3 Variational inference

This chapter covers

- Introducing KL variational inference
- Mean-field approximation
- Image denoising in the Ising model
- Mutual information maximization

In the previous chapter, we covered one of the two main camps of Bayesian inference: Markov chain Monte Carlo. We examined different sampling algorithms and approximated the posterior distribution using samples. In this chapter, we will discuss the second camp of Bayesian inference: variational inference. *Variational inference* (VI) is an important class of approximate inference algorithms; its basic idea is to choose an approximate distribution $q(x)$ from a family of tractable or easy-to-compute distributions with trainable parameters and then make this approximation as close as possible to the true posterior distribution $p(x)$.

As we will see in the mean-field section, the approximate $q(x)$ can take on a fully factored representation of the joint posterior distribution. This factorization significantly speeds up computation. We will introduce KL divergence and use it as a way to measure the closeness of our approximate distribution to the true posterior. By optimizing KL divergence, we will effectively convert VI into an optimization problem. In the following section, we will derive the evidence lower bound (ELBO) and interpret it in three different ways, which will become handy during our implementation of mean-field approximation for image denoising. The image denoising algorithm was selected because it illustrates the concepts discussed in this section via a visual example.

3.1 KL variational inference

We can use KL divergence to measure a distance between probability distributions. This is particularly useful when making an approximation to the target distribution, since we want to find out how close our approximation is. Let $q(x)$ be our approximating distribution and $p(x)$ be the target posterior distribution. Then, the reverse KL is defined as follows.

$$KL(q||p) = \sum_x \overbrace{q(x)}^{\textbf{Approximating distribution}} \underbrace{\log \frac{q(x)}{p(x)}}_{\textbf{Log ratio of approximate to actual}} \tag{3.1}$$

Consider a simple example, in which our target distribution is a standard univariate normal distribution $p(x) \sim N(0, 1)$ and our approximating distribution is a univariate normal with a mean µ and variance σ^2: $q(x) \sim N(\mu, \sigma^2)$. We can then compute $KL(q||p)$ as follows.

$$\begin{aligned}
KL(q||p) &= \int q(x) \log \frac{q(x)}{p(x)} \\
&= -\int q(x) \log p(x) + \int q(x) \log q(x) \\
&= -\int q(x) \left[-\frac{1}{2} \log 2\pi - \frac{1}{2} x^2\right] \\
&\quad + \int q(x) \left[-\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(x-\mu)^2\right] \\
&= \left[\frac{1}{2} \log 2\pi + \frac{1}{2}(\sigma^2 + \mu^2)\right] + \left[-\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2}\right] \\
&= -\frac{1}{2}\left(1 + \log \sigma^2 - \mu^2 - \sigma^2\right) \qquad (3.2)
\end{aligned}$$

We can visualize how $KL(q||p)$ changes as we vary the parameters of our approximate distribution $q(x)$. Let's fix $\sigma^2 = 4$ and vary the mean $\mu \in [-4, 4]$. We obtain figure 3.1.

Notice that the KL divergence is nonnegative and is smallest when $\mu = 0$ (i.e., the mean of the approximating distribution is equal to the mean of our target distribution $p(x) \sim N(0, 1)$). We can interpret KL divergence as a measure of distance between distributions.

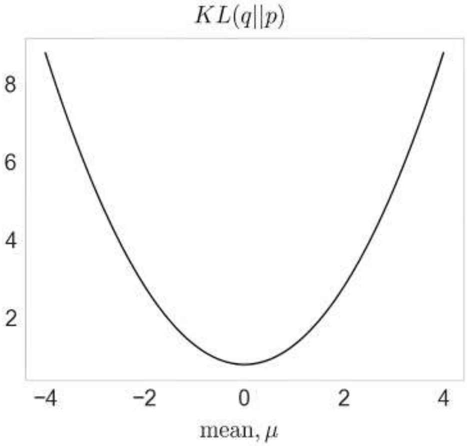


Figure 3.1 KL(q||p) for p(x)~N(0,1) and q(x)~N(μ,4)

