Inference in TrueSkill

Lukáš Žilka (lukas@zilka.me)

30.5.2013

Abstract

We explain inference in the TrueSkill model in this article, with the aim of full understanding of the inference mechanisms. Inference is based on Expectation Propagation algorithm. This model and inference in it was originally proposed by [Herbrich et al., 2007].

1 Introduction

It is boring to play against an opponent that is too bad (we knew that before), and with advent of online gaming, it started to be a problem for more people that kill their time behind TV hooked-up to Xbox, because they wanted to kill the time preferably in an entertaining way. On the Xbox platform, people play in teams against each other. The result of each game is a ranking of the teams (i.e. we know the winner, looser and everything in between). The results of the matches are recorded. The question is: based on the recorded games, how to assign a score to each player, that represents the player's skill, so that we can rank the players, and then match ones with a similar skill-score. This score is called TrueSkill and this article will be about its inference, using a statistical model.

2 Model

The result of a game in the TrueSkill model is modelled in the following way:

$$\forall i \in Players : s_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{1}$$

$$\forall i \in Players : p_i \sim \mathcal{N}(s_i, \beta^2) \tag{2}$$

$$\forall k \in Teams : t_k \sim I(t_k = \sum_{i \in Team(k)} p_i) \tag{3}$$

$$p(\mathbf{r}|\{t_1, .., t_k\}) = p(t_{r_{(1)}} > t_{r_{(2)}} > ... > t_{r_{(k)}}) \tag{4}$$

$$\tag{5}$$

3 Inference

Inference in TrueSkill model is based on the Expectation Propagation [Minka, 2001] algorithm, that is a generalization of the sum-product algorithm [Kschischang et al., 2001].

4 Take-home Message

Gaussians rule! They are very powerful because you just need 2 parameters to represent the whole distribution, and consequentially you just update these two parameters when you need to update the distribution. For example, when you multiply/divide two Gausssians, the result is also a Gaussian:

- Product of two Gaussian distributions over the same variable is a Gaussian distribution.

$$\mathcal{N}(x;\mu_1,\sigma_1^2)\cdot\mathcal{N}(x;\mu_2,\sigma_2^2)=\mathcal{N}(x;\mu,\sigma^2) \tag{6}$$

- Product of two Gaussian distributions over different variables is a multinomial Gaussian distribution with symmetric covariance matrix:

$$\mathcal{N}(x;\mu_1,\sigma_1^2)\cdot\mathcal{N}(y;\mu_2,\sigma_2^2)=\mathcal{N}(\begin{bmatrix}x\\y\end{bmatrix};\begin{bmatrix}\mu_1\\\mu_2\end{bmatrix},\begin{bmatrix}\sigma_1^2 & 0\\0 & \sigma_2^2\end{bmatrix}) \tag{7}$$

- If two random variables are from Gaussians, the sum of those variables is also a Gaussian.

$$X\sim\mathcal{N}(x;\mu_1,\sigma_1^2) \tag{8}$$

$$Y\sim\mathcal{N}(x;\mu_2,\sigma_2^2) \tag{9}$$

$$X+Y\sim\mathcal{N}(x;\mu_1+\mu_2,\sigma_1^2+\sigma_2^2) \tag{10}$$

$$\tag{11}$$

- Sum of two Gaussian distributions is NOT a Gaussian.

$$\mathcal{N}(x;\mu_1,\sigma_1)+\mathcal{N}(x;\mu_2,\sigma_2)\neq\mathcal{N}(x;\cdot) \tag{12}$$

- Integration of two Gaussian that are over different variables, where one ot them is the mean parameter for the other, can be turned into a convolution, and the result is again a Gaussian. Convolution is an integral of the form $\int f(t-x)g(x)\mathrm{d}t$.

$$p(x)=\int\mathcal{N}(x;\mu_0,\sigma_0^2)\mathcal{N}(y;x,\sigma^2)\mathrm{d}x= \tag{13}$$

  Following the equation for the Gaussian distribution, we can take out the x from the mean of the second Gaussian and put it to its argument.

$$=\int\mathcal{N}(x;\mu_0,\sigma_0^2)\mathcal{N}(y-x;0,\sigma^2)\mathrm{d}x \tag{14}$$

$$\tag{15}$$

  Now the form somehow resembles the form of convolution given above: $\int f(x)g(y-x)\mathrm{d}x$, and consequentially, we can use the result for Gaussian convolution. Its derivation can be found in [Moser, 2012].

$$=\mathcal{N}(x;\mu_0,\sigma_0^2+\sigma^2) \tag{16}$$

$$\tag{17}$$

5 References

References

[Herbrich et al., 2007] Herbrich, R., Minka, T., and Graepel, T. (2007). Trueskill$^{\mathrm{TM}}$: A bayesian skill rating system. *Advances in Neural Information Processing Systems*, 19:569.

[Kschischang et al., 2001] Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519.

[Minka, 2001] Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

[Moser, 2012] Moser, J. (2012). The Math Behind True Skill. Online: `http://dl.dropboxusercontent.com/u/1083108/Moserware/Skill/The%20Math%20Behind%20TrueSkill.pdf`.

Message passing in clique trees

June 5, 2013

- an alternative approach to Variable-Elimination
- manipulation of factors is a basic computational step
- clique tree: a global data structure for scheduling these operations
- all operations can be performed on normalized as well as unnormalized measures
  - the unnormalized measure (unnormalized joint probability): $\tilde{P_\Phi}(\mathcal{X}) = \sum_{\phi_i \in \Phi} \phi_i(\mathbf{X_i})$, where $\mathcal{X}$ is a set of all variables, Φ set of all factors and $\mathbf{X_i}$ set of variables in a scope of the factor ϕ_i

1 Message passing in clique trees $\Leftrightarrow$ Variable-Elimination

In this section I would like to show that message passing in clique trees is an approach how to compute marginals of a joint probability distribution, equivalent to variable-elimination.

1.1 Variable-Elimination algorithm

An algorithm to compute marginals from a joint probability distribution.

Example:

We would like to compute $P(C)$ for a distribution represented by a Markov Net in Figure (TODO).

Instead of computing normalized measure $P(C)$ we can put off the normalization at the very end and work with unnormalized measures ($\tilde{P}$). The marginal of C is simply a joint distribution after marginalizing out all variables except for C. Joint distribution is defined as a product of all factors.

$$\tilde{P}(C) = \sum_{A,B,D,E} \tilde{P}(A,B,C,D,E) = \sum_{A,B,D,E} \phi(A,B)\phi(B,C)\phi(B,E)\phi(C,D)\phi(C,E) =$$

We eliminate all variables except for C one after another. We pick a variable A to start with. We can group all factors which A participates in (in this case just $\phi(A,B)$), multiply them all into a single factor $\psi_1(A,B) = \phi(A,B)$ and marginalize out A by summing over values of A to get a new factor $\tau_1(B) = \sum_A \psi_1(A,B)$.

$$= \sum_{B,D,E} \phi(B,C)\phi(B,E)\phi(C,D)\phi(C,E)\sum_A \phi(A,B) = \sum_{B,D,E} \phi(B,C)\phi(B,E)\phi(C,D)\phi(C,E)\tau_1(B) =$$

We proceed by eliminating B, which again consists of a regroupin, multiplying to a new factor $\psi_2(B,C,E) = \phi(B,C)\phi(B,E)\tau_1(B)$ and marginalizing out B to get a new factor $\tau_2(C,E) = \sum_B \psi_1(B,C,E)$.

$$= \sum_{D,E} \phi(C,D)\phi(C,E)\sum_B \phi(B,C)\phi(B,E)\tau_1(B) = \sum_{D,E} \phi(C,D)\phi(C,E)\tau_2(C,E)$$

We will continue in the same manner until we eliminate all variables except for C. The resulting function of C is then an unnormalized marginal distribution $\tilde{P}(C)$.

1.2 Clique tree

We have to define a basic structure message passing processes on - a clique tree.

Definition Clique tree is an undirected tree such that:

- nodes are clusters of variables $C_i \subseteq \mathcal{X} = \{X_1, \ldots, X_n\}$
- edge between C_i and C_j is associated with sepset $S_{i,j} = C_i \cap C_j$
- Family preservation property (FPP): for each factor$\phi_k \in \Phi$, it is assigned to single cluster $C_{\alpha(k)}$, so that $Scope[\phi_k] \subseteq C_{\alpha(k)}$[1]
- Running intersection property (RIP): for each pair of clusters C_i and C_j and variable $X \in C_i \cap C_j$there exists a unique path between C_i and C_j for which all clusters and sepsets contain X

1.3 Variable-Elimination⇒Message passing in clique trees

We want to show that V-E produces a clique tree. Let us construct a clique tree as follows. The multiplied factor ψ_i produced in a single step of V-E forms a cluster $C_i = Scope(\psi_i)$. We connect clusters C_i and C_j by an edge if the factor τ_i produced by eliminating ψ_i is used in making of ψ_j.

Lemma The graph produced by V-E is a tree.

Proof Each intermediate factor ψ_i (representing a node) is used only once in computing τ_i (an edge) and never appears again.

[1] $Scope[\phi(X)] = X$, so it is a set of variables associated with a factor

Lemma The tree satisifies an FPP.

Proof Each original factor ϕ_k is used in construction of some ψ_i (a cluster $C_{\alpha(k)}$) and never reappears. All factors ϕ must be eliminated in V-E.

Lemma The tree satisfies an RIP.

Proof Suppose, we have clusters C,$C`$ and C_X and a variable X, which is in the scope of each of the three clusters. Furthermore, C_X is a cluster where X is eliminated (or the cluster that remains at the end of V-E).
C must take place earlier in the algorithm than C_X. X is eliminated in C_X, i.e. all factors containing X are multiplied into C_X and the result of summation does not contain X in its scope. Thus, X cannot appear after C_X is created.
By assumption, $X \in Scope(C)$. X is not eliminated in C, so $X \in Scope(\tau_C)$. A neighbor in the tree just multiplies τ_C. This happens for all clusters upstream from C until X is eliminated in C_X. Thus, X appears in all clusters between C and C_X.
The same holds for $C`$, so there exist a unique $X$ - path between $C$ and $C`$.

Corollary Variable-Elimination produces a clique tree.

1.4 Message passing in a clique tree

In the following we show a simplified version of MP algorithm, in which we calculate the unnormalized marginal $\tilde{P}(Scope(C_{root}))$ of all variables associated with a given cluster C (the root). Given this marginal, we can further compute a marginal of any variables fully contained in the $Scope(C)$ by means of V-E.

1. Select a root cluster C_{root} containing all variables the marginal that we search for consists of
2. Calculate initial potentials for all C_i:
$$\psi(C_i) = \prod_{\phi_j \in C_i} \phi_j$$
3. Pass message if possible, i.e. all incoming messages have been received (except for the one from the cluster we are passing to):
$$\delta_{i \to j} = \sum_{C_i \setminus S_{i,j}} \psi_i \prod_{k \in Nb_i \setminus \{j\}} \delta_{k \to i}$$
4. After the root has received all messages, calculate belief:
$$\beta_{root} = \psi_{root} \prod_{k \in Nb_{root}} \delta_{k \to root}$$

The calculated belief represents, as we show below:

$$\beta_{root} = \tilde{P}(Scope(C_{root})) = \sum_{\mathcal{X} \setminus Scope(C_{root})} \prod_{\phi} \phi$$

1.5 Message passing in clique trees $\Rightarrow$Variable-Elimination

Our aim is to show the correctness of the algorithm, that MP over a clique tree that statisfies FPP and RIP produces the same marginals as V-E as it is schematized by the last equation. We start with an auxilliary lemma.

Lemma X is eliminated when a message from C_i to C_j is sent. Then X does not appear anywhere in the tree on the C_j side of the edge $(i \leftrightarrow j)$.

Proof A consequence of RIP. Assume by contradicton there is a node C_k containing variable X on the C_j side of the edge $(i \leftrightarrow j)$. So C_j lies on the path from C_k (containing X) to C_i (containing X). However, C_i does not contain X, which violates RIP.

Let us introduce a notation used in the last proof. $\mathcal{F}_{<(i \leftrightarrow j)}$ is a set of all factors asociated with clusters on the C_i side of the edge. $\mathcal{V}_{<(i \leftrightarrow j)}$ is a set of variables that appear on the C_i side but not in the edge $(i \leftrightarrow j)$ itself.

