Aalto University
School of Science

CS-C3160 - Data Science

Jaakko Hollmen, Pekka Marttinen,
Rohit Babbar, Arno Solin, Tolou Shadbahr

December 5, 2019

Contents of this chapter

- What was taught on the course?
- What is important?
- What does the exam look like?
- Questions

Data Science definition

- Q: What is Data Science?
- A: Data Science focuses on different ways to extract knowledge from data

Course contents, overview

- What is data? What is knowledge?
- Formalizing the data analysis problem: putting natural data into the data matrix and then applying mathematics and programming techniques on that data matrix
- Feature extraction: from natural data to data matrix
- Analysis and modelling: applying mathematics and programming to data (matrix)

Course contents: Histogram.

- Histograms;
  - Histogram is a common way to create features, by dividing the values of a variable in different “bins”.
  - The height of a box in a histogram is equal to the number of values in the corresponding bin.
  - A color histogram of an image shows the number of pixels of each color.
  - Histograms are used also for audio and text.

Course contents: Histogram.

- Histograms;
  - If the word histograms of individual documents are the columns of a data matrix X, it is called a *term-document matrix* (or document-term matrix)
  - In a term-document matrix each row represents a document. Each column represents a word.
  - Example; consider 3 sentences as follow:
    - 1) It is sunny.
    - 2) Yesterday it was cloudy.
    - 3) It is nice, is it not?

    The term frequency vectors for above sentence are [111100000$\cdot$], [10001110$\cdot$] and [220000011$\cdot$] respectively. Additionally, the term document matrix for these three sentences is;

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \cdots \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \cdots \end{bmatrix}$$

Course contents: sounds in time and frequency domain.

- The spectrum of a musical instrument's sound could be a data vector, where the vector variables correspond to frequencies.
- An analog *signal*, e.g. sound, can be described with a time-dependent continuous function $f(t)$
- The Fourier transform convert the signal from (continuous) time domain to the (discrete) frequency domain and it is as follow;

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} \mathrm{d}t$$

  where ω is the (angular) frequency and $i = \sqrt{-1}$ is the imaginary number.
- *Discrete Fourier Transform* (DFT) of a digitized signal $f_m, m = 0, ..., T-1$:

$$F_n = \sum_{m=0}^{T-1} f_m e^{-2\pi i m n / T}$$

Course contents: Frequency filtering

- It is often desirable to remove certain frequencies from the time domain signal, examples: high frequencies representing noise.
- This can be done with linear filters, which are computationally implemented with *convolution*:

$$g_k = \sum_{m=-\infty}^{\infty} f_m s_{k-m} = \ldots + f_{k-2}s_2 + f_{k-1}s_1 + f_k s_0 + f_{k+1}s_{-1} + f_{k+2}s_{-1} \ldots$$

  where f_m is the digital signal being filtered and s_j is the filter
- It can be easily shown that the Fourier transform of a convolution is

$$G_n = F_n \cdot S_n$$

  i.e. the product of the spectrums in the frequency domain
- If we wish to attenuate/amplify certain frequencies, the filter sequence s_k must be chosen so that these frequencies are weak/strong in its Fourier transform S_n

Course contents: Estimation theory

- Parametric distributions are used for providing the probabilities of occurrence of different possible outcomes in an experiment. The most common density functions are:*normal distribution* (a.k.a. Gaussian distribution)

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

exponential distribution

$$p(x) = \lambda e^{-\lambda x}$$

and *uniform distribution* (in $[a, b]$)

$$p(x) = \frac{1}{b-a} \text{ when } x \in [a, b] \text{ and 0 otherwise}$$

- Form of the distribution fixed. Parameters define the exact location and scale of the distribution.

