DD2434 Machine Learning, Advanced Course
Assignment 2

Jens Lagergren, Hedvig Kjellström and Cheng Zhang

Deadline 12.00 (noon) (CET) December 16th, 2015

You will present the assignment will by a written report that you can mail to BOTH jensl@kth.se (who corrects Tasks 2.1-2.4) and hedvig@kth.se (who corrects Tasks 2.5-2.7) before the deadline. From the report it should be clear what you have done and you need to support your claims with results. You are supposed to write down the answers to the specific questions detailed for each task. This report should clearly show how you have drawn the conclusions and come up with the derivations. Your assumptions, if any, should be stated clearly. For the practical part of the task you should not show any of your code but rather only show the results of your experiments using images and graphs together with your analysis.

Being able to communicate your results and conclusions is a key aspect of any scientific practitioner. It is up to you as a author to make sure that the report clearly shows what you have done. Based on this, and only this, we will decide if you pass the task. No detective work should be needed on our side. Therefore, neat and tidy reports please!

I very much recommend you to get used to LaTeX to write your report. It is an amazing tool that you will find very useful in your further endeavors as a scientist.

The grading of the assignment will be as follows,

E Completed Tasks 2.1, 2.2, and 2.5.

D E + Completed one of Tasks 2.3, 2.4, 2.6, and 2.7.

C E + Completed two of Tasks 2.3, 2.4, 2.6, and 2.7.

B E + Completed three of Tasks 2.3, 2.4, 2.6, and 2.7.

A Completed all tasks.

These grades are valid for review December 18th, 2015. See the course web page, HT 2015 - Assignments in the menu, for grading of delayed assignments.

Abstract

This assignment containts two parts. In the first part, you will get experience with different types of graphical models and with inference over graphical models. In the second part, you will be acquainted with two types of latent representation models where there are assumptions about the data that makes it efficient to use non-Gaussian priors over the distributions in the latent space. To summarize, all methods in this assignment make use of different kinds of knowledge about the structure of the data.

I Graphical Models

2.1 Qualitiative effects in a Directed Graphical Model (DGM)

Consider the Directed Acyclical Graph (DAG) of a DGM shown in Figure 1. The variables are binary-valued. The conditional probability densities are not known but, in contrast, available information reveal how each variable qualitatively influences its children. The interpretation of the influences, which are denoted $\overset{+}{\to}$ and $\overset{-}{\to}$, are:

$\overset{+}{\to}$ means $p(y^1|x^1, \mathbf{c}) > p(y^1|x^0, \mathbf{c})$, for all values $\mathbf{c}$ of Y's parents.

$\overset{-}{\to}$ means $p(y^1|x^1, \mathbf{c}) < p(y^1|x^0, \mathbf{c})$, for all values $\mathbf{c}$ of Y's parents.

You should also assume the parents in any V-structure are conditionally dependent given the the common child. Consider the following pairs of conditional probabilities:

1. $p(t^1|d^1)$ and $p(t^1)$
2. $p(d^1|t^0)$ and $p(d^1)$
3. $p(h^1|e^1, f^1)$ and $p(h^1|e^1)$
4. $p(c^1|f^0)$ and $p(c^1)$
5. $p(c^1|h^0)$ and $p(c^1)$
6. $p(c^1|h^0, f^0)$ and $p(c^1|h^0)$
7. $p(d^1|h^1, e^0)$ and $p(d^1|h^1)$
8. $p(d^1|e^1, f^0, w^1)$ and $p(d^1|e^1, f^0)$
9. $p(t^1|w^1, f^0)$ and $p(t^1|w^1)$.

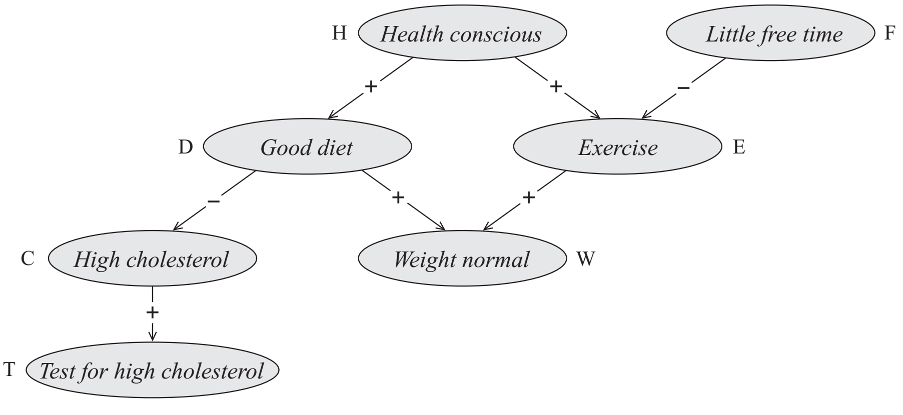


Figure 1: The DAG with qualitative influence information.

