Knowledge Discovery from Data

Knowledge Discovery From Data (KDD)

Knowledge Discovery from Data (KDD): the process of discovering useful **patterns** or knowledge from (large) data sources.

Data sources:

- databases
- text
- images
- World Wide Web
- streaming data (video, audio)
- structures (graphs, etc.)

Knowledge discovery from data is often used as a synonym for the term **data mining**. In this course, use the term **KDD** to refer to a wider range of processes. For us, **KDD** incorporates:

- **Pattern Matching and Discovery**: the techniques, methods and algorithms for finding patterns in structured data.
- **Machine Learning**: the techniques, methods and algorithms for making predictions from data
- **Information Retrieval:** the techniques, methods, algorithms and data models for finding information in unstructured (primarily, but not always, textual) data. A larger area of "sensemaking" in textual data is called **Natural Language Processing**.

Knoweldge Discovery from Data is a multidisciplinary field combining the approaches and methodologies from the following fields:

- **Databases:** KDD activities happen on **very large datasets**. The field of databases deals with efficient storage and management of large quantities of data.
- **Statistics:** the **original** field of *data analysis*. Statistics provides methodology for staging experiments and assessing results. It also provides some basic building blocks for KDD procedures. In addition, a family of KDD methods is based on the use of **probability theory**.
- **Artificial Intelligence:** machine learning, a sub-area of AI studies computer algorithms that improve automatically through experience[1]. The concepts of *supervised learning* (*classification*) and *unsupervised learning* (*clustering*), now the integral part of **data mining**, originated from machine learning and AI.
- **Visualization:** *itself, a multidisciplinary area*, visualization studies the means of clear and understandable representation of information for human consumption.
- **Linguistics:** and **natural language processing** provide rich supply of "building blocks" for analysis of textual data, the same way machine learning and statistics provide building blocks for analysis of structured data.

The Many Faces of KDD

Data is a by-product of human activity. Simple analysis of data can be performed by *querying databases* or *performing statistical analyses* on data. KDD methods seek to provide answers to *more complex* questions about the data and to *discover* knowledge that was not "expected".

KDD processes and activities are all around us:

- Google, (Bing);
- Grocery store discount cards;
- Coupons in the mail;
- amazon.com's *"People who bought this book also bought..."*
- Netflix's movie recommendations (complete with predictions of how much you would like them)
- Spam filters
- all the data **you** generate on Facebook
- Total Information Awareness

[1] Tom Mitchell, *Machine Learning*, McGraw Hill, 1997.

- ...

We need to understand a number of aspects of **Knowledge Discovery from Data**:

- **the technical aspect:** as scientists and engineers we want to know *how KDD works*.
- **the applied aspect:** businesses want to know which KDD methods can help them address their needs.
- **the sinister aspect:** we generate data as a by-product of our activities. We need to be aware of who uses this data and how they are using it.

KDD Process in a Nutshell

The process of **knowledge discovery from data** typically proceeds in **three** steps:

1. **Pre-processing.** Selection of data sources, transformation of raw data into suitable format, data cleaning/filtering,...
2. **Knowledge discovery.** A KDD algorithm (or algorithms) is (are) run on the data.
3. **Post-processing.** Output of the KDD algorithm(s) is analyzed, filtered (if necessary), evaluated and visualized.

What we will study

To a large dgree, **Knowledge Discovery from Data** takes a *cookbook* approach to its structure. It is home to a large number of diverse problems, which are similar only in that they deal with search for interesting information in large data collections.

Of the various problems that exist under the **extended** KDD umbrella, we will consider the following (not necessarily in order in which they will appear in our course):

- **Association Rules Mining.** Search of associative patterns in *market basket* datasets.
- **Supervised Learning (Classification).** Determination whether incoming data belongs to a specific class (classes) of objects, based on prior information about these object classes (categories).
- **Unsupervised Learning (Clustering).** Analysis of a collection of data items targeted at combining these items into groups (clusters) based on their percieved similarity.
- **Collaborative Filtering and Recommender Systems.** Formulation of recommendations (predictions) based on similarity patterns discovered in data.

- **Information Retrieval.** Search of textual document collections for documents relevant to user-specified queries.
- **Link Analyisis.** Analysis of graph structures targeted at identifying ”important” components within the graphs.

