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 2.1-2.3) and hedvig@kth.se (who corrects Tasks 2.4-2.6) 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 one of Tasks 2.1-2.3 and one of Tasks 2.4-2.6.

D Completed two of Tasks 2.1-2.3 and one of Tasks 2.4-2.6 **OR**
one of Tasks 2.1-2.3 and two of Tasks 2.4-2.6

C Completed 4 of the 6 tasks.

B Completed 5 of the 6 tasks.

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 Bayes Net

TBA shortly.

2.2 Expectation-Maximization

TBA shortly.

2.3 Viterbi

TBA shortly.

II Non-Gaussian Latent Representations

2.4 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. It is certainly possible to perform this task without having done Task 1.4, but it might be useful to think about the relations. 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.

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

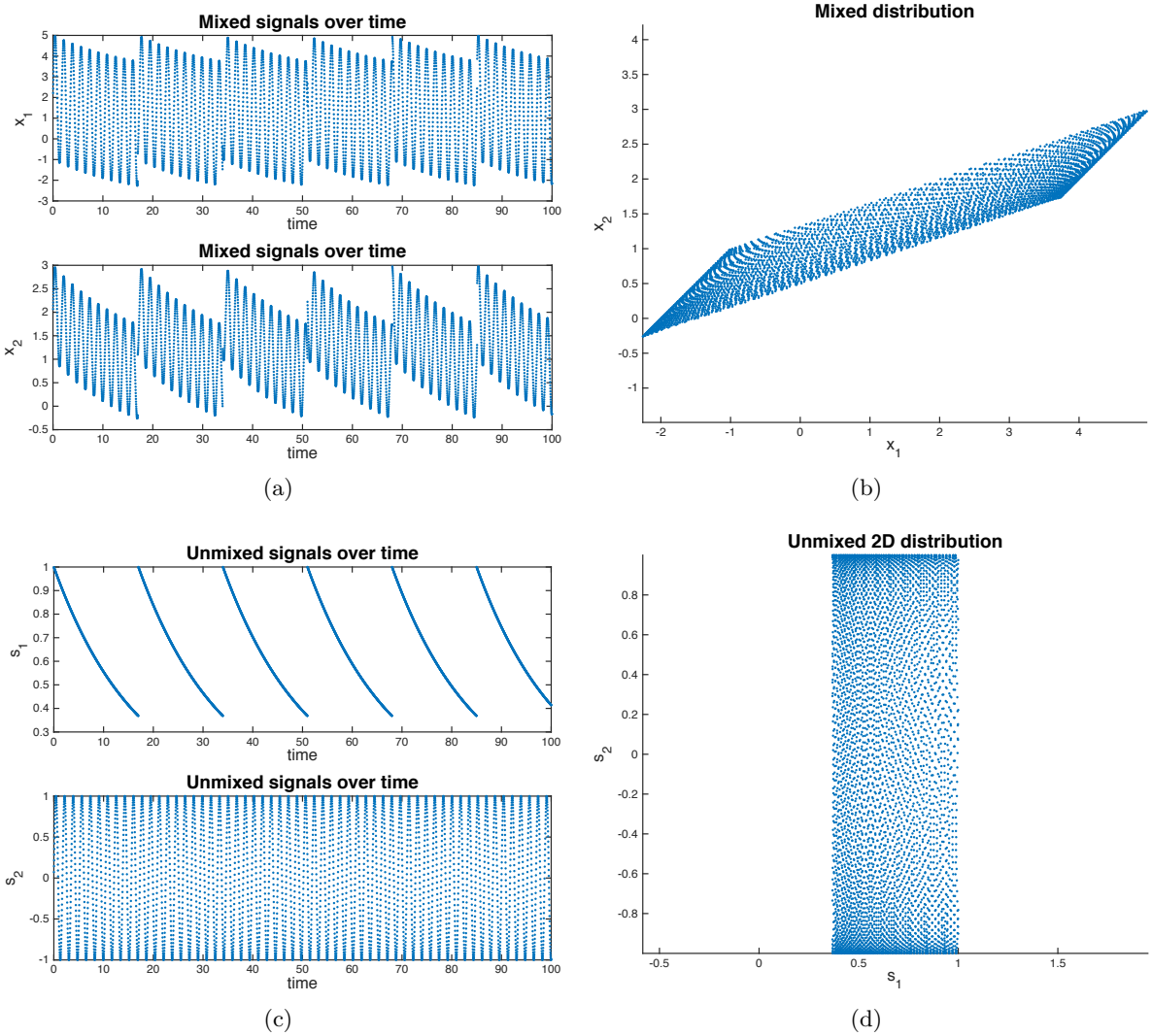

Figure 1: The dataset found in `DataICA.txt`.

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 1(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 1(b).

Two microphones record two different linear combinations of the two signals. The linear combinations over time are shown in Figure 1(c), and their 2D point representations $\mathbf{x}_i$ in Figure 1(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 1: *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 2: *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 3: *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 1(d).

2.5 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 2 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 2, Bishop Section 11.3 for an introduction to Gibbs sampling, as well as Blei and Lafferty 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

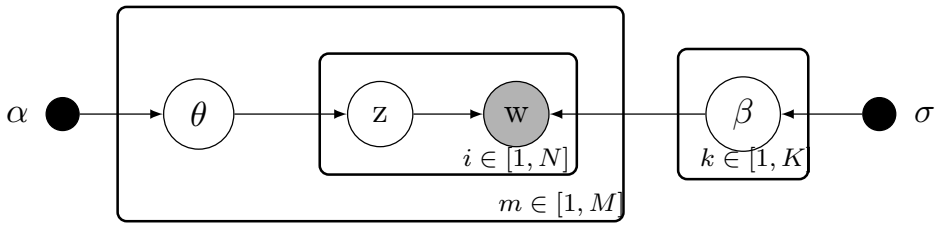

Figure 2: Graphic representation of LDA. Words w are observed for each document. α and σ are the hyperparameters, which are manually set with Gibbs sampling.

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 2, and train it with this document collection using Gibbs sampling. (You are NOT allowed to copy code from readymade code packages.)

Question 4: *Implement LDA with Gibbs sampling and run it with the given R3 data. Vary the settings of the three parameters K, α and σ. What effect do they have on the learned topic space? (In particular, you should try $K = 3$, i.e., one topic for each class.) In the report, you should show a list of the 20 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 5: *Now use a $K > 3$, for example $K = 10$ or $K = 15$. For each test document $m_{\mathbf{test}}$, infer the topic distribution $\theta_{m_{\mathbf{test}}}$, using the per topic word distributions β learned in the training phase. 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. Are the correctly classified documents more typical for their class?*

2.6 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 6: *Please derive Equation (3) from the dependencies represented in Figure 2. 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!