



UNIVERSIDAD
DE GRANADA

STATISTICAL MODELS WITH VARIATIONAL METHODS

LUIS ANTONIO ORTEGA ANDRÉS

End-of-Degree Project
Computer Science and Mathematics

Tutor
Serafín Moral Callejón

FACULTY OF SCIENCE
H.T.S. OF COMPUTER ENGINEER AND TELECOMMUNICATIONS

Granada, Saturday 20th June, 2020

ABSTRACT

Some introduction about how important Variational methods are nowadays and what this project is about.

CONTENTS

I BASIC CONCEPTS

1	PROBABILITY	6
2	DISTRIBUTIONS	11
2.1	Discrete Distributions	13
2.2	Continuous Distributions	13
2.3	Kullback-Leibler Divergence	16
3	GRAPH THEORY	18

II VARIATIONAL INFERENCE

4	INTRODUCTION	21
5	MEAN-FIELD VARIATIONAL INFERENCE	24
5.1	The mean-field variational family	24
5.2	CAVI Algorithm	25
6	EXPONENTIAL FAMILY	27
6.1	Latent variable and conditionally conjugate models	28
6.2	CAVI in conditionally conjugate models	30

III GRAPHICAL MODELS

7	BAYESIAN NETWORKS	33
8	MARKOV RANDOM FIELDS	37

IV BAYESIAN NETWORKS LEARNING

9	INTRODUCTION	40
9.1	Utility	42
10	MAXIMUM LIKELIHOOD TRAINING	43
10.1	ML and KL divergence	43
11	BAYESIAN TRAINING	45
11.1	Global and local parameter independence	45
11.2	Learning binary variables	46
11.3	Learning discrete variables	48
11.3.1	No parents	48
11.3.2	Parents	49
12	STRUCTURE LEARNING	51
12.1	PC Algorithm	51
12.2	Independence Learning	52
13	MISSING VARIABLES	53
14	VARIATIONAL INFERENCE IN BAYESIAN NETWORKS	55
14.1	Expectation Maximization	55
14.1.1	General case	55
14.1.2	EM increases the marginal likelihood	57

14.1.3	Bayesian Networks case	58
14.2	Distribution mixture example with EM	59
14.3	EM Extensions	61
14.3.1	Partial steps	61
14.4	Variational Bayes	61
14.4.1	VB is a generalization of the EM algorithm	63
14.5	Comparison	64
15	VARIATIONAL MESSAGE PASSING	65
15.1	Variational Message Passing Algorithm	67
 V COMMONLY STUDIED LATENT VARIABLE MODELS		
16	GAUSSIAN MIXTURE	70
17	LATENT DIRICHLET ALLOCATION	72
18	PROBABILISTIC PRINCIPAL COMPONENTS ANALYSIS	75
18.1	Artificial Neural networks	76
18.2	Non-linear PCA	76
18.3	Variational Auto-encoder	77
 VI CASE STUDY		
19	INFERPY USAGE	79
19.1	Installation	80
19.2	Usage guide with PCA	80
20	GAUSSIAN MIXTURE	84

Part I

BASIC CONCEPTS

In this part we will introduce the underlying concepts of probability and graph theory that we will need.

PROBABILITY

All our theory will be made under the assumption that there is a *referential set* Ω , set of all possible outcomes of an experiment. Any subset of Ω will be called an *event*.

Definition 1. Let $\mathcal{P}(\Omega)$ be the power set of Ω . Then, $\mathcal{F} \subset \mathcal{P}(\Omega)$ is called a σ -algebra if it satisfies:

- $\Omega \in \mathcal{F}$.
- \mathcal{F} is closed under complementation.
- \mathcal{F} is closed under countable unions.

From these properties it follows that $\emptyset \in \mathcal{F}$ and that \mathcal{F} is closed under countable intersections.

The tuple (Ω, \mathcal{F}) is called a *measurable space*.

Definition 2. A *probability* P over (Ω, \mathcal{F}) is a mapping $P : \mathcal{F} \rightarrow [0, 1]$ which satisfies

- $P(\alpha) \geq 0 \quad \forall \alpha \in \mathcal{F}$.
- $P(\Omega) = 1$.
- P is countably additive, that is, if $\{\alpha_i\}_{i \in \mathbb{N}} \subset \mathcal{F}$, is a countable collection of pairwise disjoint sets, then

$$P\left(\bigcup_{i \in \mathbb{N}} \alpha_i\right) = \sum_{i \in \mathbb{N}} P(\alpha_i).$$

The first condition guarantees non negativity. The second one states that the *trivial event* has the maximal possible probability of 1. The third condition implies that given a set of pairwise disjoint events, the probability of either one of them occurring is equal to the sum of the probabilities of each one.

From these conditions follows

- $P(\emptyset) = 0$.
- $P(\alpha \cup \beta) = P(\alpha) + P(\beta) - P(\alpha \cap \beta)$.

The triple (Ω, \mathcal{F}, P) is called a *probability space*.

Definition 3. Given two events $\alpha, \beta \in \mathcal{F}$, with $P(\beta) \neq 0$, the conditional probability of α given β is defined as the quotient of the probability of the joint events and the probability of β :

$$P(\alpha | \beta) = \frac{P(\alpha \cap \beta)}{P(\beta)}.$$

Theorem 1. (Bayes' theorem). Let α, β be two events of an experiment, given that $P(\beta) \neq 0$. Then

$$P(\alpha | \beta) = \frac{P(\beta | \alpha)P(\alpha)}{P(\beta)}.$$

Example 1. Consider a study where the relation of a disease d and an habit h is being investigated. Suppose that $P(d) = 10^{-5}$, $P(h) = 0.5$ and $P(h | d) = 0.9$. What is the probability that a person with habit h will have the disease d ?

$$P(d | h) = \frac{P(d \cap h)}{P(h)} = \frac{P(h | d)P(d)}{P(h)} = \frac{0.9 \times 10^{-5}}{0.5} = 1.8 \times 10^{-5}.$$

If we set the probability of having habit h to a much lower value as $P(h) = 0.001$, then the above calculation gives approximately $1/100$. Intuitively, a smaller number of people have the habit and most of them have the disease. This means that the relation between having the disease and the habit is stronger now compared with the case where more people had the habit.

Definition 4. We say that two events $\alpha, \beta \in \mathcal{F}$ are *independent* if knowing one of them does not give any extra information about the other. Mathematically,

$$P(\alpha \cap \beta) = P(\alpha)P(\beta), \quad P(\alpha | \beta) = P(\alpha).$$

Let $\gamma \in \mathcal{F}$, we say that α and β are *conditionally independent* on γ , $\alpha \perp\!\!\!\perp \beta | \gamma$ if and only if

$$P(\alpha \cup \beta | \gamma) = P(\alpha | \gamma)P(\beta | \gamma).$$

Otherwise, they are said to be *conditionally dependent* on γ , $\alpha \not\perp\!\!\!\perp \beta | \gamma$.

Now we are going to introduce the concept of *random variable* and some properties as we have done with events.

Definition 5. A function $f : \Omega_1 \rightarrow \Omega_2$ between two measurable spaces $(\Omega_1, \mathcal{F}_1)$ and $(\Omega_2, \mathcal{F}_2)$ is said to be *measurable* if $f^{-1}(\alpha) \in \mathcal{F}_1$ for every $\alpha \in \mathcal{F}_2$.

Definition 6. A *random variable* is a measurable function $X : \Omega \rightarrow E$ from a probability space (Ω, \mathcal{F}, P) to a measurable space (E, \mathcal{F}') verifying $X(\omega) \in \mathcal{F}' \forall \omega \in \Omega$.

The probability of X taking a value on a measurable set $S \in E$ is written as

$$P_X(S) = P(X \in S) = P(\{a \in \Omega \mid X(a) \in S\}).$$

We could make a question like “How likely is that the value of X equals a ?”. This is the same as asking for the probability of the set $\{\omega \in \Omega \mid X(\omega) = a\}$.

We will set the following notation that is going to be used, that is: random variables will be denoted with an upper case letter like X and a set of variables with a bold symbol like \mathbf{X} . The meaning of $P(\text{state})$ will be clear without a reference to the variable. Otherwise $P(X = \text{state})$ will be used. Using a lower case letter like $P(x)$ will denote the probability of the corresponding upper case variable X taking a specific value.

Definition 7. The *cumulative distribution function* of a real-valued random variable X is defined as

$$F_X(x) = P(X \leq x),$$

where the right-hand side represents the probability of the random variable taking value below or equal to x .

Definition 8. When the image of a random variable X is countable, the random variable is called a *discrete random variable* and its *probability mass function* p gives the probability of it being equal to some value:

$$p(x) = P(X = x).$$

In case the image is uncountable and real, then X is called a *continuous random variable* and if there exists a non-negative Lebesgue-integrable f such that

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f(u)du,$$

then it is called its *probability density function*.

A *mixed random variable* is a random variable who is neither discrete nor continuous, it can be realized as the sum of a discrete and continuous random variables. An example of a random variable of mixed type would be based on an experiment where a coin is flipped and a random positive number is chose only if the result of the coin toss is heads, -1 otherwise.

From now on, $P(x)$ will denote $f_X(x)$ when X is a continuous random variable. We define the *probability distribution* P_X of a random variable X over the probability space (Ω, \mathcal{F}, P) as the pushforward measure of it, that is, $P_X = PX^{-1}$.

Integrate notation will be used in both continuous and discrete cases, where the last one can be interpreted as integration with respect to the *counting measure* defined as

$$\#(dx) = \sum_{n \in \mathcal{I}} \delta(x - n)dx,$$

where \mathcal{I} is the set of values X can take, and δ is the Dirac distribution. Given this measure, integration corresponds to summation as

$$\int_x P(x)\#(dx) = \sum_{n \in \mathcal{I}} \int_x P(x)\delta(x - n)dx = \sum_{n \in \mathcal{I}} P(n).$$

Where we used that $\int f(x)\delta(x - x_0) = f(x_0)$. Given this, from now on, we will use the integration notation for both discrete and continuous variables given that the integrals will be respect to the counting measure when needed.

Definition 9. As we did for events, we can define the *conditional probability* over random variables, let X, Y be two random variables, then

$$P(x | y) = \frac{P(x, y)}{P(y)}.$$

It is required that $P(y) \neq 0$ for the conditional probability to be defined.

We can also enunciate the *Bayes' theorem* as

$$P(x, y) = \frac{P(y | x)P(x)}{P(y)}$$

Clearly an arbitrary number of variables can be considered in both cases.

Definition 10. The *marginal distribution* of a subset of random variables is the probability distribution of the variables contained in that subset.

Let X, Y be two random variables, it follows that

$$P(x) = \int_y P(x, y).$$

Definition 11. Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ be a set of random variables, the *joint probability distribution* for \mathbf{X} is function that gives the probability of each random variable X_i falling in a particular range or discrete set of values for that variable. It is called a *multi-variate distribution*.

When using only two random variables, then is called a *bi-variate distribution*.

This distribution can be expressed either in terms of a joint cumulative distribution function

$$F_{\mathbf{X}}(\mathbf{x}) = F_{X_1, \dots, X_n}(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_n \leq x_n)^1,$$

or using a probability density or mass function.

Definition 12. We say that two random variables X and Y are *independent* if knowing one of them doesn't give any extra information about the other. Mathematically,

$$P(x, y) = P(x)P(y).$$

From this it follows that if X and Y are independent, then $P(x | y) = P(x)$.

Definition 13. Let X, Y and Z be three random variables, then X and Y are *conditionally independent* given Z if and only if

$$P(x, y | z) = P(x | z)P(y | z),$$

in that case we will denote $X \perp\!\!\!\perp Y | Z$. If X and Y are not conditionally independent, they are *conditionally dependent* $X \not\perp\!\!\!\perp Y | Z$

¹ Where $\mathbf{x} = (x_1, \dots, x_n)$

Both independence definitions can be made over sets of variables \mathbf{X}, \mathbf{Y} and \mathbf{Z} in a straight forward way.

Definition 14. We say that a set of n random variables $\{X_1, \dots, X_n\}$ defined to assume values in $I \subset \mathbb{R}$ are *independent and identically distributed (i.i.d)* if and only if they are independent, i.e,

$$F_{X_1, \dots, X_n}(x_1, \dots, x_n) = F_{X_1}(x_1) \dots F_{X_n}(x_n) \quad \forall x_1, \dots, x_n \in I,$$

and are identically distributed

$$F_{X_1}(x_1) = F_{X_k}(x_k) \quad \forall k \in \{2, \dots, n\} \text{ and } \forall x \in I.$$

Definition 15. A *multi-variate random variable* or *random vector* is a column vector $\mathbf{X} = (X_1, \dots, X_n)^T$ whose components are random variables that can be defined over different probability spaces.

Note that we use the same symbol \mathbf{X} for random vectors and sets of variables, but the meaning will be clear within the context.

DISTRIBUTIONS

In this section we will summarize some concepts concerning probability distributions among with some of the most used ones.

From now on, let X be a random variable and P its probability distribution.

Definition 16. The *mode* X_* of the probability distribution P is the state of X where the distribution takes it's highest value

$$X_* = \arg \max_x P(x).$$

A distribution could have more than one mode, in this case we say it is *multi-modal*.

Definition 17. The notation $\mathbb{E}[X]$ is used to denote the *average* or *expectation* of the values a real-valued variable takes respect to its distribution. It is worth mentioning that it might not exists. If X is non-negative, it is defined as

$$\mathbb{E}[X] = \int_{\Omega} X(\omega) dP(\omega)^1.$$

For a general variable X it is defined as $\mathbb{E}[X] = \mathbb{E}[X^+] - \mathbb{E}[X^-]$. Where

$$X^+(\omega) = \max(X(\omega), 0), \quad X^-(\omega) = \min(X(\omega), 0).$$

Suppose now that X is a real-valued random variable, in case it is also continuous, the expectation is

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} xf(x)dx.$$

And if it is discrete, let x_i be the values X can take, the expectation takes the form

$$\mathbb{E}[X] = \sum_{i=1}^{+\infty} x_i p(x_i)dx.$$

Let $g : \mathbb{R} \rightarrow \mathbb{R}$ be a measurable function, then $g \circ X$ is another random variable and we can talk about $\mathbb{E}[g(X)]$, so in case X is continuous, we have that

$$\mathbb{E}[g(X)] = \int_{-\infty}^{+\infty} g(x)f(x)dx.$$

¹ P is a measure over Ω

In the above definition, the expectation is calculated over the probability distribution P of the random variable, in future sections this distribution is unknown and a guessed distribution Q will be used. In this cases when the distribution is not clear from the context the notation $\mathbb{E}_Q[X]$ will be used.

Definition 18. We define the k^{th} moment of a distribution as the average of X^k over the distribution

$$\mu_k = \mathbb{E}[X^k]$$

For $k = 1$ it is typically denoted as μ . Note μ_k can also denote the k^{th} element in the mean vector of a multi-variate variable.

Definition 19. The *variance* of a distribution is defined as

$$\text{Var}(X) = \sigma^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X]^2 - \mathbb{E}[X^2]$$

The square root of the variance σ is called the *standard deviation*.

When using a multi-variate distribution $\mathbf{X} = (X_1, \dots, X_n)^T$ we can talk about the *covariance matrix* Σ whose elements are

$$\begin{aligned} \Sigma_{ij} &= \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] \\ &= \mathbb{E}[X_i X_j] - \mathbb{E}[X_i] \mathbb{E}[X_j] \end{aligned}$$

The following result will be helpful later on.

Proposition 1. Let $\mathcal{X} = \{X_1, \dots, X_n\}$ be a set of random variables, $\mathcal{X}_0 \subset \mathcal{X}$ and $P(\mathcal{X}), P(\mathcal{X}_0)$ their probability distributions. It follows that the expectation of a function g over \mathcal{X}_0 , verifies

$$\mathbb{E}_{P(\mathcal{X})}[g(\mathcal{X}_0)] = \mathbb{E}_{P(\mathcal{X}_0)}[g(\mathcal{X}_0)].$$

That is, we only need to know the marginal distribution of the subset in order to carry out the average.

Proof. Let $\mathcal{I} = (i_1, \dots, i_k)$ be the indexes corresponding to \mathcal{X}_0 , then

$$\begin{aligned} \mathbb{E}_{P(\mathcal{X})}[g(\mathcal{X}_0)] &= \int_{x_1} \cdots \int_{x_n} g(x_{i_1}, \dots, x_{i_k}) f(x_1, \dots, x_n) \\ &= \int_{x_{i_1}} \cdots \int_{x_{i_k}} g(x_{i_1}, \dots, x_{i_k}) \int \cdots \int f(x_1, \dots, x_n) dx_1, \dots, dx_n \\ &= \int_{x_{i_1}} \cdots \int_{x_{i_k}} g(x_{i_1}, \dots, x_{i_k}) f(x_{i_1}, \dots, x_{i_k}) = \mathbb{E}_{P(\mathcal{X}_0)}[g(\mathcal{X}_0)]. \end{aligned}$$

Where in the second-last equality we used marginalization. □

We are going to discuss now some examples of probability distributions that are going to be used from now on.

2.1 DISCRETE DISTRIBUTIONS

Bernoulli Distribution

The Bernoulli distribution describes a discrete binary variable X that takes the value 1 with probability p and the value 0 with probability $1 - p$.

$$P(x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases}.$$

Categorical Distribution

A generalization of the Bernoulli Distribution when the variable can take more than two states is the *Categorical Distribution*. Let $\text{Dom}(X) = \{1, \dots, N\}$, then X follows a categorical distribution of parameters $\theta = (\theta_1, \dots, \theta_N)$ if and only if

$$P(x \mid \theta) = \prod_{i=1}^N \theta_i^{\mathbb{I}[x=i]} \text{ and } \sum_{i=1}^N \theta_i = 1.$$

Binomial Distribution

The binomial distribution describes the number of successes in a sequence of independent Bernoulli Trials. A discrete binary random variable X follows a *binomial distribution* of parameters $n \in \mathbb{N}$ and $p \in [0, 1]$, denoted as $X \sim B(n, p)$ if and only if

$$P(x = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

Multinomial Distribution

The multinomial distribution is a generalization of the binomial distribution which describes the result of a sequence of independent trials in a categorical distribution. A discrete random variable X follows a *multinomial distribution* of parameters $n \in \mathbb{N}$, $\mathbf{p} = (p_1, \dots, p_K)$ such that $\sum p_k = 1$ if and only if

$$P(\mathbf{x} = x_1, \dots, x_K) = \begin{cases} \frac{n!}{x_1! \dots x_K!} \prod_{k=1}^K p_k^{x_k} & \text{if } \sum_k x_k = n \\ 0 & \text{otherwise} \end{cases}$$

2.2 CONTINUOUS DISTRIBUTIONS

Uni-Variate Normal Distribution

The *normal* or *Gaussian distribution* is a type of continuous probability distribution for real-valued random variables.

Definition 20. We say the real valued random variable X follows a *normal distribution* of parameters $\mu, \sigma \in \mathbb{R}$, denoted as $X \sim N(\mu, \sigma)$ if and only if, its probability density function exists and is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$

The parameter μ is the mean or expectation of the distribution and σ is its standard deviation.

The simplest case of a normal distribution is known as *standard normal distribution*, denoted as Z . It is a special case where $\mu = 0$ and $\sigma = 1$, then its density function is

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$

One of the properties of the normal distribution is that if $X \sim N(\mu, \sigma)$, $a, b \in \mathbb{R}$ and $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined as $f(x) = ax + b$, then $f(X) \sim N(\mu + b, a^2\sigma)$.

Multi-Variate Normal Distribution

This distribution plays a fundamental role in this project so we will discuss its properties in more detail.

This distribution is an extension of the uni-variate one when having a multi-variate random variable.

Definition 21. We say that a random vector $\mathbf{X} = (X_1, \dots, X_p)$ follows a *multi-variate normal distribution* of parameters $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma} \in \mathbb{M}_n(\mathbb{R})$, denoted as $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ if and only if its probability density function is

$$f(\mathbf{x}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}.$$

Where $\boldsymbol{\mu}$ is the mean vector of the distribution, and $\boldsymbol{\Sigma}$ the covariance matrix. The inverse matrix $\boldsymbol{\sigma}^{-1}$ is called *precision*. It also satisfies that

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}], \quad \boldsymbol{\Sigma} = \mathbb{E}\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\right].$$

As $\boldsymbol{\Sigma}$ is a real symmetric matrix, it can be eigendecomposed

$$\boldsymbol{\Sigma} = \mathbf{E}\boldsymbol{\Delta}\mathbf{E}^T,$$

where $\mathbf{E}^T \mathbf{E} = \mathbf{I}$ and $\boldsymbol{\Delta} = \text{diag}(\lambda_1, \dots, \lambda_n)$.

Using the transformation

$$\mathbf{y} = \boldsymbol{\Delta}^{\frac{1}{2}} \mathbf{E}^T (\mathbf{x} - \boldsymbol{\mu}),$$

we get that

$$(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma} (\mathbf{x} - \boldsymbol{\mu}) = \mathbf{y}^T \mathbf{y}.$$

Using this, the multi-variate Normal Distribution reduces to a product of n uni-variate standard normal distributions.

Gamma Distribution

Another continuous distribution that we are going to use is the *Gamma distribution*.

Definition 22. We say that a continuous random variable X defined on \mathbb{R}^+ follows a *Gamma distribution* of parameters $\alpha, \beta > 0$, denoted as $X \sim \text{Gamma}(\alpha, \beta)$ if and only if its density function is

$$f(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)},$$

where Γ is the Gamma function defined as

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx.$$

The mean is given by $\mathbb{E}[X] = \frac{\alpha}{\beta}$.