Theorem Let $\delta_{i \to j}$ be a message from C_i to C_j. Then

$$\delta_{i \to j} = \sum_{\mathcal{V}_{<(i \leftrightarrow j)}} \prod_{\phi \in \mathcal{F}_{<(i \leftrightarrow j)}} \phi$$

Proof By induction from leafs to the root.

If C_i is a leaf, the equation follows from the definition of a message. If C_i is not a leaf and C_k, $k \in \{i_1, \ldots, i_m\} = Nb_i \setminus \{j\}$ are the neighbours of C_i (excluding C_j) then:

$$\begin{aligned} \mathcal{F}_{<(i \leftrightarrow j)} &= \mathcal{F}_{<(i_1 \leftrightarrow i)} \cup \ldots \cup \mathcal{F}_{<(i_m \leftrightarrow i)} \cup \mathcal{F}_i;\ \mathcal{F}_{<(i_1 \leftrightarrow i)} \cap \ldots \cap \mathcal{F}_{<(i_m \leftrightarrow i)} \cap \mathcal{F}_i = \emptyset \\ \mathcal{V}_{<(i \leftrightarrow j)} &= \mathcal{V}_{<(i_1 \leftrightarrow i)} \cup \ldots \cup \mathcal{V}_{<(i_m \leftrightarrow i)} \cup Y_i;\ \mathcal{V}_{<(i_1 \leftrightarrow i)} \cap \ldots \cap \mathcal{V}_{<(i_m \leftrightarrow i)} \cap Y_i = \emptyset \end{aligned}$$

where Y_i is a set of variables eliminated in C_i itself. Due to this property we can partition sums and products in the equation as follows:

$$\sum_{\mathcal{V}_{<(i \leftrightarrow j)}} \prod_{\phi \in \mathcal{F}_{<(i \leftrightarrow j)}} \phi = \sum_{Y_i} \Big(\sum_{\mathcal{V}_{<(i_m \leftrightarrow i)}} \ldots \Big(\sum_{\mathcal{V}_{<(i_1 \leftrightarrow i)}} \prod_{\phi \in \mathcal{F}_{<(i_1 \leftrightarrow i)}} \phi \Big) \ldots \prod_{\phi \in \mathcal{F}_{<(i_m \leftrightarrow i)}} \phi \Big) \prod_{\phi \in \mathcal{F}_i} \phi =$$

After reordering the sums and products:

$$= \sum_{Y_i} \Big(\prod_{\phi \in \mathcal{F}_i} \phi \Big) \sum_{\mathcal{V}_{<(i_1 \leftrightarrow i)}} \Big(\prod_{\phi \in \mathcal{F}_{<(i_1 \leftrightarrow i)}} \phi \Big) \ldots \sum_{\mathcal{V}_{<(i_m \leftrightarrow i)}} \Big(\prod_{\phi \in \mathcal{F}_{<(i_m \leftrightarrow i)}} \phi \Big) =$$

Using the induction step:

$$= \sum_{Y_i} \psi_i \delta_{i_1 \to i} \dots \delta_{i_m \to i} = \delta_{i \to j}$$

Corollary $\beta_i = \tilde{P}(Scope(C_i)) = \sum_{\mathcal{X} \setminus Scope(C_i)} \prod_\phi \phi$

Proof From the definition of belief:

$$\beta_i = \psi_i \prod_{k \in Nb_i} \delta_{k \to i} =$$

Using the theorem above:

$$= \psi_i \prod_{k \in Nb_i} \sum_{\mathcal{V}_{<(k \leftrightarrow i)}} \prod_{\phi \in \mathcal{F}_{<(k \leftrightarrow i)}} \phi =$$

Variational Inference: Unknown mean and variance of a Gaussian

Máme data $\mathcal{D} = \{x_1, \ldots, x_N\}$, která jsou nezávislá a pochází z normálního rozdělení s neznámou střední hodnotou a přesností.

Log likelihood parametrů μ, τ je

$$p(\mathcal{D} \mid \mu, \tau) = \prod_{i=1}^{N} \sqrt{\frac{\tau}{2\pi}} \exp\left(\frac{-\tau(x_i - \mu)^2}{2}\right) \tag{1}$$

$$\log p(\mathcal{D} \mid \mu, \tau) = \sum_{i=1}^{N} \frac{1}{2} \log \frac{\tau}{2\pi} - \frac{\tau(x_i - \mu)^2}{2} \tag{2}$$

$$= \frac{N}{2} \log \tau - \sum_{i=1}^{N} \frac{\tau(x_i - \mu)^2}{2} + C \tag{3}$$

$$= \frac{N}{2} \log \tau - \frac{\tau}{2}\left(N(\mu - \bar{x})^2 + \sum_{i=1}^{N}(x_i - \bar{x})^2\right) + C \tag{4}$$

Pravděpodobnostní model reprezentuje sdruženou pravděpodobnost $p(\boldsymbol{X}, \boldsymbol{Z})$ a my potřebujeme nalézt aproximaci aposteriorní distribuce $p(\boldsymbol{Z} \mid \boldsymbol{X})$ a také pravděpodobnost pozorování $p(\boldsymbol{X})$.

$$\log p(\boldsymbol{X}) = \mathcal{L}(q) + \mathrm{KL}(q \| p) \tag{5}$$

$$\mathcal{L}(q) = \int q(\boldsymbol{Z}) \log\left(\frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})}\right) \mathrm{d}\boldsymbol{Z} \tag{6}$$

$$\mathrm{KL}(q \| p) = -\int q(\boldsymbol{Z}) \log\left(\frac{p(\boldsymbol{Z} \mid \boldsymbol{X})}{q(\boldsymbol{Z})}\right) \mathrm{d}\boldsymbol{Z} \tag{7}$$

Aposteriorní distribuci faktorizujeme

$$q(\boldsymbol{Z}) = \prod_{i}^{M} q_i(\boldsymbol{Z}_i) \tag{8}$$

Likelihood s faktorizovanou distribucí pak bude

$$\mathcal{L}(q) = \int \left(\prod_i q_i\right) \left(\log p(\boldsymbol{X}, \boldsymbol{Z}) - \sum_i \log q_i\right) \mathrm{d}\boldsymbol{Z} \tag{9}$$

$$= \int q_j \left(\int \log p(\boldsymbol{X}, \boldsymbol{Z}) \prod_{i \neq j} q_i \ \mathrm{d}\boldsymbol{Z}_i\right) \mathrm{d}\boldsymbol{Z}_j - \tag{10}$$

$$- \int q_j \log q_j \ \mathrm{d}\boldsymbol{Z}_j + C \tag{11}$$

$$= \int q_j \log \tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j) \mathrm{d}\boldsymbol{Z}_j - \int q_j \log q_j \mathrm{d}\boldsymbol{Z}_j + C \tag{12}$$

$$\log \tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j) = \mathbb{E}_{i \neq j}[\log p(\boldsymbol{X}, \boldsymbol{Z})] + C \tag{13}$$

Předpokládejme, že všechny $\{q_{i \neq j}\}$ jsou zafixované a maximizujeme $\mathcal{L}(q)$ pro distribuci $q_j(\boldsymbol{Z}_j)$. To provedeme jednoduše když vezmeme v úvahu, že rovnice (12) je negativní Kullback-Leibler divergence mezi $q_j(\boldsymbol{Z}_j)$ a $\tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j)$. Takže maximalizace (12) je to samé jako minimalizace KL divergence a minimum nastane když $q_j(\boldsymbol{Z}_j) = \tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j)$. Obecná formule pro optimální řešení $q_j^*(\boldsymbol{Z}_j)$ tedy bude

$$\log q_j^*(\boldsymbol{Z}_j) = \mathbb{E}_{i \neq j}[\log p(\boldsymbol{X}, \boldsymbol{Z})] + C \tag{14}$$

Nyní k příkladu, aposterioriní pravděpodobnost budeme aproximovat faktory:

$$q(\mu, \tau) = q_\mu(\mu) q_\tau(\tau) \tag{15}$$

Optimální faktory získáme z rovnice (14):

$$\log q_\mu^*(\mu) = \mathbb{E}_\tau[\log p(\mathcal{D} \mid \mu, \tau) + \log(p(\mu)) + \log(p(\tau))] \tag{16}$$

$$= \mathbb{E}_\tau\left[\frac{N}{2} \log \tau - \frac{\tau}{2}\Big(N(\mu - \bar{x})^2 + \sum_i^N (x_i - \bar{x})^2\Big) + \right. \tag{17}$$

$$\left. + \log \frac{1}{\sigma_\mu} + \log \frac{1}{\tau} + C\right]$$

$$= -\frac{N \mathbb{E}_\tau[\tau]}{2}(\mu - \bar{x})^2 + C \tag{18}$$

Nyní už je vidět, že $q_\mu^*(\mu)$ je ve formě normálního rozdělení

$$q_\mu^*(\mu) = N(\mu \mid \bar{x}, \lambda^{-1}) \tag{19}$$

$$\lambda = N \mathbb{E}_\tau[\tau] \tag{20}$$

Pro $q^*_\tau(\tau)$ budeme postupovat obdobně:

$$\log q^*_\tau(\tau) = \mathbb{E}_\mu[\log p(\mathcal{D} \mid \mu, \tau) + \log p(\mu) + \log p(\tau)] \tag{21}$$

$$= \mathbb{E}_\mu\Big[\frac{N}{2}\log(\tau) - \frac{\tau}{2}\Big(\sum_{i=1}^{N}(x_i - \mu)^2\Big) + \tag{22}$$

$$+ \log\frac{1}{\sigma_\mu} + \log\frac{1}{\tau} + C\Big]$$

$$= \frac{N-2}{2}\log(\tau) - \frac{\tau}{2}\Big(\mathbb{E}_\mu\Big[\sum_{i=1}^{N}(x_i^2 - 2x_i\mu + \mu^2)\Big]\Big) + C \tag{23}$$

$$= \frac{N-2}{2}\log(\tau) - \frac{\tau}{2}\Big(\sum_{i=1}^{N}x_i^2 - 2\mathbb{E}_\mu[\mu]\sum_{i=1}^{N}x_i + N\mathbb{E}_\mu[\mu^2])\Big) + C \tag{24}$$

Pokud výsledný tvar srovnáme s předpisem pro Gamma rozložení

$$Gamma(\tau \mid a, b) = b^2\frac{1}{\Gamma(a)}\tau^{a-1}\exp(-b\tau) \tag{25}$$

s parametry

$$a = \frac{N}{2} \tag{26}$$

$$b = \frac{1}{2}\Big(\sum_{i=1}^{N}x_i^2 - 2\mathbb{E}_\mu[\mu]\sum_{i=1}^{N}x_i + N\mathbb{E}_\mu[\mu^2])\Big) \tag{27}$$

Nyní můžeme oba kroky střídat a iterovat dokud nedojdeme k pevnému bodu. Při iteraci využijeme toho, že pro nalezení nové hodnoty $q_\mu(\mu)$ nám stačí znát střední hodnotu τ, což pro Gamma rozdělení není problém. Pro nalezení nové hodnoty $q_\tau(\tau)$ nám zase stačí znát $\mathbb{E}_\mu[\mu]$ a $\mathbb{E}_\mu[\mu^2]$

Expectation Propagation for the Clutter Problem – Theory and Implementation

Matěj Korvas

June 4, 2013

Abstract

These lecture notes describe the algorithm of Expectation Propagation as applied to the Clutter problem, touching on the underlying theory. Important is also the practical part where I document my implementation of the algorithm.

1 Introduction

Expectation Propagation (EP for short) was introduced in [3] as an iterated version of the previously known Assumed-Density Filtering approximate inference algorithm. In the work [3], the author also shows how EP is applied to the clutter problem.

In the next section, we describe the EP algorithm in general, in Section 3, we formulate the clutter problem and derive formulas used in EP to solve it, and the final Section 4 discusses our implementation of EP applied to the clutter problem.

2 Expectation Propagation

Expectation Propagation is an approximate inference algorithm for graphical probabilistic models that factorise as follows:

$$p(\mathbf{z}, \mathbf{e}) = \prod_i f_i(\mathbf{z}, \mathbf{e}) \tag{1}$$

where $\mathbf{z}$ is the vector of latent variables, $\mathbf{e}$ is the vector of observed variables (evidence), and f_i are factors that depend on a non-empty subset of $\mathbf{z}$ and a subset of $\mathbf{e}$. This factorisation naturally emerges in experiments with i.i.d. observations where f_0 is the prior on $\mathbf{z}$ and f_i the posterior for the i-th observation for $i = 1, \ldots$

TODO: Draw a figure of a graphical model that is typically used for doing EP.

