Machine Learning and Data Mining
(IT4242E)

Quang Nhat NGUYEN

quang.nguyennhat@hust.edu.vn

Hanoi University of Science and Technology

School of Information and Communication Technology

Academic year 2021-2022

The course's content:

- **Introduction**
  - **Machine learning**
  - **Data mining**
  - **Practical applications**
  - **Software frameworks and tools**
- Performance evaluation of the ML/DM system
- Regression problem
- Classification problem
- Clustering problem
- Association rule mining problem

Machine learning vs. Data mining

- Similarities:
  - Need to use data, and usually a (very) large amount of data
  - Discover knowledge from data
- Differences:

	Machine learning	Data mining
Focus:	On the **learning** of the computer	On the **understanding** of the **data**
Use goal:	To make **predictions in future**	To analyze the **current (past) data**

Introduction of Machine learning

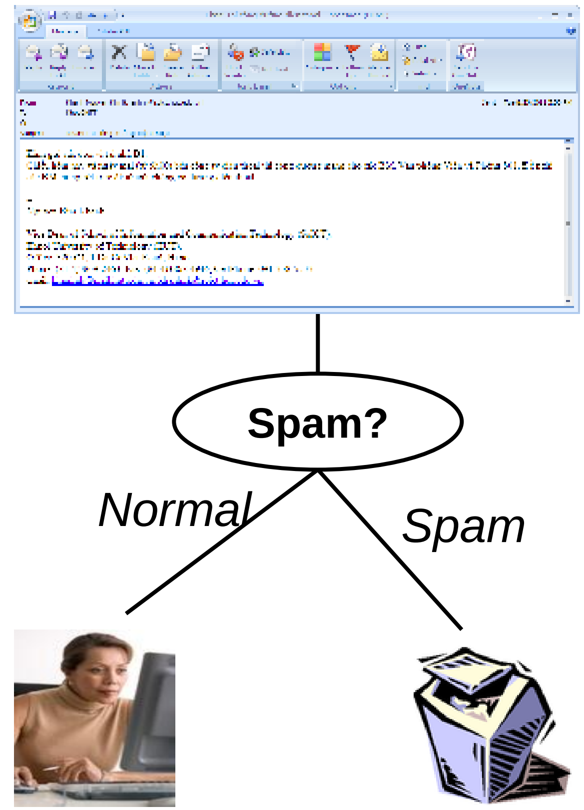
- Machine Learning (ML) is a traditional and very active field of Artificial Intelligence (AI)
- Some examples of definition of ML
  - → A process by that a system improves its performance `[Simon, 1983]`
  - → A process by that a computer program improves its performance in a task through experience `[Mitchell, 1997]`
  - → A programming of computers to improve a performance criterion based on past sample data or experience `[Alpaydin, 2004]`
- Representation of a ML problem `[Mitchell, 1997]`

  ML = Improvement of a task's efficiency through experience
  - A task ***T***
  - For the evaluation criteria of performance ***P***
  - By using some experience ***E***

Example of ML problem (1)

Email spam filtering:

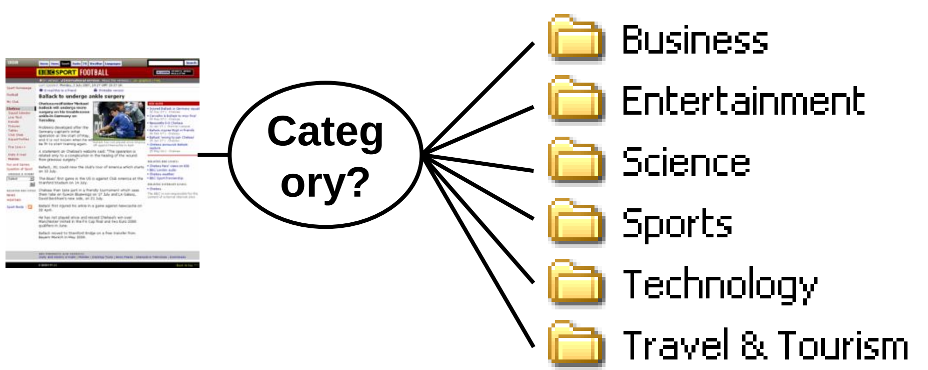
- ***T***: To predict (i.e., to filter) spam emails
- ***P***: % of correctly classified (i.e., predicted) incoming emails
- ***E***: A set of sample emails, where each email is represented by a set of attributes (e.g., a set of keywords) and its corresponding label (i.e., normal or spam)



Example of ML problem (2)

Web page categorization (classification):

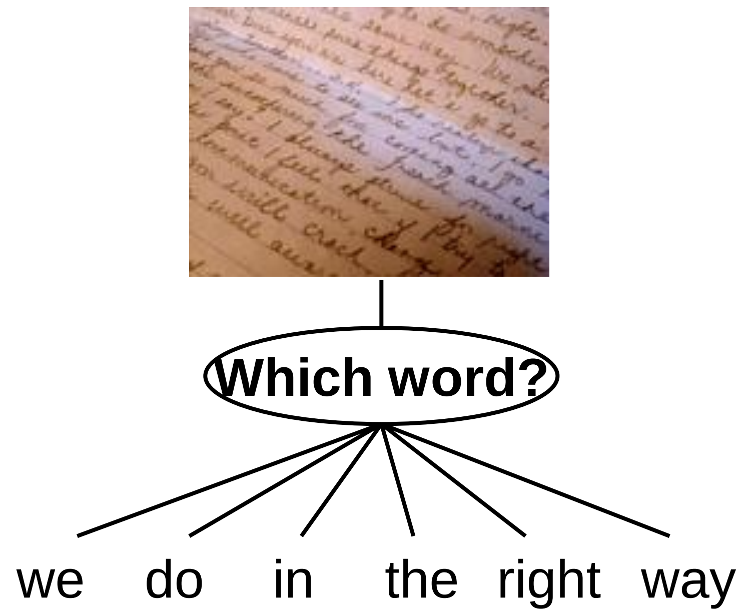
- ***T***: To categorize Web pages in predefined categories
- ***P***: % of correctly categorized Web pages
- ***E***: A set of Web pages, and each one associates with a category



Example of ML problem (3)