Let $\tilde{p}(x) = p(x)Z$ be the unnormalized distribution, and then consider the following objective function.

$$\begin{aligned} J(q) &= KL(q||\tilde{p}) \\ &= \sum_x q(x)\log\frac{q(x)}{p(x)Z} = \sum_x q(x)\log\frac{q(x)}{p(x)} - \log Z \\ &= KL(q||p) - \log Z \end{aligned} \tag{3.3}$$

Since KL divergence is nonnegative, $J(q)$ is an upper bound on the marginal likelihood.

$$J(q) = KL(q||p) - \log Z \geq -\log Z = -\log p(D) \tag{3.4}$$

When $q(x)$ equals the true posterior $p(x)$, the KL divergence vanishes. The optimal value $J(q^*)$ equals the log partition function, and for all other values of q, it yields a bound. $J(q)$ is called the *variational free energy* and can be written as follows.

$$\min_q J(q) = E_q[\log q(x)] + E_q[-\log\tilde{p}(x)] = -H(q) + E_q[E(x)] \tag{3.5}$$

The variational objective function in equation 3.5 is closely related to energy minimization in statistical physics. The first term acts as a regularizer by encouraging maximum entropy, while the second term is the expected energy and encourages the variational distribution q to explain the data.

The reverse KL that acts as a penalty term in the variational objective is also known as *information projection* or *I-projection*. In the reverse KL, $q(x)$ will typically underestimate the support of $p(x)$ and will lock onto one of its modes. This is due to $q(x) = 0$ whenever $p(x) = 0$ to ensure the KL divergence stays finite. On the other hand, the forward KL, known as *moment projection* or *M-projection* is zero avoiding $q(x)$ and will overestimate the support of $p(x)$, as shown in figure 3.2.

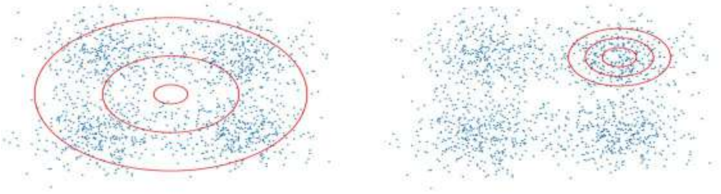


Figure 3.2 Forward KL (left) $q(x)$ overestimates the support, while reverse KL (right) $q(x)$ locks onto a mode.

Figure 3.2 shows samples from a 2D Gaussian mixture with 4 components $p(x)$) as well as density ellipses of approximating distribution $q(x)$. We can see that optimizing forward KL leads to $q(x)$ centered at zero (in the low-density region), as we over-estimate the support of $q(x)$. On the other hand, optimizing reverse KL leads to $q(x)$ centered at one of the four modes of the Gaussian mixture.

We can use Jensen's inequality to derive the ELBO, an objective that we can maximize to learn the variational parameters of our model. Let x be our data and z be the latent variables, and then we can derive our ELBO objective as follows.

$$\begin{aligned}
\log p(x) &= \log \sum_z p(x,z) = \log \sum_z \frac{q(z)}{q(z)} p(x,z) \\
&= \log E_{q(z)}\left[\frac{p(x,z)}{q(z)}\right] \geq E_{q(z)}\left[\log \frac{p(x,z)}{q(z)}\right] \\
&= \underbrace{E_{q(z)}[\log p(x,z)]}_{\textbf{Energy term}} - \underbrace{E_{q(z)}[\log q(x)]}_{\textbf{Entropy term}} = \text{ELBO}
\end{aligned} \tag{3.6}$$

Notice that the first term is the average negative energy, and the second term is the entropy. Thus, a good posterior must assign most of its probability mass to regions of low energy (i.e., high joint probability density) while also maximizing the entropy of $q(z)$. Thus, variational inference, in contrast to the MAP estimator, prevents $q(z)$ from collapsing into an atom.