EP approximates the factors f_i with factors $\tilde{f}_i$ that belong to a convenient probability distribution family. The approximation aims to minimise the KL-divergence between a distribution computed using the exact factor f_i, and a distribution using the approximate factor $\tilde{f}_i$. If the approximating distribution family is chosen from the exponential family (which it typically is), minimising the KL-divergence is reduced to *matching moments*, i.e. setting a few moments of the estimating distribution (its sufficient statistics) to the values of corresponding moments of the distribution approximated. Choosing the family from the exponential family has also other benefits, including the fact that this family is closed under the operation of product (this property being assumed in the algorithm), and that Minka [3] proved the existence of a fixed point for the solution provided the family is exponential.

Choosing the approximating family is the first thing done in the algorithm. Next, approximated factors $\tilde{f}_i$ and their product $\mathcal{Q} = \prod_i \tilde{f}_i$ are initialised to uniform. The algorithm then proceeds in iterations, iteratively updating all the approximating factors in each of the outer iterations. When convergence is reached, the normalisation coefficient, an estimate of $p(\mathbf{e})$, is computed. A more detailed exposition of the algorithm follows.

1. Initialisation

All the approximate factors are initialised to uniform, meaning the initial approximation is non-informative. The product $\mathcal{Q}$ of the factors is computed accordingly. Typically, all the factors, as well as their product, are initialised to constant 1.

Factors that already belong to the chosen family can also be computed during initialisation, as such factors are always best approximated by themselves, not needing to be updated iteratively.

2. Outer loop

Following four steps are repeated until convergence.

2.1. Choose a factor $\tilde{f}_i$

Choose a factor to approximate.

2.2. Compute the cavity distribution $\mathcal{Q}^{\setminus i}$

When updating the factor $\tilde{f}_i$, we would ideally want to minimise the KL-divergence between the true distribution and the resulting approximative distribution:

$$\arg\min_{\tilde{f}_i} \mathrm{KL}(p \mid\mid \prod_i \tilde{f}_i). \tag{2}$$

However, there we would need to compute moments of p in order to optimise for this KL-divergence. If we were able to do that, we would not need to use approximate inference in the first place, so let us assume this is intractable. In that case, we have to substitute p with an approximation. The approximation used in EP is the following:

$$\hat{p} = \frac{1}{Z_i} f_i \mathcal{Q}^{\setminus i} \tag{3}$$

where

$$\mathcal{Q}^{\setminus i} \propto \prod_{j \neq i} \tilde{f}_j \quad (= \mathcal{Q}/\tilde{f}_i). \tag{4}$$

Here, $\mathcal{Q}^{\setminus i}$ is called the *cavity distribution*, as it is a distribution over $\mathbf{z}$ obtained by multiplying all the approximate factors but the i-th one (thus creating the cavity in the distribution) and normalising (in order to make it a distribution). $\hat{p}$ is defined as a product of the *exact* factor f_i with the rest of the factors *approximated*, normalised to 1, and the cavity distribution needs to be computed in order to express $\hat{p}$.

2.3. Compute the approximative distribution $\mathcal{Q}_{\text{new}}$

Whereas the previous step was concerned with computing the cavity distribution, computing the normalisation coefficient Z_i (as $\int_{\mathbf{Z}} f_i(\mathbf{z})\mathcal{Q}^{\setminus i}(\mathbf{z})\mathrm{d}\mathbf{z}$) and the approximative posterior distribution $\hat{p}$ is reserved for this step.

Having computed $\hat{p}$, we can minimise the KL-divergence to an updated $\mathcal{Q}_{\text{new}}$ restricted to be in the approximating family $\mathcal{F}$:

$$\arg\min_{\mathcal{Q}_{\text{new}}\in\mathcal{F}} \mathrm{KL}(\hat{p} \,||\, \mathcal{Q}_{\text{new}}). \tag{5}$$

As mentioned earlier, this minimisation is achieved by matching moments of $\mathcal{Q}_{\text{new}}$ to those of $\hat{p}$.

2.4. Update the factor

We can see the relation of the f_i, which we wish to approximate, to $\mathcal{Q}_{\text{new}}$ by combining formulas (3) and (5):

$$\mathcal{Q}_{\text{new}} \approx \hat{p} = \frac{1}{Z_i} f_i \mathcal{Q}^{\setminus i}. \tag{6}$$

From here, we easily obtain the formula for the approximation of f_i:

$$f_i \approx \tilde{f}_i = Z_i \frac{\mathcal{Q}_{\text{new}}}{\mathcal{Q}^{\setminus i}}. \tag{7}$$

Thanks to the right hand side of Eq. (7) consisting of a division of distributions from the approximating family (and a coefficient), $\tilde{f}_i$ will also be from that family (provided it is closed under division). Now, the approximate factor gets updated according to Eq. (7), and the outer loop is repeated.

3. Evaluate the normalisation constant

After the algorithm has converged to a set of factors $\{\tilde{f}_i\}$, an approximate posterior $p(\mathbf{z}, \mathbf{e})$ can be computed as a product of the factors, according to the assumptions. If we are interested in $p(\mathbf{e})$, the model evidence, it can be computed now as

$$p(\mathbf{e}) \approx \int_{\mathbf{Z}} \prod_i \tilde{f}_i(\mathbf{z})\mathrm{d}\mathbf{z}. \tag{8}$$

3 The Clutter Problem

In the Clutter problem, we assume a sequence of d-dimensional i.i.d. observations being generated either from a normal distribution with an unknown mean with some probability, or from the "clutter" distribution. The model is specified by the following formulas:

$$W_i \overset{\text{i.i.d.}}{\sim} \text{Bern}(w_0) \tag{9}$$

$$\mathbf{x}_i \mid \boldsymbol{\mu} \overset{\text{ind.}}{\sim} W_i\, \mathcal{N}(\mathbf{0}_d, a\mathbf{I}_d) + (1 - W_i)\, \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_d) \tag{10}$$

The w_0 parameter determines the *proportion of clutter*, W_i select for each observation whether it was generated from the distribution of interest, $\mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_d)$, or the clutter, and finally, $\boldsymbol{\mu}$ is the unknown mean of the

distribution we are trying to estimate. When learning the model, we will not learn W_i explicitly for each i, but rather treat the observations as identically distributed with the same proportion of clutter w_0:

$$p(\mathbf{x} \mid \boldsymbol{\mu}) = w_0 \, \mathcal{N}(\mathbf{0}_d, a\mathbf{I}_d) + (1 - w_0) \, \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_d). \tag{11}$$

Finally, we adopt a broad Gaussian prior on $\boldsymbol{\mu}$:

$$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}_d, b\mathbf{I}_d). \tag{12}$$

TODO: draw the graphical model
This problem fits nicely the assumptions for EP:

1. It is intractable to do exact inference to find the value of $\boldsymbol{\mu}$. This is due to the fact that in the Bayesian network, the node for $\boldsymbol{\mu}$ has $(N + 1)$ independent parent nodes, a prior and the N likelihood factors, of which the N likelihood factors have 2 Gaussian components each. This results in the posterior for $\boldsymbol{\mu}$ consisting of 2^N N-dimensional Gaussians, corresponding to the 2^N subsets of observations that could have been generated from the true distribution (as opposed to the clutter).

2. The posterior is a product of factors that depend on a non-empty subset of the latent variables (which is $\{\boldsymbol{\mu}\}$ in this case) and a subset of the observed variables (either $\{\mathbf{x}_i\}$ for the likelihood factors, or $\emptyset$ for the prior) – exactly as required.

Instantiating the general Eq. (1) for the Clutter problem, we get the following:

$$p((\boldsymbol{\mu}), (\mathbf{x}_1, \ldots, \mathbf{x}_N)) = p(\boldsymbol{\mu}) \cdot \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\mu}). \tag{13}$$

In Eq. (13), the generic f_0 is instantiated as the prior $p(\boldsymbol{\mu})$, and the generic $f_i, i = 1, \ldots$ as the likelihood $p(\mathbf{x}_i \mid \boldsymbol{\mu})$. In the following, we may use one or the other notation, whichever is more convenient.

We choose to approximate the factors, and hence also their product, by (unnormalised) spherical Gaussians, with one stipulation: the factors approximating the likelihoods may have their σ^2 parameter negative. This is an inherent property of the algorithm, and we discuss it later in Section 4. Still, each factor $\tilde{f}_i$ can be represented by the triple $\langle \tilde{s}_i, \tilde{\mathbf{m}}_i, \tilde{v}_i \rangle$, describing its scale ($\int_{\mathbf{Z}} \tilde{f}_i(\mathbf{z}) \mathrm{d}\mathbf{z}$), mean, and variance, respectively:

$$\tilde{f}_i = \tilde{s}_i \, \mathcal{N}(\tilde{\mathbf{m}}_i, \tilde{v}_i \mathbf{I}_d). \tag{14}$$

Besides that, also the approximate posterior has the same form, and we shall denote its parameters as follows:

$$\mathcal{Q} = \mathcal{N}(\mathbf{m}, v\mathbf{I}_d). \tag{15}$$

Note that $\mathcal{Q}$ is an approximating *distribution*, i.e. it is normalised to 1.

Since $\tilde{f}_0$, the prior, already is a spherical Gaussian, its parameters can be set as part of initialisation:

$$\tilde{s}_0 = 1 \qquad \tilde{\mathbf{m}}_0 = \mathbf{0}_d \qquad \tilde{v}_0 = b. \tag{16}$$

This factor is exact and need not be updated anymore.

What remains is expressing the formulas (4), (5), (7) for a factor $\tilde{f}_i, i = 1, \ldots$, and (8). The following sections are concerned with this.

Update formula for the cavity distribution

The general formula is as follows:

$$\mathcal{Q}^{\setminus i} \propto \mathcal{Q}/\tilde{f}_i. \tag{4 – repeated}$$

After substituting the values of $\mathcal{Q}$ and $\tilde{f}_i$, represented as shown in Eqs. (15) and (14), respectively, we obtain the following:

$$\mathcal{Q}^{\setminus i} \propto \frac{\mathcal{N}(\mathbf{m}, v\mathbf{I}_d)}{\tilde{s}_i\, \mathcal{N}(\tilde{\mathbf{m}}_i, \tilde{v}_i\mathbf{I}_d)}. \tag{17}$$

The parameters of $\mathcal{Q}^{\setminus i}$ can be computed using the formula for the ratio of Gaussians,

$$\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)/\,\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = C\,\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{18}$$

where

$$\begin{aligned}
\boldsymbol{\Sigma} &= \left(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \\
\boldsymbol{\mu} &= \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \\
C &= \sqrt{\frac{|\boldsymbol{\Sigma}|\,|\boldsymbol{\Sigma}_2|}{(2\pi)^d|\boldsymbol{\Sigma}_1|}}\exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)\right\}.
\end{aligned} \tag{19}$$

As the result, we can express $\mathcal{Q}^{\setminus i}$ in terms of its parameters $\mathbf{m}^{\setminus \mathbf{i}}$ (mean) and $v^{\setminus i}$ ($v^{\setminus i}\mathbf{I}_d$ being the variance-covariance matrix) as follows:

$$\mathbf{m}^{\setminus \mathbf{i}} = v^{\setminus i}(\mathbf{m}v^{-1} - \tilde{\mathbf{m}}_i\tilde{v}_i^{-1}) \qquad\qquad v^{\setminus i} = \left(v^{-1} - \tilde{v}_i^{-1}\right)^{-1}. \tag{20}$$

Update formula for $\mathcal{Q}$

In computing $\mathcal{Q}_{\text{new}}$ according to Eq. (5), we have to compute $\hat{p}$ and then its first and second moment in order to arrive at the spherical normal distribution minimising the KL-divergence to $\hat{p}$. In the definition of $\hat{p}$ in Eq. (3), the quantity Z_i is yet to be computed. It is the normalisation constant of $f_i\mathcal{Q}^{\setminus i}$, i.e.:

$$Z_i = \int_{\mathbb{R}^d} f_i(\boldsymbol{\mu})\mathcal{Q}^{\setminus i}(\boldsymbol{\mu})\mathrm{d}\boldsymbol{\mu}. \tag{21}$$

Parameters of $\mathcal{Q}^{\setminus i}$ were obtained in the previous step, and f_i was defined as the likelihood for $\mathbf{x}_i$ (cf. Eq. (11)):

$$f_i(\boldsymbol{\mu}) = w_0\,\mathcal{N}(\mathbf{x}_i; \mathbf{0}_d, a\mathbf{I}_d) + (1 - w_0)\,\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{I}_d). \tag{22}$$