Handwritten characters recognition

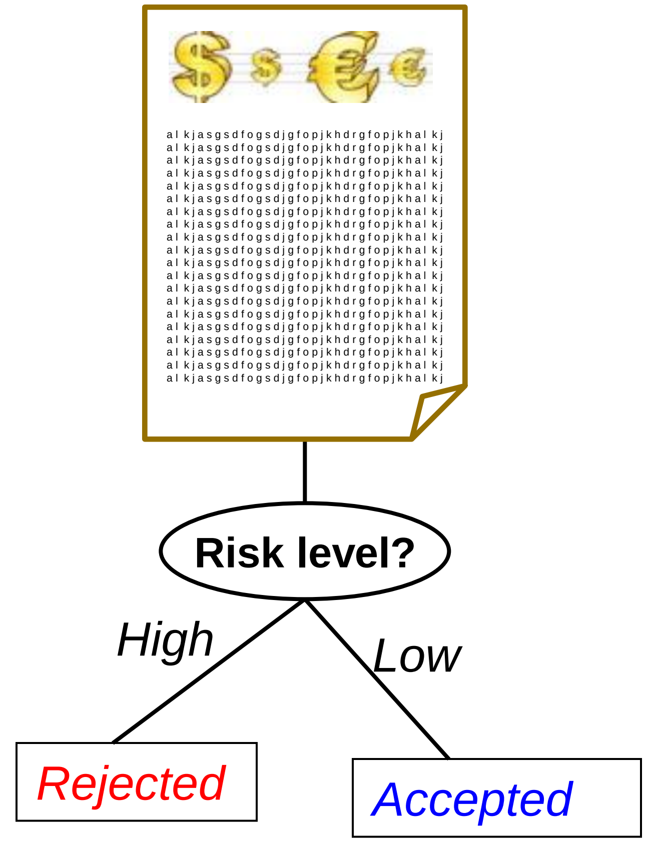
- ***T***: To recognize the words that appear in a captured image of a handwritten document
- ***P***: % of correctly recognized words
- ***E***: A set of captured images of handwritten words, where each image associates with a word's label (ID)



Example of ML problem (4)

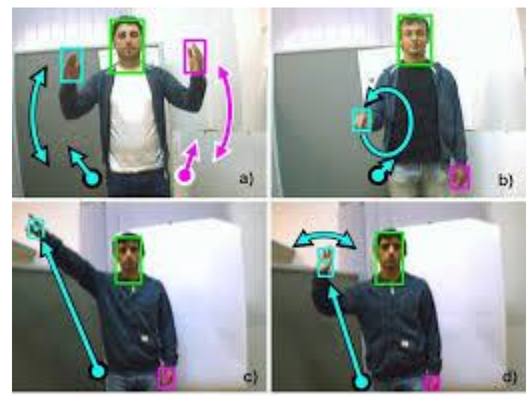
Risk estimation of loan application:

- ***T***: To estimate the level (e.g., high or low) of risk of a loan application
- ***P***: % of correctly estimated high-level-risk loan applications (i.e., those do not return the loans, or returns in a long delay)
- ***E***: A set of loan applications, where each loan application is represented by a set of attributes and a risk level value (high/low)



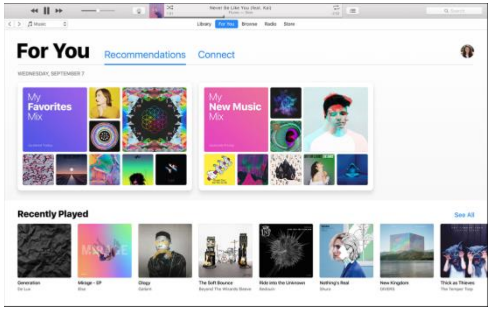
Successful applications of ML in practice (1)

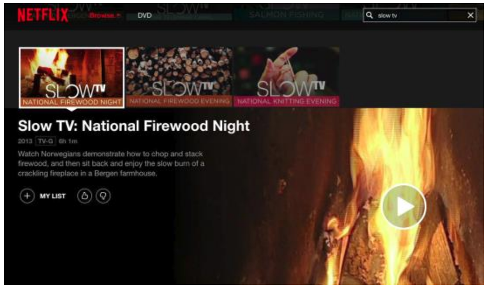
- Human-machine communication
  - Voice, Gesture, Language understanding, …



Successful applications of ML in practice (2)

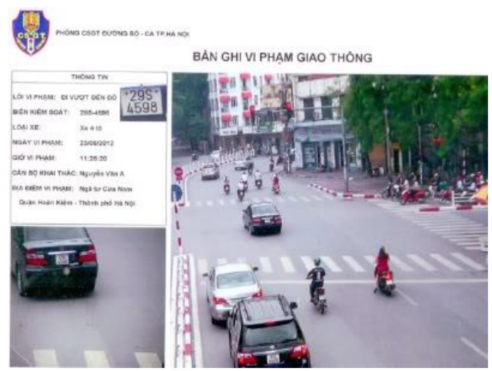
- Entertainment
  - Music, Movies, Games, News, Social networks, …





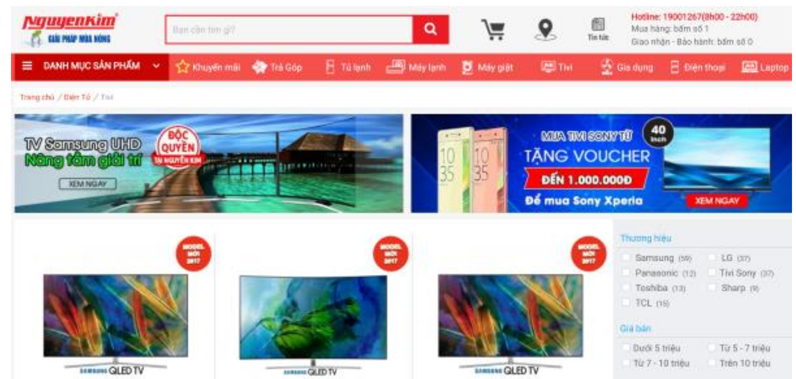
Successful applications of ML in practice (3)

- Transportation
  - Automatic car, Traffic surveillance, Car ride demand estimation, …



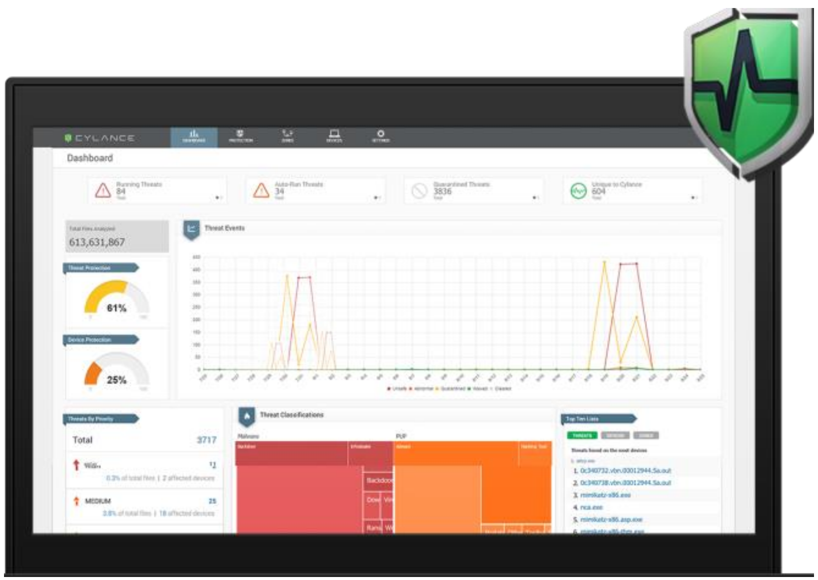
Successful applications of ML in practice (4)

- E-commerce
  - Recommendation of products and services, Customer need prediction, Promotion campaigns, …