Course contents: Parameter estimation

- Estimation of the parameters can be used to define a specific distribution for a given data matrix. Let us denote the ordered set of unknown parameters with vector Θ and the density function with $p(\boldsymbol{x}|\Theta)$. What this means: the argument of the function consist of the vector elements $x_1, \ldots, x_d$ but the function also depends on the parameters (elements of Θ)
- The general shape of the function is assumed known except for the values of the parameters Θ

Course contents: Maximum likelihood principle

- Principle of maximum likelihood: select the parameter vector Θ that maximizes the joint density of the data set, the so-called *likelihood function* $L(\Theta)$

$$L(\Theta) = p(\boldsymbol{X}|\Theta) = p(\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(n)|\Theta)$$

- Likelihood function for a normally distributed scalar (1-D dimensional) data set$L(\Theta)$ for the whole data set:

$$p(\boldsymbol{X} \mid \mu, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left[-\frac{1}{2\sigma^2}\sum_{j=1}^{n}[x(j) - \mu]^2\right]$$

- Let's take the logarithm $\ln L(\Theta)$:

$$\ln p(\boldsymbol{X} \mid \mu, \sigma^2) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}[x(j) - \mu]^2$$

- the μ and σ^2 can be estimated by setting the derivative to zero for both parameters.

Course contents: Bayes estimation

- Extension to include prior information with prior distributions. The Bayes formula tells us how to go from the prior to the posterior. If we have a data matrix X and an unknown parameter vector θ, the Bayes formula gives us;

$$p(\Theta|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\Theta)p(\Theta)}{p(\boldsymbol{X})}$$

- The Bayes formula shows us that the prior distribution is multiplied by the likelihood function and divided by the probability of the data in order to arrive at the posterior distribution.
- Both MI and Bayes estimation are used for classification and regression task.

Course contents: Least-squares method

- maximizing the log-likelihood with respect to θ is equivalent to minimizing the sum of squares ($\sum_{i=1}^{n}[y(i) - f(\boldsymbol{x}(i), \Theta)]^2$) in following;

$$\ln p(\boldsymbol{X}, Y|\Theta) = \ln p(\boldsymbol{X}) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}[y(i) - f(\boldsymbol{x}(i), \Theta)]^2 + n\ln(\text{constant})$$

  Therefor it is equal to the *least-squares method* (LSM): finding Θ that minimize following:

$$\frac{1}{n}\sum_{i=1}^{n}[y(i) - f(\boldsymbol{x}(i), \Theta)]^2$$

Course contents: Classification Problem

- Classification; the task of finding the label (class) for unlabeled test data set by using train data set as input data x and output label (class) c. This is a supervised learning as we have labels for our train examples. Pattern areas or discriminant functions are formed using the training set. The test set is given to the classifier as input. The number of wrong classifications (divided by the total) is the error of our classifier model. Because the training set is finite, it's not possible in general to create an error-free classifier.

Course contents: K-nearest neighbor

- The idea of KNN; a fresh input vector $\boldsymbol{x} \in \mathbb{R}^d$ arrives and we have to label it with its "proper" class ω
- The training set $(\boldsymbol{x}, \omega)_j$ is a set of already classified (into M classes) data $\boldsymbol{x}_{\omega i}(j) \in \mathbb{R}^d$:

$$\boldsymbol{x}_{\omega 1}(1), \ldots, \boldsymbol{x}_{\omega 1}(n_1), \ldots, \boldsymbol{x}_{\omega M}(1), \ldots, \boldsymbol{x}_{\omega M}(n_M)$$

- Let's compute the distances between the point $\boldsymbol{x}$ and the points of the training set ($n = n_1 + \ldots + n_M$) and find the k vectors closest to $\boldsymbol{x}$ (k nearest neighbors). Euclidean distance is often used as the distance metric:

$$D(\boldsymbol{x}, \boldsymbol{z}) = ||\boldsymbol{x} - \boldsymbol{z}|| = \sqrt{\sum_{i=1}^{d} (x_i - z_i)^2}$$

Course contents: K-nearest neighbor

- knn has the following hyper-parameters :
  - The distance metric - ℓ_2 vs ℓ_1 distance
  - Number of neighbours k in the algorithm. In order to avoid tie situation is better to select the odd number for k($k = 1, 3, 5, ...$).
- knn does all computations in the prediction phase, also called lazy learner.
- Not so efficient for high dimension. If the training set is large and the dimension d is very high, finding the nearest neighbors is computationally expensive.