All of this has be be achieved in a matter of 10.5 weeks!

The course will be *broad* in scope and *shallow* in depth. We will study multiple problems, but will cover only the most basic algorithms for solving them.

Welcome aboard!

Machine Learning Problems

Machine Learning Problem Description. The most general description of Machine Learning is something like this:

> *Given a number of observations, predict what other observations would look like.*

Machine Learning problems can be classified into

- Regression problems
- Classification problems
- Clustering problems
- Recommendation problems

depending on the shape of the observations, and on the form which the prediction is to take.

Before proceeding with more detailed discussion of different types of machine learning problems, we fisrt must examine the notion of *observations*.

Data for Machine Learning: Objects, Features, Targets

In most cases, data for machine learning problems is a collection of *object descriptions*, where each *object* is described as a *collection of features*.

Features. A *feature* is an individual measurable property of an object or phenomenon that is being observed. A *feature* has the following properties:

- Name. An unique identifier that distinguishes the feature from all other features considered as part of the same machine learning problem.
- Domain. A set of values the feature can take (i.e., the set of possible measurements that can be made of the feature). Feature ranges can be finite or infinite.

Types of features. Depending on the specific range of feature values, features can be classified into the following categories:

1. Numeric. Numeric features have domains that are infinite or finite (but usually rather large) sets of numbers.
2. Categorical. Categorical features are features with domains that are not numeric. Among categorical features, we identify
   (a) Nominal features. These are features whose possible values admit no order.
   (b) Ordinal features. These features have possible values that can be meaningfully ordered, and thus, can be represented as numbers $\{1, \ldots, N\}$ where N is the size of the domain of the feature.

Example. Temperature of a hospital patient patient measured in degrees Celcius is a *numeric feature* whose domain is the range $[30.0, 45.0]$.

The patient's native language is a *categorical nominal feature.* Its domain may be rather large (total number of languages in the world is around 6900), by the possible values of this feature, e.g., "English", "Spanish", "Mandarin", "Russian", etc. do not have an ascribed order.

The patient's level of pain is a *categorical ordinal feature*. Its domain may be $\{\mathsf{no}, \mathsf{slight}, \mathsf{moderate}, \mathsf{severe}\}$. There is a natural order that can be imposed on these values: $\mathsf{no} < \mathsf{slight} < \mathsf{moderate} < \mathsf{severe}$, which makes this feature *ordinal*. The domain can be represented as $\{1, 2, 3, 4\}$ (or $\{0, 1, 2, 3\}$) with numbers retaining their property of order.

Synonyms. We use the terms feature, attribute, and variable (or independent variable) interchangebly.

Observations. In *statistics*, **an observation** is a measured value, at a particular moment of time, of a specific feature.

In this class, we use the term observation in a more general way to refer to a collection of measured values of a particular set of features, that *describe a specific object or phenomenon*.

Formalizing. A given machine learning problem describes a set of objects or a set of phenomena by establishing a *set of features* $\mathcal{A} = \{A_1, \ldots, A_n\}$ whose values combined produce a *complete* (from the perspective of the specific machine learning problem) description of a single object/single phenomenon. The values for each feature $A_i, i = 1 \ldots n$ come from the set $D_i = dom(A_i)$.

A vector $\mathbf{x} = (x_1, \ldots, x_n)$ of values, where $(\forall i = 1 \ldots n)(x_i \in dom(A_i))$ is called a **data point**. Often, without loss of generality, we refer to data points as *objects*, *phenomena*, *entities*, *points*, *records*, *tuples*, *feature-vectors* or use a domain-specific name to identify them.

Example. Consider a collection of features $\{\mathsf{Name}, \mathsf{Language}, \mathsf{Temperature}, \mathsf{PainLevel}\}$ representing the name of a hospital patient, their native language, their temperature at admission time, and their self-reported pain level.