Question 1: *In which pairs is one value larger than the other? Explain your choices.*

Question 2: *Which pairs are equal? Explain your choices.*

Question 3: *Which pairs are incomparable (i.e., the two values can not be compared based on the information available in the DAG.) Explain your choices.*

2.2 The casino model

Consider the following generative model. There are $2K$ tables in a casino, $T_1, ..., T_K, T'_1, ..., T'_K$ of which each is equipped with a single dice (which may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$) and N players $P_1, ..., P_N$ of which each is equipped with a single dice (which may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$). Each player P_i visits K tables. In the k:th step, if the the previous table visited was T_{k-1}, the player visits T_k with probability $1/4$ and T'_k with probability $3/4$, and if the previous table visited was T'_{k-1}, the player visits T'_k with probability $1/4$ and T_k with probability $3/4$. So, in each step the probability of staying among the primed or unprimed tables is $1/4$. At table k the player i throws the player's own dice as well as the table's dice. We then observe the sum S^i_k of the two dice, while the outcome of the table's dice X_k and the player's own dice Z_k are hidden variables. So for player i, we observe $S^i = S^i_1, ..., S^i_K$, and the overall observation for N players is $S^1, ..., S^N$.

Question 4: *Provide a drawing of the graphical model* Θ.

Question 5: *Provide an implementation (in Matlab or Python) of the model* Θ.

Question 6: *Provide data generated using at least three different sets of categorical dice distributions – what does it look like for all perfect dice with uniform distributions, for example, or if all of them are perfect instead of one, or if all are bad in the same way?*

2.3 Sampling tables given dice sums

You will now design an algorithm that does inference on the casino model that you designed in Task 2.2.

Question 7: *Describe an algorithm that, given (1) the parameters* Θ *of the full casino model of Task 2.2 (so,* Θ *is all the categorical distributions corresponding to all the dice), (2) a sequence of tables* $z_1 \ldots, z_n$*, and (3) an observation of dice sums* $s_1, \ldots, s_K$*, outputs* $p(z_1, ..., z_K | s_1, \ldots, s_K, \Theta)$.

Notice, in the above DP algorithm you have to keep track of the last table visited.

Question 8: *You should also show how to sample* $z_1, \dots, z_K$ *from* $p(Z_1, ..., Z_K | s_1, \dots, s_K, \Theta)$ *as well as implement and show test runs of this algorithm. In order to design this algorithm show first how to sample* z_K *from*

$$p(Z_K | s_1, \dots, s_K, \Theta) = p(Z_K, s_1, \dots, s_K | \Theta) / p(s_1, \dots, s_K | \Theta)$$

and then z_{K-1} *from*

$$p(Z_{K-1} | z_K, s_1, \dots, s_K, \Theta) = p(Z_{K-1}, z_K, s_1, \dots, s_K | \Theta) / p(z_K, s_1, \dots, s_K | \Theta).$$

2.4 Expectation-Maximization (EM)

Consider the following simplification of the casino model from Problem 2.2. There are K tables in the casino $T_1, ..., T_K$ of which each is equipped with a single dice (which may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$) and N players $P_1, ..., P_N$ of which each is equipped with a single dice (which may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$). Each player P_i visits K tables in the order $1, ..., K$. At table k the player i throws their own dice as well as the tables dice. We then observe the sum S_k^i of the dice, while the outcome of the tables dice X_k and the player's own dice Z_k are hidden variables. So for player i, we observe $s^i = s_1^i, ..., s_K^i$, and the overall observation for N players is $s^1, ..., s^N$.

Design and describe an EM algorithm for this model. That is, an EM algorithm that given $s^1, ..., s^N$ finds locally optimal parameters for the categorical distributions (i.e., the dice), that is, the Θ maximising $P(s_1^i, ..., s_K^i | \Theta)$.