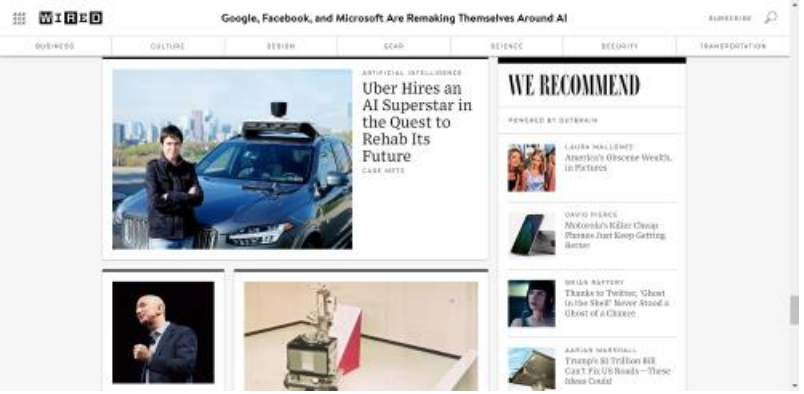
Successful applications of ML in practice (5)

- System security
  - Computer virus detection, Network intrusion detection, Spam email filtering,…

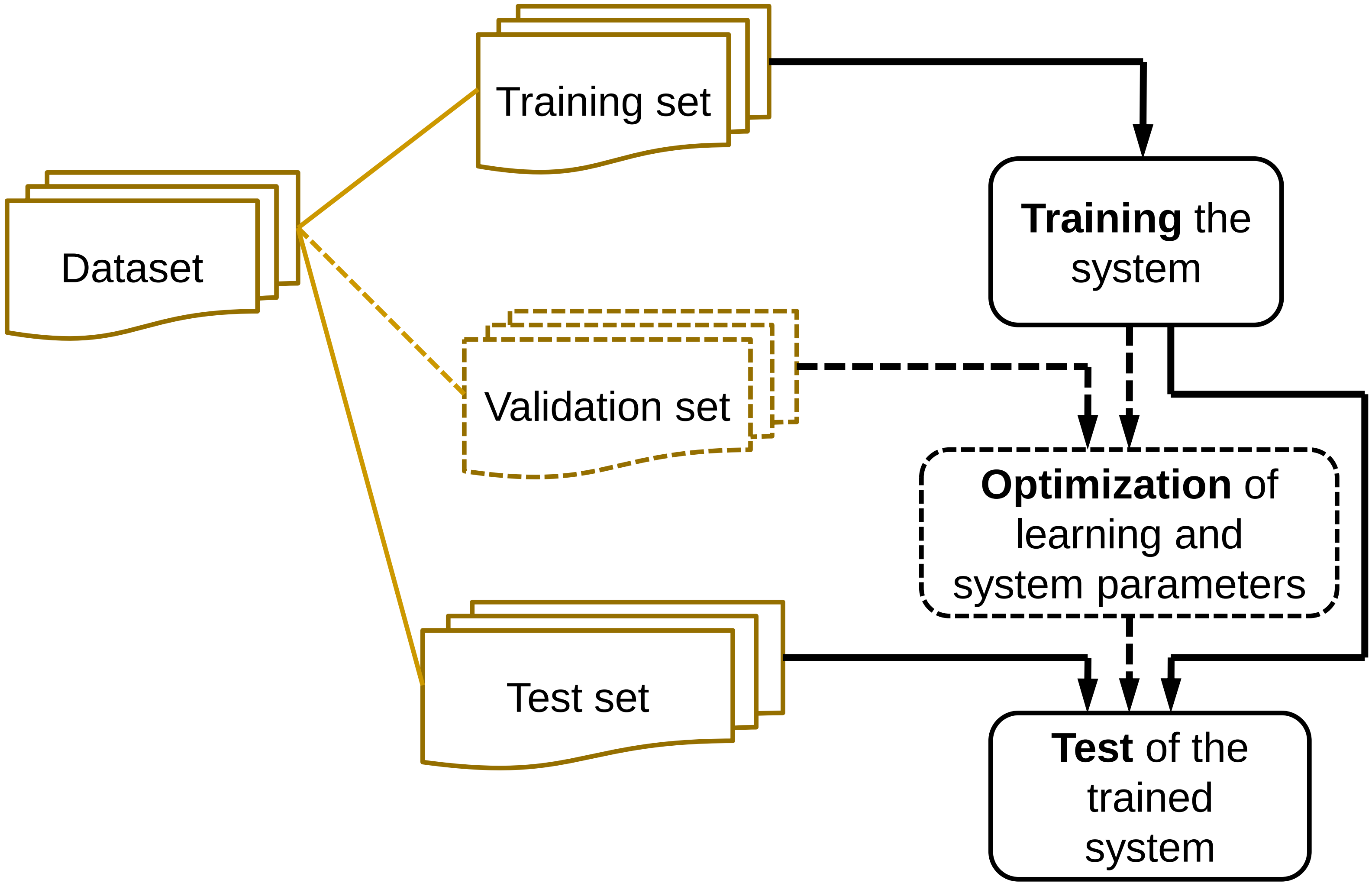


Successful applications of ML in practice (6)

- Marketing and advertisement



Machine learning process



Main elements of ML problem (1)

- Training (learning) examples
  - The training feedback is included in training examples or indirectly provided (e.g., from the working environment)
  - They are supervised or unsupervised training examples
  - The training examples should be compatible with (i.e., representative for) the future test examples
- The target function to be learned
  - F: $X \rightarrow \{0,1\}$
  - F: $X \rightarrow$ A set of class labels
  - F: $X \rightarrow R^+$ (i.e., a domain of positive real values)
  - …

Main elements of ML problem (2)

- Representation of the target function to be learned
  - A polynomial function
  - A set of rules
  - A decision tree
  - An artificial neural network
  - …
- ML algorithm that can learn *approximately* the target function
  - Regression-based
  - Rule induction
  - Decision tree learning (e.g., ID3 or C4.5)
  - Back-propagation
  - …

Challenges in ML (1)

- Learning algorithm
  - Which learning algorithms can learn approximately a given target function?
  - Under which conditions, a selected learning algorithm converges (approximately) the target function?
  - For a specific application problem and a specific example (object) representation, which learning algorithm performs best?

Challenges in ML (2)

- Training examples
  - How many training examples are enough for the training?
  - How does the size of the training set (i.e., the number of training examples) affect the accuracy of the learned target function?
  - How do error (noise) and/or missing-value examples affect the accuracy?

Challenges in ML(3)

- Learning process
  - What is the best ways of use order of training examples?
  - How does the order of using training examples vary the complexity of the ML problem?
  - How does the application problem-specific knowledge (apart from the training examples) contribute to the machine learning process?