Substituting into Eq. (21), we get

$$\begin{aligned}
Z_i &= \int_{\mathbb{R}^d}[w_0\,\mathcal{N}(\mathbf{x}_i; \mathbf{0}_d, a\mathbf{I}_d) + (1 - w_0)\,\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{I}_d)]\;\mathcal{N}(\boldsymbol{\mu}; \mathbf{m}^{\setminus \mathbf{i}}, v^{\setminus i}\mathbf{I}_d)\mathrm{d}\boldsymbol{\mu} && (23)\\
&= \int_{\mathbb{R}^d} w_0\,\mathcal{N}(\mathbf{x}_i; \mathbf{0}_d, a\mathbf{I}_d)\,\mathcal{N}(\boldsymbol{\mu}; \mathbf{m}^{\setminus \mathbf{i}}, v^{\setminus i}\mathbf{I}_d)\mathrm{d}\boldsymbol{\mu} \\
&\quad + \int_{\mathbb{R}^d}(1 - w_0)\,\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{I}_d)\,\mathcal{N}(\boldsymbol{\mu}; \mathbf{m}^{\setminus \mathbf{i}}, v^{\setminus i}\mathbf{I}_d)\mathrm{d}\boldsymbol{\mu} && (24)\\
&= w_0\,\mathcal{N}(\mathbf{x}_i; \mathbf{0}_d, a\mathbf{I}_d) + (1 - w_0)\int_{\mathbb{R}^d}\mathcal{N}(\mathbf{x}_i - \boldsymbol{\mu}; \mathbf{0}_d, \mathbf{I}_d)\,\mathcal{N}(\boldsymbol{\mu}; \mathbf{m}^{\setminus \mathbf{i}}, v^{\setminus i}\mathbf{I}_d)\mathrm{d}\boldsymbol{\mu} && (25)\\
&= w_0\,\mathcal{N}(\mathbf{x}_i; \mathbf{0}_d, a\mathbf{I}_d) + (1 - w_0)\,\mathcal{N}\Big(\mathbf{x}_i; \mathbf{m}^{\setminus \mathbf{i}}, (v^{\setminus i} + 1)\mathbf{I}_d\Big) && (26)
\end{aligned}$$

where, going from (25) to (26), we used the result [1] about convolution of Gaussians.

The mean value and variance of $\hat{p}$ can be derived for a general form of the factor f_i. Hence, we will simplify the next derivations by rewriting $\hat{p}$ in the following form:

$$\hat{p}(\boldsymbol{\mu}) = \frac{1}{Z(\mathbf{m}, \boldsymbol{\Sigma})} f(\boldsymbol{\mu})\, \mathcal{N}(\boldsymbol{\mu}; \mathbf{m}, \boldsymbol{\Sigma}) \tag{27}$$

where

$$Z(\mathbf{m}, \boldsymbol{\Sigma}) = \int_{\mathbb{R}^d} f(\boldsymbol{\mu})\, \mathcal{N}(\boldsymbol{\mu}; \mathbf{m}, \boldsymbol{\Sigma}) \mathrm{d}\boldsymbol{\mu}. \tag{28}$$

The two moments will be found from derivatives of Z:

$$\frac{\mathrm{d}Z(\mathbf{m}, \boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}} = \int_{\mathbb{R}^d} \frac{\mathrm{d}}{\mathrm{d}\mathbf{m}} \left(f(\boldsymbol{\mu})\, \mathcal{N}(\boldsymbol{\mu}; \mathbf{m}, \boldsymbol{\Sigma}) \right) \mathrm{d}\boldsymbol{\mu} \tag{29}$$

$$= \int_{\mathbb{R}^d} \frac{f(\boldsymbol{\mu})}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left\{ -\frac{1}{2} (\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{m}) \right\} \left((\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1} \right) \mathrm{d}\boldsymbol{\mu} \tag{30}$$

$$= \int_{\mathbb{R}^d} Z(\mathbf{m}, \boldsymbol{\Sigma}) \hat{p}(\boldsymbol{\mu}) \left((\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1} \right) \mathrm{d}\boldsymbol{\mu} \tag{31}$$

$$= Z(\mathbf{m}, \boldsymbol{\Sigma}) \left(\int_{\mathbb{R}^d} \boldsymbol{\mu} \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} - \int_{\mathbb{R}^d} \mathbf{m} \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} \right)^T \boldsymbol{\Sigma}^{-1} \tag{32}$$

$$= Z(\mathbf{m}, \boldsymbol{\Sigma}) \cdot \left(\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}] - \mathbf{m} \right)^T \boldsymbol{\Sigma}^{-1}, \tag{33}$$

$$\frac{\mathrm{d}Z(\mathbf{m}, \boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}} = \int_{\mathbb{R}^d} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Sigma}} \left(f(\boldsymbol{\mu})\, \mathcal{N}(\boldsymbol{\mu}; \mathbf{m}, \boldsymbol{\Sigma}) \right) \mathrm{d}\boldsymbol{\mu} \tag{34}$$

$$= \int_{\mathbb{R}^d} \frac{f(\boldsymbol{\mu})}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left\{ -\frac{1}{2} (\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{m}) \right\} \left(\frac{1}{2} \boldsymbol{\Sigma}^{-T} (\boldsymbol{\mu} - \mathbf{m}) (\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-T} \right)$$
$$- \frac{1}{2} \frac{f(\boldsymbol{\mu})}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \boldsymbol{\Sigma}^{-T} \exp\left\{ -\frac{1}{2} (\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{m}) \right\} \mathrm{d}\boldsymbol{\mu} \tag{35}$$

$$= \frac{1}{2} \boldsymbol{\Sigma}^{-1} \left[\int_{\mathbb{R}^d} \boldsymbol{\mu}\boldsymbol{\mu}^T Z(\mathbf{m}, \boldsymbol{\Sigma}) \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} - \int_{\mathbb{R}^d} \boldsymbol{\mu}\mathbf{m}^T Z(\mathbf{m}, \boldsymbol{\Sigma}) \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} \right.$$
$$\left. - \int_{\mathbb{R}^d} \mathbf{m}\boldsymbol{\mu}^T Z(\mathbf{m}, \boldsymbol{\Sigma}) \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} + \int_{\mathbb{R}^d} \mathbf{m}\mathbf{m}^T Z(\mathbf{m}, \boldsymbol{\Sigma}) \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} \right] \boldsymbol{\Sigma}^{-1}$$
$$- \frac{1}{2} \int_{\mathbb{R}^d} \boldsymbol{\Sigma}^{-1} Z(\mathbf{m}, \boldsymbol{\Sigma}) \hat{p}(\boldsymbol{\mu}) \mathrm{d}\boldsymbol{\mu} \tag{36}$$

$$= Z(\mathbf{m}, \boldsymbol{\Sigma}) \cdot \left\{ \frac{1}{2} \boldsymbol{\Sigma}^{-1} \left[\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}\boldsymbol{\mu}^T] - \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}] \mathbf{m}^T - \mathbf{m} \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T + \mathbf{m}\mathbf{m}^T \right] \boldsymbol{\Sigma}^{-1} - \frac{1}{2} \boldsymbol{\Sigma}^{-1} \right\} \tag{37}$$

where $\mathbf{X}^{-T}$ is a shorthand for $\left(\mathbf{X}^{-1}\right)^T \left(= \left(\mathbf{X}^T\right)^{-1}\right)$ and we applied matrix calculus results from [2].

Eqs. (33) and (37) give us formulas for the moments we are interested in. However, they include the term $Z(\mathbf{m}, \boldsymbol{\Sigma})$, which is an inconvenient integral to compute. Taking the derivative of the $\log$ instead will

get us rid of this term:

$$\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}} = \frac{1}{Z(\mathbf{m},\boldsymbol{\Sigma})}\frac{\mathrm{d}Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}} = \left(\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}] - \mathbf{m}\right)^T \boldsymbol{\Sigma}^{-1} \tag{38}$$

$$\begin{aligned}\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}} &= \frac{1}{Z(\mathbf{m},\boldsymbol{\Sigma})}\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}} \\ &= \frac{1}{2}\boldsymbol{\Sigma}^{-1}\left[\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}\boldsymbol{\mu}^T] - \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]\mathbf{m}^T - \mathbf{m}\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T + \mathbf{m}\mathbf{m}^T\right]\boldsymbol{\Sigma}^{-1} - \frac{1}{2}\boldsymbol{\Sigma}^{-1}\end{aligned} \tag{39}$$

The first and second moment are now obtained easily from Eqs. (38) and (39) by shuffling them a bit. We get:

$$\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}] = \mathbf{m} + \boldsymbol{\Sigma}\left(\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}}\right)^T \tag{40}$$

$$\begin{aligned}\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}\boldsymbol{\mu}^T] - \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T &= \boldsymbol{\Sigma}\left(2\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}} + \boldsymbol{\Sigma}^{-1}\right)\boldsymbol{\Sigma} - \Big[-\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]\mathbf{m}^T - \mathbf{m}\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T + \mathbf{m}\mathbf{m}\Big] \\ &\quad - \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T \end{aligned} \tag{41}$$

$$\begin{aligned}&= 2\boldsymbol{\Sigma}\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}}\boldsymbol{\Sigma} + \boldsymbol{\Sigma} \\ &\quad - \left[\left(\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}] - \mathbf{m}\right)\left(\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}] - \mathbf{m}\right)^T - \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T\right] - \mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]\mathbb{E}_{\hat{p}}[\boldsymbol{\mu}]^T \end{aligned} \tag{42}$$

$$= 2\boldsymbol{\Sigma}\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}}\boldsymbol{\Sigma} + \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\left(\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}}\right)^T \frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}}\boldsymbol{\Sigma}^T \tag{43}$$

$$= \boldsymbol{\Sigma}\left[2\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\boldsymbol{\Sigma}} - \left(\frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}}\right)^T \frac{\mathrm{d}\log Z(\mathbf{m},\boldsymbol{\Sigma})}{\mathrm{d}\mathbf{m}}\right]\boldsymbol{\Sigma} + \boldsymbol{\Sigma} \tag{44}$$

Now, what remains to be computed in order to arrive at the KL-divergence minimiser are the derivatives

of $\log Z$:

$$\frac{\mathrm{d}\log Z_i}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}} = \frac{\mathrm{d}\log\left[w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)\right]}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}} \tag{45}$$

$$= \frac{1}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)} \cdot \frac{(1-w_0)\mathrm{d}\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}} \tag{46}$$

$$= \frac{1}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)} \cdot \frac{\frac{1-w_0}{\sqrt{(2\pi)^d|(v^{\backslash i}+1)\mathbf{I}_d|}}\mathrm{d}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{m}^{\backslash i})^T((v^{\backslash i}+1)\mathbf{I}_d)^{-1}(\mathbf{x}_i-\mathbf{m}^{\backslash i})\right)}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}} \tag{47}$$

$$= \frac{\frac{1-w_0}{\sqrt{(2\pi)^d|(v^{\backslash i}+1)\mathbf{I}_d|}}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{m}^{\backslash i})^T((v^{\backslash i}+1)\mathbf{I}_d)^{-1}(\mathbf{x}_i-\mathbf{m}^{\backslash i})\right)}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)} \cdot \frac{\mathrm{d}\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{m}^{\backslash i})^T((v^{\backslash i}+1)\mathbf{I}_d)^{-1}(\mathbf{x}_i-\mathbf{m}^{\backslash i})\right)}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}} \tag{48}$$

$$= \frac{(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)} \cdot \left(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}}\right)^T\left((v^{\backslash i}+1)\mathbf{I}_d\right)^{-1} \tag{49}$$

Let us simplify the expression by introducing r as the probability of $\mathbf{x}_i$ not being generated from the clutter, and realising that multiplication by the last term is equivalent to division by $(v^{\backslash i}+1)$:

$$r := \frac{(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},(v^{\backslash i}+1)\mathbf{I}_d\big)} \tag{50}$$

$$\frac{\mathrm{d}\log Z_i}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}} = r\frac{\left(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}}\right)^T}{v^{\backslash i}+1} \tag{51}$$

