Lunds Tekniska Högskola
Institutionen för datavetenskap

EDA132
VT 2016

Tillämpad Artificiell Intelligens
Applied Artificial Intelligence
Tentamen 2016–03–16, 14.00–19.00, MA:9

You can give your answers in English or Swedish.
You are welcome to use a combination of figures and text in your answers.
100 points, 50% needed for pass.

1 Search: (11 points)

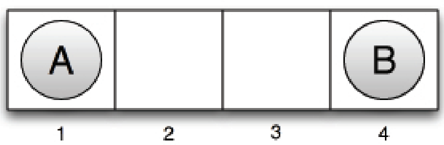
The *heuristic path algorithm*, proposed by Ira Pohl in 1977, is a best-first search in which the evaluation function is

$$f(n) = (2 - w)g(n) + wh(n).$$

For what values of w is this complete? Motivate your answer. (4p)
For what values of w is it optimal, assuming that h is admissible? Motivate your answer. (4p)
What kind of search does this perform for $w = 0$, $w = 1$, and $w = 2$? (3p)

2 Games: (14 points)

Consider a two-player game featuring a board with four locations, numbered 1 through 4 and arranged in a line. Each player has a single token. Player A starts with his token on space 1, and player B starts with his token on space 4. Player A moves first.



The two players take turns moving, and each player must move his token to an open adjacent space *in either direction*. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game is $+1$; if player B reaches space 1 first, then the value of the game is -1.

1. (6 points) Draw the complete game tree, using the following conventions:
   - Write each state as (s_A, s_B) where s_A and s_B denote the token locations.
   - Put the terminal states in square boxes, and annotate each with its game value in a circle.
   - Put *loop states* (states that already appear on the path to the root) in double square boxes. Since it is not clear how to assign values to loop states, annotate each with a "?" in a circle.
2. (3 points) Now mark each node with its backed-up minimax value (also in a circle). Explain in words how you handled the "?" values, and why.
3. (5 points) Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to 2. Does your modified algorithm give optimal decisions for all games with loops?

3 Reasoning: (15 points)

Given the following, can you prove that the unicorn is mythical? How about magical? How about horned? Can your conclusion in either of the three cases be reached using Backward Chaining? If yes, how? If no, why? Can your conclusion be reached using Resolution? If yes, how? If no, why?

> If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

4 Easter Eggs and probabilistic reasoning: (30 points)

David decided to surprise his family with homemade chocolate Easter eggs this year, as he really likes to do experiments in the kitchen. After some initial tests with more or less edible outcomes, he settles for two types of fillings, one with nougat cream, the other with some real brandy in it. In the end, he has 20 eggs with nougat and 10 with brandy, that he now wraps carefully into some nice glittery pieces of foil, some green, some silver. He does not really care about which type of egg goes into which type of foil, as he does not expect anyone in his family to have problems with any of the

fillings.
Easter Sunday approaches, David goes to visit his parents, well equipped with his homemade Easter eggs. Upon arrival, his mother tells him gladly, that some remote cousin of hers will join them for the day, and this cousin will bring her two-year-old daughter... "Uh-oh, two-year-olds and brandy - not good!" he thinks. In a frenzy, he tries to remember all he can about the foil colours and flavours of the chocolate eggs. He is pretty sure to have wrapped about half of the nougat eggs in green, the other half in silver foil, while he is definitely sure that he had only three pieces of green foil left for brandy-eggs.
The guests arrive, the egg hunt starts, and the little girl finds a silver-foil wrapped egg in some bush, which she is really eager to open and eat - she loves "sssotolates"!

a) (5p) What is the most likely type of egg the little girl picked - according to a *Maximum-Likelihood* reasoning, or

b) (5p) according to a *Maximum-A-Posteriori (MAP)* reasoning? (Motivate your answers by showing your calculations)

c) (5p) Assume that David really had some fun in the kitchen not only giving the eggs different fillings, but also using different casting moulds, so that 40% of the nougat cream eggs and 60% of the brandy eggs have some wave-like pattern, while the rest are plain. For the wrapping, David does not care about the different patterns, and the numbers for green and silver foil wrapped eggs with nougat or brandy respectively remain the same as previously mentioned. Draw the optimal Bayesian Network reflecting the overall situation, i.e., considering all three features of the eggs (filling, pattern, and wrapper colour).