Challenges in ML (4)

- Learning capability
  - Which target function the system should learn?
    - Representation of the target function: Representation capability (e.g., linear / non-linear function) vs. Complexity of the learning algorithm and learning process
  - The theorical limits for the learning capability of learning algorithms?
  - The system's capability of generalization from the training examples?
    - **Under-fitting** problem
    - **Over-fitting** problem
  - The system's capability of self-adapting its internal architectural representation?
    - To improve the system's capability of representation and learning of the target function

Challenges in ML (5)

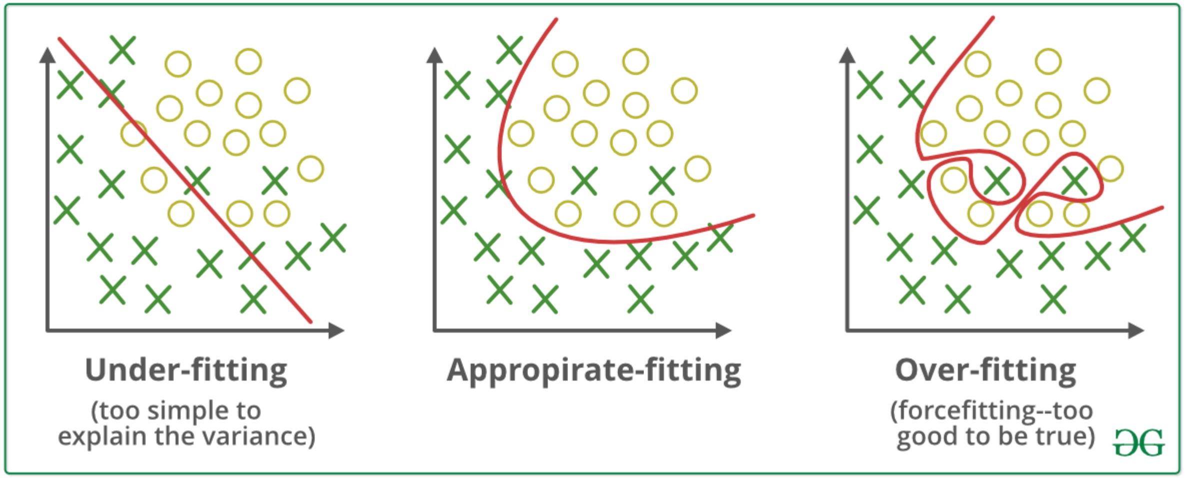
- **WHEN** should a trained model be re-trained?
  - The trained model has performed well on the past examples
  - But at a certain time, the trained model performs <u>significantly poor</u> on the newly coming examples
- **HOW** should a trained model be re-trained?
  - To adapt to the newly coming examples

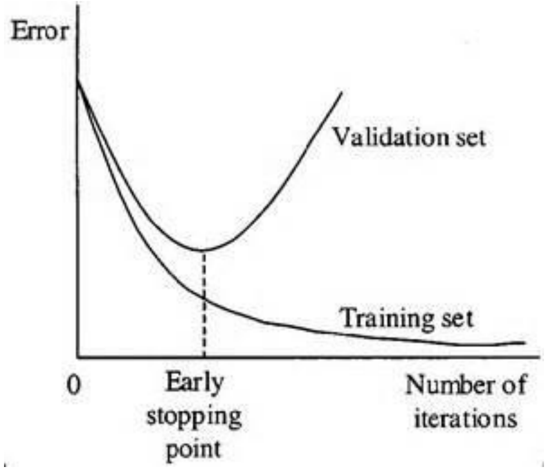
Generalization capability (1)

- Generalization shows the ability of the model to still achieve high accuracy <u>for future (unseen) data</u>
  - <u>Note</u>: We cannot use any test examples during model selection/training!
  - Use the **validation set** (often extracted from (as a small part of) the original training set) to serve as unseen data in the model training/selection
    - <u>Assumption</u>: The data characteristics are similar between the validation and test sets!

Generalization capability (2)

- 2 common (and should be avoided!) problems of generalization:
  - **Under-fitting**: Achieve low accuracy on all the training, validation and test sets
    - Often make false conclusions (i.e., the "*high bias*" characteristic)
  - **Over-fitting**: Achieve high accuracy on the training set, but low accuracy on the validation and test sets
    - Tend to make different conclusions for the same (or rather similar) examples (i.e., the "*high variance*" characteristic)





(https://towardsdatascience.com/underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6fe4a8a49dbf)

Problem of over-fit learning (1)

- A learned target function *h* is considered **over-fit** to a specific training set if there exists another target function *h'* such that:
  - *h'* produces lower accuracy than *h* for the training set, but
  - *h'* produces higher accuracy than *h* for the whole dataset (including also those examples that are evaluated after the training process)

Problem of over-fit learning (2)

- Assume that `D` is the whole dataset, and `D_train` the training set
- Assume that $\mathrm{Err}_{D}(h)$ is the error caused by the target function `h` on `D`, and $\mathrm{Err}_{D\_train}(h)$ is the error caused by the target function `h` on `D_train`
- The target function `h` is over-fit to `D_train` if there exists another target function `h'`:
  - $\mathrm{Err}_{D\_train}(h) < \mathrm{Err}_{D\_train}(h')$, and
  - $\mathrm{Err}_{D}(h) > \mathrm{Err}_{D}(h')$

Problem of over-fit learning (3)

- The problem of over-fit learning is often caused by:
  - Errors (noises) in the training set (i.e., by a collection/construction of the training set)
  - The number of training examples is too small, or not representative for the overall distribution of all the examples of the learning problem
  - The accuracy is too high/ideal (~100%) for the training set – The training process converges at a target function that is ideal/perfect for the training examples (<u>but not good for future/unseen examples</u>)

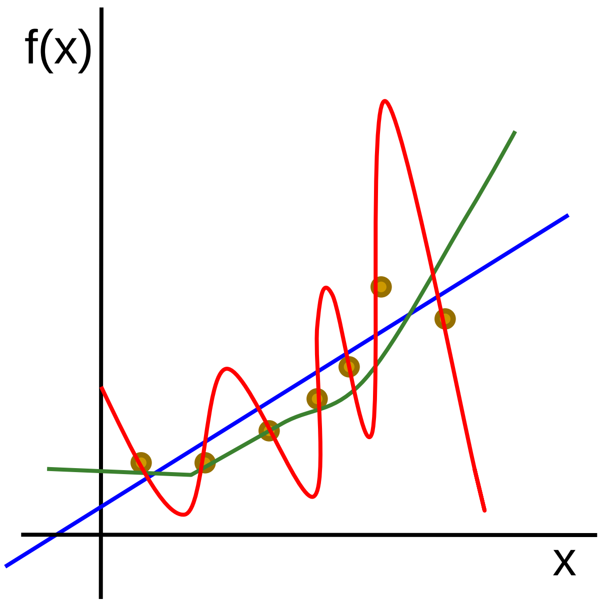
Problem of over-fit learning (4)

- Amongst those target functions learned, which one best generalizes from the training examples?