The derivative of $\log Z_i$ by the variance parameter is obtained similarly (let $\mathbf{\Sigma}$ denote the second param-

eter of Z_i, which has the value $v^{\backslash i}\mathbf{I}_d$):

$$\frac{\mathrm{d}\log Z_i}{\mathrm{d}\boldsymbol{\Sigma}} = \frac{\mathrm{d}\log\left[w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\big)\right]}{\mathrm{d}\boldsymbol{\Sigma}} \tag{52}$$

$$= \frac{1}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\big)}\cdot\frac{(1-w_0)\mathrm{d}\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\big)}{\mathrm{d}\boldsymbol{\Sigma}} \tag{53}$$

$$= \frac{(1-w_0)(2\pi)^{-d/2}}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\big)} \cdot\frac{\mathrm{d}\left[|\boldsymbol{\Sigma}+\mathbf{I}_d|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{m}^{\backslash i})^T(\boldsymbol{\Sigma}+\mathbf{I}_d)^{-1}(\mathbf{x}_i-\mathbf{m}^{\backslash i})\right)\right]}{\mathrm{d}\boldsymbol{\Sigma}} \tag{54}$$

$$\begin{aligned}= \frac{(1-w_0)(2\pi)^{-d/2}}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\big)} &\cdot\Bigg\{|\boldsymbol{\Sigma}+\mathbf{I}_d|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T(\boldsymbol{\Sigma}+\mathbf{I}_d)^{-1}(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})\right]\\ &\cdot\left(\frac{1}{2}(\boldsymbol{\Sigma}+\mathbf{I}_d)^{-T}(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T(\boldsymbol{\Sigma}+\mathbf{I}_d)^{-T}\right)\\ &-\frac{1}{2}|\boldsymbol{\Sigma}+\mathbf{I}_d|^{-1/2}(\boldsymbol{\Sigma}+\mathbf{I}_d)^{-T}\exp\left[-\frac{1}{2}(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T(\boldsymbol{\Sigma}+\mathbf{I}_d)^{-1}(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})\right]\Bigg\}\end{aligned} \tag{55}$$

$$\begin{aligned}= \frac{(1-w_0)}{w_0\,\mathcal{N}(\mathbf{x}_i;\mathbf{0}_d,a\mathbf{I}_d)+(1-w_0)\,\mathcal{N}\big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\big)} &\cdot\left[\mathcal{N}\Big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\Big)\cdot\frac{(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T}{2(v^{\backslash i}+1)^2}-\mathcal{N}\Big(\mathbf{x}_i;\mathbf{m}^{\backslash\mathbf{i}},\boldsymbol{\Sigma}+\mathbf{I}_d\Big)\cdot\frac{\mathbf{I}_d}{2(v^{\backslash i}+1)}\right]\end{aligned} \tag{56}$$

$$= \frac{r}{2(v^{\backslash i}+1)^2}\cdot\Big[(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T-(v^{\backslash i}+1)\mathbf{I}_d\Big] \tag{57}$$

Substituting into Eqs. (40) and (44), we finally arrive at the new parameters of $\mathcal{Q}$, $\mathbf{m}_{\text{new}}$ (mean) and $\boldsymbol{\Sigma}_{\text{new}}$ (variance):

$$\mathbf{m}_{\text{new}} = \mathbf{m}^{\backslash\mathbf{i}}+\boldsymbol{\Sigma}\left(\frac{\mathrm{d}\log Z_i}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}}\right)^T = \mathbf{m}^{\backslash\mathbf{i}}+\boldsymbol{\Sigma}r\frac{\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}}}{v^{\backslash i}+1} = \mathbf{m}^{\backslash\mathbf{i}}+r\frac{v^{\backslash i}}{v^{\backslash i}+1}(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}}) \tag{58}$$

$$\boldsymbol{\Sigma}_{\text{new}} = \boldsymbol{\Sigma}\left\{2\frac{\mathrm{d}\log Z_i}{\mathrm{d}\boldsymbol{\Sigma}}-\left(\frac{\mathrm{d}\log Z_i}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}}\right)^T\frac{\mathrm{d}\log Z_i}{\mathrm{d}\mathbf{m}^{\backslash\mathbf{i}}}\right\}\boldsymbol{\Sigma}+\boldsymbol{\Sigma} \tag{59}$$

$$\begin{aligned}= \boldsymbol{\Sigma}\Bigg\{&\frac{r}{(v^{\backslash i}+1)^2}\cdot\Big[(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T-(v^{\backslash i}+1)\mathbf{I}_d\Big]\\ &-\left(r\frac{\big(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}}\big)^T}{v^{\backslash i}+1}\right)^T r\frac{\big(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}}\big)^T}{v^{\backslash i}+1}\Bigg\}\boldsymbol{\Sigma}+\boldsymbol{\Sigma}\end{aligned} \tag{60}$$

$$= r\left(\frac{v^{\backslash i}}{v^{\backslash i}+1}\right)^2\cdot\Big[(1-r)(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})(\mathbf{x}_i-\mathbf{m}^{\backslash\mathbf{i}})^T-(v^{\backslash i}+1)\mathbf{I}_d\Big]+\boldsymbol{\Sigma} \tag{61}$$

where we again used the symbol $\mathbf{\Sigma}$ to denote $v^{\setminus i}\mathbf{I}_d$.

However, this $\mathbf{\Sigma}_{\text{new}}$ is generally not a variance matrix of a spherical normal, which is the form we assume for the posterior distribution. Hence, we need to find the KL-divergence minimiser of a spherical normal from a normal with the general covariance $\mathbf{\Sigma}$.[1] Let us solve this problem now, denoting the general multivariate normal with $\mathcal{Q}$ and the spherical one with $\mathcal{S}$, and assuming the mean $\mathbf{0}_d$ for both, WLOG:

$$\begin{aligned}
\arg\min_v \mathrm{KL}(\mathcal{Q} \,||\, \mathcal{S}) &= \arg\min_v \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \log \frac{\mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma})}{\mathcal{N}(\mathbf{x}; \mathbf{0}_d, v\mathbf{I}_d)} \mathrm{d}\mathbf{x} && (62)\\
&= \arg\min_v \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \log \frac{\sqrt{|v\mathbf{I}_d|}\exp\left\{-\frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}\right\}}{\sqrt{|\mathbf{\Sigma}|}\exp\left\{-\frac{1}{2}\mathbf{x}^T v^{-1}\mathbf{I}_d\mathbf{x}\right\}} \mathrm{d}\mathbf{x} && (63)\\
&= \arg\min_v \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \left[\frac{d}{2}\log v - \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\log|\mathbf{\Sigma}| + \frac{1}{2}\mathbf{x}^T v^{-1}\mathbf{I}_d\mathbf{x}\right] \mathrm{d}\mathbf{x} && (64)\\
&= \arg\min_v \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \big(d\log v + \mathbf{x}^T\mathbf{x}/v\big) \mathrm{d}\mathbf{x} && (65)\\
&=: \arg\min_v G(v) && (66)
\end{aligned}$$

Because the function we minimise here is smooth for $v > 0$, we shall find the minimum by setting the derivative equal to zero:

$$\begin{aligned}
0 &= \frac{\mathrm{d}G}{\mathrm{d}v}(v^*) && (67)\\
&= \left(\frac{\mathrm{d}}{\mathrm{d}v}\int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \big(d\log v + \mathbf{x}^T\mathbf{x}/v\big) \mathrm{d}\mathbf{x}\right)(v^*) && (68)\\
&= \left(\int_{\mathbb{R}^d} \frac{\mathrm{d}}{\mathrm{d}v}\,\mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \big(d\log v + \mathbf{x}^T\mathbf{x}/v\big) \mathrm{d}\mathbf{x}\right)(v^*) && (69)\\
&= \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x}; \mathbf{0}_d, \mathbf{\Sigma}) \big(d/v - \mathbf{x}^T\mathbf{x}/v^2\big) \mathrm{d}\mathbf{x}\Big|_{v=v^*} && (70)\\
&= \frac{d}{v} - \frac{1}{v^2}\mathbb{E}_{\mathcal{Q}}[\mathbf{x}^T\mathbf{x}]\Big|_{v=v^*} && (71)
\end{aligned}$$

Using the following identity for the product of a quadratic form with a Gaussian density function,

$$\int_{\mathbb{R}^d} (\mathbf{x} - \mathbf{x_0})^T\mathbf{F}^{-1}(\mathbf{x} - \mathbf{x_0})\,\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{\Sigma}) \mathrm{d}\mathbf{x} = (\mathbf{x_0} - \boldsymbol{\mu})^T\mathbf{F}^{-1}(\mathbf{x_0} - \boldsymbol{\mu}) + \mathrm{Tr}[\mathbf{F}^{-1}\mathbf{\Sigma}], \tag{72}$$

we can express the minimiser v^* from Eq. (71):

$$v^* = \mathrm{Tr}[\mathbf{\Sigma}]/d. \tag{73}$$

This result can be combined with Eq. (61) to give us the updated variance of the spherical Gaussian

[1] This Σ will be the $\mathbf{\Sigma}_{\text{new}}$ as given by Eq. (61).

posterior:

$$v_{\text{new}} = \text{Tr}\left[r \left(\frac{v^{\setminus i}}{v^{\setminus i}+1} \right)^2 \cdot \left[(1-r)(\mathbf{x}_i - \mathbf{m}^{\setminus \mathbf{i}})(\mathbf{x}_i - \mathbf{m}^{\setminus \mathbf{i}})^T - (v^{\setminus i}+1)\mathbf{I}_d \right] + v^{\setminus i}\mathbf{I}_d \right] / d \tag{74}$$

$$= r \left(\frac{v^{\setminus i}}{v^{\setminus i}+1} \right)^2 \cdot \left[(1-r)\text{Tr}\left[(\mathbf{x}_i - \mathbf{m}^{\setminus \mathbf{i}})(\mathbf{x}_i - \mathbf{m}^{\setminus \mathbf{i}})^T \right] / d - (v^{\setminus i}+1) \right] + v^{\setminus i} \tag{75}$$

$$= v^{\setminus i} - r\frac{(v^{\setminus i})^2}{v^{\setminus i}+1} + \frac{r(1-r)}{d} \left(\frac{v^{\setminus i}}{v^{\setminus i}+1} \right)^2 ||\mathbf{x}_i - \mathbf{m}^{\setminus \mathbf{i}}||^2 \tag{76}$$

Eqs. (26), (58) and (76) give us the updated parameters for $\mathcal{Q}$, which was the objective of this step.

Update formula for $\tilde{f}_i$

The updated $\tilde{f}_i$ is now computed according to Eq. (7) using the formula for a ratio of Gaussians, yielding:

$$\tilde{v}_i = \left((v_{\text{new}})^{-1} - (v^{\setminus i})^{-1} \right)^{-1} \tag{77}$$

$$\tilde{\mathbf{m}}_i = \tilde{v}_i \left((v_{\text{new}})^{-1}\mathbf{m}_{\text{new}} - (v^{\setminus i})^{-1}\mathbf{m}^{\setminus \mathbf{i}} \right). \tag{78}$$

We could compute the normalisation constant $\tilde{s}_i$ using the appropriate formula for a ratio of Gaussians, but we can get it in a simpler way in terms of a convolution of Gaussians if we reorganise Eq. (7) slightly:

$$\tilde{f}_i = Z_i \frac{\mathcal{Q}_{\text{new}}}{\mathcal{Q}^{\setminus i}} \tag{7 – repeated}$$

$$Z_i \mathcal{Q}_{\text{new}} = \tilde{s}_i \, \mathcal{N}(\tilde{\mathbf{m}}_\mathbf{i}, \tilde{v}_i \mathbf{I}_d) \cdot \mathcal{Q}^{\setminus i} \tag{79}$$

$$Z_i = \int_{\mathbb{R}^d} Z_i \, \mathcal{N}(\mathbf{x}; \mathbf{m}_{\text{new}}, v_{\text{new}}\mathbf{I}_d)\text{d}\mathbf{x} = \int_{\mathbb{R}^d} \tilde{s}_i \, \mathcal{N}(\mathbf{x}; \tilde{\mathbf{m}}_\mathbf{i}, \tilde{v}_i\mathbf{I}_d) \, \mathcal{N}(\mathbf{x}; \mathbf{m}^{\setminus \mathbf{i}}, v^{\setminus i}\mathbf{I}_d)\text{d}\mathbf{x} \tag{80}$$

$$= \tilde{s}_i \int_{\mathbb{R}^d} \mathcal{N}(\tilde{\mathbf{m}}_\mathbf{i} - \mathbf{x}; \mathbf{0}_d, \tilde{v}_i\mathbf{I}_d) \, \mathcal{N}(\mathbf{x}; \mathbf{m}^{\setminus \mathbf{i}}, v^{\setminus i}\mathbf{I}_d)\text{d}\mathbf{x} \tag{81}$$

$$= \tilde{s}_i \, \mathcal{N}(\tilde{\mathbf{m}}_i; \mathbf{m}^{\setminus \mathbf{i}}, (\tilde{v}_i + v^{\setminus i})\mathbf{I}_d) \tag{82}$$

Note how we moved from Eq. (79) to (80) – the former asserts the equality of measures (unnormalised distributions), hence their integral over the whole sample space must equal too, as asserted in Eq. (80).