Definition 23. The *inverse gamma distribution* is defined by the density function

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\beta/x}.$$

Beta Distribution

Definition 24. We say that a continuous random variable X defined on the interval $[0, 1]$ follows a *Beta distribution* of parameters $\alpha, \beta > 0$, denoted as $X \sim \text{Beta}(\alpha, \beta)$ if and only if its density function is

$$f(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1},$$

where $B(\alpha, \beta)$ is the *beta function* defined as

$$B(\alpha, \beta) = \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx.$$

The mean is given by $\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta}$.

Dirichlet Distribution

The Dirichlet distribution is a family of continuous multi-variate probability distributions parameterized by a vector α of positive reals. It is a multi-variate generalization of the Beta Distribution.

Definition 25. We say that a continuous random multi-variate variable \mathbf{X} with order $K \geq 2$, follows a *Dirichlet Distribution* with parameters $\alpha = (\alpha_1, \dots, \alpha_K)$, if and only if its density function is defined as

$$f(\mathbf{x}) = \frac{1}{B(\alpha)} \prod_{k=1}^K x_k^{\alpha_k-1},$$

and it satisfies that

$$\sum_{k=1}^K x_k = 1 \text{ and } x_k > 0 \forall k = 1, \dots, K.$$

Where the normalization constant is the multi-variate beta function

$$B(\alpha) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}.$$

When the vector parameter α is filled with the same value α_0 , the distribution is called Symmetric-Dirichlet with parameter α_0 .

Proposition 2. Let $(X_0, \dots, X_n) \sim \text{Dirichlet}(\alpha_0, \dots, \alpha_n)$, then $X_0 \sim \text{Beta}(\alpha_0, \alpha_1 + \dots + \alpha_n)$.

Proof. Following [Farrow (2008)], we can write the joint probability as

$$f(x_1, \dots, x_n) = f_1(x_1) f_2(x_2 | x_1) \dots f_{n-1}(x_{n-1} | x_1, \dots, x_{n-2}).$$

We do not need the last term because it is fixed given the others. In fact, let $A = \sum_i \alpha_i$, we can write it as

$$\begin{aligned} & \left(\frac{\Gamma(A)}{\Gamma(\alpha_1)\Gamma(A-\alpha_1)} x_1^{\alpha_1-1} (1-x_1)^{A-\alpha_1-1} \right) \left(\frac{\Gamma(A-\alpha_1)}{\Gamma(\alpha_2)\Gamma(A-\alpha_1-\alpha_2)} \frac{x_2^{\alpha_2-1} (1-x_1-x_2)^{A-\alpha_2-\alpha_1-1}}{(1-x_1)^{A-\alpha_1-1}} \right) \\ & \dots \left(\frac{\Gamma(A-\alpha_1-\dots-\alpha_{n-2})}{\Gamma(\alpha_{n-1})\Gamma(A-\alpha_1-\dots-\alpha_{n-1})} \frac{x_{n-1}^{\alpha_{n-1}-1} x_n^{\alpha_n-1}}{(1-x_1-\dots-x_{n-2})^{\alpha_{n-1}+\alpha_n-1}} \right). \end{aligned}$$

From this, we get that

$$f_1(x_1) = \frac{\Gamma(A)}{\Gamma(\alpha_1)\Gamma(A-\alpha_1)} x_1^{\alpha_1-1} (1-x_1)^{A-\alpha_1-1} \implies X_1 \sim \text{Beta}(\alpha_1, A-\alpha_1).$$

Making the decomposition over any other X_j , results on $X_j \sim \text{Beta}(\alpha_j, A-\alpha_j)$. \square

2.3 KULLBACK-LEIBLER DIVERGENCE

Definition 26. Let P and Q be two probability distributions over the same probability space \mathcal{X} , the *Kullback-Leibler divergence* $KL(Q | P)$ measures the “difference” between both distributions as

$$KL(Q | P) = \mathbb{E}_Q[\log Q(x) - \log P(x)].$$

The Kullback-Leibler divergence is defined if and only if for all $x \in \mathcal{X}$ such that $P(x) = 0$, then $Q(x) = 0$. In measure terms, Q is absolutely continuous with respect to P .

Proposition 3. The Kullback-Leibler divergence is always non-negative.

Proof. As the logarithm is bounded by $x - 1$, we can bound $\log \frac{P(x)}{Q(x)}$

$$\log x \leq x - 1 \implies \frac{P(x)}{Q(x)} - 1 \geq \log \frac{P(x)}{Q(x)}.$$

Since probabilities are non-negative, we can multiply by $Q(x)$ in the last inequality

$$P(x) - Q(x) \geq Q(x) \frac{\log P(x)}{\log Q(x)} = Q(x) \log P(x) - Q(x) \log Q(x).$$

Now we integrate (sum in case of discrete variables) both sides

$$0 \geq \mathbb{E}_Q \left[\log P(x) - \log Q(x) \right] \implies \mathbb{E}_Q \left[\log Q(x) - \log P(x) \right] \geq 0.$$

□

As a result, the Kullback-Leibler divergence is 0 if and only if the two distributions are equal almost everywhere.

GRAPH THEORY

Definition 27. A graph $G = (V, E)$ is a set of vertices or nodes V and edges $E \subset V \times V$ between them. If V is a set of ordered pairs then the graph is called a *directed graph*, otherwise if V is a set of unordered pairs it is called an *undirected graph*.

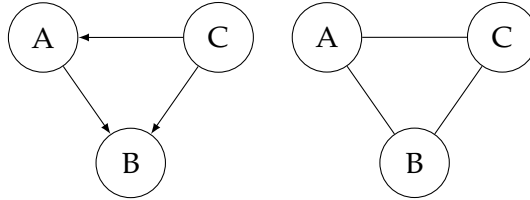


Figure 1: Example of directed and undirected graph, respectively.

Definition 28. In a directed graph $G = (V, E)$, a *directed path* $A \rightarrow B$ is a sequence of vertices $A = A_0, A_1, \dots, A_{n-1}, A_n = B$ where $(A_i, A_{i+1}) \in E \forall i \in 0, \dots, n-1$.

If G is a undirected graph, $A \rightarrow B$ is an *undirected path* if $\{A_i, A_{i+1}\} \in E \forall i \in 0, \dots, n-1$

Definition 29. Let A, B be two vertices of a directed graph G . If $A \rightarrow B$ is a directed path and $B \not\rightarrow A$ (meaning there isn't a directed path from B to A), then A is called an *ancestor* of B and B is called a *descendant* of A .

For example, in the figure 1, C is an ancestor of B .

Definition 30. A *directed acyclic graph* (DAG) is a directed graph such that no directed path between any two nodes revisits a vertex.

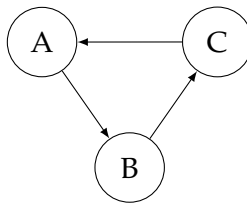


Figure 2: Example of graph which isn't a DAG.

As we can see in the figure 2, $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B$ is a path from A to B that revisits A .

Now where are going to define some relations between nodes in a DAG.

Definition 31. The *parents* of a node A is the set of nodes B such that there is a directed edge from B to A . The same applies for the *children* of a node.

The *Markov blanket* of a node is composed by the node itself, its children, its parents and the parents of its children.

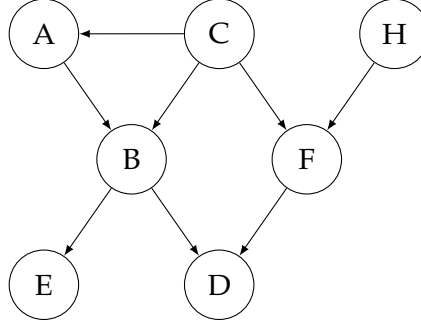


Figure 3: Directed acyclic graph

Definition 32. In a graph, the *neighbors* of a node are those directly connected to it.

We can use figure 3 to reflect on these definitions. The parents of B are $pa(B) = \{A, C\}$ and its children are $ch(B) = \{E, D\}$. Taking this into account, its neighbors are $ne(B) = \{A, C, E, D\}$ and its Markov blanket is $\{A, B, C, D, E, F\}$.

Definition 33. Let G be a DAG, U be a path between two vertex and $A \in U$

- A is called a *collider* if $\forall B \in ne(A) \cap U, (B, A) \in E$.
- A is called a *fork* if $\forall B \in ne(A) \cap U, (A, B) \in E$.

Notice, a vertex can be a collider for a path but not for others. A vertex is said to be a collider or a fork without any reference to the path when it is for any path that goes through it. This happens when the edge direction condition is satisfied for every neighbor.

For example in figure 3, D is a collider and C is a fork.

Definition 34. Let G be an undirected graph, a *clique* is a maximally connected subset of vertices. That is, all the members of the clique are connected to each others and there is no bigger clique that contains another.

Formally, $S \subset V$ is a *clique* if and only if $\forall A, B \in S, \{A, B\} \in E$ and $\nexists C \in V \setminus S$ such that $\forall A \in S, \{A, C\} \in E$.

Part II

VARIATIONAL INFERENCE

INTRODUCTION

Statistical inference is the process of using data analysis to deduce properties of an underlying distribution.

The following elements are being considered: a random variable X with its corresponding set of observations $\mathbf{x} = (x_1, \dots, x_N)$ and a *hidden or latent variable* Z , which means that it is unobserved and the values $\mathbf{z} = (z_1, \dots, z_N)$ are unknown. The samples are governed by the joint distribution $P(\mathbf{x}, \mathbf{z})$. Where \mathbf{z} represents a vector with length N with a possible configuration of the unknown values, then, it is not fixed as \mathbf{x} .

In *classical inference* the conditional is calculated as

$$P(\mathbf{z} \mid \mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{z})}{\int_{\mathbf{z}} P(\mathbf{x}, \mathbf{z})},$$

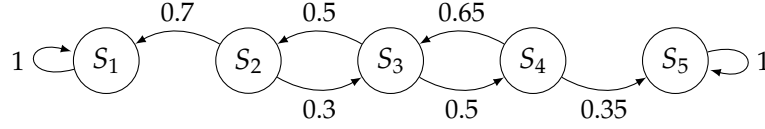
but for many models this integral is computationally hard to solve.

By comparison, *Bayesian inference* is a method of statistical inference in which Bayes' theorem is used, it derives the posterior probability $P(\mathbf{z} \mid \mathbf{x})$ as a consequence of two antecedents: a *prior probability* $P(\mathbf{z})$ and a *likelihood function* $P(\mathbf{x} \mid \mathbf{z})$ derived from a statistical model for the observed data.

Other methods as *Markov chain Monte Carlo (MCMC)* and *variational Bayesian inference* try a different approach when solving the given inference problem.

On one hand, *variational Bayesian inference* or *variational inference*, is a machine learning method whose main goal is to approximate probability distributions (Jordan *et al.* (1999), Wainwright & Jordan (2008)). On the other hand, *MCMC* approximates the posterior distribution using a Markov chain. Let us briefly introduce the main idea behind this method, we need to introduce two concepts: *Markov chain* and *MCMC*.

A *Markov Chain* is formally defined as a stochastic process, i.e, a family of random variables, that satisfies the *Markov property* also known as the *memoryless property*: *the conditional probability distribution of future states of the process (conditional on both present and past values) depends only on the present state*. To fully understand it, imagine a system with a number of possible states S_1, \dots, S_5 and the probabilities of going from one state to another as in the following diagram.



Consider a sequence of random variables X_t that symbolize the current state at the step t . The Markov property means that the probability of moving to the next state depends only on the present one, i.e.,

$$P(X_{n+1} = x \mid X_1 = x_1 \dots, X_n = x_n) = P(X_{n+1} = x \mid X_n = x_n).$$

We need two concepts to define the process of MCMC:

- **Ergodic Markov chain.** A Markov chain where it exists a number $N \in \mathbb{N}$ such that any state can be reached from any other state in any number of steps less or equal than N .
- **Stationary Distribution.** The probability distribution to which the process converges over time.

In MCMC, we consider an ergodic Markov chain over the latent variable \mathbf{Z} whose stationary distribution is the posterior $P(\mathbf{z} \mid \mathbf{x})$, samples are taken from the chain to approximate the posterior with them.

Instead, *variational inference* exchanges the inference problem with an optimization one. We fix a family of distributions \mathcal{Q} over the latent variable \mathbf{Z} and find the element that minimizes its Kullback-Leibler divergence with the posterior $P(\mathbf{z} \mid \mathbf{x})$.

$$Q^{opt} = \arg \min_{Q \in \mathcal{Q}} KL(Q(\mathbf{z}) \mid P(\mathbf{z} \mid \mathbf{x})).$$

Compared to *Markov Chain Monte Carlo (MCMC)*, variational inference tends to be faster and scale easier to large data (Blei *et al.* (2017)), it has been applied to different problems such as computer vision, computational neuroscience and document analysis.

We can further analyze the Kullback-Leibler divergence we want to minimize, it may be decomposed the following way:

$$\begin{aligned} KL(Q(\mathbf{z}) \mid P(\mathbf{z} \mid \mathbf{x})) &= \mathbb{E}_{Q(\mathbf{z})} [\log Q(\mathbf{z})] - \mathbb{E}_{Q(\mathbf{z})} [\log P(\mathbf{z} \mid \mathbf{x})] \\ &= \mathbb{E}_{Q(\mathbf{z})} [\log Q(\mathbf{z})] - \mathbb{E}_{Q(\mathbf{z})} [\log P(\mathbf{x}, \mathbf{z})] + \log P(\mathbf{x}). \end{aligned}$$

Although the Kullback-Leibler divergence cannot be computed as long as $P(\mathbf{z} \mid \mathbf{x})$ is unknown, we can optimize an equivalent objective: we can use its positiveness to set the following lower bound to the evidence, defined as *evidence lower bound* or *ELBO*.

$$\log P(\mathbf{x}) \geq \underbrace{-\mathbb{E}_{Q(\mathbf{z})} [\log Q(\mathbf{z})]}_{\text{Entropy}} + \underbrace{\mathbb{E}_{Q(\mathbf{z})} [\log P(\mathbf{x}, \mathbf{z})]}_{\text{Energy}} = \text{ELBO}(Q).^1$$

¹ Energy and Entropy terms come from a statistical physics terminology

Minimizing the Kullback-Leibler divergence is equivalent to maximize the ELBO. We can rewrite this bound as

$$\begin{aligned}\text{ELBO}(Q) &= \mathbb{E}_{Q(z)} [\log P(z)] + \mathbb{E}_{Q(z)} [\log P(\mathbf{x} | z)] - \mathbb{E}_{Q(z)} [\log Q(z)] \\ &= \mathbb{E}_{Q(z)} [\log P(\mathbf{x} | z)] - \text{KL}(Q(z) | P(z)),\end{aligned}$$

where it is expressed as the sum of the log likelihood of the observations and the Kullback-Leibler divergence between the prior $P(z)$ and $Q(z)$.

The *Expectation Maximization algorithm* is designed to optimize this lower bound in order to solve the optimization problem we are focusing. We will go deeper into this algorithm in Section 14.1.

MEAN-FIELD VARIATIONAL INFERENCE

5.1 THE MEAN-FIELD VARIATIONAL FAMILY

The *mean-field variational family* \mathcal{Q} is defined as the family of distributions where the variables are mutually independent, i.e, any $Q \in \mathcal{Q}$ verifies

$$Q(\mathbf{z}) = \prod_{n=1}^N Q_n(z_n),$$

where $\mathbf{z} = \{z_1, \dots, z_N\}$ is the considered set of variables. The mean-field family is commonly used to model the family of distributions over the latent variables in our optimization problem. Notice that each *factor* Q_n can be different and this family does not depend on the observed data.

The mean-field family can capture any marginal of the latent variables but not correlation between them, as it assumes they are independent. For example, consider a two dimensional Gaussian distribution where a high percentage of the density is inside the blue ellipse shown in the following figure. Any mean-field approximation would factorize as a product of two Gaussian distributions, condensing its density in a circle as shown in purple.

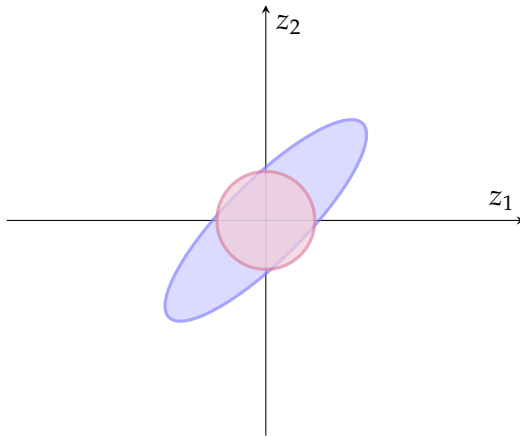


Figure 4: Mean-field family distribution (purple) approximating a Gaussian distribution (blue)

Notice the parametric form of each factor Q_n is not specified and the appropriate configuration depends on the variable. For example, a continuous variable might have a Gaussian factor and a categorical variable have a categorical factor.

Algorithm 1: Coordinate Ascent Variational Inference**Data:** A distribution $P(\mathcal{V}, \mathbf{z})$ with a dataset \mathcal{V} **Result:** A distribution of the mean-field family $Q(\mathbf{z}) = \prod_{n=1}^N Q_n(z_n)$ Initialize $Q(\mathbf{z})$;**while** *Convergence stop criteria* **do** **for** $n \in 1, \dots, N$ **do** $\mathbf{z}_{\setminus n} = (z_1, \dots, z_{n-1}, z_{n+1}, \dots, z_M)$; Set $Q_n(z_n) \propto \exp \mathbb{E}_{Q_{\setminus n}} [\log P(z_n | \mathbf{z}_{\setminus n}, \mathcal{V})]$; **end** Compute $ELBO(Q)$;

// Used for convergence criteria.

end**return** Q ;

5.2 CAVI ALGORITHM

In this section, we describe a widely used algorithm to solve the optimization problem we discussed in the previous section using the mean-field family. It is *coordinate ascent variational inference* or *CAVI* and its procedure is to iteratively optimize each factor of the mean-field family distribution, while fixing the others. With this, the ELBO reaches a local optimum.

Consider the set $\mathbf{z} = \{z_1, \dots, z_N\}$ comes from a set of i.i.d variables $\mathbf{Z} = \{Z_1, \dots, Z_N\}$, and fix the n^{th} variable Z_n , let $\setminus n$ denote the full set of indexes without the n^{th} , then $\mathbf{Z}_{\setminus n}$ is the full set of variables without the focused one. Let the factors $Q_m, m \neq n$ be fixed. Under this assumptions, the ELBO has the form

$$\begin{aligned}
 ELBO(Q) &= \mathbb{E}_Q [\log P(\mathbf{x}, \mathbf{z})] - \mathbb{E}_Q [\log Q(\mathbf{z})] \\
 &\stackrel{1}{=} \mathbb{E}_{Q_n} [\mathbb{E}_{Q_{\setminus n}} [\log P(\mathbf{x}, \mathbf{z})]] - \mathbb{E}_{Q_n} [\log Q_n(z_n)] + \text{const.} \\
 &\stackrel{2}{=} \mathbb{E}_{Q_n} [\mathbb{E}_{Q_{\setminus n}} [\log P(z_n | \mathbf{z}_{\setminus n}, \mathbf{x}) + \log P(\mathbf{z}_{\setminus n}, \mathbf{x})]] - \mathbb{E}_{Q_n} [\log Q_n(z_n)] + \text{const.} \\
 &\stackrel{3}{=} \mathbb{E}_{Q_n} [\mathbb{E}_{Q_{\setminus n}} [\log P(z_n | \mathbf{z}_{\setminus n}, \mathbf{x})]] - \mathbb{E}_{Q_n} [\log Q_n(z_n)] + \text{const.} \\
 &\stackrel{4}{=} -KL(Q_n(z_n) | \exp \mathbb{E}_{Q_{\setminus n}} [\log P(z_n | \mathbf{z}_{\setminus n}, \mathbf{x})]) + \text{const.}
 \end{aligned}$$

1. The expectations in the ELBO formula are separated. The logarithm factorizes as $\log Q(\mathbf{z}) = \sum_{n=1}^N \log Q_n(z_n)$. The constant term comes from $\mathbb{E}_{Q_{\setminus n}} [\log Q_{\setminus n}(\mathbf{z}_{\setminus n})]$.
2. P is separated as $P(\mathbf{z}, \mathbf{x}) = P(z_n | \mathbf{z}_{\setminus n}, \mathbf{x})P(\mathbf{z}_{\setminus n}, \mathbf{x}) \implies \log P(\mathbf{z}, \mathbf{x}) = \log P(z_n | \mathbf{z}_{\setminus n}, \mathbf{x}) + \log P(\mathbf{z}_{\setminus n}, \mathbf{x})$.
3. $\mathbb{E}_{Q_n} [\mathbb{E}_{Q_{\setminus n}} [\log P(\mathbf{z}_{\setminus n}, \mathbf{x})]] = \mathbb{E}_{Q_{\setminus n}} [\log P(\mathbf{z}_{\setminus n}, \mathbf{x})]$ is constant.
4. Applied Kullback-Leibler definition.

Minimizing the ELBO is equivalent to maximize the given Kullback-Leibler divergence, which corresponds to updating Q_n^{new} as:

$$Q_n^{new}(z_n) \propto \exp \mathbb{E}_{Q_{\setminus n}} [\log P(z_n | \mathbf{z}_{\setminus n}, \mathcal{V})].$$

Notice that the proportionality restriction is enough to fully determine the distribution as its integral is normalized. Equivalently, the distribution is proportional to

$$Q_n^{new}(z_n) \propto \exp \mathbb{E}_{Q_{\setminus n}} \left[\log P(z_n, \mathbf{z}_{\setminus n}, \mathcal{V}) \right].$$

As the ELBO is generally a non-convex function and the CAVI algorithm converges to a local optimum, the initialization values of the algorithm play an important role on its performance. The convergence criteria is usually a threshold for the ELBO.

EXPONENTIAL FAMILY

The exponential family is a parametric set of probability distributions of a certain form. This form is chosen based on some useful algebraic properties and generality, table 5 shows some examples of common distributions in the exponential family.

