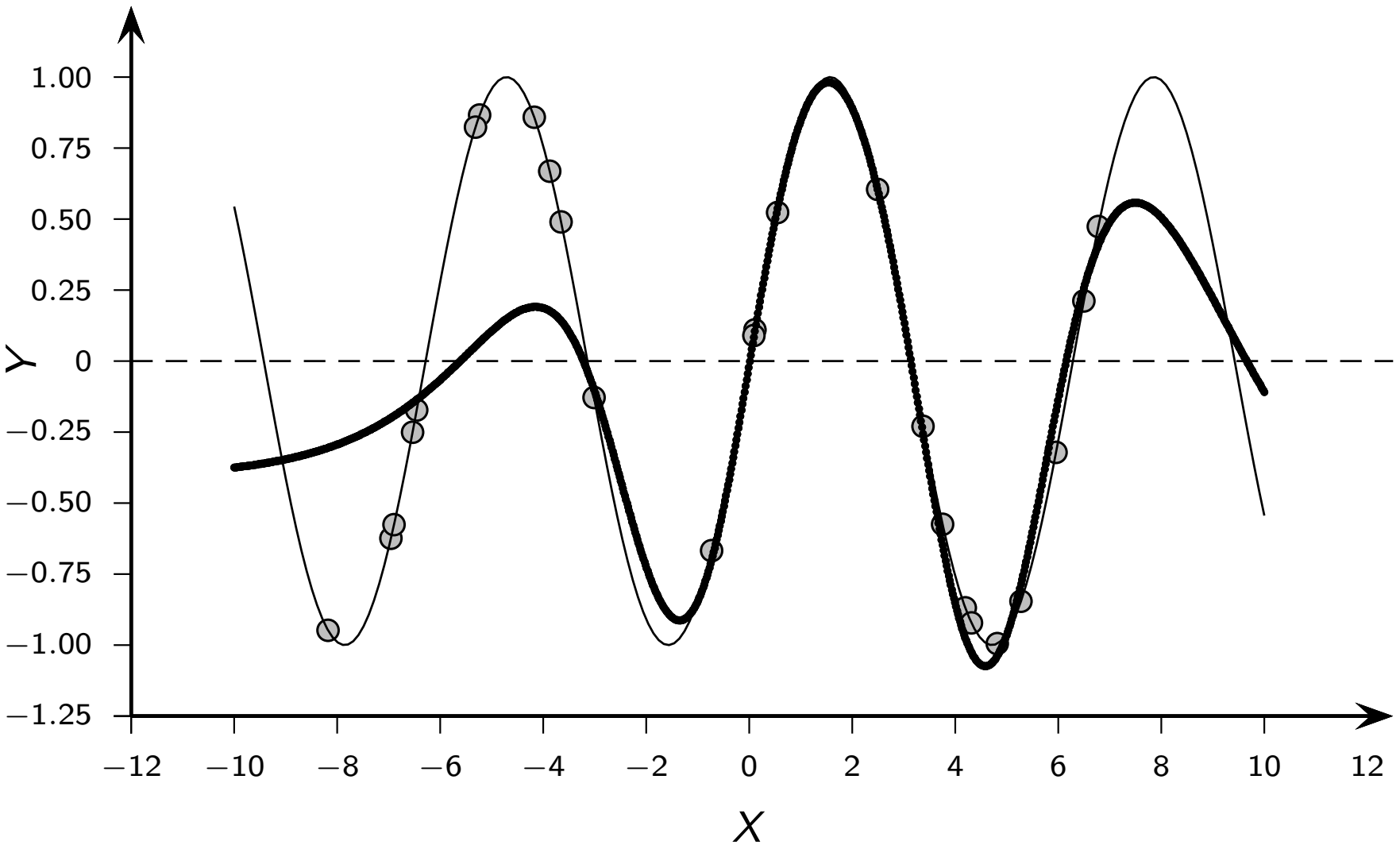
MLP for sine curve

$t = 1000$



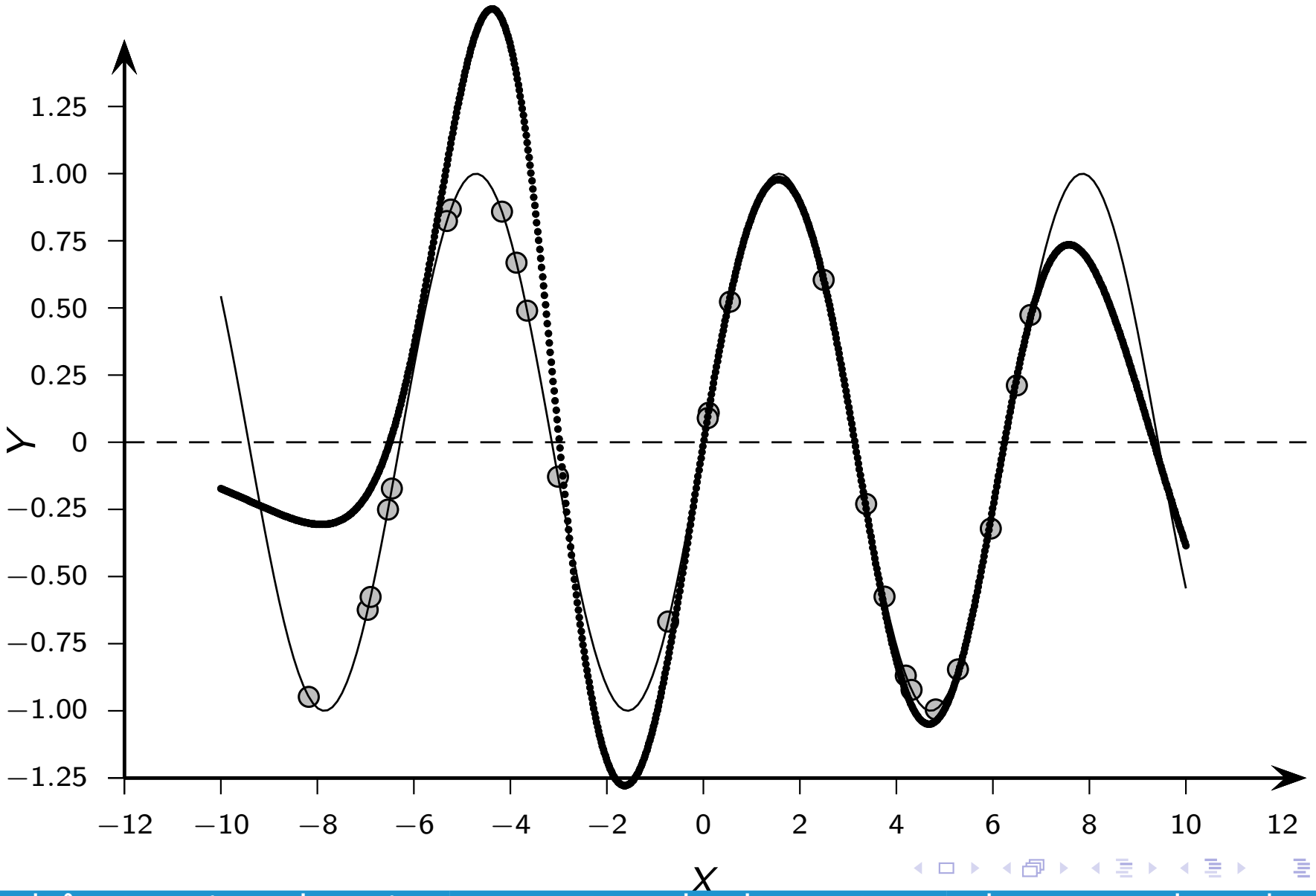
MLP for sine curve

$t = 5000$



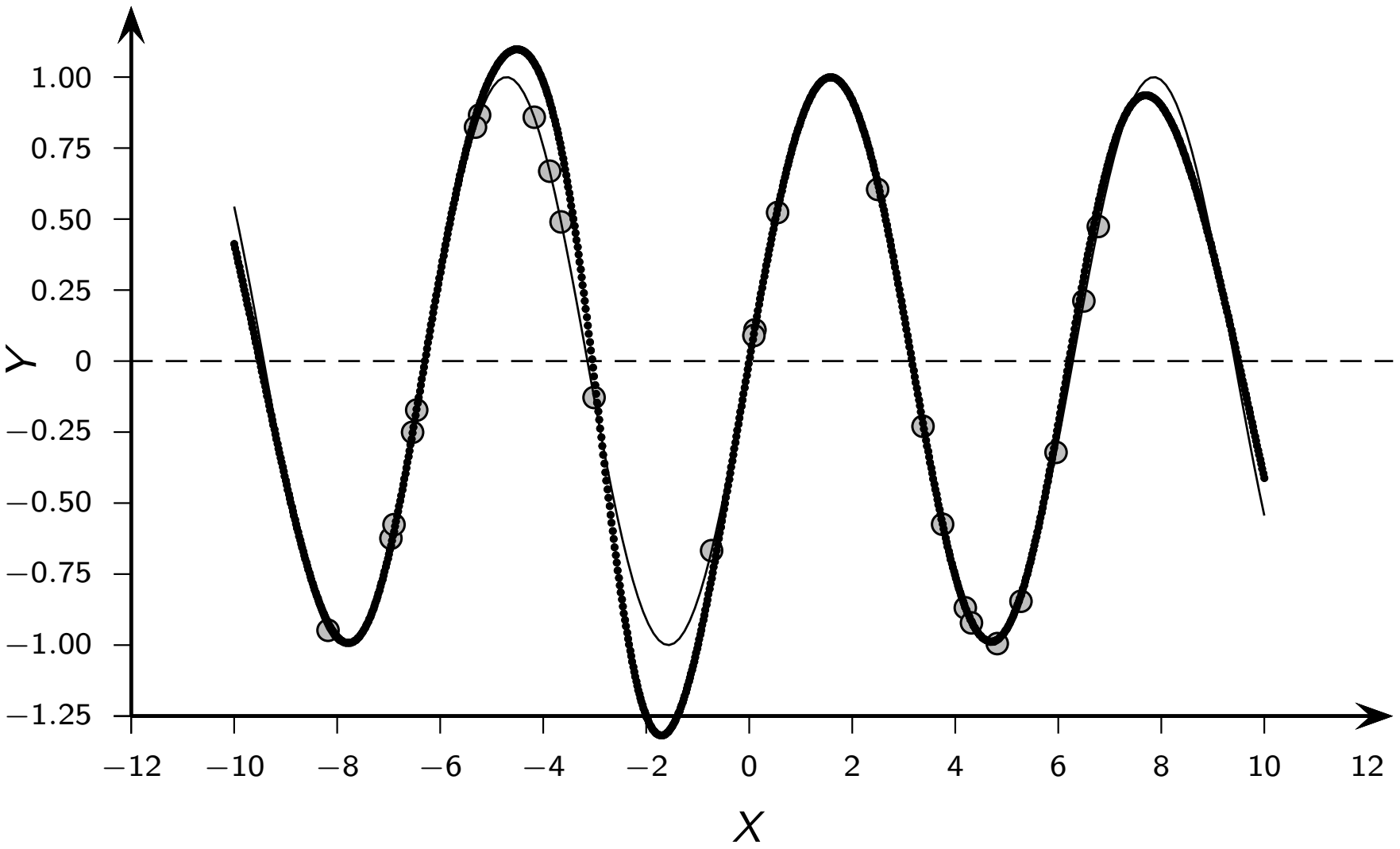
MLP for sine curve

$t = 10000$



MLP for sine curve