From the equality of (80) and (82), we easily obtain the value of $\tilde{s}_i$ as

$$\tilde{s}_i = \frac{Z_i}{\mathcal{N}(\tilde{\mathbf{m}}_i; \mathbf{m}^{\setminus \mathbf{i}}, (\tilde{v}_i + v^{\setminus i})\mathbf{I}_d)}. \tag{83}$$

Formula for the normalisation constant

According to Eq. (8), here we need to evaluate the normalisation constant of a product of $(N+1)$ (spherical) Gaussians $\tilde{f}_i, i = 0, \ldots, N$. The general formulas for a product of a number of Gaussians are the following:

$$\prod_{i=1}^{N} \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{Z} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{84}$$

where

$$\boldsymbol{\Sigma} = \left(\sum_{i=1}^{N} \boldsymbol{\Sigma}_i^{-1} \right)^{-1} \tag{85}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma} \left(\sum_{i=1}^{N} \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i \right) \tag{86}$$

$$Z = \frac{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}}{\prod_{i=1}^{N} (2\pi)^{d/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp\left\{ \frac{1}{2} \left(\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} - \sum_{i=1}^{N} \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \right) \right\}. \tag{87}$$

In the case of spherical Gaussians, using the notation introduced for the Clutter problem, the normalisation constant Z is expressed as follows:

$$Z = (2\pi v)^{d/2} \exp(B/2) \prod_{i=0}^{N} \tilde{s}_i (2\pi \tilde{v}_i)^{-d/2} \tag{88}$$

where

$$B = \mathbf{m}^T v^{-1} \mathbf{m} - \sum_{i=0}^{N} \tilde{\mathbf{m}}_i^T \tilde{v}_i^{-1} \tilde{\mathbf{m}}_i. \tag{89}$$

TODO: how the results are read off

4 Implementation

TODO: how some parameters were instantiated, how uniform factors were expressed as Gaussians (infinite variance)

TODO: general properties of the implementation: Python, uses numpy, can be configured inside the source code, can be asked to draw plots interactively

TODO: example pictures from the algorithm

TODO: problems encountered (negative variance, infinities, unsensible normalisation constants)... but the algorithm converges well. Include some statistics of convergence properties (or, say, distance of the estimated mean from the true one)

References

[1] BROMILEY, P. Products and convolutions of Gaussian distributions. Internal Report 2003-003, TINA Vision, 2003.

[2] FACKLER, P. L. Notes on matrix calculus, 2005.

[3] MINKA, T. P. Expectation Propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence* (2001), Morgan Kaufmann Publishers Inc., pp. 362–369.

Expectation Propagation for a Probit Regression Model

Ondřej Dušek

May 31, 2013

1 The probit model

Suppose we have independent data points $\mathbf{x}_{i=1}^{n}$, each consisting of d features, thus building a matrix $X \in \mathbb{R}^{n \times d}$. Each data point $\mathbf{x}_i$ has a label $y_i \in \{-1, 1\}, i = 1 \dots n$, which gives a vector of labels $\mathbf{y}$.

We want to model this data using a *probit model*: $P(y_i|\mathbf{x}_i, \mathbf{w}) = \Phi(y_i \cdot \mathbf{w}^T \mathbf{x}_i)$. Φ denotes a standard Gaussian cumulative distribution function, i. e. $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} \, \mathrm{d}t$.

In order to obtain a posterior estimate of the unknown parameters $\mathbf{w}$ ("weights vector") given our data $\{X, \mathbf{y}\}$, we use the standard Bayesian estimation scheme:

$$\text{posterior} = \frac{\text{prior} \cdot \text{likelihood}}{\text{normalization}} \tag{1}$$

Since we can choose our own prior on $\mathbf{w}$, we take the path of least resistance. We select a Gaussian prior on $\mathbf{w}$ with a zero mean and known variance v_0 and assume independence of w_j in the individual dimensions:

$$P(\mathbf{w}) \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}|\mathbf{0}, I \cdot v_0) \stackrel{\text{indep}}{=} \prod_{j=1}^{d} \mathcal{N}(w_j|0, v_0) \tag{2}$$

The form of likelihood is given by the probit model (remember that the data points are assumed to be independent, identically distributed):

$$P(\mathbf{y}|X, \mathbf{w}) \stackrel{\text{iid}}{=} \prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \Phi(y_i \cdot \mathbf{w}^T \mathbf{x}_i) \tag{3}$$

The posterior then has the following form (where Z is a normalization constant):

$$P(\mathbf{w}|X, \mathbf{y}) = \frac{1}{Z} \cdot P(\mathbf{w}) \cdot \prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w}) \tag{4}$$

Computing the posterior in this form is not tractable due to the product of probits in the likelihood. Therefore, we must approximate the posterior by a simpler distribution. We can use the Expectation Propagation (EP) algorithm to do that.

2 Expectation propagation algorithm

The EP algorithm assumes that we are given a joint distribution $P(X, \mathbf{y}, \mathbf{w})$ over observed data and unknown parameters, i.e. likelihood $\cdot$ prior, in the form of a product of factors:

$$P(X, \mathbf{y}, \mathbf{w}) = \prod_{i} f_i(\mathbf{w}) \tag{5}$$

The EP then tries to find an approximation $q(\mathbf{w})$ of the true posterior distribution $p(\mathbf{w}) \stackrel{\text{def}}{=} P(\mathbf{w}|X, \mathbf{y})$ by minimizing the Kullback-Leibler (KL) Divergence:

$$KL(p(\mathbf{w})||q(\mathbf{w})) = \int_{-\infty}^{\infty} p(\mathbf{w}) \log \left(\frac{p(\mathbf{w})}{q(\mathbf{w})} \right) \mathrm{d}\mathbf{w} \tag{6}$$

It does so by gradually refining one of the factors $\hat{f}_i(\mathbf{w}), i = 1 \dots n$ while keeping rest of $q(\mathbf{w})$ fixed. This is repeated until convergence (i.e. until the refined factors are undistinguishable from the original factors) and requires several passes over all factors in general.

The general flow of the algorithm looks like this:

1. Select a form of a distribution from the *exponential family* for your approximate posterior $q(\mathbf{w})$. It must be possible to express it as a product of approximate factors $\hat{f}_i(\mathbf{w})$, each approximating a factor $f_i(\mathbf{w})$ of the true posterior. We use the exponential family since it works nicely with KL divergence minimization (see below).
2. Initialize the approximate factors to some (arbitrary, but reasonable) values. You now have the first approximation of the posterior.
3. In several passes, select one factor $\hat{f}_i(\mathbf{w})$ to refine; keep the rest of the factors intact:
   (a) Take factor $\hat{f}_i(\mathbf{w})$ out of the current posterior approximation $q(\mathbf{w})$ to create a *cavity distribution* $q^{\setminus i}(\mathbf{w})$.
   (b) Now create a new approximation $\hat{f}_i^{\text{new}}(\mathbf{w})$ of the true factor $f_i(\mathbf{w})$ by minimizing:
   $$KL(f_i(\mathbf{w}) \cdot q^{\setminus i}(\mathbf{w})||\hat{f}_i(\mathbf{w}) \cdot q^{\setminus i}(\mathbf{w})) \tag{7}$$
   Note that we are not minimizing the distance to the true posterior, but to a distribution composed of the exact factor $f_i(\mathbf{w})$ and approximations of the rest, i.e. we are approaching the true factor in the context of our current approximation.
   The exponential family is very convenient here since we may use *moment matching*: we just compute the *sufficient statistics*[1] of the target distribution $f_i(\mathbf{w}) \cdot q^{\setminus i}(\mathbf{w})$ and use them for our approximation $\hat{f}_i(\mathbf{w}) \cdot q^{\setminus i}(\mathbf{w})$. If an approximation from the exponential family has the same sufficient statistics as the target distribution, it must have the lowest KL divergence.
   (c) Now replace your old posterior approximation with $q^{\text{new}}(\mathbf{w}) \propto \hat{f}_i^{\text{new}}(\mathbf{w}) \cdot q^{\setminus i}(\mathbf{w})$.
4. Repeat previous step until convergence.

3 Form of the approximation in the probit model

We now return to our probit model. We denote our posterior distribution on weights (4) as $p(\mathbf{w})$ and its individual factors as $f_i(\mathbf{w}), i = 0 \dots n$ (i.e. some functions of $\mathbf{w}$):

$$f_0(\mathbf{w}) \stackrel{\text{def}}{=} P(\mathbf{w}) \tag{8}$$

$$f_i(\mathbf{w}) \stackrel{\text{def}}{=} P(y_i|\mathbf{x}_i, \mathbf{w}) \quad i = 1 \dots n \tag{9}$$

Note that f_0 corresponds to the prior and $f_i, i = 1 \dots n$ correspond to the individual data points.

We now try to find an approximation $q(\mathbf{w})$ of $p(\mathbf{w})$. We choose the shape of $q(\mathbf{w})$ ourselves, the only requirement is that it has to be in the exponential family (see Section 2). A Gaussian with independent dimensions is the best way to keep things simple:

$$q(\mathbf{w}) \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}|\mathbf{m}, I \cdot \mathbf{v}) \stackrel{\text{indep}}{=} \prod_{j=1}^{d} \mathcal{N}(w_j|m_j, v_j) \tag{10}$$

[1] Sufficient statistics is a set of moments that uniquely define a distribution from the exponential family. For Gaussians, it is mean and variance.

Note that $\mathbf{m} = \{m_j\}_{j=1}^d$ and $\mathbf{v} = \{v_j\}_{j=1}^d$ denote means and variances in the individual dimensions.

We also want our $q(\mathbf{w})$ to be a product of factors of a similar form to (8,9), but simpler. We thus denote:

$$q(w) \stackrel{\text{def}}{=} \frac{1}{Z} \prod_{i=0}^{n} \hat{f}_i(w) \tag{11}$$

Where:

$$\hat{f}_0(\mathbf{w}) \stackrel{\text{def}}{=} f_0(\mathbf{w}) = P(\mathbf{w}) \tag{12}$$

$$\hat{f}_i(\mathbf{w}) \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}|\mathbf{m}_i, I \cdot \mathbf{v}_i) \cdot \mathbf{s}_i = \prod_{j=1}^{d} \mathcal{N}(\mathbf{w}_j|\mathbf{m}_{ij}, I \cdot \mathbf{v}_{ij}) \cdot s_{ij} \quad i = 1 \dots n \tag{13}$$

I.e. we use the exact prior (since it is a plain Gaussian) and choose the other factors $\hat{f}_i(\mathbf{w}), i = 1 \dots n$ as unnormalized Gaussians with independent dimensions. We know that the original factors $f_i(\mathbf{w}), i = 1 \dots n$ are not normalized with respect to $\mathbf{w}$ (since they are normalized with respect to y_i), but want them to have a simple form. We therefore use a Gaussian multiplied by a "de-normalization constant" s_{ij}.

We now aim to find $q(\mathbf{w})$ with such parameters $\mathbf{m}, \mathbf{v}$ that it is as close to $p(\mathbf{w})$ as possible. This is the task of the EP algorithm.

4 EP initialization step

We initialize our approximation $q(\mathbf{w})$ by setting $\hat{f}_0(\mathbf{w})$ to the prior and $\hat{f}_i(\mathbf{w})$ to uniform distributions.[2] The parameters of the approximate factors then look as follows:

$$m_{0j} := 0, \quad v_{0j} := v_0 \qquad j = 1 \dots d \tag{14}$$

$$m_{ij} := 0, \quad v_{ij} := \infty \qquad j = 1 \dots d, \quad i = 1 \dots n \tag{15}$$

Now our posterior approximation is in fact equal to our prior (if we view it as prior $\cdot \prod_{i=1}^n$ uniform).

5 Refining one factor

We select an approximate factor $\hat{f}_i(\mathbf{w})$ to be refined. The order of factors selected for refining is arbitrary and all factors should be refined multiple times.