Let X be a random variable and θ a set of parameters. A family of distributions is said to belong to the exponential family if its probability distribution has the form

$$P(x | \theta) = h(x) \exp \left(\sum_{i=1}^S \eta_i(\theta) T_i(x) - \psi(\theta) \right),$$

where $h(x)$, $T_i(x)$, $\eta_i(\theta)$ and $\psi(\theta)$ are known functions such that h is called a *base measure*, $\eta_i(\theta)$ are called the *distribution parameters*, $T_i(x)$ the *test statistics* and ψ is the *log normalizer* as it ensures logarithmic normalization due to

$$\begin{aligned} 1 &= \int_x h(x) \exp \left(\sum_{i=1}^S \eta_i(\theta) T_i(x) - \psi(\theta) \right) \\ &= \int_x e^{-\psi(\theta)} h(x) \exp \left(\sum_{i=1}^S \eta_i(\theta) T_i(x) \right) \\ &= e^{-\psi(\theta)} \int_x h(x) \exp \left(\sum_{i=1}^S \eta_i(\theta) T_i(x) \right), \end{aligned}$$

so ψ verifies

$$\psi(\theta) = \log \int_x h(x) \exp \left(\sum_{i=1}^S \eta_i(\theta) T_i(x) \right).$$

Naming η and T the corresponding vector functions, the parameters can always be transformed as $\theta^{new} = \eta(\theta)$, in which case we say the distribution into its *canonical form* (notice ψ has changed but we are not differentiating it from the previous one since it is fully determined by the other functions):

$$P(x | \theta) = h(x) e^{\theta^T T(x) - \psi(\theta)}.$$

In this form, it is easier to see that T is the sufficient statistic for θ . This is a consequence of *Fisher–Neyman factorization theorem* which says that T is sufficient for θ if and only if the probability distribution P can be factored into a product such that one factor, h , does not depend on θ and the other factor, which does depend on θ , depends on x only through T .

Distribution	Parameter set θ	Base measure h	Parameter function η	Statistics T
Bernoulli	p	1	$\log \frac{p}{1-p}$	x
Binomial	p	$\binom{n}{x}$	$\log \frac{p}{1-p}$	x
Categorical	p_1, \dots, p_K	1	$\begin{pmatrix} \log p_1 \\ \vdots \\ \log p_K \end{pmatrix}$	$\begin{pmatrix} [x = 1] \\ \vdots \\ [x = K] \end{pmatrix}$
Gaussian	μ, σ^2	$\frac{1}{\sqrt{2\pi}}$	$\begin{pmatrix} \frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2} \end{pmatrix}$	$\begin{pmatrix} x \\ x^2 \end{pmatrix}$
Beta	α, β	$\frac{1}{x(1-x)}$	$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\begin{pmatrix} \log x \\ \log(1-x) \end{pmatrix}$
Dirichlet	$\alpha_1, \dots, \alpha_K$	1	$\begin{pmatrix} \alpha_1 - 1 \\ \vdots \\ \alpha_K - 1 \end{pmatrix}$	$\begin{pmatrix} \log x_1 \\ \vdots \\ \log x_K \end{pmatrix}$

Figure 5: Some distributions in the exponential family

An important property of the exponential family is that they have *conjugate priors*, when the posterior distribution is in the same probability distribution family as the prior distribution, they are called *conjugate distributions*, and the prior is called a *conjugate prior* of the likelihood distribution.

6.1 LATENT VARIABLE AND CONDITIONALLY CONJUGATE MODELS

One important case of exponential family are *latent variable models* or *LVMs*. In this models, the following assumptions are made:

1. There set of i.i.d random variables $\{X_1, \dots, X_N\}$ and a set of observations $\mathbf{x} = (x_1, \dots, x_N)$.
2. Both global θ and local parameters $\mathbf{Z} = \{Z_1, \dots, Z_n\}$ govern the data (both seen as latent variables). We refer to *global latent variables* when they affect the whole distribution and *local latent variables* when they affect only to a subset, in this case each Z_n affects only x_n . Given this, the joint probability is

$$P(\mathbf{x}, \theta, \mathbf{z}) = P(\theta) \prod_{n=1}^N P(z_n, x_n \mid \theta).$$

3. The n^{th} observation x_n and the n^{th} local hidden variable z_n are conditionally independent, given the global variables θ , of all other observations and local hidden variables,

$$P(x_n, z_n \mid \theta, \mathbf{x}_{\setminus n}, \mathbf{z}_{\setminus n}) = P(x_n, z_n \mid \theta).$$

Remark 1. These models are widely used to discover patterns in data sets (Blei (2014)). LVMs include popular models like *Latent Dirichlet Allocation* models used to uncover the hidden topics in text corpora (Blei *et al.* (2003)), mixture of Gaussian models to discover hidden clusters in data (Bishop (2006)) and probabilistic principal component analysis for dimensionality reduction (Tipping & Bishop (1999)).

One important case of LVMs and exponential family models are *conditionally conjugate models*, where the following assumptions are made.

1. The prior for the global latent variable $P(\theta)$ is in the exponential family with an hyper-parameter $\alpha = [\alpha_1, \alpha_2]$, where α_1 is a vector and α_2 is a scalar, and statistics that concatenate the global latent variable and its log normalizer $[\theta, -\psi(\theta)]$,

$$P(\theta) = h(\theta) \exp \left(\alpha^T [\theta, -\psi(\theta)] - \psi(\alpha) \right).$$

2. Each local term $P(z_n, x_n | \theta)$ is in the exponential family of the form

$$P(z_n, x_n | \theta) = h(z_n, x_n) \exp \left(\theta^T T(z_n, x_n) - \psi(\theta) \right).$$

3. The complete conditional of a local latent variable verifies

$$P(z_n | \theta, x, z_{\setminus n}) = P(z_n | x_n, \theta)$$

and is also in the exponential family

$$P(z_n | x_n, \theta) = h(z_n) \exp \left(\eta(\theta, x_n)^T T(z_n) - \psi(\theta, x_n) \right).$$

As conjugacy holds, the posterior $P(\theta | \mathcal{V}, z)$ is in the same family with parameter

$$\bar{\alpha} = [\alpha_1 + \sum_{n=1}^N T(z_n, x_n), \alpha_2 + N]^T.$$

This can be demonstrated as follows,

$$\begin{aligned} P(\theta | x, z) &= \frac{P(x, z | \theta) P(\theta)}{P(x, z)} \propto P(x, z | \theta) P(\theta) = P(\theta) \prod_{n=1}^N h(x_n, z_n) \exp \left(\theta^T T(z_n, x_n) - \psi(\theta) \right) \\ &\propto h(\theta) \exp \left(\alpha^T [\theta, -\psi(\theta)] \right) \prod_{n=1}^N h(x_n, z_n) \exp \left(\theta^T T(z_n, x_n) - \psi(\theta) \right) \\ &\propto h(\theta) \exp \left(\alpha^T [\theta, -\psi(\theta)] + \sum_{n=1}^N \theta^T T(z_n, x_n) - \psi(\theta) \right) \\ &\propto h(\theta) \exp \left(\alpha_1^T \theta - \alpha_2^T \psi(\theta) - N \psi(\theta) + \sum_{n=1}^N T(z_n, x_n)^T \theta \right) \\ &\propto h(\theta) \exp \left(\left(\alpha_1 + \sum_{n=1}^N T(z_n, x_n) \right)^T \theta - (\alpha_2 + N)^T \psi(\theta) \right). \end{aligned}$$

As the inverse reasoning can be done, if the complete conditional of the global hidden variable is in the exponential family with hyper-parameter $\bar{\alpha}$, the prior is in the same family with the corresponding hyper-parameter α .

6.2 CAVI IN CONDITIONALLY CONJUGATE MODELS

Set aside the conditionally conjugate models and consider the following situation where we fit a distribution $Q(z) = \prod_{n=1}^N Q_n(z_n)$ in the mean-field family, using an exponential family distribution for the marginal $P(z_n | z_{\setminus n}, \mathbf{x})$:

$$P(z_n | z_{\setminus n}, \mathbf{x}) = h(z_n) \exp \left(\eta_n(z_{\setminus n}, \mathbf{x})^T T(z_n) - \psi(z_{\setminus n}, \mathbf{x}) \right).$$

The update of the CAVI algorithm is then given by

$$\begin{aligned} Q(z_n) &\propto \exp \mathbb{E}_{Q(z_{\setminus n})} \left[\log P(z_n | z_{\setminus n}, \mathbf{x}) \right] \\ &= h(z_n) \exp \left(\mathbb{E}_{Q(z_{\setminus n})} \left[\eta_n(z_{\setminus n}, \mathbf{x}) \right]^T T(z_n) - \mathbb{E}_{Q(z_{\setminus n})} \left[\psi(z_{\setminus n}, \mathbf{x}) \right] \right) \\ &\propto h(z_n) \exp \left(\mathbb{E}_{Q(z_{\setminus n})} \left[\eta_n(z_{\setminus n}, \mathbf{x}) \right]^T T(z_n) \right) = h(z_n) e^{v_n^T T(z_n)}, \end{aligned} \quad (1)$$

where

$$v_n = \mathbb{E}_{Q(z_{\setminus n})} \left[\eta_n(z_{\setminus n}, \mathbf{x}) \right].$$

Summarizing, the factor is in the exponential family with the same base measure h and updating it is equivalent to setting the distribution parameter η to the expected one of the complete conditional $\mathbb{E} \left[\eta_n(\mathbf{x}, z_{\setminus n}) \right]$. This expression facilitates deriving CAVI algorithm for many complex models.

Going back to the conditionally conjugate models, our main goal is to set both the local and global hidden variable update. Our variational distribution is in the mean-field family with $Q(\boldsymbol{\theta} | \lambda)$ where λ is called the *global variational parameter*, and for each local variable, the distribution is $Q(z_n | \gamma_n)$, where γ_n is called a *local variational parameter*:

$$\begin{aligned} Q(z_n | \gamma_n) &\propto h(z_n) e^{\gamma_n^T T(z_n)}, \\ Q(\boldsymbol{\theta} | \lambda) &= h(\boldsymbol{\theta}) \exp \left(\lambda^T [\boldsymbol{\theta}, -\psi(\boldsymbol{\theta})] - \psi(\lambda) \right). \end{aligned}$$

CAVI iteratively updates each local variational parameter and the global variational parameter.

Following the steps done in 1, the local variational parameter update is

$$\gamma_n^{new} = \mathbb{E}_{Q(\boldsymbol{\theta} | \lambda)} \left[\eta(\boldsymbol{\theta}, x_n) \right],$$

In this case, the local hidden variable conditional does not depend on the other local hidden variables, neither other data-points.

The global variational parameter update is calculated as

$$\begin{aligned} Q^{new}(\boldsymbol{\theta} | \lambda) &\propto \exp \mathbb{E}_{Q(z)} \left[\log P(\boldsymbol{\theta} | \mathbf{z}, \mathbf{x}) \right] \propto h(\boldsymbol{\theta}) \exp \mathbb{E}_{Q(z)} \left[\left[\lambda_1 + \sum_{n=1}^N T(x_n, z_n), \lambda_2 + N \right]^T [\boldsymbol{\theta}, \psi(\boldsymbol{\theta})] \right] \\ &\propto h(\boldsymbol{\theta}) \exp \left(\left[\lambda_1 + \sum_{n=1}^N \mathbb{E}_{Q(z)} \left[T(x_n, z_n) \right], \lambda_2 + N \right]^T [\boldsymbol{\theta}, -\psi(\boldsymbol{\theta})] \right) \\ &= Q(\boldsymbol{\theta}, \lambda^{new}). \end{aligned}$$

Then, the variational parameter updated is

$$\lambda^{new} = \left[\lambda_1 + \sum_{n=1}^N \mathbb{E}_{Q(z_n|\gamma_n)} [T(x_n, z_n)], \lambda_2 + N \right].$$

We can compute the ELBO at each iteration up to a constant that does not depend on the variational parameters,

$$\begin{aligned} ELBO &= \left(\lambda_1 + \sum_{n=1}^N \mathbb{E}_{Q(z_n|\gamma_n)} [T(x_n, z_n)] \right)^T \mathbb{E}_{Q(\theta|\lambda)} [\theta] - (\lambda_2 + N) \mathbb{E}_{Q(\theta|\lambda)} [\psi(\theta)] \\ &\quad + \lambda^T \mathbb{E}_{Q(\theta, \lambda)} [T(\theta)] - \psi(\lambda) + \sum_{n=1}^N \gamma_n^T \mathbb{E}_{Q(z_n, \gamma_n)} [z_n] - \psi(\gamma_n) + \text{const.} \end{aligned}$$

The calculations are the following:

$$\begin{aligned} ELBO(Q(\theta, z)) &= \mathbb{E}_{Q(\theta, z)} [\log P(\theta, z, x)] - \mathbb{E}_{Q(\theta, z)} [\log Q(\theta, z)] \\ &= \mathbb{E}_{Q(z|\gamma)} [\mathbb{E}_{Q(\theta|\lambda)} [\log P(\theta, z, x)]] - \mathbb{E}_{Q(\theta, z)} [\log Q(\theta, z)] \\ &= \mathbb{E}_{Q(\theta|\lambda)} [\log P(\theta)] + \mathbb{E}_{Q(z|\gamma)} [\mathbb{E}_{Q(\theta|\lambda)} [\sum_{n=1}^N \log P(z_n, x_n | \theta)]] \\ &\quad - \mathbb{E}_{Q(\theta, z)} [\log Q(\theta, z)] \end{aligned}$$

The first term is

$$\begin{aligned} \mathbb{E}_{Q(\theta|\lambda)} [\log P(\theta)] &= \mathbb{E}_{Q(\theta|\lambda)} [\lambda_1 \theta - \lambda_2 \psi(\theta) - \psi(\lambda)] \\ &= \lambda_1 \mathbb{E}_{Q(\theta|\lambda)} [\theta] - \lambda_2 \mathbb{E}_{Q(\theta|\lambda)} [\psi(\theta)] - \psi(\lambda). \end{aligned}$$

The middle term is

$$\begin{aligned} \mathbb{E}_{Q(z|\gamma)} [\mathbb{E}_{Q(\theta|\lambda)} [\log P(\theta, z, x)]] &= \mathbb{E}_{Q(z|\gamma)} [\mathbb{E}_{Q(\theta|\lambda)} [\log P(\theta) + \sum_{n=1}^N \log P(z_n, x_n | \theta)]] \\ &= \mathbb{E}_{Q(z|\gamma)} [\mathbb{E}_{Q(\theta|\lambda)} [\log P(\theta)]] + \mathbb{E}_{Q(z|\gamma)} [\mathbb{E}_{Q(\theta|\lambda)} [\sum_{n=1}^N \log P(z_n, x_n | \theta)]] \\ &= \left(\lambda_1 + \sum_{n=1}^N \mathbb{E}_{Q(z_n|\gamma_n)} [T(x_n, z_n)] \right)^T \mathbb{E}_{Q(\theta|\lambda)} [\theta] - (\lambda_2 + N) \mathbb{E}_{Q(\theta|\lambda)} [\psi(\theta)] \end{aligned}$$

The last term is

$$\begin{aligned} \mathbb{E}_{Q(\theta, z)} [\log Q(\theta, z)] &= \mathbb{E}_{Q(\theta)} [\log Q(\theta)] + \mathbb{E}_{Q(z)} [\sum_{n=1}^N \log Q(z_n)] \\ &= \mathbb{E}_{Q(\theta)} [\lambda^T T(\theta) - \psi(\lambda)] + \sum_{n=1}^N \mathbb{E}_{Q(z_n)} [\gamma_n^T z_n - \psi(\gamma_n)] \\ &= \lambda^T \mathbb{E}_{Q(\theta, \lambda)} [T(\theta)] - \psi(\lambda) + \sum_{n=1}^N \gamma_n^T \mathbb{E}_{Q(z_n, \gamma_n)} [z_n] - \psi(\gamma_n). \end{aligned}$$

Part III

GRAPHICAL MODELS

A *graphical model* is a statistical model for which a graph expresses the conditional dependence structure between random variables.

Commonly, they provide a graph-based representation for encoding a multi-dimensional distribution representing a set of independences that hold in the specific distribution. The most commonly used are *Bayesian networks* and *Markov random fields*, which differ in the set of independences they can encode and the factorization of the distribution that they include.

BAYESIAN NETWORKS

Consider we have N variables X_1, \dots, X_N with the corresponding distribution $P(x_1, \dots, x_N)$. Let \mathcal{E} be a set of indexes such as $\text{evidence} = \{X_e = x_e \mid e \in \mathcal{E}\}$. Inference could be made by brute force:

$$P(X_i = x_i \mid \text{evidence}) = \frac{\int_{j \notin \mathcal{E}, j \neq i} P(\text{evidence}, x_j, X_i = x_i)}{\int_{j \notin \mathcal{E}} P(\text{evidence}, x_j)}.$$

Let us suppose all these variables are binary, this calculation will require $O(2^{N-\#\mathcal{E}})$ operations. Also, all entries of a table $P(x_1, \dots, x_N)$ take $O(2^N)$ space.

This is unpractical when taking into account millions of variables. The underlying idea of bayesian networks is to specify which variables are independent of others, factorizing the joint probability distribution.

Definition 35. Let $G = (V, E)$ be a graph where $V = \{X_1, \dots, X_n\}$ is a set of random variables. We say that the joint probability $P(x_1, \dots, x_n)$ *factorizes* according to G if and only if

$$P(x_1, \dots, x_N) = \prod_{i=1}^N P(x_i \mid \text{pa}(x_i)).$$

Definition 36. A *belief network* or *Bayesian network* is a pair (G, P) where P factorizes over G . It is a probabilistic graphical model that represents conditional dependencies of a set of variables X_1, \dots, X_n .

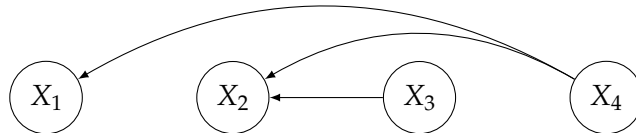


Figure 6: Bayesian Network factorizing $P(x_1, x_2, x_3, x_4) = P(x_1 \mid x_4)P(x_2 \mid x_3, x_4)P(x_3)P(x_4)$

Given this definition, the representation of any Bayesian network is a directed acyclic graph.

Any probability distribution can be written as a Bayesian Network, even though it may end up been a fully-connected "cascase" DAG, which means that each variable X_n is a parent of any X_m with $m > n$. Using four variables:

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2 | x_1)P(x_3 | x_2, x_1)P(x_4 | x_3, x_2, x_1).$$

To set the specification of the Belief network, we need to define all elements of the probability tables $P(x_i | pa(x_i))$. When the number of variables is large, this is still intractable so the tables are generally parameterized in a low dimensional manner.

Bayesian Networks are good for encoding conditional independence over the variables, but are not for encoding dependence. For example, with the following network

$$P(x, y) = P(y | x)P(x).$$

represented as $x \rightarrow y$ in a DAG (a Bayesian network can be defined giving the graph or the joint probability). It may appear to encode dependence between both variables but the conditional $P(y | x)$ could happen to equal $P(y)$, giving $P(x, y) = P(x)P(y)$.

How could we check if two variables are conditionally independent given a Bayesian network? For example in figure 3, $X_1 \perp\!\!\!\perp X_2 | X_4$ as¹:

$$\begin{aligned} P(x_2|x_4) &= \frac{1}{P(x_4)} \int_{x_1, x_3} P(x_1, x_2, x_3, x_4) = \frac{1}{P(x_4)} \int_{x_1, x_3} P(x_1|x_4)P(x_2|x_3, x_4)P(x_3)P(x_4) \\ &= \int_{x_3} P(x_2|x_3, x_4)P(x_3), \\ P(x_1, x_2|x_4) &= \frac{1}{P(x_4)} \int_{x_3} P(x_1, x_2, x_3, x_4) = \frac{1}{P(x_4)} \int_{x_3} P(x_1|x_4)P(x_2|x_3, x_4)P(x_3)P(x_4) \\ &= P(x_1|x_4) \int_{x_3} P(x_2|x_3, x_4)P(x_3) = P(x_1|x_4)P(x_2|x_4). \end{aligned}$$

Now we are going to define two central concepts to determine conditional independence in any Bayesian network, these are *d-connection* and *d-separation*.

Definition 37. Let G be a DAG where X, Y and Z are disjoint sets of vertices. We say that X and Y are *d-connected* by Z if and only if there exists an undirected path U from any vertex in X to any vertex in Y such that:

- For any collider C , itself or any it's descendants is in Z
- No non-collider on U is on Z

Definition 38. Let G be a DAG where X, Y and Z are disjoint sets of vertices. X and Y are *d-separated* by Z if and only if they are not d-connected by Z in G

For example, in figure 7 d d-separates a and c (e is a collider in the path that is not in $\{d\}$), and $\{d, e\}$ d-connect them.

¹ Continuous variable notation is used

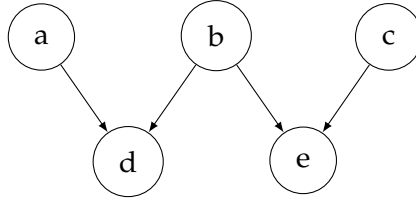


Figure 7: D-separation example

Theorem 2 (Pearl & Dechter (2013)). Let G be a DAG where X, Y and Z are disjoint sets of vertices. If X and Y are d-separated by Z , then they are independent conditional on Z in all probability distributions that factorize over G .