Question 9: *Present the algorithm written down in a formal manner (using both text and mathematical notation, but not pseudo code).*

Question 10: *Provide an implementation (in Matlab or Python) of the algorithm, and insert comments to explain the code.*

Question 11: *Test the implementation with data generated in Task 2.2, and provide graphs or tables of the results of testing it with the data.*

II Non-Gaussian Latent Representations

2.5 Independent Component Analysis (ICA)

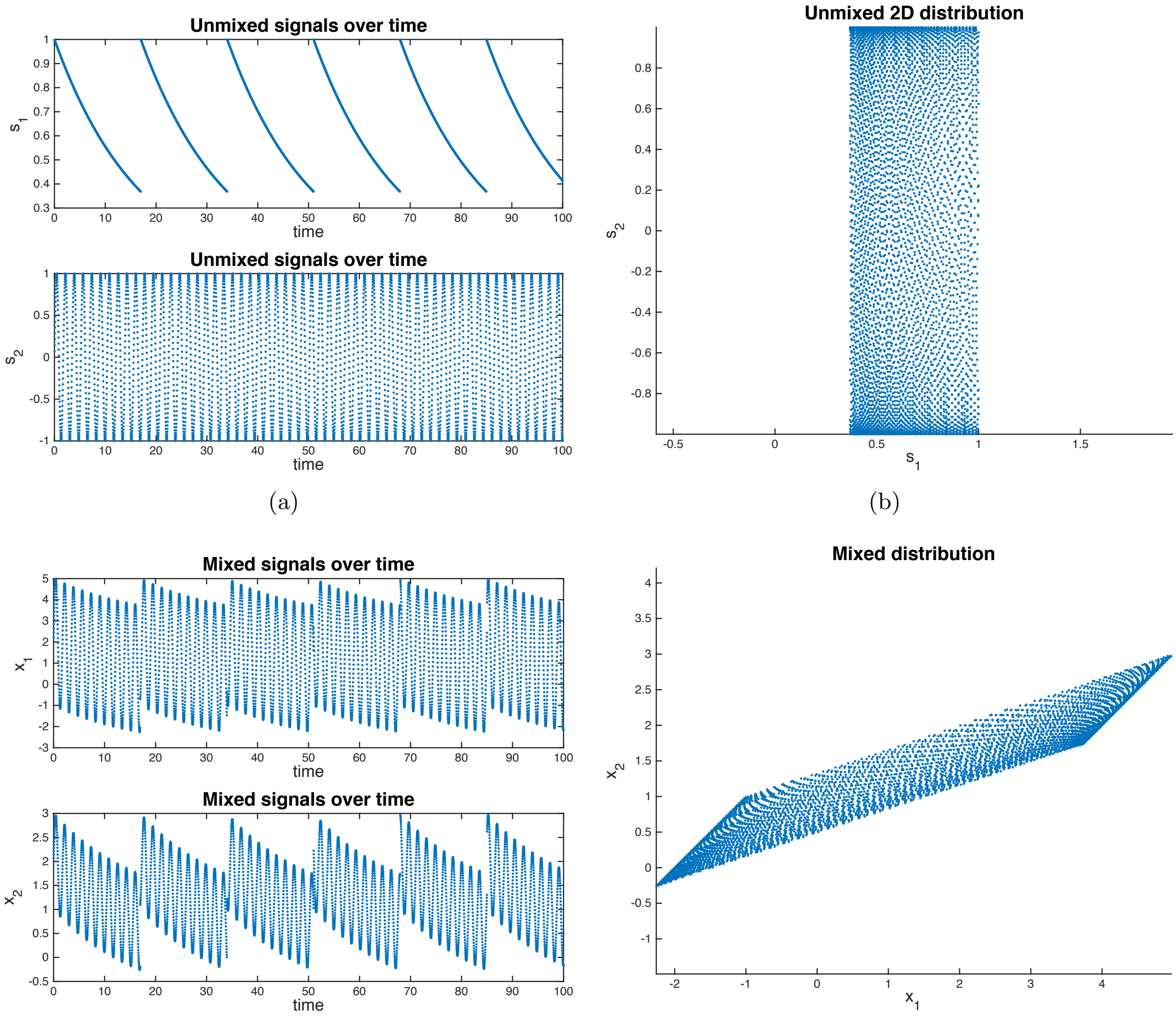
In this task we will pick up the thread from Task 1.4 in Assignment 1, but now in an unsupervised learning setting. In Task 1.4, you essentially implemented Probabilistic Principal Component Analysis, PPCA (Bishop, Section 12.2) using an iterative solution rather than the closed-form solution found in Bishop, Section 12.2. Here you will implement a similar transform, but with a different prior assumption.