One form of ELBO emphasizes that the lower bound becomes tighter, as the variational distribution better approximates the posterior.

$$\begin{aligned}
\text{ELBO} &= E_{q(z)}\left[\log \frac{p(x,z)}{q(z)}\right] = E_{q(z)}\left[\log \frac{p(z|x)p(x)}{q(z)}\right] \\
&= \underbrace{-KL(q(z)||p(z|x))}_{\substack{\textbf{Distance between the approximating} \\ \textbf{q(z) and the posterior p(z|x)}}} + \log p(x)
\end{aligned} \tag{3.7}$$

Therefore, we can improve the ELBO by improving the model log evidence log $p(x)$ through the prior $p(z)$ or the likelihood $p(z|x)$ or by improving the variational posterior approximation $q(z)$.

Finally, we can write the ELBO as follows.

$$\text{ELBO} = \frac{1}{n}\sum_{i=1}^{n}\left[\underbrace{E_{q(z)}\left[\log p(x_i|z_i)\right]}_{\textbf{Sample likelihood}} - \overbrace{KL(q(z_i)||p(z_i))}^{\substack{\textbf{Distance between the approximating} \\ \textbf{q(z) and the prior p(z) for sample i}}}\right] \tag{3.8}$$

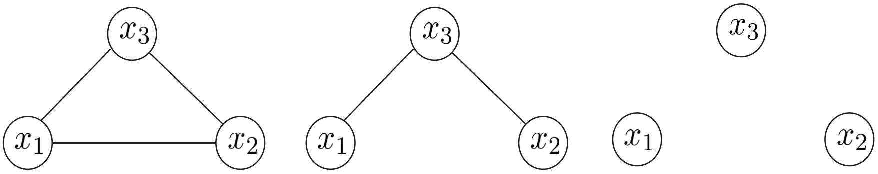
This version emphasizes a likelihood term for the `i`-th observation and KL divergence term between each approximating distribution and the prior. In all the preceding cases, the expectation with respect to $q(z)$ can be computed by sampling from our approximating distribution. Let's look at one of the most common variational approximations in the next section.

3.2 Mean-field approximation

One of the most popular forms of variational inference is the *mean-field approximation*, where we assume the posterior is a fully factorized approximation of the form.

$$q(x) = \prod_{i} q_i(x_i) \tag{3.9}$$

Here, we optimize over the parameters of each marginal distribution $q_i(x_i)$. We can visualize a fully factored distribution, as in figure 3.3.



True posterior $p(x_1, x_2, x_3)$

Structured approximation

Fully factorized approximation $q(x_1)\, q(x_2)\, q(x_3)$

Figure 3.3 True posterior (left), structured approximation (middle), and fully factored approximation (right)

For a distribution with three random variables, x_1, x_2, and x_3, we have the true posterior $p(x_1, x_2, x_3)$ that we are attempting to approximate by a fully factored distribution $q(x_1)\,q(x_2)\,q(x_3)$.

Our goal is to minimize variational free energy $J(q)$ or, equivalently, maximize the lower bound.

$$L(q) = -J(q) = \sum_x q(x) \log \frac{\tilde{p}(x)}{q(x)} \tag{3.10}$$

We can rewrite the objective for each marginal distribution q_j, keeping the rest of the terms as constants, as demonstrated in section 21.3 of *Probabilistic Machine Learning* by Kevin Murphy (2012).