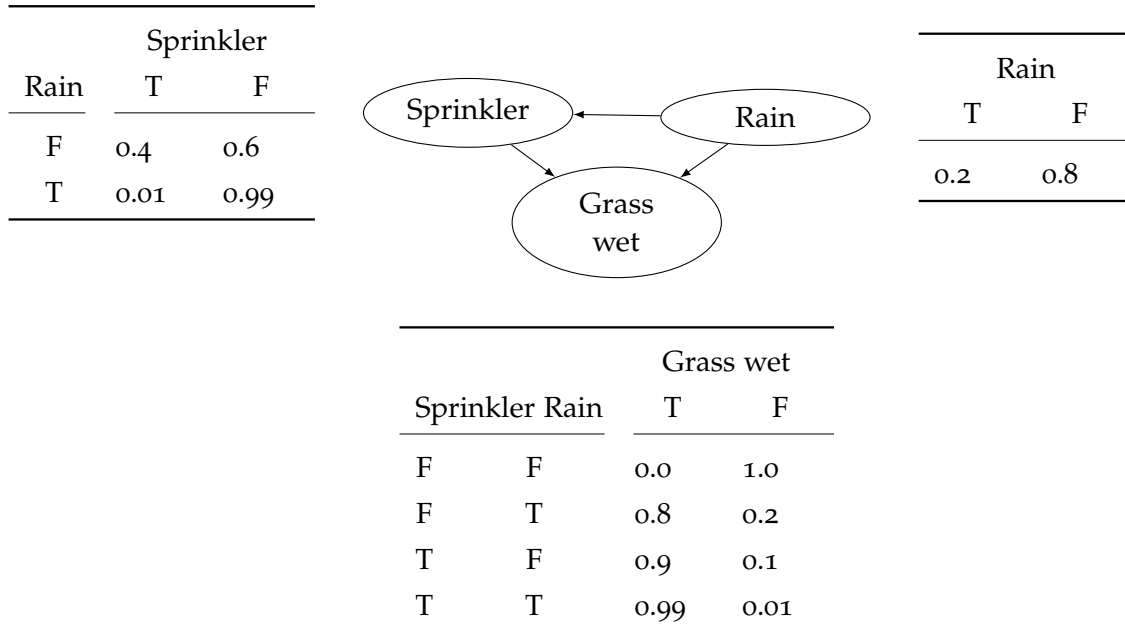
The Bayes Ball algorithm (Shachter (2013)) provides a linear time complexity algorithm that computes conditional independent using this theorem.

Example 2. In this example we are modeling three discrete random variables: Sprinkler (S), Rain (R) and Grass wet (G).

The joint probability function is:

$$P(s, r, g) = P(s|r)P(g|s, r)P(r)$$

The following DAG illustrates the Bayesian network along with the probability tables we are using.



This model can answer questions about the presence of a cause given the presence of an effect. For example, What is the probability that it has been raining given the grass is wet?

$$P(R = T | G = T) = \frac{P(G = T, R = T)}{P(G = T)} = \frac{\sum_s P(G = T, R = T, s)}{\sum_{r,s} P(G = T, r, s)}$$

Using the expression of the joint probability among with the tables we can compute every term. For example:

$$\begin{aligned} P(G = T, R = T, S = T) &= P(S = T | R = T) P(G = T | R = T, S = T) P(R = T) \\ &= 0.01 * 0.99 * 0.2 = 0.00198 \end{aligned}$$

In some situations our Bayesian networks will contain a number of nodes that are essentially the same but repeated a number of times, in this cases we are using the *plate notation*. Suppose we have the situation that figure 8 shows on the left. Then we can collapse all B_i variables in a box, indicating there number of variables inside it.

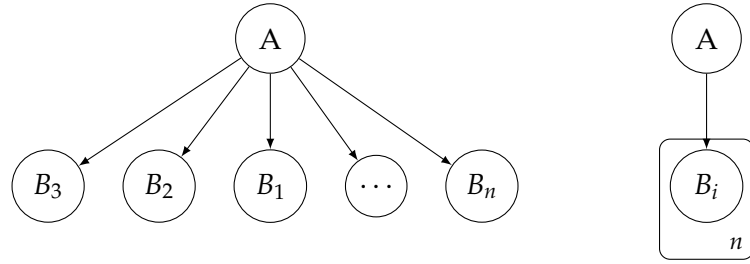


Figure 8: Plate notation example. Standard notation on the left and plate on the right

MARKOV RANDOM FIELDS

Definition 39. A *potential* ϕ is a non-negative function. It is worth to mention that a probability distribution is a special case of a potential.

Definition 40. Let \mathbf{X} be a set of random variables, G an undirected graph, $\mathbf{X}_c, c \in \{1, \dots, C\}$ be the maximal cliques of G and P a probability distribution over \mathbf{X} . The pair (G, P) is said to be a *Markov network* or *Markov random field* if, and only if

$$P(x_1, \dots, x_n) = \frac{1}{Z} \prod_{c=1}^C \phi_c(\mathbf{X}_c)$$

where Z is a constant that ensures normalization.

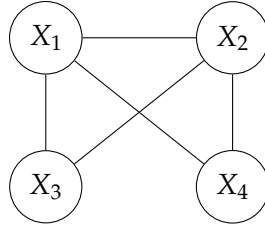


Figure 9: Markov Network $P(x_1, x_2, x_3, x_4) = \phi(x_1, x_2, x_3)\phi(x_2, x_3, x_4)/Z$

In figure 9 we can see an example of the factorization, without giving any reference of the potentials.

Let (G, P) be a Markov network, then it satisfies the following properties known as Markov properties:

- Pairwise Markov property. Any two non-adjacent variables are conditionally independent given all other variables.
- Local Markov property. A variable is conditionally independent over all other variables given it's neighbors. That is,

$$P(x_i \mid x_{\setminus i}) = p(x_i \mid ne(x_i))$$

- Global Markov property. Any two subsets of variables are conditionally independent given a separating subset (any path from one set to the other passes through this one).

Remark 2. A Markov network can also be defined as a pair (G, P) such as all Markov properties are satisfied. The clique factorization definition is a special case of these properties.

Definition 41. Let G be an undirected graph and P a probability distribution over a set of random variables \mathbf{X} . The pair (G, P) is called a Markov Random Field if and only if it follows the local Markov Property.

Part IV

BAYESIAN NETWORKS LEARNING

INTRODUCTION

In this chapter we are focusing on Bayesian Learning training, both Bayesian and Variational inference are considered. Firstly, the network structure is assumed and no hidden variables are considered. As all variables are visible, we are denoting \mathcal{D} the set of observations and θ the set of parameters of the distributions.

The goal now is to determine the posterior $P(\theta \mid \mathcal{D})$ using a prior belief $P(\theta)$ and the likelihood function $P(\mathcal{D} \mid \theta)$,

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})}.$$

We are using the following example to illustrate the situation, in it we will try to learn the bias of a coin, given a set of tossing results.

Example 3. Let $\mathcal{D} = \{d_n\}_{n \in 1, \dots, N}$ be the results of tossing a coin $N \in \mathbb{N}$ times, let 1 symbolize *heads* and 0 *tails*.

Our objective is to estimate the probability θ that the coin will be head $P(d_n = 1 \mid \theta)$, for this we have the i.i.d random variables d_1, \dots, d_n and θ , and we require a model $P(d_1, \dots, d_n, \theta)$. We have a Belief Network shown in figure 10

$$P(\mathcal{D}, \theta) = P(\theta) \prod_{n=1}^N P(d_n \mid \theta).$$

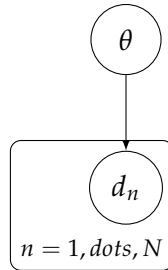


Figure 10: Belief network for coin tossing

We want to calculate

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})},$$

to do so, we need to specify the prior $P(\theta)$, we are using a discrete model where

$$P(\theta = 0.2) = 0.1, \quad P(\theta = 0.5) = 0.7 \quad \text{and} \quad P(\theta = 0.8) = 0.2.$$

This means that we have a 70% belief that the coin is fair, a 10% belief that is biased to tails and 20% that is biased to heads. The parameter models the probability of a datapoint as

$$P(d_n = 1 \mid \theta) = \theta \quad \text{and} \quad P(d_n = 0 \mid \theta) = 1 - \theta.$$

Let n_h be the number of heads in our observed data and n_t the number of tails

$$n_h = \#\{d = 1\} \quad \quad \quad n_t = \#\{d = 0\},$$

then the posterior has the form

$$P(\theta \mid \mathcal{D}) = \frac{P(\theta)}{P(\mathcal{D})} \theta^{n_h} (1 - \theta)^{n_t}.$$

Suppose that $n_h = 2$ and $n_t = 8$, then

$$\begin{aligned} P(\theta = 0.2 \mid \mathcal{D}) &= \frac{1}{P(\mathcal{D})} \times 0.1 \times 0.2^2 \times 0.8^8 = \frac{1}{P(\mathcal{D})} \times 6.71 \times 10^{-4}, \\ P(\theta = 0.5 \mid \mathcal{D}) &= \frac{1}{P(\mathcal{D})} \times 0.7 \times 0.5^2 \times 0.5^8 = \frac{1}{P(\mathcal{D})} \times 6.83 \times 10^{-4}, \\ P(\theta = 0.8 \mid \mathcal{D}) &= \frac{1}{P(\mathcal{D})} \times 0.2 \times 0.2^2 \times 0.8^8 = \frac{1}{P(\mathcal{D})} \times 3.27 \times 10^{-7}. \end{aligned}$$

Now, we can compute

$$\frac{1}{P(\mathcal{D})} = 6.71 \times 10^{-4} + 6.83 \times 10^{-4} + 3.27 \times 10^{-7} = 0.00135.$$

Therefore, the posterior is

$$\begin{aligned} P(\theta = 0.2 \mid \mathcal{D}) &= 0.4979, \\ P(\theta = 0.5 \mid \mathcal{D}) &= 0.5059, \\ P(\theta = 0.8 \mid \mathcal{D}) &= 0.00024. \end{aligned}$$

These are the posterior parameter beliefs of our experiment. Given this, if we were to choose a single value for the posterior it would be $\theta = 0.5$, as it is the one with the highest probability.

This result is intuitive, we had a strong belief of the coin being fair and even though the number of tails was quite bigger than heads, it was not enough to make the difference. Obviously the posterior of the coin being biased to tails is now bigger than the prior.

In this example, we have used a discrete prior for the parameter, a continuous prior may be chosen instead. Suppose a uniform prior distribution so that $P(\theta) = k \implies \int_0^1 P(\theta) d\theta = k = 1$ due to normalization.

Using the previous calculations we have

$$P(\theta \mid \mathcal{D}) = \frac{1}{P(\mathcal{D})} \theta^{n_h} (1 - \theta)^{n_t},$$

where

$$P(\mathcal{D}) = \int_0^1 \theta^{n_h} (1 - \theta)^{n_t} d\theta.$$

This implies that

$$P(\theta | \mathcal{D}) = \frac{\theta^{n_h} (1 - \theta)^{n_t}}{\int_0^1 u^{n_h} (1 - u)^{n_t} du} \implies \theta | \mathcal{D} \sim \text{Beta}(n_h + 1, n_t + 1).$$

A Beta distribution may be also considered as the prior distribution:

$$\theta \sim \text{Beta}(\alpha, \beta) \implies P(\theta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}.$$

then, repeating the same as before we get that

$$P(\theta, \mathcal{D}) = \frac{1}{B(\alpha + n_h, \beta + n_t)} \theta^{\alpha+n_h-1} (1 - \theta)^{\beta+n_t-1} \implies (\theta, \mathcal{D}) \sim \text{Beta}(\alpha + n_h, \beta + n_t)$$

Remark 3. We have shown that the Beta distribution (prior and posterior) is conjugate of the Binomial distribution (likelihood).

9.1 UTILITY

The Bayesian posterior says nothing about how to benefit from the beliefs it represents, in order to do this we need to specify the utility of each decision.

With this idea we define an utility function over the parameters

$$U(\theta, \theta_{true}) = \alpha \mathbb{I}[\theta = \theta_{true}] - \beta \mathbb{I}[\theta \neq \theta_{true}],$$

where $\alpha, \beta \in \mathbb{R}$. This symbolizes the gains or losses of choosing the parameter θ , when the true value of the parameter is supposed to be θ_{true} . Therefore, the expected utility of a parameter θ_0 is calculated as

$$U(\theta = \theta_0) = \sum_{\theta_{true}} U(\theta = \theta_0, \theta_{true}) P(\theta = \theta_{true} | \mathcal{D}).$$

We might as well define an utility function over the previous example:

$$U(\theta, \theta_{true}) = 10 \mathbb{I}[\theta = \theta_{true}] - 20 \mathbb{I}[\theta \neq \theta_{true}],$$

where we interpret that the loss of choosing the wrong parameter is twice as important as the gains from doing it right.

The expected utility of the decision that the parameter is $\theta = 0.2$ in our discrete example would be

$$\begin{aligned} U(\theta = 0.2) &= U(\theta = 0.2, \theta_{true} = 0.2) P(\theta_{true} = 0.2 | \mathcal{D}) \\ &\quad + U(\theta = 0.2, \theta_{true} = 0.5) P(\theta_{true} = 0.5 | \mathcal{D}) \\ &\quad + U(\theta = 0.2, \theta_{true} = 0.8) P(\theta_{true} = 0.8 | \mathcal{D}) \\ &= 10 \times 0.4979 - 20 \times 0.5059 - 20 \times 0.00024 \\ &= -5.1438, \\ U(\theta = 0.5) &= -4.9038, \\ U(\theta = 0.8) &= -20.0736. \end{aligned}$$

MAXIMUM LIKELIHOOD TRAINING

In this section we will introduce two concepts, Maximum Likelihood and Maximum a Posteriori, showing that *training* a model's parameter to maximize the Maximum Likelihood equals to take the empirical distribution.

Let $\{X_1, \dots, X_N\}$ be a set of real i.i.d random variables, $\mathcal{D} = \{x_1, \dots, x_N\}$ be the set of observations and θ the considered parameters of the model.

Definition 42. Maximum Likelihood refers to the value of the parameter θ for which the observed data better fits the model:

$$\theta^{ML} = \arg \max_{\theta} P(\mathcal{D} \mid \theta).$$

Definition 43. Maximum A Posteriori refers to the value of the parameter θ that better fits the data:

$$\theta^{MAP} = \arg \max_{\theta} P(\mathcal{D} \mid \theta)P(\theta) = \arg \max_{\theta} P(\theta \mid \mathcal{D}).$$

The decision of taking the Maximum A Posteriori can be motivated using an utility that equals zero for all but the correct parameter

$$U(\theta, \theta_{true}) = \mathbb{I}[\theta = \theta_{true}],$$

using this, the expected utility of a parameter $\theta = \theta_0$ is

$$U(\theta = \theta_0) = \sum_{\theta_{true}} \mathbb{I}[\theta_{true} = \theta_0]P(\theta = \theta_{true} \mid \mathcal{D}) = P(\theta_0 \mid \mathcal{D}).$$

This means that the maximum utility decision is to take the value θ_0 with the highest posterior value.

Remark 4. When using a flat prior($P(\theta)$ is constant), $\theta^{ML} = \theta^{MAP}$.

10.1 ML AND KL DIVERGENCE

Now, we are going to show the relation between the Maximum Likelihood and the Kullback-Leibler divergence of the empirical distribution and our model. Firstly, we define the empirical distribution as a distribution whose probability mass function Q is

$$Q(x) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}[x = x_n].$$

We may calculate the Kullback-Leibler divergence between the empirical and our considered model $P(x | \theta)$ and study their functional independence,

$$KL(Q | P) = \mathbb{E}_Q[\log(Q(x))] - \mathbb{E}_Q[\log(P(x | \theta))].$$

Notice the term $\mathbb{E}_Q[\log(Q(x))]$ is a constant and the log likelihood under Q takes the form

$$\mathbb{E}_Q[\log(P(x | \theta))] = \frac{1}{N} \int_x \sum_{n=1}^N \mathbb{I}[x = x_n] \log P(x | \theta) = \frac{1}{N} \sum_{n=1}^N \log P(x_n | \theta).$$

As the logarithm is a strictly increasing function, maximizing the log likelihood equals to maximize the likelihood itself, in conclusion, it is equivalent to minimize the Kullback-Leibler divergence between the empirical distribution Q and our distribution P .

$$\begin{aligned} \arg \min_{\theta} KL(Q | P) &= \arg \min_{\theta} -\mathbb{E}_Q[\log(P(x | \theta))] = \arg \max_{\theta} \mathbb{E}_Q[\log(P(x | \theta))] = \\ &= \arg \max_{\theta} \frac{1}{N} \sum_{n=1}^N \log P(x_n | \theta) = \arg \max_{\theta} \frac{1}{N} \sum_{n=1}^N P(x_n | \theta) = \arg \max_{\theta} \sum_{n=1}^N P(x_n | \theta) = \theta^{ML}. \end{aligned}$$

In case $P(x | \theta)$ is unconstrained, the optimal choice is $P(x | \theta) = Q(x)$, that is, the **maximum likelihood distribution corresponds to the empirical distribution**.

For a Belief Network we know there is the following constraint

$$P(x_1, \dots, x_N | \theta) = \prod_{n=1}^N P(x_n | pa(x_n), \theta).$$

In this case N variables are being considered so the empirical distribution counts the number of occurrences of a configuration of these variables.

We now aim to minimize the Kullback-Leibler divergence between the empirical distribution $Q(x_1, \dots, x_N)$ and $P(x_1, \dots, x_N | \theta)$ in order to get the Maximum Likelihood value:

$$\begin{aligned} KL(Q | P) &= -\mathbb{E}_Q\left[\sum_{n=1}^N \log P(x_n | pa(x_n), \theta)\right] + \mathbb{E}_Q\left[\sum_{n=1}^N \log Q(x_n | pa(x_n))\right] \\ &= -\sum_{n=1}^N \mathbb{E}_Q[\log P(x_n | pa(x_n), \theta)] + \sum_{n=1}^N \mathbb{E}_Q[\log Q(x_n | pa(x_n))]. \end{aligned}$$

We might use proposition 1 on $\log P(x_n | pa(x_n), \theta)$ and $Q(x_n, pa(x_n))$, resulting:

$$\begin{aligned} KL(Q | P) &= \sum_{n=1}^N \mathbb{E}_{Q(x_n, pa(x_n))}[\log Q(x_n | pa(x_n))] - \mathbb{E}_{Q(x_n, pa(x_n))}[\log P(x_n | pa(x_n), \theta)] \\ &= \sum_{n=1}^N \mathbb{E}_{Q(x_n, pa(x_n))}[KL(Q(x_n | pa(x_n)) | P(x_n | pa(x_n), \theta))]. \end{aligned}$$

The optimal setting is then

$$P(x_n | pa(x_n), \theta) = Q(x_n | pa(x_n)),$$

in terms of the initial data it is to set $P(x_n | pa(x_n))$ to the number of times the state appears in it.

BAYESIAN TRAINING

A Bayesian approach where we set a distribution over the parameters is an alternative to Maximum Likelihood training of a Bayesian Network, as we did in the coin tossing example 3.

Consider the following scenario where a disease D and two habits A and B are being studied. Consider the following i.i.d variables $\{A_1, \dots, A_N\}$, $\{B_1, \dots, B_N\}$ and $\{D_1, \dots, D_N\}$ governed by the parameters θ_A, θ_B and θ_D as shown in figure 11. Let $N = 7$ be the number of observations of the variables as shown in the table below and $\mathcal{D} = \{(a_n, b_n, d_n), n = 1, \dots, N\}$ the set of observations.

All the variables are binary satisfying

$$P(A_n = 1 \mid \theta_A) = \theta_A, \quad P(B_n = 1 \mid \theta_B) = \theta_B, \quad P(D_n = 1 \mid A_n = 0, B_n = 1, \theta_D) = \theta_1,$$

$$\theta_D = (\theta_0, \theta_1, \theta_2, \theta_3).$$

Where we are using a binary to decimal transformation between the states of A and B and the sub-index of θ .

Summarizing, we are considering a Bernoulli distribution on A_n, B_n and D_n conditioned on the others.

The graph gives the following joint probability distribution over A_n, B_n and D_n :

$$P(a_n, b_n, d_n, \theta_A, \theta_B, \theta_D) = P(d_n \mid a_n, b_n, \theta_D)P(a_n \mid \theta_A)P(b_n \mid \theta_A).$$

We need to specify a prior and since dealing with multidimensional continuous distributions is computationally problematic, it is usual to use univariate distributions.

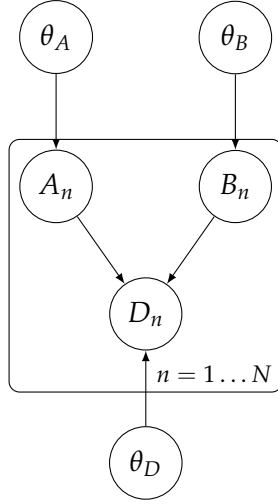
11.1 GLOBAL AND LOCAL PARAMETER INDEPENDENCE

A convenient assumption is that the prior factorizes, this is usually called *global parameter independence*:

$$P(\theta_A, \theta_B, \theta_D) = P(\theta_A)P(\theta_B)P(\theta_D).$$

Using this, the joint probability factorizes as

$$P(\mathcal{D}, \theta_A, \theta_B, \theta_D) = P(\theta_A)P(\theta_B)P(\theta_D) \prod_n P(a_n \mid \theta_A)P(b_n \mid \theta_B)P(d_n \mid a_n, b_n, \theta_D).$$



A	B	D
1	1	1
1	0	0
0	1	1
0	1	0
1	1	1
0	0	0
1	0	1

Figure 11: Bayesian parameter model for the relation between A, B, D

Table 1: Set of observations, where 1 means true and 0 means false.