Note that we, according to tradition, and to coincide with Hyvärinen and Oja, use a different notation than Assignment 1: the data variable is here denoted $\mathbf{x}$ *while the underlying latent variable is denoted* $\mathbf{s}$ *(or* $\mathbf{z}$ *in Bishop). Discrepancies in representation is a necessary evil that you will come across many times in your professional life.*

Before proceeding, read Bishop, Section 12.4.2, as well as Hyvärinen and Oja, Sections 1-6, which give an introduction to ICA and the types of data when it is applicable. (I can also recommend the concise Wikipedia page on FastICA.) Essentially, PPCA and ICA differ in the type of assumption they make about the prior distribution over the latent variable. In PPCA it is assumed that the latent variable is normally distributed:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_z, \Sigma_z) \ , \tag{1}$$

where $\mathbf{m}_z$ is the mean and Σ_z the covariance of the latent distribution. However, ICA instead makes



(a) (b) (c) (d)

Figure 2: The dataset found in `DataICA.txt`.

the assumption that the dimensions in the latent space all are independent:

$$p(\mathbf{s}) = \prod_j p(s_j) \ . \tag{2}$$

The reason for the terminology $\mathbf{s}$ is that the original application of ICA was ”the cocktail party problem”, i.e., factorization of a multidimensional sound signal $\mathbf{x}$, each x_k corresponding to the recorded sound from one microphone k, into the individual signals $\mathbf{s}$, each s_j corresponding to the speech signal from one speaker j.

You will now work with a signal that is constructed to have the same properties as the signals in the cocktail party problem. Figure 2(a) shows two individual, independent signals over time. If the order with respect to time is disregarded, the measurement of both signals at a certain time i can be represented as a point $\mathbf{s}_i$ in a 2D space, see Figure 2(b).

Two microphones record two different linear combinations of the two signals. The linear combinations over time are shown in Figure 2(c), and their 2D point representations $\mathbf{x}_i$ in Figure 2(d). These are found in `DataICA.txt`, which are linked from the course page, HT 2015 mladv15 > Assignments. The task is now to recreate the separated signals using ICA. (You are NOT allowed to copy code from readymade code packages.)

Question 12: *First, whiten the data as described in Section 5 of Hyvärinen and Oja. Show plots that illustrate both the two obtained eigenvectors and their eigenvalues, and a plot of the whitened pointset* $\{\tilde{\mathbf{x}}_i\}$*, and describe (using both text and mathematical notation, but not pseudo code) how you obtained it.*

Question 13: *Then, describe (using both text and mathematical notation, but not pseudo code) why a PPCA transform can not recover the independent components in the data.*

Question 14: *Finally, recover the two independent components using FastICA as described in Section 6 of Hyvärinen and Oja. Show plots that illustrate both the two obtained mixing vectors, and a plot of the decorrelated pointset* $\{\mathbf{s}_i\}$*, and describe (using both text and mathematical notation, but not pseudo code) how you obtained it.*

Two notes in connection the the last question: See Wikipedia for missing information about g'. Moreover, you can only recover $\{\mathbf{s}_i\}$ up to a scale factor, so do not be worried if you obtain a scaled copy of Figure 2(b).

2.6 Implementation of Latent Dirichlet Allocation (LDA)

Another type of latent representation model with non-Gaussian priors, developed for representation of text documents, is Latent Dirichlet Allocation (LDA). Figure 3 shows a graphical representation of this model. A document m, observed as a bag of words $\{w_{mi}\}$ (i.e., a multinomial distribution over the language with V words in which the document is written) can be represented as a mixture θ_m of k topics. Since $k \ll V$, θ_m is a very compact low-dimensional latent representation of the document m. The prior assumption on the latent topic space θ is that it is Dirichlet distributed.

Before proceeding further, read Bishop page 363 for an explanation of the plate notation in Figure 3, Bishop Section 11.3 for an introduction to Gibbs sampling, as well as Blei and Lafferty, and Griffith for an introduction to LDA. I can also recommend the Wikipedia pages on Gibbs sampling and LDA.