$$\begin{aligned}
L(q_j) &= \sum_x \overbrace{\prod_i q_i(x_i)}^{\text{Mean-field approx}} \left[\log \tilde{p}(x) - \overbrace{\sum_k \log q_k(x_k)}^{\text{Mean-field approx}}\right] \\
&= \sum_{x_j} \sum_{x_{-j}} q_j(x_j) \prod_{i \neq j} q_i(x_i) \left[\log \tilde{p}(x) - \sum_k \log q_k(x_k)\right] \leftarrow \text{Factors out qj} \\
&= \sum_{x_j} q_j(x_j) \log f_j(x_j) \\
&\quad - \sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \left[\sum_{k \neq j} \log q_k(x_k) + \log q_j(x_j)\right] \\
&= \sum_{x_j} q_j(x_j) \log f_j(x_j) - \sum_{x_j} q_j(x_j) \log q_j(x_j) + \text{const} \leftarrow \text{Treats non-qj terms as constant}
\end{aligned} \tag{3.11}$$

Here, we defined the following.

$$\log f_j(x_j) = \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log \tilde{p}(x) = E_{-q_j}[\log \tilde{p}(x)] \tag{3.12}$$

Since we are replacing the values by their mean value, the method is known as the *mean field*. We can rewrite $L(q_j) = -KL(q_j \,||\, f_j)$ and, therefore, maximize the objective by setting $q_j = f_j$ or, equivalently, equation 3.13.

$$\log q_j(x_j) = \log f_j(x_j) = E_{-q_j}[\log \tilde{p}(x)] \tag{3.13}$$

Here, the functional form of q_j will be determined by the type of variables x_j and their probability model. We will use this result in the next section to derive the image denoising algorithm from scratch.

3.3 Image denoising in an Ising model

The Ising model is an example of a Markov random field (MRF) and has its origins in statistical physics. A Markov random field is a set of random variables with a Markov property described by an undirected graph, in which the nodes represent random variables and the edges encode conditional independence. The Ising model assumes we have a grid of nodes, where each node can be in one of two possible states. The state of each node depends on the neighboring nodes through interaction potentials. In the case of images, this translates to a smoothness constraint (i.e., a pixel prefers to be of the same color as the neighboring pixels). In the image denoising problem, we assume we have a 2D grid of noisy pixel observations of an underlying true image and we would like to recover the true image.

Let y_i be noisy observations of binary latent variables $x_i \in \{-1, +1\}$. We can write down the joint distribution as follows.

$$\begin{aligned} p(x, y) = p(x)p(y|x) &= \prod_{(s,t)\in E} \Psi_{st}(x_s, x_t) \prod_{i=1}^{n} p(y_i|x_i) \\ &= \prod_{(s,t)\in E} \exp\{x_s w_{st} x_t\} \prod_{i=1}^{n} N\left(y_i|x_i, \sigma^2\right) \end{aligned} \tag{3.14}$$

In equation 3.14, the interaction potentials are represented by Ψ_{st} for every pair of nodes x_s and x_t in a set of edges E, and the observations y_i are Gaussian with mean x_i and variance σ^2. Here, w_{st} is the coupling strength and assumed to be constant and equal to $J > 0$, indicating a preference for the same state as neighbors (i.e., the potential $\Psi(x_s, x_t) = \exp\{x_s J x_t\}$ is higher when x_s and x_t are both either +1 or −1).

To fit the model parameters using variational inference, we first need to maximize the ELBO.

$$\begin{aligned} \text{ELBO} &= E_{q(x)}\left[\log p(x, y)\right] - E_{q(x)}\left[\log q(x)\right] \\ &= E_{q(x)}\left[\sum_{(s,t)\in E} x_s w_{st} x_t + \sum_{i=1}^{n} \log N\left(x_i; \sigma^2\right)\right] \\ &\quad - \sum_{i=1}^{n} E_{q_i(x)}\left[\log q_i(x)\right] \end{aligned} \tag{3.15}$$

Here, we are using the mean-field assumption of a fully factored approximation $q(x)$.

$$q(x) = \prod_{i=1}^{n} q(x_i; \mu_i) \tag{3.16}$$

Using the result derived in equation 3.12, we state that $q(x_i;\ \mu_i)$, which minimizes the KL divergence, is given by the following.