Therefore, learning corresponds to making inference,

$$\begin{aligned}
 P(\theta_A, \theta_B, \theta_D \mid \mathcal{D}) &= \frac{P(\mathcal{D} \mid \theta_A, \theta_B, \theta_D) P(\theta_A, \theta_B, \theta_D)}{P(\mathcal{D})} = \frac{P(\mathcal{D} \mid \theta_A, \theta_B, \theta_D) P(\theta_A) P(\theta_B) P(\theta_D)}{P(\mathcal{D})} \\
 &= \frac{1}{P(\mathcal{D})} P(\theta_A) P(\theta_B) P(\theta_D) \prod_n P(a_n \mid \theta_A) P(b_n \mid \theta_B) P(d_n \mid a_n, b_n, \theta_D) \\
 &= P(\theta_A \mid \mathcal{D}_A) P(\theta_B \mid \mathcal{D}_B) P(\theta_D \mid \mathcal{D}).
 \end{aligned}$$

Where \mathcal{D}_A is the subset of observations related to the variable A . If we further assume that $P(\theta_D)$ factorizes as $P(\theta_D) = P(\theta_0)P(\theta_1)P(\theta_2)P(\theta_3)$, which is called *local parameter independence*, $P(\theta_D \mid \mathcal{D})$ factorizes as

$$P(\theta_D \mid \mathcal{D}) = P(\theta_0 \mid \mathcal{D}) P(\theta_1 \mid \mathcal{D}) P(\theta_2 \mid \mathcal{D}) P(\theta_3 \mid \mathcal{D}).$$

11.2 LEARNING BINARY VARIABLES

The simplest cases to continue are $P(\theta_A \mid \mathcal{D}_A)$ and $P(\theta_B \mid \mathcal{D}_B)$ since they require only a uni-variate prior distribution $P(\theta_A)$ or $P(\theta_B)$. The procedure is shown using θ_A and it is analogous when using θ_B .

The posterior is

$$P(\theta_A \mid \mathcal{D}_A) = \frac{1}{P(\mathcal{D}_A)} P(\theta_A) \theta_A^{\#(a=1)} (1 - \theta_A)^{\#(a=0)}.$$

The most convenient choice for the prior is a Beta distribution as conjugacy will hold:

$$\theta_A \sim \text{Beta}(\alpha_A, \beta_A) \implies P(\theta_A) = \frac{1}{B(\alpha_A, \beta_A)} \theta_A^{\alpha_A-1} (1 - \theta_A)^{\beta_A-1}.$$

Therefore, it follows that

$$\theta_A \mid \mathcal{D}_A \sim \text{Beta}(\alpha_A + \#(A = 1), \beta_A + \#(A = 0)).$$

The marginal is then

$$\begin{aligned}
 P(A = 1 \mid \mathcal{D}_A) &= \frac{P(A = 1, \mathcal{D}_A)}{P(\mathcal{D}_A)} = \int_{\theta_A} \frac{P(A = 1, \mathcal{D}_A, \theta_A)}{P(\mathcal{D}_A)} = \int_{\theta_A} \frac{P(A = 1 \mid \mathcal{D}_A, \theta_A) P(\mathcal{D}_A, \theta_A)}{P(\mathcal{D}_A)} \\
 &= \int_{\theta_A} \frac{P(A = 1 \mid \mathcal{D}_A, \theta_A) P(\theta_A \mid \mathcal{D}_A) P(\mathcal{D}_A)}{P(\mathcal{D}_A)} \\
 &= \int_{\theta_A} P(\theta_A \mid \mathcal{D}_A) \theta_A = \mathbb{E}[\theta_A \mid \mathcal{D}_A] \\
 &= \frac{\alpha_A + \#(A = 1)}{\alpha_A + \#(A = 1) + \beta_A + \#(A = 0)}.
 \end{aligned}$$

Where the last equality is given by the expected value of a Beta distribution.

For $P(d \mid a, b)$ the situation is more complex, the simplest approach is to specify a Beta prior for each of the components of θ_D . Focus on θ_2 , notice the parameters α and β we used before now do depend on A and B , these are called *hyperparameters*.

$$\theta_2 \sim \text{Beta}\left(\alpha_D(1, 0) + \#(D = 1, A = 1, B = 0), \beta_D(1, 0) + \#(D = 0, A = 1, B = 0)\right).$$

Repeating the procedure we used with A we get that

$$P(D = 1 \mid A = 1, B = 0, \mathcal{D}) = \frac{\alpha_D(1, 0) + \#(D = 1, A = 1, B = 0)}{\alpha_D(1, 0) + \beta_D(1, 0) + \#(A = 1, B = 0)}.$$

In case we had no preference, we could set all hyperparameters to the same value, where a complete ignorance prior would correspond to set them to 1.

Once we get to this situation there are two limit possibilities depending on the amount of data we got.

- **No data.** The marginal probability corresponds to the prior, which in the last case is

$$P(D = 1 \mid A = 1, B = 0, \mathcal{D}) = \frac{\alpha_D(1, 0)}{\alpha_D(1, 0) + \beta_D(1, 0)}.$$

Note that equal hyperparameters would give a result of 0.5.

- **Infinite data.** When infinite data is available, the marginal is generally dominated by it, this corresponds to the Maximum Likelihood solution.

$$P(D = 1 \mid A = 1, B = 0, \mathcal{D}) = \frac{\#(D = 1, A = 1, B = 0)}{\#(A = 1, B = 0)}.$$

This happens unless the prior has a pathologically strong effect.

Consider the data given in the table in figure 11, and equal parameters and hyperparameters 1 and a prior belief that any setting is equally probable, i.e, $P(A = 1) = 0.5$.

We may illustrate the different results that are obtained using using Bayesian inference and Maximum likelihood training. The former is

$$P(A = 1 \mid \mathcal{D}) = \frac{1 + \#(A = 1)}{2 + N} = \frac{5}{9} \approx 0.556.$$

and the latter is $4/7 = 0.571$. In conclusion, the Bayesian result is more prudent than this one, which fits in with our prior belief.

11.3 LEARNING DISCRETE VARIABLES

The natural generalization to more than two states is using a Dirichlet distribution as prior, assuming i.i.d data, local and global prior independence. We are considering two different scenarios, firstly one where the variable has no parents, as the case for A and B in the previous example. Secondly, we will consider a variable with a non void set of parents.

11.3.1 No parents

Consider a variable $X \sim \text{Categorical}(\theta)$ with $\text{Dom}(X) = \{1, \dots, I\}$ and $\theta = (\theta_1, \dots, \theta_I)$ so that

$$P(x) = \prod_{i=1}^I \theta_i^{\mathbb{I}[x=i]} \text{ with } \sum_{i=1}^I \theta_i = 1.$$

So that the posterior (considering N observations of the variable $\mathcal{D} = (x_1, \dots, x_N)$) is

$$P(\theta \mid \mathcal{D}) = \frac{1}{P(\mathcal{D})} P(\theta) \prod_{n=1}^N \prod_{i=1}^I \theta_i^{\mathbb{I}[x_n=i]} = \frac{1}{P(\mathcal{D})} P(\theta) \prod_{i=1}^I \theta_i^{\sum_n \mathbb{I}[x_n=i]}.$$

Then, assuming a prior $\theta \sim \text{Dirichlet}(\mathbf{u})$ with $\mathbf{u} = (u_1, \dots, u_I)$

$$P(\theta) = \frac{1}{B(\mathbf{u})} \prod_{i=1}^I \theta_i^{u_i-1} \implies P(\theta \mid \mathcal{D}) = \frac{1}{B(\mathbf{u})P(\mathcal{D})} \prod_{i=1}^I \theta_i^{u_i-1+\sum_n \mathbb{I}[x_n=i]}.$$

Therefore, defining $\mathbf{c} = (\sum_{n=1}^N \mathbb{I}[x_n = i])_{i=1, \dots, I}$, the posterior follows a Dirichlet distribution

$$\theta \mid \mathcal{D} \sim \text{Dirichlet}(\mathbf{u} + \mathbf{c}).$$

It may not be easy to decompose $B(\mathbf{u} + \mathbf{c})$ as $B(\mathbf{u})P(\mathcal{D})$, but using that it ensures normalization of the Beta distribution we may use that

$$\int_{\theta} P(\theta \mid \mathcal{D}) = 1 \implies \int_{\theta} \prod_{i=1}^I \theta_i^{u_i+c_i-1} = B(\mathbf{u} + \mathbf{c}) = B(\mathbf{u})P(\mathcal{D}).$$

Remark 5. Summarizing the above information, we just proved that the Dirichlet distribution is the conjugate prior of the Categorical distribution.

The marginal is then given by

$$\begin{aligned} P(X = i \mid \mathcal{D}) &= \int_{\theta} P(X = i \mid \theta) P(\theta \mid \mathcal{D}) = \int_{\theta} \theta_i P(\theta \mid \mathcal{D}) \\ &= \int_{\theta_i} \theta_i P(\theta_i \mid \mathcal{D}) = \mathbb{E}[\theta_i \mid \mathcal{D}]. \end{aligned}$$

Where we used that

$$\int_{\theta_{j \neq i}} \theta_i P(\theta \mid \mathcal{D}) = \theta_i \prod_{k \neq j} P(\theta_k \mid \mathcal{D}) \int_{\theta_j} P(\theta_j \mid \mathcal{D}) = \theta_i \prod_{k \neq j} P(\theta_k \mid \mathcal{D})$$

As we already know from Proposition 2, the uni-variate marginal of a Dirichlet distribution is a Beta distribution of parameters

$$\theta_i \mid \mathcal{D} \sim \text{Beta}(u_i + c_i, \sum_{j \neq i} u_j + c_j).$$

Using the expectation of a Beta distribution we get that

$$P(X = i \mid \mathcal{D}) = \frac{u_i + c_i}{\sum_j u_j + c_j}.$$

11.3.2 Parents

Consider now that X has a set of parent variables $pa(X)$, in this case, we want to compute the marginal given a state of its parents and the data:

$$P(X = i \mid pa(X) = \mathbf{j}, \mathcal{D}).$$

We are using the following notation for the parameters

$$P(X = i \mid pa(X) = \mathbf{j}, \theta) = \theta_{i,\mathbf{j}}, \quad \theta_{\mathbf{j}} = (\theta_{1,\mathbf{j}}, \dots, \theta_{L,\mathbf{j}}).$$

Local independence means that

$$P(\theta) = \prod_j P(\theta_j).$$

As we did before, we consider a Dirichlet prior

$$\theta_{\mathbf{j}} \sim \text{Dirichlet}(\mathbf{u}_{\mathbf{j}}),$$

the posterior is then

$$\begin{aligned} P(\theta \mid \mathcal{D}) &= \frac{P(\theta)P(\mathcal{D} \mid \theta)}{P(\mathcal{D})} = \frac{1}{P(\mathcal{D})} \left(\prod_j P(\theta_j) \right) P(\mathcal{D} \mid \theta) \\ &= \frac{1}{P(\mathcal{D})} \left(\prod_j \frac{1}{B(\mathbf{u}_{\mathbf{j}})} \prod_i \theta_{i,\mathbf{j}}^{u_{i,\mathbf{j}}-1} \right) P(\mathcal{D} \mid \theta) \\ &= \frac{1}{P(\mathcal{D})} \left(\prod_j \frac{1}{B(\mathbf{u}_{\mathbf{j}})} \prod_i \theta_{i,\mathbf{j}}^{u_{i,\mathbf{j}}-1} \right) \left(\prod_n \prod_j \prod_i \theta_{i,\mathbf{j}}^{\mathbb{I}_{[x_n=i, pa(x_n)=\mathbf{j}]}} \right) \\ &= \frac{1}{P(\mathcal{D})} \prod_j \frac{1}{B(\mathbf{u}_{\mathbf{j}})} \prod_i \theta_{i,\mathbf{j}}^{u_{i,\mathbf{j}}-1 + \#(X=i, pa(X)=\mathbf{j})}. \end{aligned}$$

Naming $\mathbf{v}_{\mathbf{j}} = \mathbf{u}_{\mathbf{j}} + \#(X = i, pa(X) = \mathbf{j})$ and using the same argument as we did in the *no parents* case with the normalization constants, the posterior is

$$(\theta \mid \mathcal{D}) \sim \prod_j \text{Dirichlet}(\mathbf{v}_{\mathbf{j}}).$$

Denoting $v_{i,\mathbf{j}}$ the components of $\mathbf{v}_{\mathbf{j}}$, the marginal is

$$P(X = i, pa(X) = \mathbf{j}, \mathcal{D}) = \frac{v_{i,\mathbf{j}}}{\sum_i v_{i,\mathbf{j}}}.$$

Notice all the above has been done using a fixed variable X , so that all the parameters depend on that variable.

We can define the data likelihood under a model, usually called the *model likelihood* using the same calculations as we did with $P(\boldsymbol{\theta} \mid \mathcal{D})$ but applied to all the variables of the model

$$\begin{aligned} P(\mathcal{D} \mid \mathcal{M}) &= \int_{\boldsymbol{\theta}} P(\boldsymbol{\theta}) P(\mathcal{D} \mid \boldsymbol{\theta}, \mathcal{M}) = \prod_x \prod_j \frac{1}{B(\mathbf{u}_j)} \int_{\boldsymbol{\theta}} \prod_i \theta_{i,j}^{u_{i,j}-1+\#(x=i, pa(x)=j)} \\ &= \prod_x \prod_j \frac{B(\mathbf{v}_j)}{B(\mathbf{u}_j)}. \end{aligned}$$

STRUCTURE LEARNING

So far, we have assumed the Bayesian Network was an hypothesis of the problem, however, this structure is not always given and may be learned also from the data.

Even considering complete data (no missing observations), there are some problems that need to be taken into account.

- The number of Bayesian networks is exponential over the number of variables so a brute force algorithm would not be viable.
- Testing dependencies requires a large amount of data. Therefore a threshold must be set to measure when a dependence is significant.
- A Bayesian network or a Markov network may not be enough to represent the data due to the existence of unobserved variables.

12.1 PC ALGORITHM

An approach to learn the structure is the PC algorithm, it starts with a complete graph G and tries to remove as many links as possible studying the independence of the variables.

The algorithm [2](#) iterates over neighborhoods, from smaller to bigger ones. It chooses a linked pair of variables $(X, Y) \in E$ and a subset $S_{XY} \subset ne(X)$, verifying $Y \notin S_{XY}$. If $X \perp\!\!\!\perp Y \mid S$, then the link is removed and S_{XY} is stored.

The main idea behind the algorithm is that a set of independencies is faithful to a graph if there is no link between two nodes X and Y if and only if there exists a subset of $ne(X)$ such that they are independent given this subset.

This gives the skeleton of the Bayesian Network, no more edges will be removed or added. The directed graph may be constructed following two rules:

1. For any undirected link $X - Y - Z$, if $Y \notin S_{XZ}$ then set $X \rightarrow Y \leftarrow Z$ (we are creating a collider for that path).
2. The rest of links may be oriented arbitrarily not creating cycles or colliders.

The reasoning behind this is the d-separation theorem [2](#), if $Y \notin S_{XZ}$ then $X \not\perp\!\!\!\perp Z \mid Y$ so Y must d-connect them, to get this we set it as a collider. On the other hand, if $Y \in S_{XZ}$ and $X \perp\!\!\!\perp Z \mid S_{XZ}$ then we want S_{XZ} to d-separate them, that is, using any configuration that doesn't create a collider in S_{XZ} .

Algorithm 2: PC Algorithm**Data:** Complete undirected graph G , with vertices \mathcal{V} **Result:** G with removed links $i = 0;$ **while** all nodes have $\leq i$ neighbors **do** **for** $X \in \mathcal{V}$ **do** **for** $Y \in ne(x)$ **do** **if** $\exists S \subset ne(X) \setminus Y$ such that $\#S = i$ and $X \perp\!\!\!\perp Y \mid S$ **then** Remove $X - Y$ from G ; $S_{XY} = S$; **end** **end** **end** $i = i + 1$;**end**

12.2 INDEPENDENCE LEARNING

Our main concern now is, given three variables X, Y and Z , measure $X \perp\!\!\!\perp Y \mid Z$. One approach is to measure the empirical *conditional mutual information* of the variables.

Definition 44. Given two random variables X, Y , we define their *mutual information* as the Kullback-Leibler divergence of their joint distribution and the product of their marginals,

$$MI(X; Y) = KL(P_{X,Y} \mid P_X P_Y).$$

Definition 45. Given three random variables X, Y, Z we define the *conditional mutual information* of X and Y over Z as

$$MI(X; Y \mid Z) = \mathbb{E}_Z \left[KL(P_{X,Y \mid Z} \mid P_{X \mid Z} P_{Y \mid Z}) \right].$$

Where $MI(X; Y \mid Z) \geq 0$ and $MI(X; Y \mid Z) = 0 \iff P_{X,Y \mid Z} = P_{X \mid Z} P_{Y \mid Z} \iff X \perp\!\!\!\perp Y \mid Z$. We can estimate this using the empirical distributions, however, this *empirical* mutual information will be typically greater than 0 even when $X \perp\!\!\!\perp Y \mid Z$, therefore a threshold must be established.

A Bayesian approach would consist on comparing the model likelihood under independence and dependence hypothesis. That is, computing the model likelihood for the below joint distributions assuming both local and global parameter independence

$$P_{indep}(x, y, z) = P(x \mid z, \theta_1) P(y \mid z, \theta_2) P(z \mid \theta_3) P(\theta_1) P(\theta_2) P(\theta_3),$$

$$P_{dep}(x, y, z) = P(x, y, z \mid \theta) P(\theta).$$

MISSING VARIABLES

Until this moment we have assumed that the data we are given is completed but in practice this data is not in two different ways. There may be unobserved or *hidden* variables that affect the visible ones, and there may be *missing* information, that is, states of visible variables that are missing.

Think about the example with the disease and the two habits we used in the last section, missing data would be a row in the table where some entry is missing, for example $x_3 = \{D = 1, A = 1\}$, where we know that this person got the disease and had habit A but we have no information about habit B , this is an example of *missing* data.

One approach to handle this situation would be marginalizing over that variable:

$$P(x_3 | \theta) = \int_b P(d_3, a_3, b | \theta) = P(a_3 | \theta_A) \int_b P(b | \theta_B) P(d_3 | b, a_3, \theta_D).$$

This leads to a non-factorized form of the posterior which is computationally difficult to handle, notice the problem is not conceptual but computational. Using the marginal to handle missing information does not always lead to this situation, in fact, marginalizing over a collider (D in our example) would mean losing that variable as the integral simply equals 1.

$$P(x_3 | \theta) = \int_d P(d, a_3, b_3 | \theta) = P(a_3 | \theta_A) P(b_3 | \theta_B) \int_d P(d | b_3, a_3, \theta_D) = P(a_3 | \theta_A) P(b_3 | \theta_B);$$

There are three main types of missing data:

- **Missing completely at random (MCAR).** If the events that lead to any particular data to be missing is independent from both the observed and the unobserved variables, and occur at random.
- **Missing at random (MAR).** When the absence is not random but can be explained with the observed variables.
- **Missing not at random (MNAR).** The missing data is related with the reason why it is missing. For example, skipping a question in a survey for being ashamed of the answer.

To express this mathematically, split the variables \mathcal{X} into visible \mathcal{X}_{vis} and hidden \mathcal{X}_{hid} , let M be a variable denoting that the state of the hidden variables is known (0) or unknown (1). So the difference between the three types resides on how $P(M = 1 | x_{vis}, x_{hid}, \theta)$ simplifies.

When data is *missing at random*, we assume that we can explain the missing information with the visible one, so the probability of being missing only depends on the visible data, that is

$$P(M = 1 \mid x_{vis}, x_{hid}, \theta) = P(M = 1 \mid x_{vis}),$$

so that,

$$P(x_{vis}, M = 1 \mid \theta) = P(M = 1 \mid x_{vis})P(x_{vis} \mid \theta)$$

Assuming the data is *missing completely at random* is stronger, as we are supposing that there is no reason behind the missing data, so that it being missing is independent from the visible and hidden data:

$$P(M = 1 \mid x_{vis}, x_{hid}, \theta) = P(M = 1),$$

so now

$$P(x_{vis}, M = 1 \mid \theta) = P(M = 1)P(x_{vis} \mid \theta).$$

In both cases we may simply use the marginal $P(x_{vis} \mid \theta)$ to assess parameters as $P(x_{vis}, M = 1 \mid \theta)$ does not depend on the missing variables.

In case data is *missing not at random*, no independence assumption is made over the probability of the data being unknown, meaning it depends on both the visible and the hidden information. From now on, we will assume missing information is either MAR or MCAR, even though this could lead to a misunderstanding of the problem as in the following simple example.

Example 4. Consider a situation where data is obtained from a survey where people are asked to choose between 3 options A, B and C . Assume that no one chose option C because they are ashamed of the answer, and the answers are uniform between A, B and not answering.

Normalizing the missing information would lead to setting $P(A \mid \mathcal{V}) = 0.5 = P(B \mid \mathcal{V})$ and $P(C \mid \mathcal{V}) = 0$ when the reasonable result is that not answering equals to choosing C so that $P(A \mid \mathcal{V}) = P(B \mid \mathcal{V}) = P(C \mid \mathcal{V}) = \frac{1}{3}$

VARIATIONAL INFERENCE IN BAYESIAN NETWORKS

In this section we come back to variational inference, in this case we are reviewing two main algorithms, *expectation maximization* and *variational bayes*.

The chapter uses the notation and information described in [Barber \(2007\)](#) chapter 11.2.

14.1 EXPECTATION MAXIMIZATION

The *expectation maximization* algorithm is an iterative method to find maximum likelihood or maximum a posteriori estimates of parameters in statistical models, where the model depends on hidden variables. The main idea is to set a lower bound to the marginal likelihood, then using an iterative method to increase this lower bound, and the marginal likelihood with it.