  Important Note: The goal of machine learning is to achieve high accuracy in prediction for future examples, not for the training ones

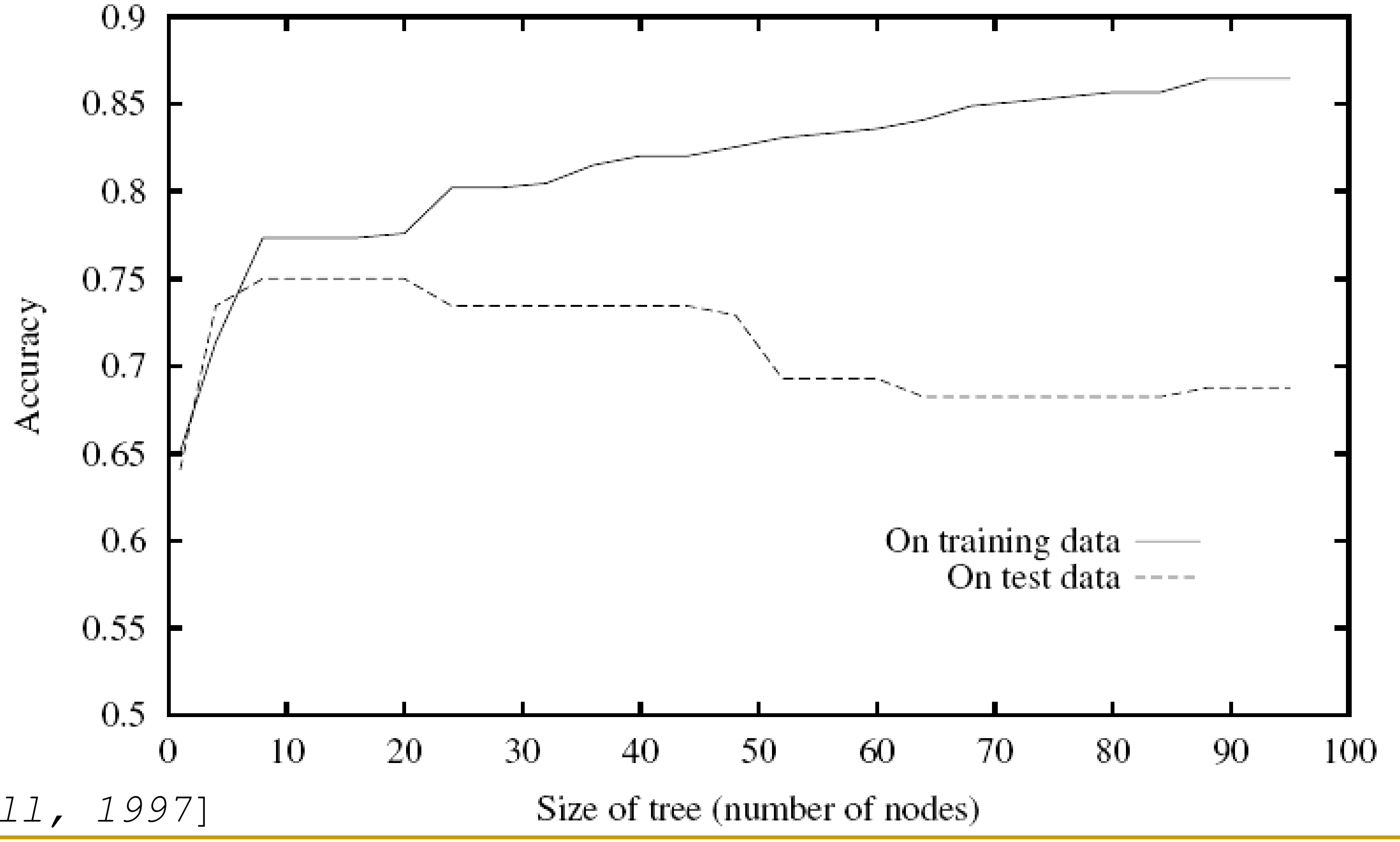
- **Occam's razor**: To select the simplest suitable target function (not necessarily perfect) for the training examples
  - → A better generalization
  - → Easier for explanation/interpretation
  - → Lower in computing cost

Which target function *f(x)* achieves a highest accuracy for future examples?



Example of over-fit learning

Continuing the Decision Tree learning process decreases the accuracy on the test set though increases the accuracy on the training set



[*Mitchell, 1997*]

Data mining: Why?

- An explosive growth of data: From a level of terabytes to another level of petabytes
  - Data collection and availability
    - Tool for automated data collection, database systems, World Wide Web, digital societies
  - Plentiful data sources
    - Business: Internet, E-commerce, Commercial transactions, Stocks,…
    - Science: Sensor signals, Bio-informatics, Simulation experiments,…
    - Society: News, Digital cameras, Social networks
- We are overwhelmed by data – But we lack of (i.e., need) knowledge
- Data mining: To automatically analyze very large datasets to discover knowledge

Data mining: Definition

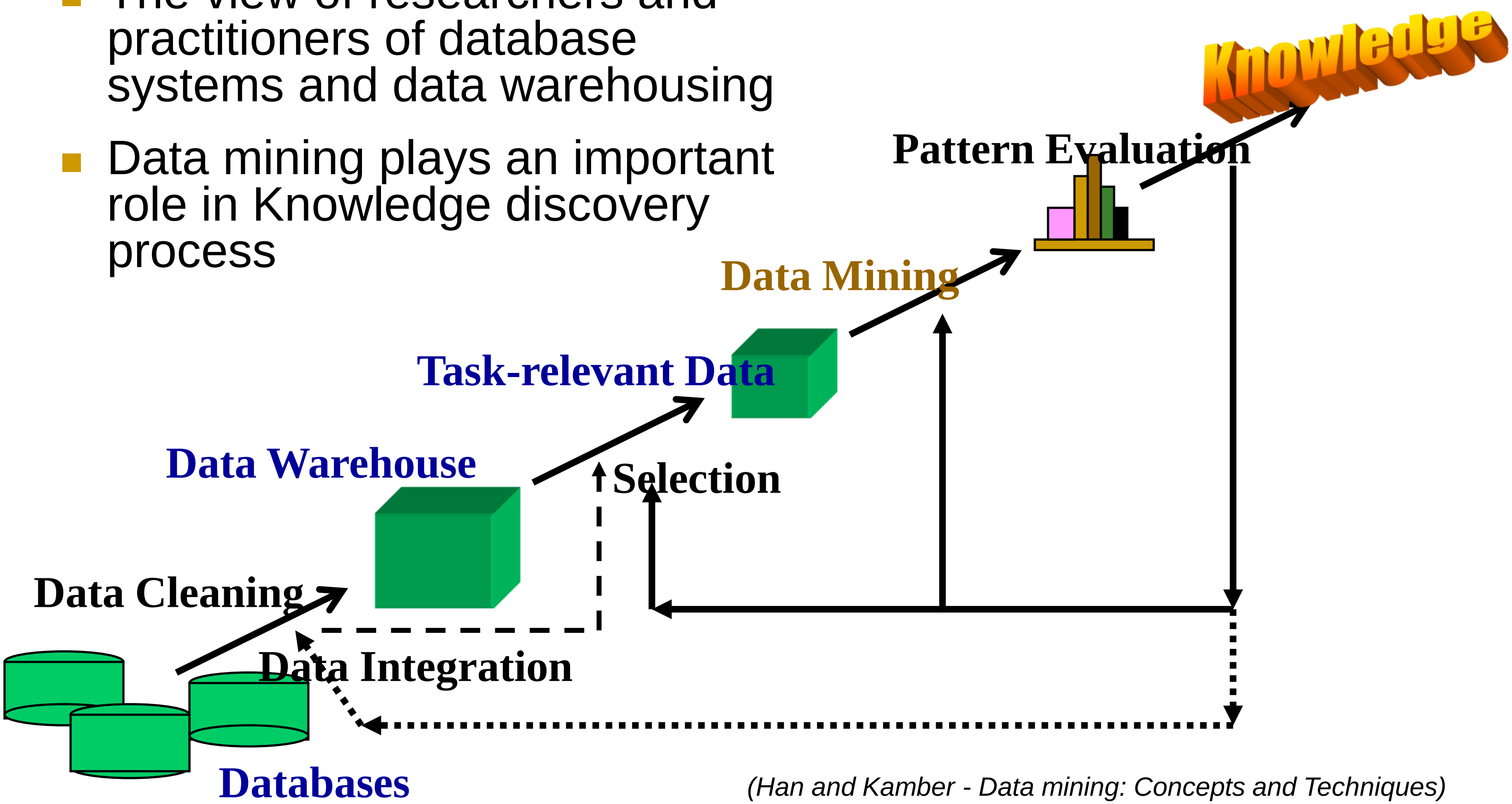
- Data mining (DM): Knowledge discovery from data
  - To extract *important* patterns or knowledge from a (very) large amount of data
    - *Important* = non-trivial, hidden, unknown, and potentially useful
- Other names:
  - Knowledge discovery in databases (KDD)
  - Knowledge extraction
  - Data/pattern analysis
  - …
- Data mining is different from…
  - Information retrieval
  - Processing SQL queries to databases