Course contents: Curse of dimensionality

- 10 bins in d-dimensional space is equal to 10^d total number of bins. With d = 79 we have 10^{79} bins, also the estimated total number of atoms in the universe. Any set of vectors in a high-dimensional space is extremely sparse. This is the *curse of dimensionality*: it seems that we need an enormous amount of high-dimensional data to be able to construct any kind of a model.
- Data in high dimensions can be tricky to visualize. Therefore it is desirable to reduce the data dimensionality and still being able to capture important facets of the data.
- There are strong dependencies between columns of a vector in data matrix (e.g. adjacent pixels in an image). By removing these dependencies we are able to greatly lower the dimensionality.

Course contents: Principal Component Analysis

- PCA does the following :
  - Removes the correlations between variable(features) cells.
  - Finds the rotation in vector space so that the new (rotated) coordinates have maximal variance (energy).
- Steps in calculating the PCA:
  - Removing the mean of features (Columns) in order to have zero average for vectors **x**.
  - Dividing each feature (columns) by its standard deviation so all features are in the same scale.
  - **x** is linearly transformed into another vector **y** of m columns which $m \leq d$, in a way that the redundancy due to correlations is removed.

Course contents: Principal Component Analysis

- Each component is a linear combination (sum) of original features of data matrix, where each feature of the original data matrix receive a weight (coefficient) in the projection vector (Eigen vector). All the components are orthogonal to each other.
- PCA components are ordered from highest variance explained to the smallest. The components with small variance can be ignored (dimension reduction).
- The first component of PCA indicates the direction of the highest variance of the original data matrix.
- The PCA in calculated by Eigen decomposition of the covariance matrix of the zero mean data matrix.

Course contents: Cluster analysis

- When the train data matrix in not labeled, the cluster analysis can be used. Since the train data set doesn't include labels, therefore, clustering is unsupervised learning.
- Every cluster contains vectors that are quite similar to each other but very different from vectors in other clusters
- Different clustering algorithms can produce a very different clustering for the same dataset.
- We have again n vectors $\mathbf{x}(1), \cdots, \mathbf{x}(n)$ of dimension d (columns of a data matrix). These vectors are unlabeled. We want to find c groups (sets, clusters) $C_1, \cdots, C_c$ so that they do not intersect (no vector belongs to two or more clusters) and their union is the whole set of vectors (all vectors belong to a cluster).

Course contents: Hierarchical clustering

- We start with each vector in its own cluster. In other words, initially there are n clusters.
- Next, we combine clusters close to each other (neighbors). This goes on until all vectors are in one cluster.
- The clustering can be represented as a hierarchical cluster tree. The tree can be cut off at a suitable level.
- different metric can be used to determine the distances between clusters; such as;
  - Single linkage: minimizes the distance between the closest observations of pairs of clusters.
  - Maximum or complete linkage: minimizes the maximum distance between observations of pairs of clusters.
  - Average linkage: minimizes the average of the distances between all observations of pairs of clusters.

Course contents: k-means clustering

- In k-mean algorithm we have to guess the number of clusters C. Each group C_i is represented by the mean vector m_i and defined such as;

$$C_i = \{x \text{ such that} ||x - m_i|| \leq ||x - m_j||\}, j \neq i$$

  which is the set of those vectors closer to the center (mean vector) of C_i than any other center.
- The clustering rule is minimize $J = \sum_{i=1}^{c} \sum_{\mathbf{x} \in c_i} ||\mathbf{x} - \mathbf{m}_i||^2 = \sum_{i=1}^{c} J_i$
- The k-means algorithm is as follow:
  - Pick c points from $\mathbf{x}(1), \cdots, \mathbf{x}(n)$ at random as the points $\mathbf{m}_1, \cdots, \mathbf{m}_c$
  - Repeat while the points $\mathbf{m}_1, \cdots, \mathbf{m}_c$ change:
    - Place each point $\mathbf{x}(1), \cdots, \mathbf{x}(n)$ in the cluster C_i that has the closest center $\mathbf{m}_i$.
    - Compute new centers:

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in c_i}$$

      where n_i is the number of points in C_i.