d) (5p) In case David lost the exact original counts for nougat vs brandy fillings, which algorithm could he use to figure out these numbers again, assuming that he has at least some vague idea and the network structure? (Explain the algorithm's basic idea)

e) (10p) Explain the following terms and their connections or applications in the context of probabilistic reasoning and learning: *Naïve Bayes Classifier*, *Prediction*, *Optimal Bayes' learner*, *Tracking*, *MAP-hypothesis*, *Learning*, *Stationary process*, *Categorisation*, *Markov assumption*, *Conditional independence.*

idx	x	y	idx	x	y	idx	x	y
1	0.062135	0.111182	11	0.064943	0.106858	21	0.065294	0.103548
2	0.064422	0.108051	12	0.062583	0.107718	22	0.064737	0.105254
3	0.064468	0.105109	13	0.063954	0.107142	23	0.064228	0.108701
4	0.065491	0.108279	14	0.065658	0.104777	24	0.064724	0.109501
5	0.063367	0.109766	15	0.064643	0.108543	25	0.062059	0.110587
6	0.067720	0.116663	16	0.068590	0.114463	26	0.066394	0.113737
7	0.068642	0.114763	17	0.067256	0.113341	27	0.069109	0.111732
8	0.067264	0.113108	18	0.070089	0.112148	28	0.068804	0.116476
9	0.069999	0.117790	19	0.070432	0.114835	29	0.068468	0.114312
10	0.068106	0.117104	20	0.069234	0.115607	30	0.066331	0.119834

Table 1: A dataset of 30 observations: Relative frequencies of the letters a (x axis) and e (y axis) in texts

5 Unsupervised learning: (30 points)

In this question, you will implement an unsupervised learning algorithm: k-means clustering.

5.1 Introduction

1. Describe what unsupervised learning is.
2. Table 1 shows a dataset extracted from texts in different languages that consists of the relative frequencies of the letters a and e in these texts. Figure 1 shows a picture of it. How many clusters (and languages) do you identify visually? Justify your choice.
3. Estimate roughly the means in x and y (barycenters) of these clusters (Give the means for each cluster).

5.2 K-Means

5.2.1 Overview

The k-means algorithm creates automatically clusters from a dataset. The result is a partition: Each observation belongs to one cluster and there is no overlap. The algorithm consists of an initialization and a loop, where the loop has two steps. Before the algorithm starts clustering, you need to choose the number of clusters k, for instance $k = 3$.

Initialization: Select randomly k seeds from the dataset. These seeds define the k first centroids: The means of the clusters. If $k = 3$, the algorithm will select three different points (means) from the dataset;

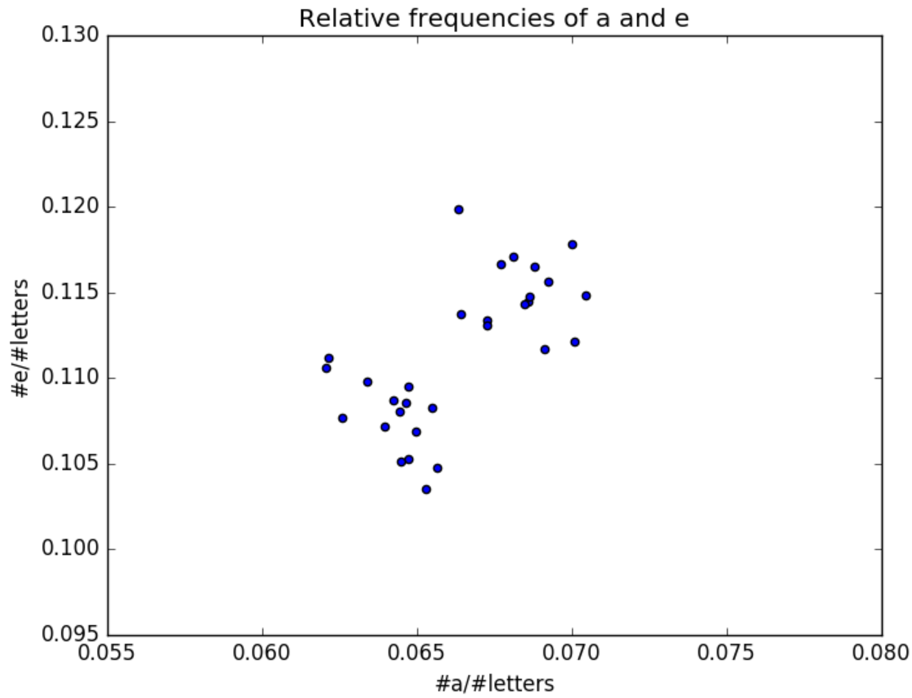


Figure 1: Plot of the relative frequencies of the letters a (x axis) and e (y axis) in texts

Loop: Repeat until the assignment does not change:

- **Assignment:** Assign each point in the dataset to the cluster defined by its closest centroid. If $k = 3$, each point will be assigned to one of three clusters;
- **Compute new centroids:** Compute the new centroid of each cluster. A centroid is defined as the mean (barycenter) of the points in the cluster. If $k = 3$, you will have to compute three new centroids (three new means) at each iteration.