Steps of Knowledge discovery

1. Analysis of the application problem
   - Goals of the application problem, the application problem's domain-specific knowledge
2. Contruction (or collection) of an appropriate dataset
3. Cleaning and pre-processing of the data
4. Reduction and transformation of the data
   - To determine the important attributes, To reduce the number of dimensions (i.e., attributes), invariant representation
5. Selection of data mining function
   - Summarization, Classification, Regression (prediction), Association, Clustering
6. Selection (or development) of appropriate data mining algorithm(s)
7. Execution of the data mining process
8. Evaluation of the discovered patterns and Knowledge representation
   - Visualization, Removing redundant patterns, …
9. Use of the discovered knowledge

Knowledge discovery process (1)

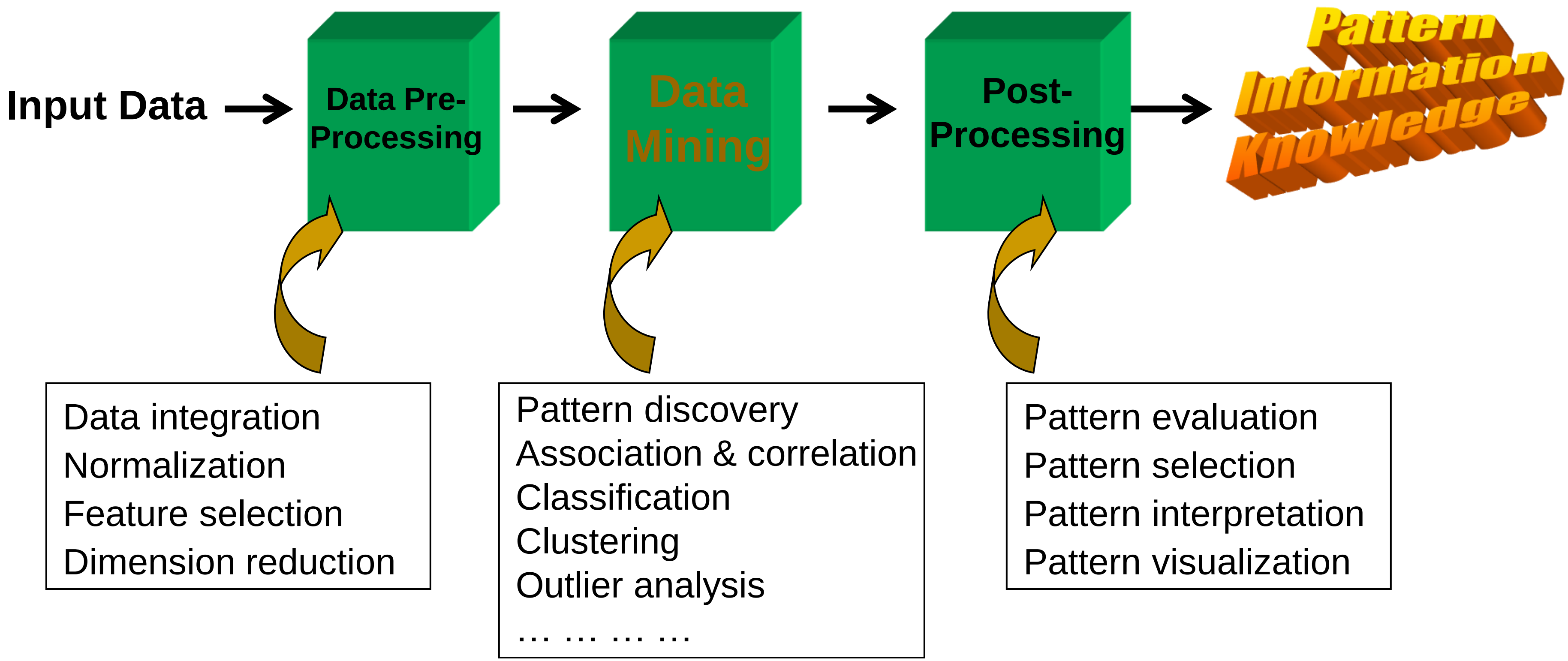
- The view of researchers and practitioners of database systems and data warehousing
- Data mining plays an important role in Knowledge discovery process



(Han and Kamber - Data mining: Concepts and Techniques)