14.1.1 General case

Consider we have only two variables, one visible V and one hidden H . The lower bound is defined using the Kullback-Leibler divergence between the variational distribution $Q(z | x)$ the a parametric one $P(v | h, \theta)$.

$$\begin{aligned} KL(Q(h | v) | P(h | v, \theta)) &= \mathbb{E}_Q \left[\log Q(h | v) - \log P(h | v, \theta) \right] \\ &= \mathbb{E}_Q \left[\log Q(h | v) \right] - \mathbb{E}_Q \left[\log P(h, v | \theta) \right] + \mathbb{E}_Q \left[\log P(v | \theta) \right] \\ &= \mathbb{E}_Q \left[\log Q(h | v) \right] - \mathbb{E}_Q \left[\log P(h, v | \theta) \right] + \log P(v | \theta) \geq 0. \end{aligned}$$

We got then the lower bound for $\log P(v | \theta)$

$$\log P(v | \theta) \geq \underbrace{-\mathbb{E}_Q \left[\log Q(h | v) \right]}_{\text{Entropy}} + \underbrace{\mathbb{E}_Q \left[\log P(h, v | \theta) \right]}_{\text{Energy}}.$$

Let $\{V_1, \dots, V_N\}$ and $\{H_1, \dots, H_N\}$ be two set of i.i.d random variables and $\{v_1, \dots, v_N\}$ a set of observations, and $\{h_1, \dots, h_N\}$ a set of possible values of the hidden variables, the lower bound is then written as

$$\log P(v_1, \dots, v_N | \theta) = \sum_{n=1}^N \log P(v_n | \theta) \geq \sum_{n=1}^N -\mathbb{E}_Q \left[\log Q(h_n | v_n) \right] + \mathbb{E}_Q \left[\log P(h_n, v_n | \theta) \right].$$

And we know equality holds if and only if $Q(h_n | v_n) = P(h_n | v_n, \theta) \forall n = 1, \dots, N$.

In contrast with CAVI, no mean-field family is assumed here, and the procedure to optimize the parameter consists in two steps:

- **E-step.** For a fixed θ , find the distributions that maximize the above bound, i.e, choose $Q(h_n | v_n) = P(h_n | v_n, \theta)$.
- **M-step.** For a fixed distribution Q , find the parameter θ that maximizes the bound. Since Q does not depend on the parameter, this is equivalent to maximize the energy term.

Example 5. Consider a single variable V with $\text{Dom}(V) = \mathbb{R}$ and a single hidden variable H with $\text{dom}(H) = \{1, 2\}$. Consider the model

$$P(v | h, \theta) = \frac{1}{\sqrt{\pi}} e^{-(v-\theta h)^2},$$

and $P(H = 1) = P(H = 2) = 0.5$. Suppose an observation $v = 2.75$ we want to optimize the parameter θ in the marginal

$$P(V = 2.75 | \theta) = \int_h P(V = 2.75 | h, \theta) P(h) = \frac{1}{2\sqrt{\pi}} (e^{-(2.75-\theta)^2} + e^{-(2.75-2\theta)^2}).$$

Then using a distribution $Q(h | v)$, the lower bound given to the log likelihood is

$$\begin{aligned} \log P(v | \theta) &\geq -Q(1 | 2.75) \log Q(1 | 2.75) - Q(2 | 2.75) \log Q(2 | 2.75) \\ &\quad - \mathbb{E}_Q[(v - \theta h)^2] + \text{const.} \end{aligned}$$

The M-step can be done analytically, noticing that due to the negative sign we want to minimize $\mathbb{E}_Q[(v - \theta h)^2]$

$$\frac{d}{d\theta} \mathbb{E}_Q[(v - \theta h)^2] = \mathbb{E}_Q[2vh + 2\theta h^2] = 2v\mathbb{E}_Q[h] + 2\theta\mathbb{E}_Q[h^2] = 0 \iff \theta = \frac{v\mathbb{E}_Q[h]}{\mathbb{E}_Q[h^2]},$$

$$\frac{d^2}{d^2\theta} \mathbb{E}_Q[(v - \theta h)^2] = 2\mathbb{E}_Q[h^2] \geq 0,$$

so the new parameter optimal parameter is

$$\theta_{new} = v \frac{\mathbb{E}_Q[H]}{\mathbb{E}_Q[H^2]}.$$

The E-step would set $Q_{new}(h | v) = P(h | v, \theta)$, in this case

$$Q_{new}(h) = \frac{P(V = 2.75 | h, \theta) P(H = 2)}{P(V = 2.75)} = \frac{e^{-(2.75-h\theta)}}{e^{-(2.75-\theta)} + e^{-(2.75-2\theta)}}.$$

Algorithm 3: Expectation Maximization Algorithm

Data: A distribution $P(x \mid \theta)$ and a dataset \mathcal{D} . Where X splits in visible variables V and hidden variables H

Result: Parameter θ that maximizes the likelihood

while *Convergence stop criteria* **do**

for $n \in 1, \dots, N$ **do**

$Q(h \mid v_n) = P(h \mid v_n, \theta);$

end

$\theta = \arg \max_{\theta} \sum_{n=1}^N \mathbb{E}_{Q(h|v_n)} [\log P(h, v_n \mid \theta)];$

end

return $\theta;$

14.1.2 EM increases the marginal likelihood

It is clear that the EM algorithm does increase the lower bound in each iteration but we would like it to increase not only the bound but also the marginal likelihood. We will be using a single data point as it easy holds by summation when using the full dataset.

Let θ^{new} be the value of the parameter after one iteration, we assume that Q is set to

$$Q(h \mid v) = P(h \mid v, \theta).$$

So the lower bound in terms of θ and θ^{new} is

$$ELBO(\theta^{new} \mid \theta) = \underbrace{-\mathbb{E}_{P(h|v,\theta)} [\log P(h \mid v, \theta)]}_{\text{Entropy}} + \underbrace{\mathbb{E}_{P(h|v,\theta)} [\log P(h, v \mid \theta^{new})]}_{\text{Energy}}.$$

From the definition of the Kullback-Leibler we get that

$$\log P(v \mid \theta^{new}) = ELBO(\theta^{new} \mid \theta) + KL(P(h \mid v, \theta) \mid P(h \mid v, \theta^{new})).$$

We could use θ in the above formula getting

$$\log P(v \mid \theta) = ELBO(\theta \mid \theta) + KL(P(h \mid v, \theta) \mid P(h \mid v, \theta)) = ELBO(\theta \mid \theta).$$

So we can compute the difference between the log likelihood between two consecutive iterations as

$$\log P(v \mid \theta^{new}) - \log P(v \mid \theta) = ELBO(\theta^{new} \mid \theta) - ELBO(\theta \mid \theta) + KL(P(h \mid v, \theta) \mid P(h \mid v, \theta^{new})).$$

Where we know the last term is always positive, about the difference of bounds, the M-step ensures the new parameter makes the lower bound higher or equal to the current one, so that difference is also positive.

$$\log P(v \mid \theta^{new}) - \log P(v \mid \theta) \geq 0.$$

14.1.3 Bayesian Networks case

As we did before let $\mathbf{X} = (\mathbf{V}, \mathbf{H}) = \{X_1, \dots, X_M\}$ be the set of variables partitioned in visible and hidden. Let $\{\mathbf{v}^1, \dots, \mathbf{v}^N\}$ be the set of observations and $\{\mathbf{h}_1, \dots, \mathbf{h}_N\}$ a set of possible values of the hidden variables in each observation.

As we are not using a single variable anymore, the variational distribution is now over the set of hidden variables conditioned on the visible $Q(\mathbf{H} | \mathbf{V})$. As the E-step is the same as in the general case, we are focusing on the M-step, where Q_{old} is fixed.

The *energy term* in a Bayesian Network is

$$\sum_{n=1}^N \mathbb{E}_{Q_{old}(\mathbf{h}^n | \mathbf{v}^n)} \left[\log P(\mathbf{x}^n | \theta) \right] = \sum_{n=1}^N \sum_{i=0}^M \mathbb{E}_{Q_{old}(\mathbf{h}^n | \mathbf{v}^n)} \left[\log P(x_i^n | pa(x_i^n), \theta) \right].$$

It is useful to use the following notation that defines a conditional distribution of the hidden variable when the visible one equals \mathbf{v}^n .

$$Q^n(\mathbf{x}) = Q^n(\mathbf{v}, \mathbf{h}) = Q_{old}(\mathbf{h}^n | \mathbf{v}^n) \mathbb{I}(\mathbf{v} = \mathbf{v}^n).$$

We can define the mixture distribution

$$Q(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N Q^n(\mathbf{x}).$$

Then we have that

$$\begin{aligned} \mathbb{E}_{Q(\mathbf{x})} \left[\log P(\mathbf{x} | \theta) \right] &= \int_{\mathbf{x}} Q(\mathbf{x}) \log P(\mathbf{x} | \theta) = \int_{\mathbf{x}} \frac{1}{N} \sum_{n=1}^N Q^n(\mathbf{x}) \log P(\mathbf{x} | \theta) \\ &= \frac{1}{N} \int_{\mathbf{x}} \sum_{n=1}^N Q_{old}(\mathbf{h}^n | \mathbf{v}^n) \mathbb{I}[\mathbf{v} = \mathbf{v}^n] \log P(\mathbf{x} | \theta) \\ &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{Q_{old}(\mathbf{h}^n | \mathbf{v}^n)} \left[\log P(\mathbf{h}^n, \mathbf{v}^n | \theta) \right] \\ &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{Q_{old}(\mathbf{h}^n | \mathbf{v}^n)} \left[\log P(\mathbf{x}^n | \theta) \right], \end{aligned}$$

equals the energy term. Then, using the Belief Network structure

$$\begin{aligned} \mathbb{E}_{Q(\mathbf{x})} \left[\log P(\mathbf{x} | \theta) \right] &= \sum_{i=1}^M \mathbb{E}_{Q(\mathbf{x})} \left[\log P(x_i | pa(x_i), \theta) \right] \\ &= \sum_{i=1}^M \mathbb{E}_{Q(x_i, pa(x_i))} \left[\log P(x_i | pa(x_i), \theta) \right] \\ &= \sum_{i=1}^M \mathbb{E}_{Q(pa(x_i))} \left[\mathbb{E}_{Q(x_i | pa(x_i))} \left[\log P(x_i | pa(x_i), \theta) \right] \right]. \end{aligned}$$

We add a constant to the last term so it comes with the structure of a Kullback-Leibler Divergence (notice its sign has changed)

$$\sum_{i=1}^M \mathbb{E}_{Q(pa(x_i))} \left[\mathbb{E}_{Q(x_i | pa(x_i))} \left[\log Q(x_i | pa(x_i)) \right] \right] - \mathbb{E}_{Q(x_i | pa(x_i))} \left[\log P(x_i | pa(x_i), \theta) \right] =$$

$$= \sum_{i=1}^M E_{Q(pa(x_i))} \left[KL \left(Q(x_i | pa(x_i)) | P(x_i | pa(x_i), \theta) \right) \right].$$

So maximizing the energy term is equivalent to minimize the above formula, that is, setting

$$P^{new}(x_i | pa(x_i), \theta) = Q(x_i | pa(x_i)).$$

So the first observation is that θ is not needed in order to maximize the energy term due to the Belief network structure.

14.2 DISTRIBUTION MIXTURE EXAMPLE WITH EM

In this section we are using a coin-flipping experiment as an example to show the difference between ML training and EM training in a *mixture distribution* (Do & Batzoglou (2008)).

Consider a simple coin-flipping experiment in which we are given a pair of coins A and B of unknown biases, θ_A and θ_B , respectively. Let 1 denote *heads* and 0 denote *tails*.

$$P(A = 1) = \theta_A \quad P(B = 1) = \theta_B,$$

Our goal is to estimate $\theta = (\theta_A, \theta_B)$ by repeating the following procedure five times: randomly choose one of the two coins (with equal probability), and perform ten independent coin tosses with the selected coin.

The ML approach is setting

$$\theta_A^{opt} = \frac{\text{Heads of coin } A}{\text{Total flips of coin } A}, \quad \theta_B^{opt} = \frac{\text{Heads of coin } B}{\text{Total flips of coin } B'}$$

Consider now that the identity on the coin that is being flipped is unknown. Let X be the random variable modeling the number of heads in M flips and Z the coin being flipped. Consider a dataset of size N . This time, using the ML approach is not possible.

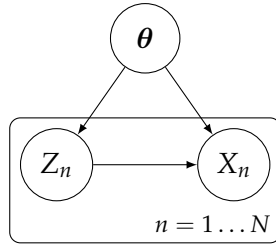


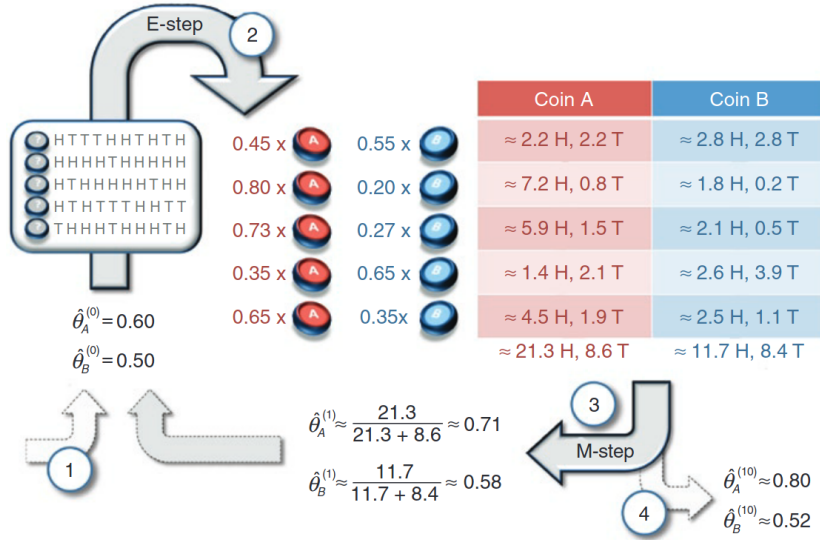
Figure 12: Bayesian Network modeling the flipping coin experiment

The lower bound is in this case

$$\sum_{n=1}^N \log P(x_n | \theta) \geq \sum_{n=1}^N -\mathbb{E}_Q \left[\log Q(z_n | x_n) \right] + \mathbb{E}_Q \left[\log P(z_n, x_n | \theta) \right],$$

Lets start with **E-step**, we already know it consist on setting

$$\begin{aligned} Q(z_n | x_n, \theta) &= P(z_n | x_n, \theta) = \frac{P(x_n | z_n, \theta) P(z_n)}{P(x_n)} \\ &= \frac{\theta_{z_n}^{x_n} (1 - \theta_{z_n})^{M-x_n}}{\theta_A^{x_n} (1 - \theta_A)^{M-x_n} + \theta_B^{x_n} (1 - \theta_B)^{M-x_n}} \quad \forall n = 1, \dots, N. \end{aligned}$$

Figure 13: EM algorithm step in Mixture [Do & Batzoglou \(2008\)](#)

The **M-step** consists on setting

$$\theta^{new} = \arg \max_{\theta} \sum_{n=1}^N \mathbb{E}_Q [P(z_n | x_n)],$$

where

$$\begin{aligned} \mathbb{E}_Q [P(z_n | x_n)] &= Q(z_n = A) (x_n \log \theta_A + (M - x_n) \log (1 - \theta_A)) \\ &+ Q(z_n = B) (x_n \log \theta_B + (M - x_n) \log (1 - \theta_B)). \end{aligned}$$

The term affected by θ_A is

$$\sum_{n=1}^N Q(z_n = A) (x_n \log \theta_A + (M - x_n) \log (1 - \theta_A)),$$

deriving and setting to 0 we get the following maxima.

$$\sum_{n=1}^N Q(z_n = A) \left(\frac{x_n}{\theta_A} - \frac{M - x_n}{1 - \theta_A} \right) = 0 \iff \theta_A = \sum_{n=1}^N Q(z_n = A) \frac{x_n}{M}.$$

The same argument is valid for θ_B .

We are now using the figure 13 to set the example in a numerical context. We are following the given data points in the diagram.

1. We are considering a prior $\theta = (0.6, 0.5)$, a set of $N = 5$ samples and $M = 10$ flips per sample.
2. This corresponds to the **E-step**, we calculate each $Q(z_n | x_n, \theta)$.

$$\begin{aligned} Q(z_1 = A) &= \frac{0.6^5 0.4^5}{0.6^5 0.4^5 + 0.5^5 0.5^5} \approx 0.45 \implies Q(z_1 = B) \approx 0.55, \\ &\vdots \\ Q(z_5 = A) &\approx 0.65 \implies Q(z_5 = B) \approx 0.35. \end{aligned}$$

3. The **M-step** corresponds to setting

$$\theta_A = \sum_{n=1}^5 Q(z_n) \frac{x_n}{10} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71,$$

$$\theta_B \approx 0.58.$$

4. Step 4 represents a possible solution after 10 iterations.

14.3 EM EXTENSIONS

14.3.1 Partial steps

Making a partial M-step consist on not using the optimal parameter for the energy term, but using one with just higher energy. Finding this values can be easier than finding the optimal one and convergence still follows as the only requirement to make the likelihood increase was to increase the lower bound.

On the other hand, when studying the increase on the likelihood, we supposed that the optimal E-step was being used. In general, we cannot guarantee that a partial step would increase the likelihood, even though it is guaranteed to increase the lower bound.

Another important factor is, that the EM algorithm assumes that the energy term is possible to calculate, which may not be. As an approach to solve this situation, we can set a class of distributions \mathcal{Q} , and minimize the KullBack-Leibler divergence between $P(h | v, \theta)$ and a distribution $Q \in \mathcal{Q}$, so we pick a distribution such that

$$Q = \arg \min_{Q \in \mathcal{Q}} KL(Q(h | v) | P(h | v, \theta)).$$

An extreme case is to choose \mathcal{Q} as delta functions, where the energy term is now a constant. And the optimal chose setting is

$$Q(h^n | v^n) = \delta(h^n, h_{opt}^n), \quad h_{opt}^n = \arg \max_h P(h, v^n | \theta).$$

This is called *Viterbi training* and does not guarantee that the log likelihood is being increased in each iteration.

14.4 VARIATIONAL BAYES

Another method to deal with hidden variables is using *Variational Bayes (VB)*, in contrast with the EM algorithm, this one uses a distribution that better represents the posterior.

Consider a simple datapoint $x = (v, h)$, in this situation we focus our interest on the posterior distribution:

$$P(\theta | v) = \frac{P(v | \theta)P(\theta)}{P(v)} = \frac{1}{P(v)} \int_h P(v, h | \theta)P(\theta).$$

Variational Bayes assumes the joint hidden and parameter posterior can be approximated as the product of two distributions, one over the hidden variable and one over the parameter such that

$$P(h, \theta | v) \approx Q(h)Q(\theta).^1$$

Remark 6. By doing this assumption, we are also assuming that the posterior can be approximated with $Q(\theta)$

$$P(\theta | v) = \int_h P(\theta, h | v) \approx \int_h Q(h)Q(\theta) = Q(\theta).$$

For this reason, the main goal of the algorithm is to make this approximation as tightest as possible with an iterative method that calculates both $Q(\theta)$ and $Q(h)$ distributions.

To achieve that, we minimize the Kullback-Leibler divergence between them.

$$KL(Q(h)Q(\theta) | P(h, \theta | v)) = \mathbb{E}_{Q(h)}[\log Q(h)] + \mathbb{E}_{Q(\theta)}[\log Q(\theta)] - \mathbb{E}_{Q(h)Q(\theta)}[\log P(h, \theta | v)].$$

We use that $\log P(v)$ is independent from $Q(h)Q(\theta)$ to get the desired inequality.

$$\begin{aligned} 0 &\leq \mathbb{E}_{Q(h)}[\log Q(h)] + \mathbb{E}_{Q(\theta)}[\log Q(\theta)] - \mathbb{E}_{Q(h)Q(\theta)}[\log P(h, \theta | v)] \\ &= \mathbb{E}_{Q(h)}[\log Q(h)] + \mathbb{E}_{Q(\theta)}[\log Q(\theta)] - \mathbb{E}_{Q(h)Q(\theta)}\left[\log \frac{P(h, \theta, v)}{P(v)}\right] \\ &= \mathbb{E}_{Q(h)}[\log Q(h)] + \mathbb{E}_{Q(\theta)}[\log Q(\theta)] - \mathbb{E}_{Q(h)Q(\theta)}[\log P(h, \theta, v)] + \mathbb{E}_{Q(h)Q(\theta)}[\log P(v)] \\ &= \mathbb{E}_{Q(h)}[\log Q(h)] + \mathbb{E}_{Q(\theta)}[\log Q(\theta)] - \mathbb{E}_{Q(h)Q(\theta)}[\log P(h, \theta, v)] + \log P(v). \end{aligned}$$

We got the following lower bound

$$\log P(v) \geq -\mathbb{E}_{Q(h)}[\log Q(h)] - \mathbb{E}_{Q(\theta)}[\log Q(\theta)] + \mathbb{E}_{Q(h)Q(\theta)}[\log P(h, \theta, v)].$$

Therefore, minimizing the Kullback-Leibler divergence is equivalent to find the tightest lower bound.

The procedure is then split in two steps to keep the structure of the EM algorithm.

- **E-step.** Given a fixed $Q(\theta)$, minimize the Kullback-Leibler divergence.

$$Q^{new}(h) = \arg \min_{Q(h)} KL(Q(h)Q(\theta) | P(h, \theta | v)).$$

- **M-step.** Given a fixed $Q(h)$, minimize the Kullback-Leibler divergence.

$$Q^{new}(\theta) = \arg \min_{Q(\theta)} KL(Q(h)Q(\theta) | P(h, \theta | v)).$$

For fixed $Q(\theta)$, the contribution from $Q(h)$ to the Kullback-Leibler divergence is, as we derived in the CAVI algorithm,

$$\begin{aligned} KL(Q(h)Q(\theta) | P(h, \theta | v)) &= \mathbb{E}_{Q(h)}[\log Q(h)] - \mathbb{E}_{Q(h)Q(\theta)}[\log P(v, h, \theta)] + \text{const.} \\ &= -KL(Q(h) | \exp \mathbb{E}_{Q(\theta)}[P(h, v, \theta)]) + \text{const.} \end{aligned}$$

¹ We use the same letter for both distributions, as they can be differenced from the context.