You will work with data in the form of text documents represented as bags of words $\{w_m^i\}$. In the 7 files `R3*.txt`, which are linked from the course page, HT 2015 mladv15 > Assignments, a subset of of the news article dataset Reuters 21578[1] is given. Each document is a news article. Our dataset is

[1] http://csmining.org/index.php/r52-and-r8-of-reuters-21578.html

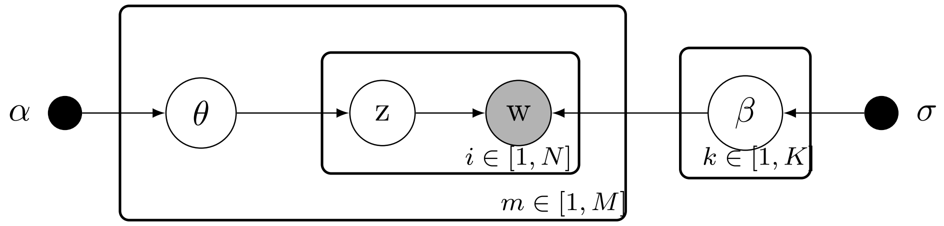


Figure 3: Graphic representation of LDA. Words w are observed for each document. α and σ are the hyperparameters, which are manually set with Gibbs sampling.

called R3 since it only contains 3 classes of news articles (crude, trade and money-fx). The original news data of these three classes are given in `R3-trn-all.txt`, where each row contain the words from a document. To ease the task, we have helped you preprocess the data so that each document is represented by a document id, followed by word id and word count of all unique words in the document. This preprocessed data is found in `R3-trn-all_run.txt`, where each row is a document in the format:

```
documentID wordID:counts wordID:counts ...
```

File `R3_all_Dictionary.txt` is the dictionary over words where the first row has wordID 0, the second wordID 1 and so on.

The task is now to implement the model shown in Figure 3, and train it with this document collection using Gibbs sampling. (You are NOT allowed to copy code from readymade code packages.)

Question 15: *Implement LDA with Gibbs sampling and run it with the given R3 data. Try a couple of settings of the parameter K, most importantly $K = 3$ when you can expect topics to correspond to the three classes of documents, but also $K = 10$ or $K = 15$ when the topics do not correspond to any semantic label but can be seen as a latent, low-dimensional representation of the document. (The hyperparameters α and σ can be set to 0.3). In the report, you should – for each model with different K – show a list of the 30 most common words in each topic k, along with their weights in the learned per topic word distribution β_k. You should also, for one of the training documents m, show the latent per document topic distribution θ_m for that document. Print out the words of the document m in the report and explain, with examples from the document, the reasons for this topic distribution θ_m.*

In `R3-tst-all.txt` and `R3-tst-all_run.txt` we provide test data for the R3 dataset. The label ground truth of the testing data is found in `R3-GT.txt`. You will also need the labeling of the training data; they can be found in `R3-Label.txt`.

Question 16: *Now use a $K > 3$, for example $K = 10$ or $K = 15$. For each test document $m_{\mathbf{test}}$, infer the topic distribution $\theta_{m_{\mathbf{test}}}$. Classify each $\theta_{m_{\mathbf{test}}}$ with kNN and the training document representations θ_m. Study a couple of correctly classified documents, and a couple of wrongly classified documents. Does it make sense, i.e., are the wrongly classified documents more atypical of their class than the correctly classified?*

2.7 Derivation of Gibbs sampling for Latent Dirichlet Allocation (LDA)

In this Gibbs sampling for LDA, we would like to sample on the topic assignment z_{mi}. We would like to compute the full conditional probability for z_{mi} to drive the update question. We can get

$$p(z_{mi} = k | w, z_{\neg mi}, \alpha, \sigma) \propto \frac{n_{k,\neg i}^{(w_{mi})} + \sigma}{\sum_{v=1}^{V}(n_{k,\neg i}^{(v)} + \sigma)} \cdot \frac{n_{m,\neg i}^{(k)} + \alpha}{(\sum_{j=1}^{K}(n_m^{(j)} + \alpha)) - 1} \;, \tag{3}$$

where $\neg i$ means that the ith word in the mth document is not counted, $n_m^{(k)}$ stands for the number of words which are assigned to the topic k from the document m, and $n_k^{(v)}$ stands for the number of words v assigned to topic (k).

Naturally, the end result of the latent parameters can be computed as:

$$\theta_{mk} = \frac{n_m^{(k)} + \alpha}{\sum_{j=1}^{K}(n_m^{(j)} + \alpha)}$$

$$\beta_{kv} = \frac{n_k^{(v)} + \sigma}{\sum_{i=1}^{V}(n_k^{(i)} + \sigma)}$$

Question 17: *Please derive Equation (3) from the dependencies represented in Figure 3. We require small steps where only one type of mathematic operation is allowed in each step. Comments should be added between each step.*

Good Luck!