5.1 Computing the cavity distribution

First, we compute the *cavity distribution* from our current posterior approximation $q(\mathbf{w})$ and the current approximate factor $\hat{f}_i(\mathbf{w})$:

$$q^{\setminus i}(\mathbf{w}) = \frac{q(\mathbf{w})}{\hat{f}_i(\mathbf{w})} \tag{16}$$

Since $q(\mathbf{w})$ and $\hat{f}_i(\mathbf{w})$ are both Gaussian from (10, 13), we can use the formulas for Gaussian identities to obtain an (unnormalized) Gaussian shape of $q^{\setminus i}(\mathbf{w})$:

$$q^{\setminus i}(\mathbf{w}) \propto \mathcal{N}(\mathbf{w}|\mathbf{m}^{\setminus i}, \mathbf{v}^{\setminus i}) \stackrel{\text{indep}}{=} \prod_{j=1}^{d} \mathcal{N}(w_j|m_j^{\setminus i}, v_j^{\setminus i}) \tag{17}$$

Where:

$$v_j^{\setminus i} = (v_j^{-1} - v_{ij}^{-1})^{-1} \tag{18}$$

$$m_j^{\setminus i} = v_j^{\setminus i}(v_j^{-1} m_j - v_{ij}^{-1} m_{ij}) \tag{19}$$

Note that m_{ij}, v_{ij} refer to the current approximation of $\hat{f}_i(\mathbf{w})$ and m_j, v_j refer to the current approximation of $q(\mathbf{w})$.

[2] Or as close to uniform distributions as we can get in practice since $\hat{f}_i(\mathbf{w})$ are assumed to be Gaussian.

5.2 Minimizing KL-divergence

Having fixed our cavity distribution, we want to minimize the KL divergence of our factor approximation in the context of the cavity distribution (7) to obtain a new, better approximation of the posterior, q^{new}. We have:

$$q^{\text{new}}(\mathbf{w}) = \arg \min_{q' \propto \hat{f}_i(w) q^{\setminus i}(\mathbf{w})} KL\left(\frac{1}{Z_i} f_i(\mathbf{w}) q^{\setminus i}(\mathbf{w}) \;||\; q'\right) \tag{20}$$

The KL-divergence for distributions in the exponential family is minimized by moment matching: setting the *sufficient statistics*, i.e. mean and variance in our case, equal to those of the distribution we want to approximate.

To do this, we will use the following clever formulas (from José's slide 15) for the moments of a Gaussian multiplied by some arbitrary factor. Given a distribution $r(\mathbf{x})$ in the following form:

$$r(\mathbf{x}) = \frac{1}{Z} t(\mathbf{x}) \mathcal{N}(\mathbf{x}|\mu, \Sigma) \text{ and } Z = \int t(\mathbf{x}) \mathcal{N}(x|\mu, \Sigma)\, \mathrm{d}\mathbf{x} \tag{21}$$

We can express its mean and variance as:

$$\mathbb{E}_r[\mathbf{x}] = \mu + \Sigma \cdot \frac{\partial \log Z}{\partial \mu} \tag{22}$$

$$\mathbb{E}_r[\mathbf{x}\mathbf{x}^T] - E_r[\mathbf{x}](E_r[\mathbf{x}])^T = \Sigma - \Sigma \cdot \left(\frac{\partial \log Z}{\partial \mu} \left(\frac{\partial \log Z}{\partial \mu} \right)^T - 2 \frac{\partial \log Z}{\partial \Sigma} \right) \cdot \Sigma \tag{23}$$

As the clever formulas are not clever enough to rid us of the normalizing constant, we must first compute Z_i:

$$Z_i = \int f_i(\mathbf{w}) q^{\setminus i}(\mathbf{w})\, \mathrm{d}\mathbf{w} = \int P(y_i|\mathbf{x}_i, \mathbf{w}) q^{\setminus i}(\mathbf{w})\, \mathrm{d}\mathbf{w} = \int \Phi(y_i \cdot \mathbf{w}^T \mathbf{x}_i) \prod_{i=1}^{d} \mathcal{N}(w_j | m_j^{\setminus i}, v_j^{\setminus i})\, \mathrm{d}\mathbf{w} \tag{24}$$

$$= \Phi\left(\frac{y_i \cdot \sum_{j=1}^{d} m_j^{\setminus i} x_{ij}}{\sqrt{\sum_{j=1}^{d} v_j^{\setminus i} x_{ij}^2 + 1}} \right) \tag{25}$$

If you know how we got the exact result, let me know. I don't. José just said it's relatively simple.

Now we can just fill in our values into (22, 23), using the value of Z_i computed in (25). We obtain the mean and the variance of the new approximate posterior $q^{\text{new}}(\mathbf{w})$:

$$m_j^{\text{new}} = m_j^{\setminus i} + v_j^{\setminus i} \cdot \frac{\partial \log Z_i}{\partial m_j^{\setminus i}} \tag{26}$$

$$v_j^{\text{new}} = v_j^{\setminus i} - \left(v_j^{\setminus i}\right)^2 \left(\left(\frac{\partial \log Z_i}{\partial m_j^{\setminus i}} \right)^2 - 2 \frac{\partial \log Z_i}{\partial v_j^{\setminus i}} \right) \tag{27}$$

5.3 Obtaining the new approximate factor

We now have the new approximate posterior $q^{\text{new}}(\mathbf{w})$ and need to obtain our new approximate factor $\hat{f}_i^{\text{new}}(\mathbf{w})$ for later use. We use an equation obtained from (20) by forcing Z_i as our new normalization constant:

$$q^{\text{new}}(\mathbf{w}) := \frac{1}{Z_i} \hat{f}_i(\mathbf{w}) q^{\setminus i}(\mathbf{w}) \tag{28}$$

$$\hat{f}_i(\mathbf{w}) = Z_i \frac{q^{\text{new}}(\mathbf{w})}{q^{\setminus i}(\mathbf{w})} \tag{29}$$

The parameters $m_{ij}^{\text{new}}, v_{ij}^{\text{new}}, s_{ij}^{\text{new}}$ are obtained from the Gaussian identities formulas:

$$v_{ij}^{\text{new}} = \left(v_j^{-1} - \left(v_j^{\backslash i} \right)^{-1} \right)^{-1} \tag{30}$$

$$m_{ij}^{\text{new}} = v_{ij}^{\text{new}} \cdot \left(m_j v_j^{-1} - m_j^{\backslash i} \left(v_j^{\backslash i} \right)^{-1} \right) \tag{31}$$

$$s_{ij}^{\text{new}} = Z_i \cdot C_j, \text{ where} \tag{32}$$

$$C_j = \sqrt{\frac{v_{ij}^{\text{new}} v_j^{\backslash i}}{(2\pi)^d v_j}} \exp\left(-\frac{1}{2} \left(m_j^2 v_j^{-1} - \left(m_j^{\backslash i} \right)^2 \left(v_j^{\backslash i} \right)^{-1} - \left(m_{ij}^{\text{new}} \right)^2 \left(v_{ij}^{\text{new}} \right)^{-1} \right) \right) \tag{33}$$

We can now use $\hat{f}_i^{\text{new}}(\mathbf{w})$ and $q^{\text{new}}(\mathbf{w})$ in the next iterations.

1. Gibbs sampling in probit regression

1.1 Introduction

In this short document we would like to introduce *Bayesian attitude* to *probit regression*. In *Bayesian theory* we are trying to derive a distribution for an unknown random variable from already known random variables.

In our example we would like to know a posterior distribution for the labels Y in the probit regression model. We will derive the posterior distribution from the features X and the prior on weights w. We will estimate the posterior distribution from observations $\{1, \ldots, N\}$ where we saw label y_i as well as features x_i for each of observation i. We also suppose that each observation from $\{1, \ldots, N\} \cup \{new\}$ are *iid*.

In fact, we are interested in predictive distribution for the label y_new for the unseen features x_new based on the knowledge of posterior distribution. Unfortunately, it is typical that the computation of the predictive as well as posterior distribution is intractable or even impossible. We will use *Gibbs sampling* as a approximation method of drawing samples from the posterior distribution. Obtaining samples from posterior distribution for Y allows us to approximate posterior distribution and later compute predictive distribution for random variable y_{new}.

1.2 Probit model

What is a probit model?

> Probit model is a type of regression where the dependent variable can only take two values.

[Wikipedia, 2013b] In our example, the values will be $\{0, 1\}$.

Definition 1.1. In **probit model** we assume that we can describe binary class variable y_i as function of features x_i and parameters w:

$$\forall i \in \{1, \ldots, N\} \Pr(y_i = 1 \mid x_i) = \Phi(x_i' * w) \tag{1.1}$$

$$\text{or as vectors: } \Pr(Y = 1 \mid X) = \Phi(X' * w) \tag{1.2}$$

where Φ is CDF of standard normal distribution. [1]

For given features x_i we have the distribution $P(y_i = 1 \mid x_i, w)$ which tells us how is probably that the observation falls in class 1 and not in class 0 given the features x_i and weights w.

Usually, the parameters w are estimated by maximum likelihood. Denote them w^*. For new features x_{new} we would give label y_{new} according weights $w*$

[1] I will be using x' instead of x^T as mark for vector or matrix transposition.

and the features x_{new}. The problem is that we do not consider any other values for $w*$ even if they are just slightly less probably than $w*$.

However, in Bayesian theory we put prior on w, which means that we consider every possible value for weights w, because w is a random variable. Instead of computing maximum likelihood from already seen data, we compute posterior distribution in equation 1.3.

$$P(Y \mid X, w) \tag{1.3}$$

In our case we suppose that w is normally distributed.

$$w \sim N(\mu_0, \Sigma_0) \tag{1.4}$$

Usually, the posterior probability 1.3 is estimated from the likelihood probability as

$$P(Y \mid X, w) = \frac{P(w \mid X, Y)}{P(w)}, \tag{1.5}$$

where we know $P(w)$ from the prior 1.4. With the posterior probability we may be able to compute predictive distribution $P(x_{new} \mid X) = \int_w P(x_{new}) * P(w \mid X) \mathrm{d}w$.

The problem is that we are not able to compute the posterior efficiently. Not even for the latent variable model, which we will introduce in section 1.3.

However, with Gibbs sampler and latent model we will be able to draw samples from posterior distribution.

Let's recapitulate what is what in probit model on example from Figure 1.1. The weights, w, are d-dimensional radnom variable. In case of Figure 1.1 $d = 2$.

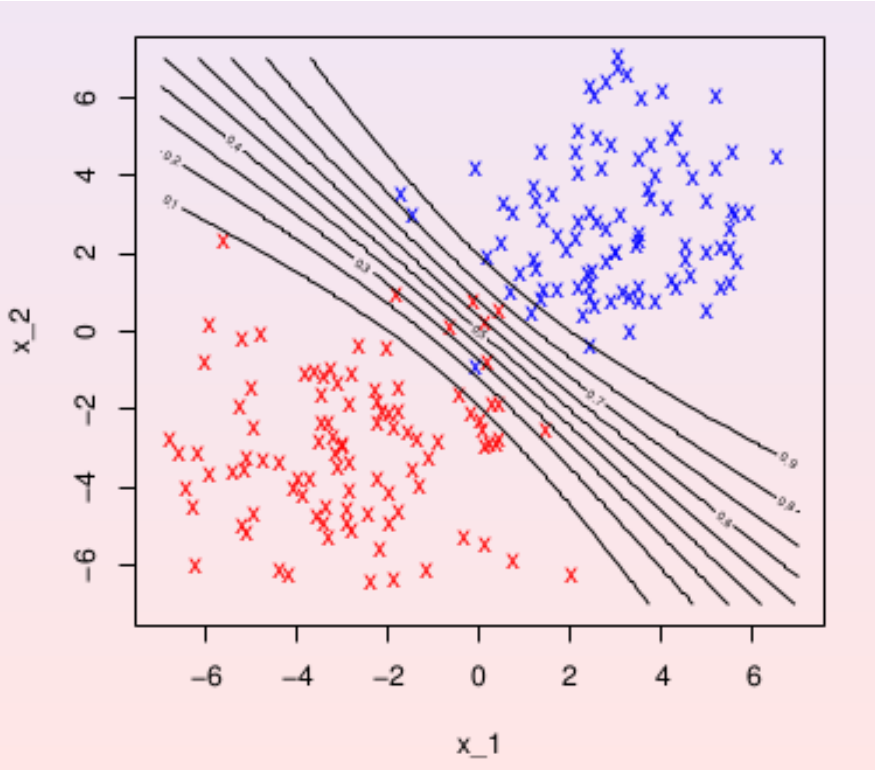

Figure 1.1: Probit regression in dimensions x_1 and x_2. The separation line is determined by random variable w. As expected, the separation line is located with the highest probability between the two differently labeled clusters.