Course contents: frequent itemsets

- Many observation vectors $\boldsymbol{x}(1), \ldots, \boldsymbol{x}(n)$, each contains d-dimensional variables (feature). Value of variable i in observation vector $\boldsymbol{x}(j)$ is denoted with $x(i, j)$ or x_{ij}.
- Search for all variable sets $\{a_1, a_2, \ldots, a_k\}$ so that there are at least N observations $\boldsymbol{x}(i)$ where $x(a_1, i) = 1$ and $x(a_2, i) = 1$ and ... and $x(a_k, i) = 1$
- Let's call this variable set $\{a_1, a_2, \ldots, a_k\}$ a *frequent itemset*
- How to search for frequent itemsets:
  - Trivial approach: brute force search through all subsets of variables. However we end up with huge search space; d variables $\Rightarrow 2^d$ subsets of variables.
  - we should be more clever. A set of variables $\{a_1, a_2, \ldots, a_k\}$ can be frequent only if all of its subsets are frequent. The frequentness of subsets is a necessary but not sufficient condition.

Course contents: Breadth-first search

- *Breadth-first search* (aka levelwise algorithm, a priori algorithm) is a simple but effective algorithm for frequent itemset search
- First, we look for frequent itemsets of size 1, then size 2, etc. The algorithm is repeated until no new candidate sets can be found.
- The basic observation we made earlier becomes very useful: a set can be frequent only if all its subsets are frequent

Course contents: Authority and hub

- A good hub points to good authorities and good authorities are pointed to by good hubs. We can get rid of the circular definition with an iterative algorithm.
- Firstly, we choose some of pages, let's call this the *root set S*. Then we form the set T: S + the pages linking to a page in S + the pages linked to by a page in S. Each page s in T is given a hub weight h_s and an authority weight a_s;

$$h_s = a_s = \frac{1}{\sqrt{n}}, \forall s \in T$$

- Iterative update rules:

$$a_s \leftarrow \sum_{t \in T, (t,s) \in E} h_t$$

$$h_s \leftarrow \sum_{t \in T, (s,t) \in E} a_t$$

- After each iteration the squared sums of the weights are scaled to 1:

$$\sum_{s \in T} a_s^2 = 1, \sum_{s \in T} h_s^2 = 1$$

Course contents: PageRank

- A page is relevant if relevant pages link to it.
- If page t links to pages $s_1, s_2, \ldots, s_k$, the relevance r of page t is distributed among these with the weight $1/k$
- Now we have a relevance vector ("PageRank") $\boldsymbol{r}$ as $\boldsymbol{r} \leftarrow \boldsymbol{F}\boldsymbol{r}$, where the matrix $\boldsymbol{F}$ is defined by

$$\boldsymbol{F}_{s,t} = \begin{cases} 1/\deg(t), & \text{if } t \text{ links to } s \\ 0, & \text{otherwise} \end{cases}$$

  where $\deg(t)$ is the number of links (h)
- If a page has no links, we add links to everywhere. Here n is the number of pages.

$$\boldsymbol{F}_{s,t} = \begin{cases} 1/\deg(t), & \text{if } t \text{ links to } s \\ 1/n, & \text{if } t \text{ has no outbound links} \\ 0, & \text{otherwise} \end{cases}$$

- With suitable assumptions $\boldsymbol{r}$ the iteration converges. The $\boldsymbol{r}$ is the eigenvector of $\boldsymbol{F}$ corresponding to the largest eigenvalue.

Exam

- Test the understanding of basic concepts and their relation to other concepts
- Questions are from the lecture slides and demonstration exercises
- Multiple choice questions
- You are allowed to have simple function calculator.
- No cheat sheet is allowed.

Questions and answers

Any questions?

best of luck for your exams.