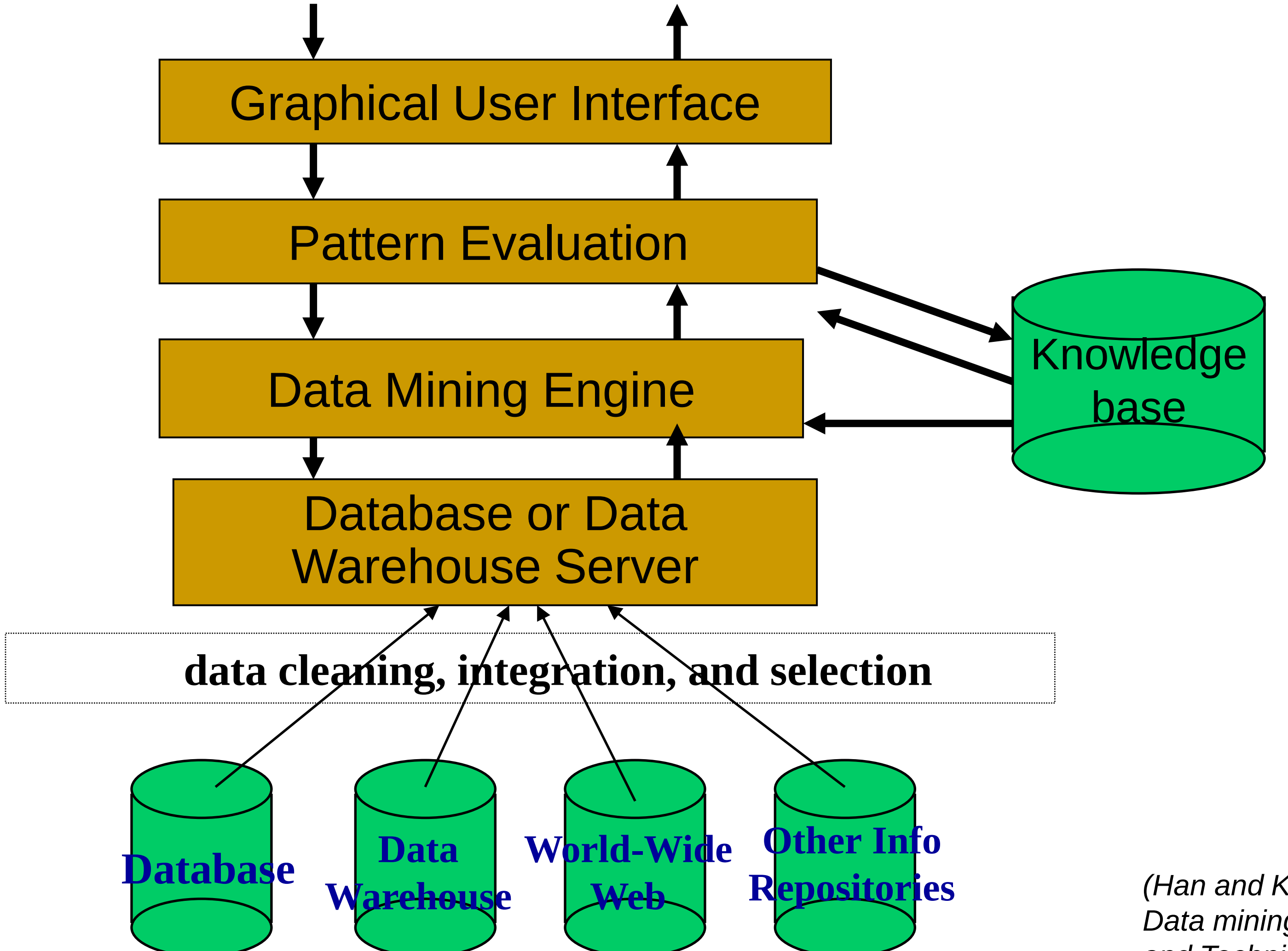
Knowledge discovery process (2)

(Han and Kamber - Data mining: Concepts and Techniques)



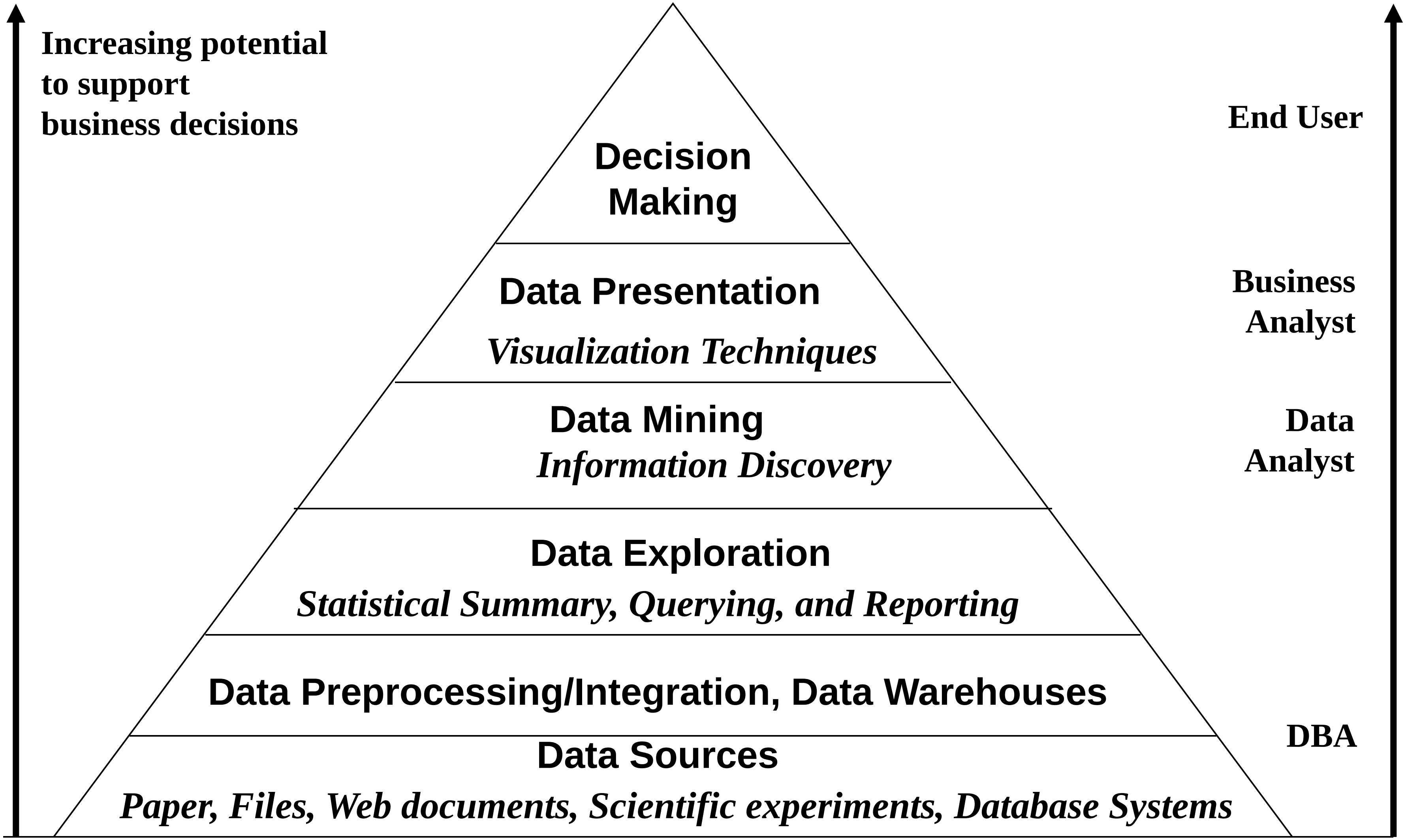
- The view of researchers and practitioners of machine learning and statistics

Architecture of a data mining system



(Han and Kamber - Data mining: Concepts and Techniques)

Data mining for business



(Han and Kamber - Data mining: Concepts and Techniques)

Data mining: Related fields

- Database technology
- Algorithm
- Statistics
- Machine learning
- Pattern recognition
- Visualization
- High-performance computing

Data mining: Different view points

- *Data* to be mined
  - Relational data, Data warehouse, Transactional data, Data stream, Object-oriented data, Spatial data, Time-series data, Textual data, Multimedia data, Heterogeneous data, WWW data, …
- *Knowledge* to be discovered
  - Summarization (characteristics), Differentiation, Association rule, Classification, Clustering, Trend, Outlier analysis
- *Technique* to be used
  - Database, Data warehouse analysis, Machine learning, Statistics, Visualization, …
- *Application* domains
  - Retail business, Telecommunication, Banking, Financial fraud detection, Bio-informatic data mining, Stock market analysis, Text mining, Web mining, …

DM: Association and correlation analysis

- Frequent (i.e., large) patterns or itemsets
  - E.g., which product items are usually purchased together by the customers of the BigC super-market?
- Association, correlation, and causality
  - Example of an association rule:
    - Bread → Milk [0.5%, 75%] (support, confidence)
  - Is it true that highly associated items are also highly correlated ones?
- How to discover such patterns (i.e., rules) in large datasets?