The X is $(d * N)$-dimensional matrix, where column i contains d-dimensional features x_i of observation i. The Y is N-dimensional random vector with output values $y_i \in \{0, 1\}$, where N is the number of already seen data (x_i and y_i). The observation x_i on coordinates x_1_i, x_2_i is labeled with 1 if it is blue or 0 if it is red.

In the Figure 1.2 we are illustrating on graphical model the Bayesian attitude to the probit model. As you can see each observation y_i of N observation depends on its features xi and also on the weights w. We put prior on weights as random variable according a distribution with parameters μ_0 and Σ_0. The graphical also models depicts that we observe N times the features x_i and the class label y_i for different iid observations i.

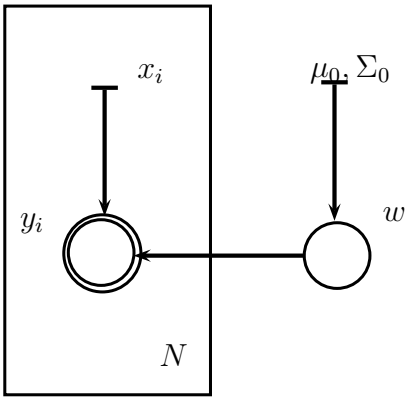

Figure 1.2: Graphical model for Probit

1.3 Introducing latent variable $\widetilde{Y}$

In fact, the model with latent variable is just technical trick. Later, in section 1.4.1 we will find out that the Gibbs sampler needs to draw samples from marginal distributions for each variable of joint distribution we want to sample from. In the probit model we do not know how to compute the marginal distributions.

However we are able to draw samples from marginal distributions of the model with latent variable, which we will introduce in this section. Furthermore, we can easily convert the latent model to probit model. So if we can draw samples from latent model we can draw samples from the probit model, which is our goal.

Let us introduce the latent model and remind priors on distribution for the random variables.

$$\widetilde{Y} = X' * w + \varepsilon, \tag{1.6}$$

where X is the matrix of features,
we suppose that weights are distributed according the equation 1.4
and we suppose noisy labeling without loss of generality [2]

$$\varepsilon \sim N(0, 1). \tag{1.7}$$

[2]Without loss of generality we can assume $N(0, 1)$ instead of $\varepsilon \sim N(0, S)$. If we needed to estimate weights w for $N(0, S)$ introduced in the graphical model we will estimate weights w^h for $N(0, 1)$ and set $w = S * w^h$

Now realize that we can draw samples from the probit model if we can draw samples from $\widetilde{y_i}$ by applying following rule.

We claimed that we can derive the distribution of probit model from distribution of the latent model:

$$y_i = \begin{cases} 1 & \text{if } \widetilde{y_i} > 0 \text{ i.e. } -\varepsilon < x_i' * w, \\ 0 & \text{otherwise.} \end{cases} \tag{1.8}$$

So, let us prove it:

$$\begin{aligned} P(y_i = 1 \mid x_i) = P(\widetilde{y_i} > 0) &= P(x_i' * w + \varepsilon > 0) && (1.9)\\ &= P(\varepsilon > -x_i' * w) && (1.10)\\ &= P(\varepsilon > -x_i' * w) && (1.11)\\ &\text{by symmetry of normal distribution} && (1.12)\\ &= P(\varepsilon < x_i' * w) && (1.13)\\ &= \Phi(x_i' * w) && (1.14) \end{aligned}$$

1.4 Sampling from model with latent variable

Let us shortly recapitulate facts. We know that feature vectors x_i are represented compactly in matrix X. There is label $y_i \in \{0, 1\}$ for each of vector x_i from the features vectors. Labels are compactly represented as Y. We assume that ε, w random variables are normally distributed and we do not know parameters for w yet.

Firstly, we will introduce the Gibbs sampling algorithm. Secondly, we will derive the marginal distributions needed by Gibbs sampling. Finally, at the end we will summarize what can we deduce from the samples about distributions which interest us.

Gibbs sampler

> The point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. [Wikipedia, 2013a]

We will describe Gibbs sampling algorithm and claim some facts. We will not proved the properties. See the references into literature for details. However, we will derive the in detail the marginal distributions which are necessary for running Gibbs sampler.

How does Gibbs sampling work?

Suppose we want to sample from $Z = \{z_1, z_2, \ldots, z_l\}$ with distribution $P(z_1, z_2, \ldots, z_l)$. Let us denote the k^{th} sample from Z by $Z^k = \{z_1^k, z_2^k, \ldots, z_l^k\}$. The Gibbs algorithm will sample Z^k according pseudo code below. For large enough k the distribution, from which the samples $Z^k, Z^{k+1}, \ldots$ are drawn, will converge to distribution $P(z_1, z_2, \ldots, z_l)$.

Algorithm 1.1 Gibbs algorithm

```
k=0
Z^k = InitValues()                          ▷ Somehow initialize the vector : Z^k
while Samples did not converge do
    k++                                     ▷ We are sampling k^th sample
    for j ∈ {1, ..., l} do
        z_j^k ~ P(z_j^k | z_1^k, z_2^k, ..., z_{j-1}^k, z_{j+1}^{k-1}, ..., z_n^{k-1})   ▷ Sampling z_j^k
    end for
end while
```

In following section we will focus on how to sample from the marginal distributions. Now let us deal with the unspecified parts in the algorithm description apart from sampling from marginal distributions.

Firstly, initializing the values is not very problematic. Gibbs sampling is guaranteed to converge for any values. On the other hand, reasonable settings like mean or mode, if we know them, gives faster convergence than outliers. [Walsh, 2004]

Secondly, the stopping condition for Gibbs algorithm is much harder problem. There are several heuristics like autocorrelation metrics or measuring discrepancy between sampled and the desired distribution which are trying to detect whether the samples have converged to stable distribution. It can be shown that for Gibbs sampler the stable distribution is in fact the desired $P(Z)$ distribution. [George, 1992]

1.4.1 Sampling from marginal distribution in Probit example

At the beginning we will give again short and dense overview of facts in few equations. Later we will compute marginals for joint probability $P(w, \widetilde{Y} \mid X, Y)$. The marginals which we need to compute are

- $P(\widetilde{Y} \mid w, X, Y)$.
- $P(w \mid X, \widetilde{Y}, Y)$.

We assume probit model for Y binary random variable.

$$Y = \Phi(X' * w) \tag{1.15}$$

where is w with parameters from equation 1.4. Further, without loss of generality we can assume that the latent model is normally distributed with covariance 1 as described in equation 1.16. [3]

$$\widetilde{y_i} \mid x_i, w \overset{iid}{\sim} N(x_i' * w, 1) \tag{1.16}$$

[3] We can compensate different variance from 1 by setting different variance in prior on weights w *TODO: add equation*

From transformation between labels y_i and latent random variable $\widetilde{y_i}$ in equation 1.8 we derive distribution described in terms of truncated normal. [4]

$$y_i \mid x_i, w \overset{ind}{\sim} 1(\widetilde{y_i} \geq 0) \tag{1.17}$$

Theorem 1.2. *All full marginal distributions with variables w, $\widetilde{Y}$ for joint probability $P(w, \widetilde{Y} \mid X, Y)$ are distributed according*

1. $$w \mid \widetilde{Y}, X, Y \sim N(\hat{w}, \hat{\Sigma}) \tag{1.18}$$
   where $\hat{\Sigma} = (\Sigma_0^{-1} + X' * X)^{-1}$ *and* $\hat{w} = \hat{\Sigma}(\Sigma_0^{-1} * \mu_0 + X' * \widetilde{y})$
2. $$\widetilde{Y} \mid w, X, Y \text{ has density } \prod_{i=1}^{N} Z * N(x_i, 1) * (1(\widetilde{y_i} \geq 0))^{y_i} * (1(\widetilde{y_i} < 0))^{1-y_i} \tag{1.19}$$
   where Z is normalizing constant.

Proof. Let us prove the marginal distribution from equation 1.18 We start with

$$P(w \mid \widetilde{Y}, X, Y) \tag{1.20}$$

and by applying the Bayes rule and omitting the denominator we get

$$\propto P(\widetilde{Y}, Y \mid X, w) * P(w). \tag{1.21}$$

We further expand the first probability by chain rule

$$\propto P(\widetilde{Y} \mid X, w) * P(Y \mid \widetilde{Y}, X, w) * P(w). \tag{1.22}$$

The middle term will disappear, because it is our deterministic way how to convert hidden model to probit model.

$$\propto P(w) * P(\widetilde{Y} \mid X, w) \tag{1.23}$$

kWe can write down the prior on weights. Because of *iid* distributed observations we can write the product of their distributions at points $\widetilde{y_i}$. Further, we can *TODO: without loss of generality assume that the latent model has variance 1 for each observation* .

$$= N(\mu_0, \Sigma_0) * \prod_{i=1}^{N} N(\widetilde{y_i} \mid w' * x_i, 1) \tag{1.24}$$

$$= N(w \mid \hat{w}, \hat{\Sigma}) \tag{1.25}$$

TODO: explain it where $\hat{\Sigma} = (\Sigma_0^{-1} + X'X)^{-1}$ and $\hat{w} = \hat{\Sigma}(\Sigma_0^{-1}\mu_0 + X'y)$ □

[4] More about truncated normal distribution on Wikipedia

Proof. Let us prove the second marginal distribution from equation 1.19

$$P(\widetilde{Y} \mid X, Y, w) \propto \tag{1.26}$$

We continue by using Bayes rule in form $P(A \mid C, B) = \frac{P(B|A,C)*P(A|C)}{P(B)}$

$$\propto P(\widetilde{Y} \mid X, w) * P(Y \mid X, \widetilde{Y}, w) \tag{1.27}$$

Now we can substitute for the conditional probabilities with distributions

$$P(\widetilde{Y} \mid X, w) = \prod_{i=1}^{N} N(\widetilde{y_i} \mid w^T * x_i, 1) \text{ and } P(Y \mid X, \widetilde{Y}, w) = \prod_{i=1}^{N} (1_{(\widetilde{y_i} \geq 0)})^{y_i} * (1_{(\widetilde{y_i} < 0)})^{1-y_i}. \tag{1.28}$$

It leads to

$$\propto \prod_{i=1}^{N} N(\widetilde{y_i} \mid w^T * x_i, 1) * \prod_{i=1}^{N} (1_{(\widetilde{y_i} \geq 0)})^{y_i} * (1_{(\widetilde{y_i} < 0)})^{1-y_i}. \tag{1.29}$$

It means that distributions $P(\widetilde{y_i} \mid x_i, y_i, w)$ are conditionally independent with truncated normal distribution. □

1.5 Summary

At the beginning we wanted to compute the posterior probability 1.3. We come up with the idea of Gibbs sampling which has the advantage that it needs only all full marginal distributions to sample from the joint distribution or some of the marginals. The marginal distributions are generally in more simpler form. Further,sampling is usually much more feasible approximation than other alternatives.

In our example, we used a latent variable model, because we were not able to derive the distributions for the marginals in probit model. We computed the marginals for latent variable model and then we used the Gibbs sampling method to draw sample from the posterior distribution.

Usually, we are not satisfied with the posterior distribution and we want to compute the predictive distribution. From the samples we estimate the parameters of the posterior distribution and then use it to compute predictive distribution. *TODO: PREDICTIVE Distribution*

References

- Git repository for this article `https://github.com/bayesian-inference/notes`
- This article was written for the Bayesian Inference course at MFF UK
- This article is based on Sara's Wade talk for the Bayesian Inference course recorded on Youtube
- The Figure 1.1 is taken from slides of Jose Miguel Hernandez-Lobato given for Bayesian Inference course

Bibliography

[George, 1992] George, G. C. E. I. (1992). Explaining the gibbs sampler. [Online; accessed 22-July-2013].

[Walsh, 2004] Walsh, B. (2004). Markov chain monte carlo and gibbs sampling. [Online; accessed 22-July-2013].

[Wikipedia, 2013a] Wikipedia (2013a). Gibbs sampling — Wikipedia, the free encyclopedia. [Online; accessed 22-July-2013].

[Wikipedia, 2013b] Wikipedia (2013b). Probit model — Wikipedia, the free encyclopedia. [Online; accessed 22-July-2013].