5.2.2 Programming K-Means

You will now program the k-means algorithm. Be sure that your program works for any number of clusters as well as with any number of observations of any dimensions. Please, use the variable k for the number of clusters, q for the number of observations, and N for the dimension of each observation. In Table 1, we have $q = 30$ and $N = 2$. Your program should run for any value of q and N, for instance $q = 3000$ and $N = 10$.

The programming languages you can use are: Python (preferred if you know this language)/Perl/Ruby, C/C++/Java, or Prolog. If you need a built-in function and you do not know exactly its name or parameters, invent something plausible.

1. Write an `initialize` function that selects randomly k different observations from the dataset as initial centroids. You will store the results in a list called `centroids`;
2. Write a `distance` function that computes the Euclidian distance between two points in a N-dimensional space;
3. Write an `assign` function that takes a list of centroids (`centroids`) and a point as inputs and that returns the closest centroid (or its index);
4. Write a `partition` function that takes a list of centroids and the dataset as input and that returns the clusters. As data structure, you will store the clusters in a dictionary (a hashmap) or a list of lists. You will call this data structure `clusters`;
5. Write a `means` function that computes the means (barycenter) of a set of points.
6. Write the program that carries out the k-means clustering. You will define yourself a stop condition.

5.2.3 Finding the Optimal K

The value of k must be given as input to the clustering procedure. In practice, this value is searched manually using the sum of squared distances between each point and the centroid of its cluster.

1. Given a clustering, `clusters`, and a list of centroids, `centroids`, compute the sum of squared distances between each point and the centroid of its cluster.
2. Write a loop that computes the sum of squared distances for each value of k in a list: `[1..kmax]`.

To decide the optimal k value, you would plot the sum of squared distances as a function of k and you would select k, where you observe a twist in the curve (an inflection). This method is called the **elbow method**. Figure 2 shows the plot for the dataset in Table 1. What is the optimal k? Justify your answer.

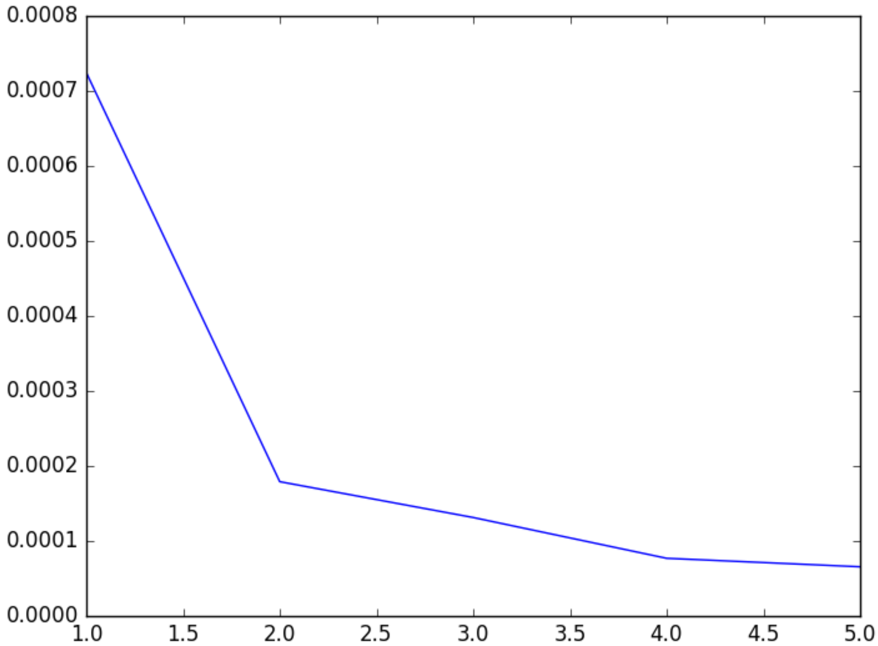


Figure 2: The plot of the sum of squared distances as a function of k.

Good Luck!