Therefore, the update is proportional to

$$Q^{new}(h) \propto \exp \mathbb{E}_{Q(\theta)} [\log P(v, h, \theta)] \propto \exp \mathbb{E}_{Q(\theta)} [\log P(v, h | \theta)].$$

Following a similar argument,

$$Q^{new}(\theta) \propto \exp \mathbb{E}_{Q(h)} [\log P(v, h, \theta)] \propto \exp \mathbb{E}_{Q(h)} [\log P(v, h | \theta)].$$

As in the case of the EM algorithm, each iterations guarantees an increase in the lower bound of the marginal likelihood, but increasing the marginal likelihood itself is not guaranteed.

When using an i.i.d dataset \mathcal{V}, \mathcal{H} , we may assume that $Q(\mathcal{H})$ is in the mean-field family:

$$Q(h_1, \dots, h_N) = \prod_{n=1}^N Q(h_n).$$

The lower bound to the marginal likelihood is then written as a summation of the bounds on each datapoint.

$$\log P(\mathcal{V}) \geq \sum_n -\mathbb{E}_{Q(h_n)} [\log Q(h_n)] - \mathbb{E}_{Q(\theta)} [\log Q(\theta)] + \mathbb{E}_{Q(h_n)Q(\theta)} [\log P(v_n, h_n, \theta)]$$

14.4.1 VB is a generalization of the EM algorithm

We start considering a distribution over the parameter that summarizes the information in the optimal point, let θ_{opt} be the optimal value of θ .

$$Q(\theta) = \delta(\theta - \theta_{opt})$$

The lower bound takes the form

$$\log P(v) \geq -\mathbb{E}_{Q(h)} [\log Q(h)] + \mathbb{E}_{Q(h)} [\log P(h, v, \theta_{opt})] + \text{const.}$$

The M-step is then picking the optimal parameter θ_{opt} given a fixed $Q(h)$:

$$\begin{aligned} \theta_{opt} &= \arg \max_{\theta} \left(\mathbb{E}_{Q(h)} [\log P(v, h, \theta)] \right) \\ &= \arg \max_{\theta} \left(\mathbb{E}_{Q(h)} [\log P(v, h | \theta) P(\theta)] \right) \\ &= \arg \max_{\theta} \left(\mathbb{E}_{Q(h)} [\log P(v, h | \theta)] + \log P(\theta) \right) \end{aligned}$$

If we take a flat prior ($P(\theta)$ constant), this term is equivalent to the energy one in the EM bound $\mathbb{E}_{Q(h|v)} [\log P(h, v | \theta)]$.

The VB E-step consists on minimizing $KL(Q(h) | P(h, \theta_{opt} | v))$ over $Q(h)$, as

$$\begin{aligned} KL(Q(h) | P(h | \theta_{opt}, v)) &= \mathbb{E}_{Q(h)} [Q(h)] - \mathbb{E}_{Q(h)} [P(h | \theta_{opt}, v)] \\ &= \mathbb{E}_{Q(h)} [Q(h)] - \mathbb{E}_{Q(h)} \left[\frac{P(h, \theta_{opt} | v)}{P(\theta_{opt})} \right] \\ &= \mathbb{E}_{Q(h)} [Q(h)] - \frac{1}{P(\theta_{opt})} \mathbb{E}_{Q(h)} [P(h, \theta_{opt} | v)] \end{aligned}$$

where $P(\theta_{opt})$ is a constant and

$$KL(Q(h) \mid P(h, \theta_{opt} \mid v)) = \mathbb{E}_{Q(h)}[Q(h)] - \mathbb{E}_{Q(h)}[P(h, \theta_{opt} \mid v)]$$

Then, as the E-step of the EM algorithm consisted on minimizing the Kullback-Leibler divergence $KL(Q(h \mid v) \mid P(h \mid v, \theta))$, both steps are equivalent.

Remark 7. Expectation Maximization algorithm is a special case of Variational Bayes with a flat prior and a delta function as the parameter posterior approximation.

14.5 COMPARISON

Variational Bayes is often compared with expectation maximization as their numerical procedure is quite similar. Their main differences are the following:

EM computes point estimates of posterior distribution of those random variables that can be categorized as "parameters", but only estimates the posterior distribution of the latent variables. On the other hand, VB computes a posterior distribution for all the variables.

For example, in a gaussian mixture, EM would directly estimate optimum values for the mean and variance of each gaussian and VB would fit a distribution to these parameters.

VARIATIONAL MESSAGE PASSING

"However, Monte Carlo methods are computationally very intensive, and also suffer from difficulties in diagnosing convergence"

"Expectation propagation is limited to certain classes of model for which the required expectations can be evaluated, is also not guaranteed to converge in general, and is prone to finding poor solutions in the case of multi-modal distributions."

"VMP allows variational inference to be applied automatically to a large class of Bayesian networks, without the need to derive application-specific update equations"

In this section, we review the *Variational Message Algorithm* or *VMA* as a variational Bayes application to Bayesian networks where the exponential family is considered using a message passing procedure between the nodes of a graphical model.

The full set of variables is $\mathcal{X} = X_1 \dots, X_N$, where we are considering both hidden variables $\mathcal{H} = \{H_1, \dots, H_J\}$ and visible ones \mathcal{V} . A variational distribution Q in the mean-field family

$$Q(\mathcal{H}) = \prod_{j=1}^J Q(h_j).$$

The optimized factor is given by

$$\log Q^{new}(h_n) = \mathbb{E}_{Q_{\setminus j}} \left[\log P(\mathcal{V}, \mathcal{H}) \right] + \text{const.}$$

Using the bayesian network structure, the update is given by

$$\log Q^{new}(h_j) = \mathbb{E}_{Q_{\setminus j}} \left[\sum_{n=1}^N \log P(x_n \mid pa(x_n)) \right] + \text{const.}$$

Any term that does not depend on H_j will be constant under its expectation, these leaves the terms $P(h_j \mid pa(h_j))$ and the conditionals of all its children:

$$\log Q^{new}(h_j) = \mathbb{E}_{Q_{\setminus j}} \left[\log P(h_j \mid pa(h_j)) \right] + \sum_{X_k \in ch(H_j)} \mathbb{E}_{Q_{\setminus j}} \left[\log P(x_k \mid pa(x_k)) \right] + \text{const.}$$

This shows how the update of a hidden variable only depends on its Markov blanket. The optimization of Q_j is therefore computed as the sum of a term involving H_j and its parent

nodes, along with a term for each children. This terms can be interpreted as "messages" from the corresponding nodes.

The exact form of the messages will depend on the functional form of the conditional distributions in the model. Important simplifications to the variational update equations occur when the conditional distribution of a node given its parents is in the exponential family.

The following notation $pa_X = pa(X)$ will be used to give clearance to the notation.

Consider a variable X and $Y \in pa_X$, such as Y is a hidden variable. Then,

$$P(y | pa_y) = h_Y(pa_Y) \exp \left(\eta_Y(pa_Y)^T T_Y(y) - \psi_Y(pa_Y) \right).$$

$$P(x | pa(x)) = h_X(pa_X) \exp \left(\eta_X(pa_X)^T T_X(x) - \psi_X(pa_X) \right).$$

The distribution $P(Y | pa_Y)$ can be though as a prior over Y , and $P(X | pa_X)$ as a (contribution to) likelihood of Y .

Conjugacy requires that these two conditionals have the same functional form with respect to Y , so the latter has to be rewritten in terms of $T_Y(y)$ by defining functions η_{XY} and λ as

$$\log P(x | y, cp_Y) = \eta_{XY}(x, cp_Y)^T T_Y(y) + \lambda(x, cp_Y),$$

where $cp_Y = pa_X - \{Y\}$.

Example 6. If X is Gaussian distributed with mean Y and variance β , the log conditional is

$$\log P(x | y, \beta) = \begin{pmatrix} \beta y \\ -\beta/2 \end{pmatrix}^T \begin{pmatrix} x \\ x^2 \end{pmatrix} + \frac{1}{2} (\log \beta - \beta Y^2 - \log 2\pi).$$

We may rewrite it the conditional as

$$\log P(x | y, \beta) = \begin{pmatrix} \beta x \\ -\beta/2 \end{pmatrix}^T \begin{pmatrix} y \\ y^2 \end{pmatrix} + \frac{1}{2} (\log \beta - \beta x^2 - \log 2\pi),$$

where

$$\eta_{XY}(x, \beta) = \begin{pmatrix} \beta x \\ -\beta/2 \end{pmatrix}^T, \quad T_Y(y) = \begin{pmatrix} y \\ y^2 \end{pmatrix}.$$

From this results, it can be seen that $\log P(x | y, cp_Y)$ is linear in $T_X(x)$ and $T_Y(y)$, and, by the same reasoning, linear in any sufficient statistic of any parent of X . This is a general result for any variable X in this kind of models: *For any variable X in a conjugate-exponential model, the log conditional under its parents must be multi-linear of the statistics of X and its parents.*

Returning to the variational update for a node Y :

$$\log Q^{new}(y) = \mathbb{E} \left[\log P(y | pa_Y) \right] = \sum_{X_k \in ch(Y)} \mathbb{E} \left[\log P(x_k | pa_{X_k}) \right] + \text{const.},$$

where the expectations are over the variational distribution of all other hidden variables. All the expectations in the right hand side can be calculated in terms of $T_Y(y)$:

$$\begin{aligned}\log Q^{new}(y) &= \mathbb{E} \left[\log(h_Y(pa_y)) + \boldsymbol{\eta}_Y(pa_y)^T T_Y(y) + \psi_Y(pa_y) \right] \\ &\quad + \sum_{X_k \in ch(Y)} \mathbb{E} \left[\boldsymbol{\eta}_{X_k, Y}(x_k, cp(x_k))^T T_Y(y) + \lambda_k(x_k, cp(x_k)) \right] + \text{const.} \\ &= \left[\mathbb{E} \left[\boldsymbol{\eta}_Y(pa_Y)^T \right] + \sum_{X_k \in ch(Y)} \mathbb{E} \left[\boldsymbol{\eta}_{X_k, Y}(x_k, cp(x_k))^T T_Y(y) \right] \right]^T T_Y(y) \\ &\quad + \log h(pa_y) + \text{const.}\end{aligned}$$

It follows that $Q^{new}(Y)$ is in the exponential family of the same form as $P(y \mid pa_y)$ but with parameter function

$$\boldsymbol{\eta}_Y^{new} = \mathbb{E} \left[\boldsymbol{\eta}_Y(pa_y) \right] + \sum_{X_k \in ch(Y)} \mathbb{E} \left[\boldsymbol{\eta}_{X_k, Y}(X_k, cp(x_k)) \right].$$

As the expectations of $\boldsymbol{\eta}_Y$ and $\boldsymbol{\eta}_{X_k, Y}$ are multilinear functions of the expectations of the statistic functions of their corresponding variables, it is possible to reparameterize these functions in terms of these expectations

$$\begin{aligned}\bar{\boldsymbol{\eta}}_Y(\{\mathbb{E}[T_{X_k}(x_k)]\}_{X_k \in pa_Y}) &= \mathbb{E} \left[\boldsymbol{\eta}_Y(pa_Y) \right] \\ \bar{\boldsymbol{\eta}}_{X_k, Y}(\mathbb{E}[T_{X_k}(x_k)], \{\mathbb{E}[T_{X_j}(x_j)]\}_{X_j \in cp_{X_k}}) &= \mathbb{E} \left[\boldsymbol{\eta}_{X_k, cp_{X_k}}(x_k, cp_{X_k}) \right]\end{aligned}$$

15.1 VARIATIONAL MESSAGE PASSING ALGORITHM

The message from a parent node Y to a child node X is the expectation under Q of its statistic vector

$$\mathbf{m}_{Y \rightarrow X} = \mathbb{E} \left[T_Y(y) \right].$$

The message from a child node X to a parent node Y is:

$$\mathbf{m}_{X \rightarrow Y} = \boldsymbol{\eta}_{\bar{X}, Y}(\mathbb{E} [T_X(x)], \{\mathbf{m}_{X_k \rightarrow X}\}_{X_k \in cp_Y})$$

which relied on having received all messages from all the co-parents. If a node X is observed, the messages defined above are defined as $T_A(a)$ instead of $\mathbb{E} [T_A(a)]$.

Example 7. If X is Gaussian distributed and Y, β are its parents, the messages are:

$$\mathbf{m}_{X \rightarrow Y} = \begin{pmatrix} \mathbb{E} [\beta] \mathbb{E} [X] \\ -\mathbb{E} [\beta] / 2 \end{pmatrix}, \quad \mathbf{m}_{X \rightarrow \beta} = \begin{pmatrix} -\frac{1}{2} \left(\mathbb{E} [x^2] - 2\mathbb{E} [x] \mathbb{E} [y] + \mathbb{E} [y^2] \right) \\ \frac{1}{2} \end{pmatrix}.$$

And the messages from X to any of its child nodes is

$$\begin{pmatrix} \mathbb{E} [\beta] \mathbb{E} [x] \\ -\mathbb{E} [x^2] \end{pmatrix}.$$

When a node Y has received all messages from its parents and children, we can compute the updated parameter $\tilde{\eta}_Y^{new}$ as

$$\tilde{\eta}_Y^{new} = \tilde{\eta}_Y(\{\mathbf{m}_{X_k \rightarrow Y}\}_{X_k \in pa_Y}) + \sum_{X_k \in ch_Y} \mathbf{m}_{X_k \rightarrow Y}.$$

Winn & Bishop (2005)

Part V

COMMONLY STUDIED LATENT VARIABLE MODELS

GAUSSIAN MIXTURE

A Gaussian mixture model (figure 15) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. We will discuss another mixture example with the EM algorithm in Section 14.2.

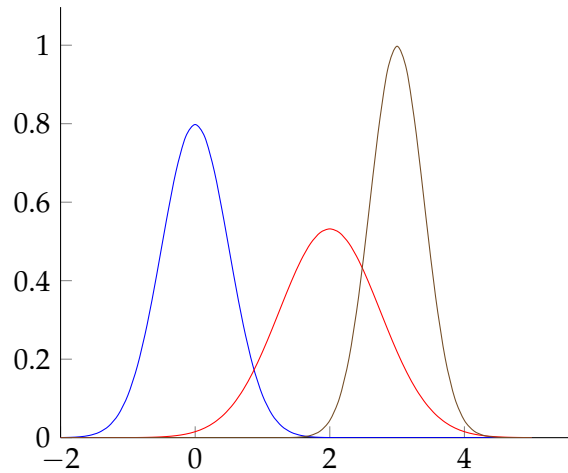


Figure 14: One-dimensional Gaussian mixture with 3 clusters.

The following elements are being considered (using [Bishop \(2006\)](#) notation):

- K mixture components and N observations.
- A set of i.i.d real valued random variables $\mathbf{X} = \{X_1, \dots, X_N\}$ and a corresponding set of observations $\mathbf{x} = \{x_1, \dots, x_n\}$.
- The cluster assignment latent variables $\mathbf{Z} = \{Z_1, \dots, Z_N\}$, where each z_n indicates the cluster to which x_n belongs.
- We choose a Dirichlet distribution over the mixing coefficients $\boldsymbol{\pi}$

$$\boldsymbol{\pi} \sim \text{Symmetric-Dirichlet}(\alpha_0) \implies P(\boldsymbol{\pi}) \propto \prod_{k=1}^K \pi_k^{\alpha_0-1}$$

The hyper-parameter α_0 is the effective prior of each mixture component. Then $\pi = \{\pi_1, \dots, \pi_K\}$ are the mixture weights, i.e, prior probability of a particular component k .

$$P(z | \pi) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{\mathbb{I}[z_n=k]} \implies (Z_n | \pi) \sim \text{Categorical}(\pi)$$

- $\mu = \{\mu_1, \dots, \mu_K\}$ and $\Lambda = \{\Lambda_1, \dots, \Lambda_K\}$ are the distribution parameters of each observation full conditional

$$(X | Z, \mu, \Lambda) \sim \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(\mu_k | \Lambda_k)^{\mathbb{I}[z_n=k]}$$

- The prior governing μ and Λ is an independent Gaussian, Inverse-Gamma distribution with hyper parameters m_0, β_0, w_0 and v_0 :

$$P(\mu, \Lambda) = P(\mu | \Lambda)P(\Lambda) \text{ where } \begin{cases} (\mu | \Lambda) \sim \prod_{k=1}^K \mathcal{N}(m_0, \beta_0 \Lambda_k) \\ \Lambda \sim \prod_{k=1}^K \text{Inverse-Gamma}(w_0, v_0) \end{cases}$$

The joint probability factorizes as

$$P(x, z, \pi, \mu, \Lambda) = P(x | z, \mu, \Lambda)P(z | \pi)P(\pi)P(\mu | \Lambda)P(\Lambda).$$

[Bishop \(2006\)](#) gives the explicit update for the CAVI algorithm.

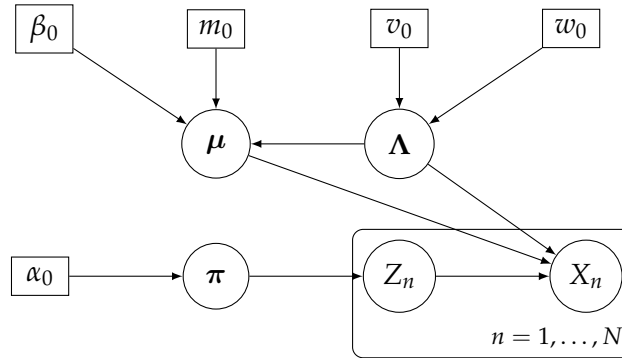


Figure 15: Gaussian mixture model. Squares represent hyper-parameters.

LATENT DIRICHLET ALLOCATION

Latent Dirichlet allocation or *LDA* is a conditionally conjugate model (figure 16) in natural language processing that allows set of observations to be explained by unobserved groups that explain why some parts of the data are similar.

For example, observations may be words in a document, which is a mixture of a small number of topics and each word's presence is attributable to one of the document's topics. Learning corresponds to extract information as the set of topics, their associated word probabilities, the topic of each word, and the particular topic mixture of each document.

The considered elements are (using Hoffman *et al.* (2013) and Blei *et al.* (2003) notation):

- K number of topics, V number of words in the vocabulary, M number of documents, N_d number of words in document d and N total number of words.
- $\beta = \{\beta_1, \dots, \beta_K\}$, where β_k is the distribution of words in topic k . Each component $\beta_{k,n}$ is the probability of the n^{th} word in topic k .
- Each document d is associated with a vector of topic proportions θ_d , which is a $K - 1$ simplex. Then each component $\theta_{d,k}$ is the probability of topic k in document d . Denote $\theta = \{\theta_1, \dots, \theta_K\}$.
- Each word in each document is assumed to be related with a single topic. The variable $Z_{d,n}$ indexes the topic of the n^{th} word in the d^{th} document.