A description of a signle patient may be a vector of values

$$\mathbf{x} = (\text{"Mary Smith"}, \text{"English"}, 38.2, \text{"moderate"}).$$

We can refer to $\mathbf{x}$ in a number of ways:

- *vector of feature values*, *feature-vector,* or simply *vector*
- *data point*
- *observation*
- *object* (although in this case it is a somewhat awkwards term)
- *record* or *tuple*
- *patient record* (this is a domain-specific term that carries knowledge of the semantics of the data)
- *patient*

Dataset. Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be a set of features. Let $X = \{\mathbf{x_1}, \ldots, \mathbf{x_m}\}$, where

$$(\forall i = 1 \ldots m)\mathbf{x_i} = (x_{i1}, \ldots, x_{in}),$$

and

Dataset. Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be a set of features. Let $X = \{\mathbf{x_1}, \ldots, \mathbf{x_m}\}$, where

$$(\forall i = 1 \ldots m)\mathbf{x_i} = (x_{i1}, \ldots, x_{in}),$$

and

$$(\forall j = 1 \ldots n)x_{ij} \in range(A_j).$$

X is called a *collection of datapoints,* or a *dataset*.

X is called a *collection of datapoints,* or a *dataset*.

Algebraic View of Datasets

Dataset as a matrix. A dataset $X = \{\mathbf{x_1}, \ldots, \mathbf{x_m}\}$ can be viewed as a *matrix* X constructed as follows:

- The *rows* of X are vectors $\mathbf{x_1}, \ldots \mathbf{x_m}$.
- The *columns* of X are individual features $A_1, \ldots, A_n$, with each column containing the values of a single feature from all data points of the dataset.

Dataset X can be visualized as a matrix as follows:

$$X = \left(\begin{array}{c|cccc} & A_1 & A_2 & \dots & A_n \\ \hline \mathbf{x_1} & x_{11} & x_{12} & \dots & x_{1n} \\ \mathbf{x_2} & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_m} & x_{m1} & x_{m2} & \dots & x_{mn} \end{array}\right)$$

Size of a dataset. The **size** of a dataset $X = \{\mathbf{x_1}, \dots \mathbf{x_n}\}$, denoted $|X|$ is the number of data points in it, i.e.,

$$|X| = m$$

The **dimensionality** of the dataset X, denoted $dim(X)$ is the length of the vectors in it. If

$$\mathbf{x_i} = (x_{i1}, \dots x_{in}),$$

then

$$dim(X) = n.$$

Numeric vector-spaces. Let $\mathcal{A} = \{A_1, \dots, A_n\}$ and let $(\forall i \in 1 \dots n) dom(A_i) \subseteq \mathbb{R}$. In this case, given a dataset $X = \{\mathbf{x_1}, \dots, \mathbf{x_n}\}$ of points in the feature-space $\mathcal{A}$, we can write for each data point $\mathbf{x_i}$ that

$$\mathbf{x_i} = (x_{i1}, \dots, x_{in}) \in \mathbb{R}^n,$$

i.e., $\mathbf{x_i}$ is a point in n-dimensional real space.

Alternatively, we can view $\mathbf{x_i}$ as an n-dimensional vector (vectors in real space are considered to be columns):

$$\mathbf{x_i} = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{pmatrix} = (x_{i1}, \dots, x_{in})^T.$$

Linear Representation of vectors. Let

$$I = \left(\begin{array}{cccc} \mathbf{e_1} & \mathbf{e_2} & \dots & \mathbf{e_n} \\ \hline 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{array}\right)$$

be a matrix of unit vectors

$$\mathbf{e_i} = (0, 0, \dots 1_i, 0, \dots, 0)^T.$$

The vectors $\mathbf{e_1}, \dots \mathbf{e_n}$ are said to form the *standard basis* of $\mathbb{R}^n$.

Vector $\mathbf{x_i}$ can be represented as a linear combination of $e_1, \dots, e_n$ as

$$\mathbf{x_i} = x_{i1}\mathbf{e_1} + x_{i2}\mathbf{e_2} + \dots + x_{in}\mathbf{e_n} = \sum_{j=1}^{n} x_{ij}\mathbf{e_j}.$$

More notation. From the above, we can denote/represent the dataset $X = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ as follows:

$$X = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{pmatrix} = \begin{pmatrix} -\mathbf{x_1}^T- \\ -\mathbf{x_2}^T- \\ \vdots \\ -\mathbf{x_m}^T- \end{pmatrix} = \begin{pmatrix} | & | & & | \\ A_1 & A_2 & \ldots & A_n \\ | & | & & | \end{pmatrix}$$