DM: Classification and Regression

- Classification and Regression
  - To build (i.e., learn) the model (i.e., the target function) based on training examples
  - To describe and differentiate the class labels (i.e., concepts) for future prediction
  - Classification: To assign a class label for a new example
  - Regression: To assign a real value for a new example
- Typical techniques
  - Decision tree learning, Naïve Bayes classification, Support vector machine, Artificial neural networks, Rule induction, Linear regression, …
- Typical applications
  - Credit card fraud detection, Target marketing, Disease classification/prediction, Web page classification, …

DM: Cluster and outlier analysis

- Cluster analysis
  - Unsupervised learning: Without class label information
  - To assign the examples to appropriate clusters
  - Rule: To maximize the similarity between examples in the same cluster, but to minimize the similarity between examples in different clusters
  - A lot of clustering techniques and application problems
- Outlier analysis (detection)
  - Outlier: Such an example that is very different from the others in its cluster
  - A data noise in the dataset, or an outlier?
  - Techniques: Clustering, Regression analysis, …
  - Very useful for the problem of fraud (fake) detection, or analysis of rare events

DM: Trend and evolution analysis

- Sequence, trend, and evolution analysis
  - Analysis of trend and shift away from trend
  - Discovery of sequential patterns
    - E.g., First buy a digital camera, then buy large capacity SD cards, …
  - Periodicity analysis
  - Analysis of time-series data and bio-informatic data
  - Similarity-based analysis
- Discovery of data streams
  - Ordered, Change over time, possibly infinite

DM: Network and structure analysis

- Data graph mining
  - To find data sub-graphs, XML data trees, Web data sub-structures … that frequently occur
- Information network analysis
  - Social networks: Actors (objects, nodes) and relations (links)
    - E.g., A network of scholars in the AI field
  - Heterogenous networks
    - E.g., A person may participate in different networks (of friends, family, class/school-mate, similar music/movie tastes,…)
  - The links have much of semantic information: Link mining
- Web mining
  - WWW is a very huge information network: PageRank (Google)
  - Analysis of Web information networks
    - Web communities detection, Opinion mining, Web usage mining

Are all discovered patterns important?

- A data mining process may result in a large number of discovered patterns – But not all of these patterns are important
- Criteria for evaluation of the importance of discovered patterns
  - Easy to user, Still true (up to a certain level) for new data, Useful, Novel, or Help confirm a hypothesis
- Objective vs. subjective evaluation
  - Objective evaluation: Based on statistics and pattern structures
    - E.g., Based on support values, confidence values
  - Subjective evaluation: Based on the user's confidence to the data
    - E.g., Surprise, Novelty, … for a user

Evaluation of the importance of discovered patterns

- Simplicity
  - Lengths of the discovered association rules
  - Size of the learned decision tree
- Certainty (confidence)
  - Confidence values of the discovered association rules
  - Accuracy of the learned classification model
- Utility (of the discovered patterns)
  - Support values of the discovered association rules
  - Noise level for the learned classification model
- Novelty: New (i.e., never been known) patterns

To find all important patterns?

- Finding all important patterns: Completeness
  - Can a data mining system find *all* important patterns?
  - Do we need to find *all* important patterns?
  - Search: Exhaustive vs. heuristic
- Finding all important patterns: Optimization
  - Should a data mining system find *only* important patterns?
  - Different ways:
    - First just generate (find) all the patterns, and then remove those unimportant patterns
    - In the data mining process, only generate (find) important patterns

Visualization of discovered patterns

- Different users and different use purposes require different visualization types for the discovered patterns
  - Visualized by: rules, tables, comparison charts, …
- Concepts taxonomy
  - The discovered knowledge may be easier to understand if it is represented at a higher level of abstraction
  - A concepts taxonomy allows to view the data in different views
- Different knowledge types require different knowledge representations (for the discovered patterns)
  - Association rule,
  - Classification,
  - Cluster,
  - …

DM: Potential applications

- Data analysis for decision making support
  - Market analysis
    - Target marketing, Customer relation management (CRM), Basket analysis, Cross-selling, Market segmentation
  - Business risk analysis
    - Prediction, Customer retention, Competitiveness analysis
  - Frauds (outliers) detection
- Other applications
  - Text mining (news group, email, document)
  - Web mining
  - Biological and bio-informatic data analysis
  - …*(And many other practical applications!)*

DM: Issues and challenges

- The efficiency and the scalability of data mining algorithms
- *Parallel*, *distributed*, *stream*, and *incremental* data mining approaches
- Mining of high dimensional (i.e., number of attributed) data
- Mining of noise, uncertain, incomplete data
- Integration of constraints, expert knowledge, background knowledge into the data mining process
- Pattern evaluation and knowledge integration
- Mining of different data types (bio-informatic, Web, information network,…)
- Integration of data mining into operational devices
- Ensuring security, integrity, privacy in data mining

Frameworks and tools for ML and DM (1)

- **Scikit-learn** (https://scikit-learn.org)
  - OS: Linux, Mac OS, Windows
  - Programming language: Python
- **TensorFlow** (www.tensorflow.org)
  - OS: Linux, Mac OS, Windows, Android
  - Programming languages: Python, C++, Java
- **PyTorch** (pytorch.org), **Caffe2** (caffe2.ai)
  - On March, 2018, Caffe2 and PyTorch is merged into a single platform
  - OS: Linux, Mac OS, Windows, iOS, Android, Raspbian
  - Programming languages: C++, Python
- **Keras** (keras.io)
  - OS: Linux, Mac OS, Windows
  - Programming languages: Python
- **Caffe** (caffe.berkeleyvision.org)
  - OS: Linux, Mac OS, Windows
  - Programming languages: Python, Matlab
- **Theano** (deeplearning.net/software/Theano)
  - OS: Linux, Mac OS, Windows
  - Programming languages: Python

Frameworks and tools for ML and DM (2)

- **CNTK** (www.microsoft.com/en-us/research/product/ cognitive-toolkit/)
  - OS: Windows, Linux
  - Programming languages: Python, C++, C#
- **Deeplearning4j** (deeplearning4j.org)
  - OS: Linux, Mac OS, Windows, Android
  - Programming languages: Java, Scala, Clojure, Python
- **Apache Mahout** (mahout.apache.org)
  - OS: Any OSs with JVM installed
  - Programming languages: Java, Scala
- **MLlib** of Apache Spark (https://spark.apache.org/mllib/)
  - OS: Any OSs with JVM installed
  - Programming languages: Java, Python, Scala, R
- **Weka** (http://www.cs.waikato.ac.nz/ml/weka/)
  - OS: Any OSs with JVM installed
  - Programming languages: Java

References

- E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2004.
- T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- H. A. Simon. *Why Should Machines Learn?* In R. S. Michalski, J. Carbonell, and T. M. Mitchell (Eds.): Machine learning: An artificial intelligence approach, chapter 2, pp. 25-38. Morgan Kaufmann, 1983.