LDA model assumes that each document is generated with the following generative process:

1. Draw topics from a Dirichlet distribution, for each $k = 1, \dots, K$:

$$P(\beta_k) = \frac{1}{B(\eta)} \prod_{v=1}^V \beta_{k,v}^{\eta_v-1} \implies \beta_k \sim \text{Symmetric-Dirichlet}_V(\eta)$$

2. For each document $d = 1, \dots, D$:

- a) Draw topic proportions,

$$P(\theta_d) = \frac{1}{B(\alpha)} \prod_{k=1}^K \theta_{d,k}^{\alpha_k-1} \implies \theta_d \sim \text{Symmetric-Dirichlet}_K(\alpha)$$

- b) For each word in the document $n = 1, \dots, N_d$:

i. Draw a topic,

$$P(z_{d,n} \mid \theta_d) = \prod_{k=1}^K \theta_{d,k}^{\mathbb{I}[z_{d,n}=k]} \implies (Z_{d,n} \mid \theta_d) \sim \text{Categorical}(\theta_d).$$

ii. Draw a word from the topic

$$P(w_{d,n} \mid z_{d,n}, \beta) = \prod_{v=1}^V \beta_{z_{d,n},v}^{\mathbb{I}[w_{d,n}=v]} \implies (W_{d,n} \mid Z_{d,n}, \beta) \sim \text{Categorical}(\beta_{Z_{d,n}})$$

The joint probability distribution is then

$$\begin{aligned} P(\theta, z, w, \beta) &= P(\beta) \prod_{d=1}^D P(\theta_d \mid \alpha) \prod_{n=1}^{N_d} P(z_{d,n} \mid \theta_d) P(w_{d,n} \mid z_{d,n}, \beta) \\ &= \left(\prod_{k=1}^K P(\beta_k) \right) \prod_{d=1}^D P(\theta_d) \prod_{n=1}^{N_d} P(z_{d,n} \mid \theta_d) P(w_{d,n} \mid z_{d,n}, \beta) \end{aligned}$$

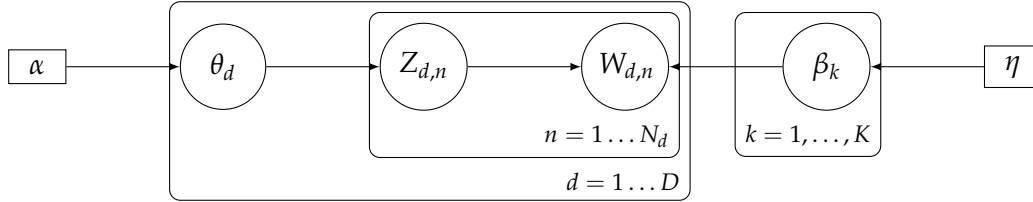


Figure 16: Latent Dirichlet Allocation model. Squares represent hyper-parameters

We can compute the posterior distributions for this model, starting with the complete conditional distribution of the local hidden variables. They only depend on other variables in the local context (same document) and the global variables

$$\begin{aligned} P(z_{d,n} \mid w_{d,n}, \theta_d, \beta) &= \frac{P(z_{d,n}, w_{d,n}, \theta_d, \beta)}{\int_{z_{d,n}} P(z_{d,n}, w_{d,n}, \theta_d, \beta)} = \frac{P(z_{d,n} \mid \theta_d) P(w_{d,n} \mid z_{d,n}, \beta)}{\int_{z_{d,n}} P(z_{d,n} \mid \theta_d) P(w_{d,n} \mid z_{d,n}, \beta)} \\ &= \frac{\prod_{k=1}^K \theta_{d,k}^{\mathbb{I}[z_{d,n}=k]} \prod_{v=1}^V \beta_{z_{d,n},v}^{\mathbb{I}[w_{d,n}=v]}}{\int_{z_{d,n}} \prod_{k=1}^K \theta_{d,k}^{\mathbb{I}[z_{d,n}=k]} \prod_{v=1}^V \beta_{z_{d,n},v}^{\mathbb{I}[w_{d,n}=v]}} = \frac{\theta_{d,z_{d,n}} \beta_{z_{d,n},w_{d,n}}}{\sum_{k=1}^K \theta_{d,k} \beta_{k,w_{d,n}}} \end{aligned}$$

Naming

$$\gamma_{d,n} = \frac{\theta_{d,z_{d,n}} \beta_{z_{d,n},w_{d,n}}}{\sum_{k=1}^K \theta_{d,k} \beta_{k,w_{d,n}}}$$

and $\gamma = \{\gamma_{d,n}\}_{d=1,\dots,D} \ n=1,\dots,N_d$, we get

$$(Z_{d,n} \mid w_{d,n}, \theta_d, \beta) \sim \text{Categorical}(\gamma).$$

On the other hand, the complete conditional of the topic proportions θ_d is only affected by the topic appearances, since $z_{d,n}$ is an indicator vector, the k^{th} element of the parameter to

this Dirichlet is the sum of the hyperparameter α and the number of words assigned to topic k in document d :

$$\begin{aligned} P(\theta_d \mid \mathbf{z}_d, \mathbf{w}_d, \beta) &= \frac{P(\theta_d, \mathbf{z}_d, \mathbf{w}_d, \beta)}{\int_{\theta_d} P(\theta_d, \mathbf{z}_d, \mathbf{w}_d, \beta)} = \frac{P(\theta_d) \prod_{n=1}^{N_d} P(z_{d,n} \mid \theta_d)}{\int_{\theta_d} P(\theta_d) \prod_{n=1}^{N_d} P(z_{d,n} \mid \theta_d)} \\ &\propto \prod_{k=1}^K \theta_{d,k}^{\alpha-1} \prod_{n=1}^{N_d} \theta_{d,k}^{\mathbb{I}[z_{d,n}=k]} = \prod_{k=1}^K \theta_{d,k}^{\alpha-1 + \sum_{n=1}^{N_d} \mathbb{I}[z_{d,n}=k]}. \end{aligned}$$

The complete conditional depends only on the topic assignments

$$(\theta_d \mid \mathbf{z}_d) \sim \text{Dirichlet}_K \left(\alpha + \sum_{n=1}^{N_d} \mathbb{I}[z_{d,n} = 1], \dots, \alpha + \sum_{n=1}^{N_d} \mathbb{I}[z_{d,n} = K] \right).$$

In words, the probability of a topic in document d is updated with the number of times that topic appears in the document. The complete conditional of a topic depends on the words and topics assignments of the entire collection

Using a similar reasoning, the words distribution in a topic k , β_k , is updated with the number of appearances in all documents of the given topic.

$$(\beta_k \mid \mathbf{w}, \mathbf{z}) \sim \text{Dirichlet}_V \left(\eta + \sum_{d=1}^D \sum_{n=1}^{N_d} \mathbb{I}[z_{d,n} = k] \mathbb{I}[w_{d,n} = 1], \dots, \eta + \sum_{d=1}^D \sum_{n=1}^{N_d} \mathbb{I}[z_{d,n} = k] \mathbb{I}[w_{d,n} = V] \right).$$

PROBABILISTIC PRINCIPAL COMPONENTS ANALYSIS

LVMs have usually been restricted to the exponential family because, in this case, inference is feasible. But recent advances in variational inference have enabled LVMs to be extended with neural networks. For example, *Variational Auto-encoders* or VAE (Kingma & Welling (2013)) are the most influential models combining both concepts.

VAEs extent the classical technique of *principal components analysis* for data representation in lower-dimensional spaces. Suppose we have a D -dimensional representation of a data point x and z is its latent K -dimensional representation ($K < D$). PCA computes an affine transformation \mathbf{W} , represented by a $K \times D$ matrix.

A probabilistic view of PCA can be modeled with an LVM (Tipping & Bishop (1999)), with the following elements:

- $\mathbf{X} = \{X_1, \dots, X_N\}$ i.i.d \mathbb{R}^D -valued random variables and the corresponding observations $\mathbf{x} = \{x_1, \dots, x_N\}$.
- $\mathbf{Z} = \{Z_1, \dots, Z_N\}$ i.i.d latent \mathbb{R}^K -valued random variables, where Z_n models the K -dimensional representation of x_n .
- A global latent $K \times D$ -dimensional random variable \mathbf{W} .
- A noise hyper-parameter σ^2 .

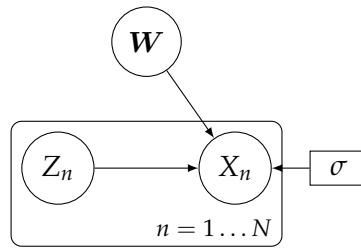


Figure 17: Probabilistic PCA model

We assume the priors are normally distributed:

$$Z_n \sim N_K(0, I) \quad \forall n = 1, \dots, N \quad \text{and} \quad \mathbf{W} \sim N_{K \times D}(0, I).$$

The data points are considered generated via a projection,

$$(X_n | Z_n, \mathbf{W}) \sim N(\mathbf{W}^T Z_n, \sigma^2 I) \quad \forall n = 1, \dots, N.$$

The probabilistic model extends the classical one in the way that the latter assumes the noise is infinitesimally small, i.e., $\sigma^2 \rightarrow 0$. The *expectation-maximization algorithm* (Section 14.1) is commonly used to solve this variational inference problem.

18.1 ARTIFICIAL NEURAL NETWORKS

An *artificial neural network* or ANN with L hidden layers can be seen as a deterministic non-linear function f parameterized by a set of matrix $\mathbf{W} = \{\mathbf{W}_0, \dots, \mathbf{W}_L\}$ and non-linear activation functions $\{r_0, \dots, r_L\}$. Given an input x the output y is calculated as

$$h_0 = r_0(\mathbf{W}_0^T x), \quad \dots, \quad h_l = r_l(\mathbf{W}_l^T h_{l-1}) \quad \dots \quad y = r_L(\mathbf{W}_L^T h_{L-1}).$$

Deep neural networks or DNNs are ANNs where the number of hidden layers is higher. Commonly, any neural network with more than 2 hidden layers is considered deep. Given a dataset $\{(x_1, y_1), \dots, (x_N, y_N)\}$ and a loss function $l(y, y^*)$ that defines how well the output $y^* = f_{\mathbf{W}}(x)$ returned by the model matches the real output y , learning reduces to the optimization problem

$$\mathbf{W}^{opt} = \arg \min_{\mathbf{W}} \sum_{n=1}^N l(y_n, f_{\mathbf{W}}(x_n)).$$

This problem is usually solved by applying a variant of the stochastic gradient descent method, which involves the computation of the gradient of the loss function with respect to the parameters of the network. The algorithm for computing this gradient is known as *back-propagation*, which is based on a recursive application of the chain-rule of derivatives. This can be implemented using the computational graph on the network.

The main idea of a computational graph is to express a deterministic function, as is the case of a neural network, using an acyclic directed graph. It is composed of input, output and operation nodes, where model data and parameters are shown as input nodes.

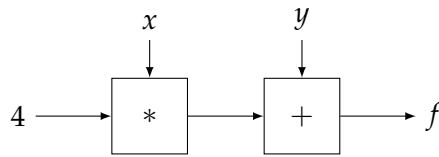


Figure 18: Computational graph example of function $f(x, y) = 4x + y$

18.2 NON-LINEAR PCA

Non-linear PCA or NLPCA, extends the classical PCA where the relation between the low dimensional space and the observed data is governed by a DNN instead of a linear transformation. It can be seen as a non-linear probabilistic PCA model.

The model is quite similar to the one presented for the PCA, the difference comes from the conditional distribution of X , that depends on Z through a fully-connected ANN with a single hidden layer.

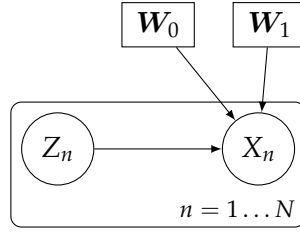


Figure 19: Non-linear PCA model. \mathbf{W}_0 and \mathbf{W}_1 together with two activation functions r_0, r_1 represent an ANN.

The prior of the latent variable is a centered Gaussian

$$Z_n \mid N(0, I) \quad \forall n \in 1, \dots, N$$

Let D be the dimension of the data X and K the dimension of the hidden variable Z . Let f a single hidden layer ANN with input dimension Z and output dimension D . Where the output is the mean value of the normal distribution under X .

As we are considering a single hidden layer, let \mathbf{W}_0 and \mathbf{W}_1 be the matrixes governing that ANN and r_0, r_1 the activation functions, the ANN f is

$$f(z_n) = r_1(\mathbf{W}_1(r_0(\mathbf{W}_0(z_n)))) ,$$

and the data-points are then generated as

$$(X_n \mid Z_n) \sim N(f(Z_n), I) \quad \forall n \in 1, \dots, N.$$

Where no noise is being considered this time.

18.3 VARIATIONAL AUTO-ENCODER

Similarly to the models PCA and NLPCA, a *variational autoencoder* or VAE, allows to perform dimensionality reduction. However a VAE will contain a neural network in the P model (decoder) and another one in the variational model Q (encoder).

The P model has the same structure as the nonlinear PCA. On the other hand, the distribution Q is defined with a reverse ANN, with input dimension D and output dimension K .

The neural networks can be defined to give an output of double the dimension, that is, the encoder would give a point in with $2K$ components, so the first K are used for the mean and the last K for the variance of the normal distribution.

Part VI

CASE STUDY

INFERPY USAGE

InferPy (Cózar *et al.* (2019)) is a high-level API for probabilistic modeling written in Python and capable of running on top of Edward and Tensorflow. InferPy's API is strongly inspired by Keras and it has a focus on enabling flexible data processing, easy-to-code probabilistic modeling, scalable inference and robust model validation.

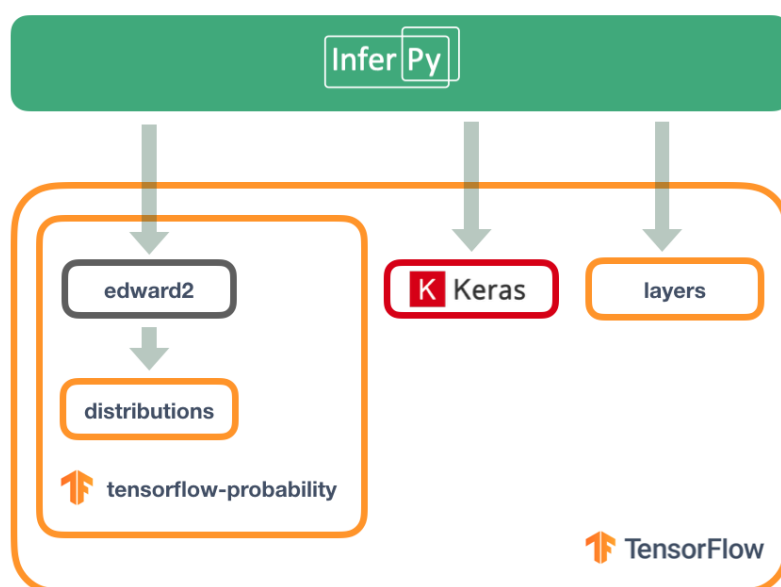


Figure 20: InferPy architecture.

The main features of InferPy are:

- Allows a simple definition over probabilistic models containing or not neural networks.
- All models that can be defined using Edward2 can also be defined in InferPy, whose probability distributions are mainly inherited from tensorflow-probability.
- All the models parameters can be defined using the standard Python types (Numpy compatibility).
- InferPy relies on top of Edward's inference engine, therefore, it includes all the inference algorithms available in that package.
- An additional advantage of using Edward and TensorFlow as inference engine is that all the parallelization details are hidden to the user. Moreover, the same code will run either in CPUs or GPUs.

19.1 INSTALLATION

InferPy has the following requirements:

- Python ≥ 3.5 and < 3.8 .
- Tensorflow $\geq 1.12.1$ and < 2.0 .
- Tensorflow-probability 0.7.0.
- NetworkX $\geq 2.2.0$ and < 3.0 .

InferPy is available at Pip and can be installed with the following command:

```
pip install inferpy
```

19.2 USAGE GUIDE WITH PCA

In this section we are constructing a PCA inference model with InferPy. Programming and Python knowledge are assumed, the guidance will be over the API usage.

Let us remember the variables we had in the model, consider X_1, \dots, X_N i.i.d \mathbb{R}^D variables, these are the observed ones. Then there are Z_1, \dots, Z_N hidden variables that correspond to the hidden representation of the data in \mathbb{R}^K . There is also a global hidden variable \mathbf{W} modeling the linear transformation from one space to the other. We are also considering another global variable \mathbf{W}_0 that will allow the model to generate non centered points.

The hidden variables prior is a centered gaussian and the data is supposed to be generated as

$$X_n \mid z_n, \mathbf{w}, \mathbf{w}_0 \sim \mathcal{N}(\mathbf{w}^T z_n + \mathbf{w}_0, I)$$

No noise is being considered in this example.

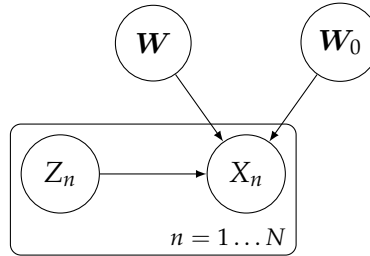


Figure 21: Probabilistic PCA model. No noise considered.

The database we are using is Mnist, which is a largely used dataset on a set of handwritten digits.

The full model definition would be the following:

```
@inf.probmodel
def pca(k,d):
    w = inf.Normal(loc=tf.zeros([k,d]),
                  scale=1, name="w")           # shape = [k,d]
    w0 = inf.Normal(loc=tf.zeros([d]),
                  scale=1, name="w0")         # shape = [d]
    with inf.datamodel():
```

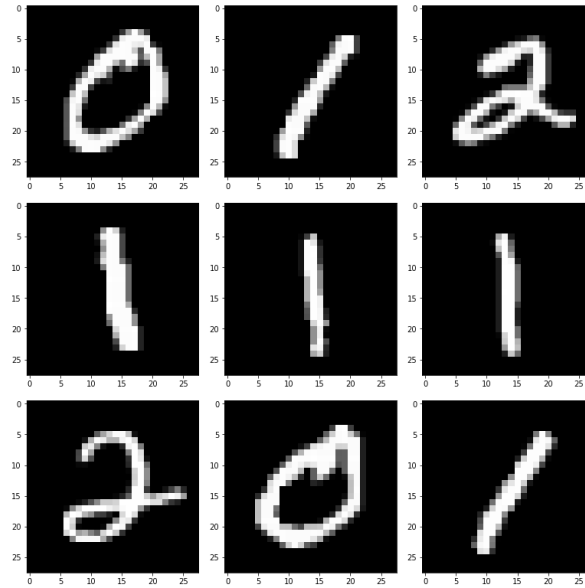



Figure 22: Mnist dataset example.

```
z = inf.Normal(tf.zeros([k]), 1, name="z")          # shape = [N,k]
x = inf.Normal(np.dot(z,w) + w0, 1, name="x")        # shape = [N,d]
```

Models are defined by decorating a Python functions with `@inf.probmodel`. In this case we are defining a model with 2 parameters:

```
@inf.probmodel
def pca(k, d):
```

Now we declare the global hidden variables, we use the distributions inside InferPy which is assumed to be imported as `inf`, in this case `inf.Normal`. This distribution has three arguments, the mean (`loc`), the standard deviation (`scale`), and the name of the variable, this last parameter is needed as its how this model and the variational will communicate.

If either the mean or the standard deviation are given as an array, a constant value on the other parameter would be interpreted as a constant array of the corresponding size.

```
w = inf.Normal(loc=tf.zeros([k,d]), scale=1, name="w")
```

Here we are defining a $K \times D$ Gaussian distributed variable with mean 0 and standard deviation 1, named "w".

We have reached the point where we have to define a set of i.i.d random variables, one for each observation, InferPy gives a explicit syntaxis for this, variables defined inside with `inf.datamodel(size)` are replicated and i.i.d of each other. The size can be omitted as it is calculated from the dataset. This is how we declare the hidden local variables in our model

```
with inf.datamodel():
    z = inf.Normal(tf.zeros([k]), 1, name="z")
```

Our generative model is now defined, now we might define the variational one:

```
@inf.probmodel
def q(k,d):
```

```

qw_loc = inf.Parameter(tf.zeros([k,d]), name="qw_loc")
qw_scale = tf.math.softplus(inf.Parameter(tf.ones([k,d]), name="qw_scale"))
qw = inf.Normal(qw_loc, qw_scale, name="w")

qw0_loc = inf.Parameter(tf.ones([d]), name="qw0_loc")
qw0_scale = tf.math.softplus(inf.Parameter(tf.ones([d]), name="qw0_scale"))
qw0 = inf.Normal(qw0_loc, qw0_scale, name="w0")

with inf.datamodel():
    qz_loc = inf.Parameter(np.zeros([k]), name="qz_loc")
    qz_scale = tf.math.softplus(inf.Parameter(tf.ones([k]), name="qz_scale"))
    qz = inf.Normal(qz_loc, qz_scale, name="z")

```

In this case, among the variables, we must define each parameter. Let us focus on "w" whose variational distribution is a Gaussian distribution, we must define two parameters, one for the mean and other for the standard deviation.

```

qw_loc = inf.Parameter(tf.zeros([k,d]), name="qw_loc")
qw_scale = tf.math.softplus(inf.Parameter(tf.ones([k,d]), name="qw_scale"))
qw = inf.Normal(qw_loc, qw_scale, name="w")

```

As the standard deviation value must not reach a negative value, we use a `softplus` function to smoothly approximate a rectifier function.

The same argument is applied to the rest of variables.

When both models are defined and instantiated in a variable, we need to create an inference object:

```
VI = inf.inference.VI(qmodel)
```

By default, the inference object uses el ELBO and loss function and uses `AdamOptimizer` from TensorFlow. An amount of epochs for the fitting might be set.

Given a training dataset `X_train`, the model is trained indicating which set of observations corresponds to each observed variable using the following syntax:

```
pmodel.fit({"x": X_train}, VI)
```

Once the model is trained, a variable posterior can be taken as

```
pmodel.posterior("z").parameters()
```

We can also take a sample from the parameter posterior

```
post = {"z": pmodel.posterior("z").sample() }
```

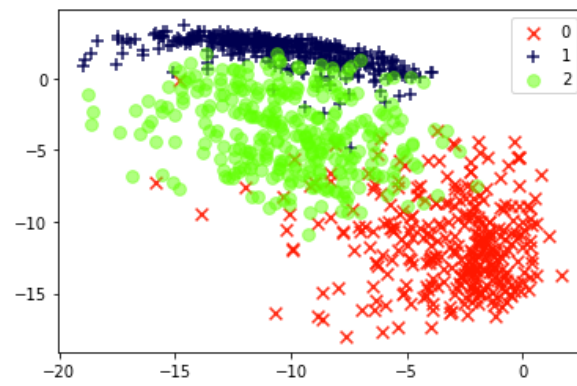


Figure 23: PCA posterior sample of MNIST.

GAUSSIAN MIXTURE

BIBLIOGRAPHY

- Barber, David. 2007. *Bayesian Reasoning and Machine Learning*. Cambridge University Press.
- Bishop, Christopher M. 2006. *Pattern recognition and machine learning*. springer.
- Blei, David M. 2014. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, **1**, 203–232.
- Blei, David M, Ng, Andrew Y, & Jordan, Michael I. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, **3**(Jan), 993–1022.
- Blei, David M, Kucukelbir, Alp, & McAuliffe, Jon D. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association*.
- Cózar, Javier, Cabañas, Rafael, Salmerón, Antonio, & Masegosa, Andrés R. 2019. InferPy: Probabilistic Modeling with Deep Neural Networks Made Easy. *arXiv preprint arXiv:1908.11161*.
- Do, Chuong B, & Batzoglou, Serafim. 2008. What is the expectation maximization algorithm? *Nature biotechnology*, **26**(8), 897–899.
- Farrow, Malcolm. 2008. MAS3301 Bayesian Statistics.
- Hoffman, Matthew D, Blei, David M, Wang, Chong, & Paisley, John. 2013. Stochastic variational inference. *The Journal of Machine Learning Research*, **14**(1), 1303–1347.
- Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, & Saul, Lawrence K. 1999. An introduction to variational methods for graphical models. *Machine learning*.
- Kingma, Diederik P, & Welling, Max. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Koller, Daphne, & Friedman, Nir. 2009. *Probabilistic Graphical Models, Principles and Techniques*. The MIT Press.
- Pearl, Judea, & Dechter, Rina. 2013. Identifying Independences in Casual Graphs with Feedback.
- Shachter, Ross D. 2013. Bayes-Ball: The Rational Pastime.
- Tipping, Michael E, & Bishop, Christopher M. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **61**(3), 611–622.
- Wainwright, Martin J., & Jordan, Michael I. 2008. *Graphical Models, Exponential Families and Variational Inference*. Now Publishers Inc.
- Winn, John, & Bishop, Christopher M. 2005. Variational message passing. *Journal of Machine Learning Research*, **6**(Apr), 661–694.