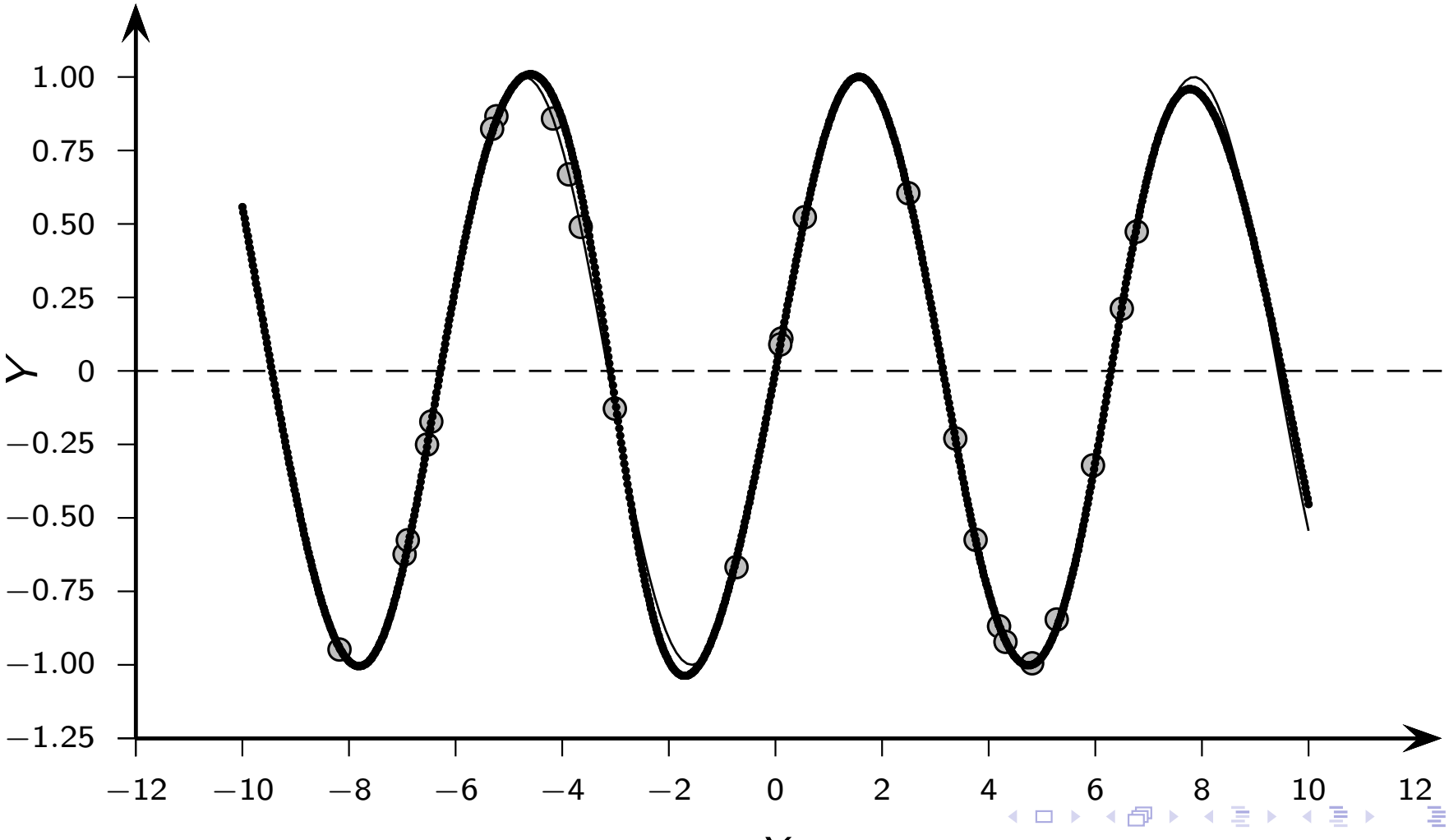
$t = 15000$



MLP for sine curve

$t = 30000$

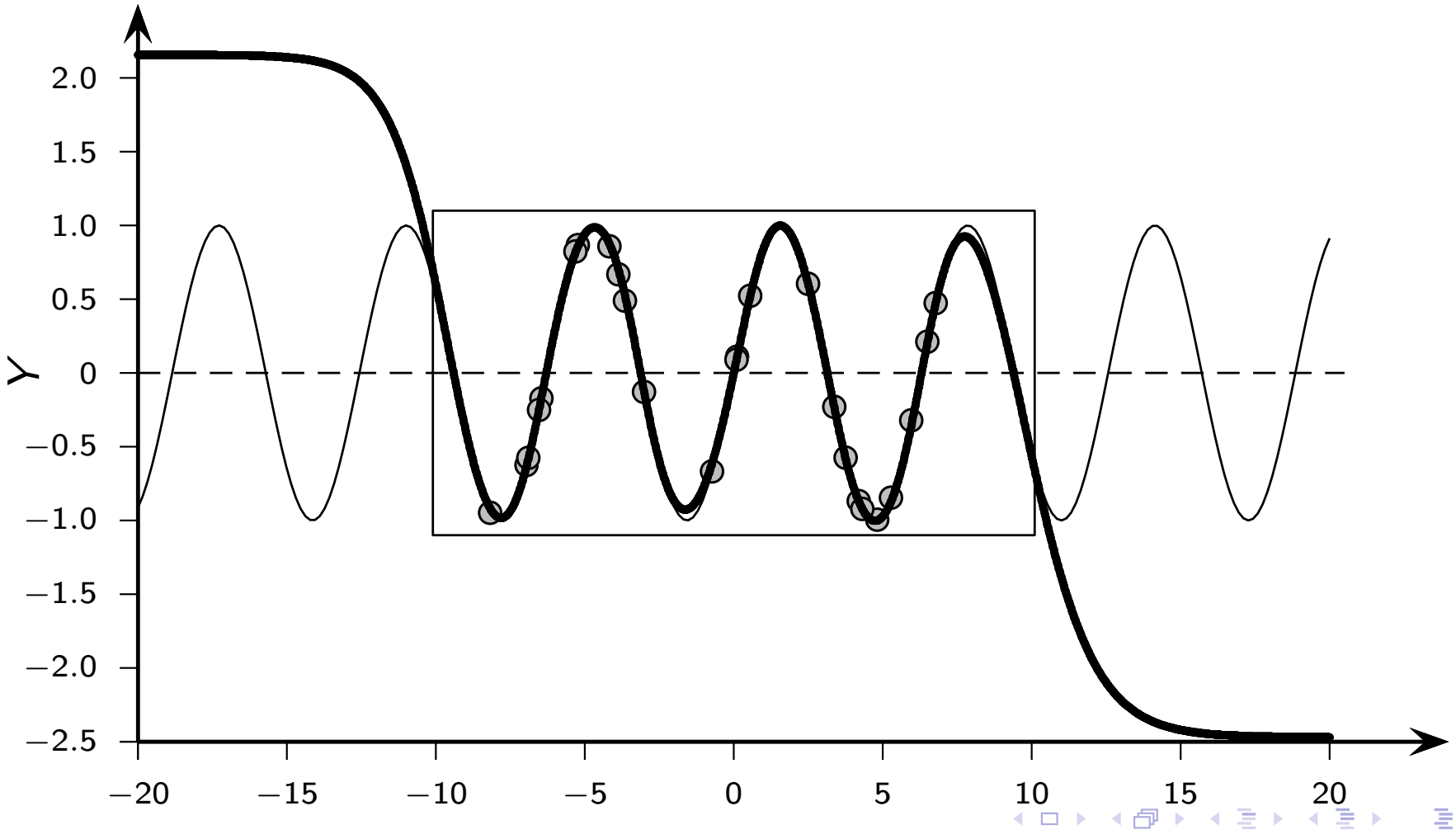
We can observe that, even with a very small training data of 25 points sampled randomly from the sine curve, the MLP is able to learn the desired function.



MLP for sine curve

Test range $[-20, 20]$

The MLP model has not really learned the sine function; rather, it has learned to approximate it only in the specified range $[-10, 10]$.



Summary

- Data availability is not a problem anymore, but data quality assurance as well as analyzing and understanding them still is.
- Techniques may be organized into paradigms according to their principles, that is, a single paradigm may support several techniques.
- Paradigms are complementary and may be combined for a single technique.
- Understanding the paradigms and their characteristics is key to mine data effectively.
- Understanding the paradigms may also help significantly regarding grasping, exploiting and developing new techniques.