CSE 291: Unsupervised learning

Homework 7

This homework is due on Thursday June 1 at 11.59pm.

- All homeworks must be typewritten and uploaded to Gradescope.
- No late homeworks will be accepted.

1. *Genes, diseases, and outcomes.* There is a fictitious disease (D) that can cause paralysis (P). Whether or not an individual gets this disease depends in large part on two genes, G_1 and G_2. Here is a summary:
   - Each of these genes G_1, G_2 can either be normal (also known as *wild-type*), represented by $G_i = 0$, or *mutant*, represented by $G_i = 1$.
   - For a person with $G_1 = G_2 = 0$ (both genes wild-type) the probability of acquiring the disease is zero. If G_1 is mutant and G_2 is wild-type, the probability is 0.2. If G_1 is wild-type and G_2 is mutant, it is 0.1. And if both genes are mutant, then the disease probability is 0.5.
   - For an individual with the disease, the probability of paralysis is 0.5. Without the disease, there is still a small probability of paralysis (from other causes): 0.01.
   - G_1 is mutant with probability 0.1 and G_2 is mutant with probability 0.2; these are independent.

   (a) Draw a suitable Bayes net over the variables G_1, G_2, D, P, and give all relevant conditional probability tables. Let $D = 1$ if the disease is present and $D = 0$ if it isn't; likewise with paralysis.

   (b) Which of the following independence statements is true? (You don't need to explain your answers.)
   - $G_1 \perp\!\!\!\perp G_2$
   - $G_1 \perp\!\!\!\perp G_2 \mid D$
   - $G_1 \perp\!\!\!\perp G_2 \mid P$
   - $G_1 \perp\!\!\!\perp P \mid D$

   (c) What is the probability of disease ($D = 1$)?

   (d) What is the probability of paralysis ($P = 1$)?

   (e) We observe that a person has paralysis. What is the probability that they have the disease?

   (f) We observe that a person has the disease ($D = 1$). What is the probability that they have the mutant form of gene G_1?

2. Probability distribution P over binary variables $x_1, \ldots, x_6 \in \{0, 1\}$ has the following form:

$$P(x_1, \ldots, x_6) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_2, x_3)\psi(x_3, x_4)\psi(x_4, x_5)\psi(x_5, x_6)\psi(x_6, x_1)$$

   where for $a, b \in \{0, 1\}$,

$$\psi(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0.5 & \text{otherwise} \end{cases}$$

(a) Draw the minimal undirected graph G over which P factors.

(b) For what setting(s) $x \in \{0,1\}^6$ is $P(x)$ maximized?

(c) For what setting(s) $x \in \{0,1\}^6$ is $P(x)$ minimized?

3. The joint distribution P over (x_1, x_2, x_3) is given by the following table.

x_1	x_2	x_3	Pr
0	0	0	1/3
0	0	1	1/3
1	0	0	1/6
1	1	1	1/6

(a) One of these variables is a deterministic function of the other two. Identify this relationship.

(b) Draw the minimal Bayes net G over which this distribution factors.

(c) Now show how P factors over G by explicitly giving the three (conditional) probability tables, one per x_i.

(d) Draw the minimal *undirected* graph G' over which distribution P factors.

(e) Let $\mathcal{P}$ denote the set of distributions that factor over G and let $\mathcal{P}'$ be the set of the distributions that factor over G'. What is the relationship between $\mathcal{P}$ and $\mathcal{P}'$?

4. *Word embeddings.*

The large number of English words can make language-based applications daunting. To cope with this, it is helpful to have a *clustering* or *embedding* of these words, so that words with similar meanings are clustered together, or have embeddings that are close to one another.

But how can we get at the meanings of words? John Firth (1957) put it thus:

> *You shall know a word by the company it keeps.*

That is, words that tend to appear in similar contexts are likely to be related. In this mini-project , you will investigate this idea by coming up with an embedding of words that is based on co-occurrence statistics.

The description here assumes you are using Python with NLTK.

- First, download the Brown corpus (using `nltk.corpus`). This is a collection of text samples from a wide range of sources, with a total of over a million words. Calling `brown.words()` returns this text in one long list, which is useful.
- Remove stopwords and punctuation, make everything lowercase, and count how often each word occurs. Use this to come up with two lists:
  - A *vocabulary* V, consisting of a few thousand (e.g., 5000) of the most commonly-occurring words.
  - A shorter list C of at most 1000 of the most commonly-occurring words, which we shall call *context words.*
- For each word $w \in V$, and each occurrence of it in the text stream, look at the surrounding window of four words (two before, two after):

$$w_1 \;\; w_2 \;\; w \;\; w_3 \;\; w_4.$$

Keep count of how often context words from C appear in these positions around word w. That is, for $w \in V, c \in C$, define

$$n(w,c) = \# \text{ of times } c \text{ occurs in a window around } w.$$

Using these counts, construct the probability distribution $\Pr(c|w)$ of context words around w (for each $w \in V$), as well as the overall distribution $\Pr(c)$ of context words. These are distributions over C.

- Represent each vocabulary item w by a $|C|$-dimensional vector $\Phi(w)$, whose c'th coordinate is:

$$\Phi_c(w) = \max\left(0,\ \log \frac{\Pr(c|w)}{\Pr(c)}\right).$$

This is known as the (positive) *pointwise mutual information*, and has been quite successful in work on word embeddings.

Now, use this matrix to come up with a 100*-dimensional representation of words.*

(a) *Give a description of your 100-dimensional embedding.* The description should be concise and clear, and should make it obvious exactly what steps you took to obtain your word embeddings. Below, we will denote these as $\Psi(w) \in \mathbb{R}^{100}$, for $w \in V$. Also clarify exactly how you selected the vocabulary V and the context words C.

(b) *Investigate your embedding using nearest neighbor.* Pick a collection of 25 words $w \in V$. For each w, return its nearest neighbor $w' \neq w$ in V. A popular distance measure to use for this is *cosine distance*:

$$1 - \frac{\Psi(w) \cdot \Psi(w')}{\|\Psi(w)\|\|\Psi(w')\|}.$$

Do the results make any sense?

(c) *Investigate your embedding by clustering the vocabulary.* Using the vectorial representation $\Psi(\cdot)$, cluster the words in V into 100 groups. Clearly specify what algorithm and distance function you use for this, and the reasons for your choices. Look over the resulting 100 clusters. Do any of them seem even moderately coherent? Pick out a few of the best clusters and list the words in them.

Note: The Brown corpus is very small. Current work on word embeddings uses data sets that are several orders of magnitude larger.