$$q_i(x_i) = \frac{1}{Z_i} \exp\left[E_{-q_i}\{\log p(x)\}\right] \tag{3.17}$$

Here, E_{-qi} denotes the expectation over every q_j except for $j = i$. To compute $q_i(x_i)$, we only care about the terms that involve x_i (i.e., we can isolate them as shown in equation 3.18).

$$\begin{aligned} E_{-q_i}\{\log p(x)\} &= E_{-q_i}\{x_i \sum_{j \in N(i)} w_{ij} x_j + \log N(x_i, \sigma^2) + \text{const}\} = \\ &= x_i \sum_{j \in N(i)} J \times \mu_j + \log N(x_i, \sigma^2) + \text{const} \end{aligned} \tag{3.18}$$

Here, $N(i)$ denotes the neighbors of node i and μ_j is the mean of a binary random variable.

$$\mu_j = E_{q_j}[x_j] = q_j(x_j = +1) \times (+1) + q_j(x_j = -1) \times (-1) \tag{3.19}$$

Figure 3.4 shows the parametric form of our mean-field approximation for the Ising model.

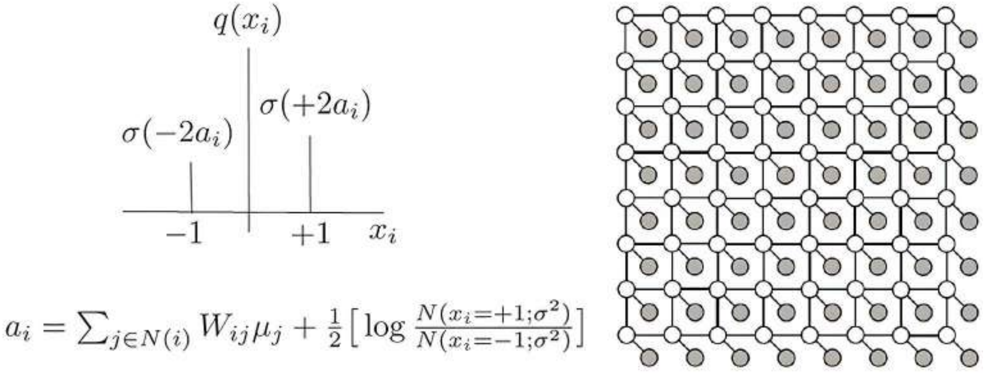

$q(x_i)$
$\sigma(+2a_i)$
$\sigma(-2a_i)$
-1
$+1$
x_i
$a_i = \sum_{j \in N(i)} W_{ij}\mu_j + \frac{1}{2}\left[\log \frac{N(x_i=+1;\sigma^2)}{N(x_i=-1;\sigma^2)}\right]$

Figure 3.4 Ising model and its approximating distribution *q*(*x*)

To compute this mean, we need to know the values of $q_j(x_j = +1)$ and $q_j(x_j = -1)$. Let $m_i = \Sigma_{\{j \in\ N(i)\}}\ w_{ij}\ \mu_j$ be the mean value of neighbors and let $L_i^+ = N\,(x_i = +1, \sigma^2)$ and $L_i^- = N\,(x_i = -1, \sigma^2)$; then, we can compute the mean as follows.

$$\begin{aligned} q_i(x_i = +1) &= \frac{\exp\left\{m_i + L_i^+\right\}}{\exp\left\{m_i + L_i^+\right\} + \exp\left\{-m_i + L_i^-\right\}} \\ &= \frac{1}{1 + \exp\left\{-2m_i + L_i^- - L_i^+\right\}} \\ &= \frac{1}{1 + \exp\left\{-2a_i\right\}} = \sigma(2a_i) \end{aligned} \tag{3.20}$$

Here, $a_i = m_i + \frac{1}{2}\left(L_i^+ L_i^-\right)$ and $\sigma(x)$ is a sigmoid function. Since $q_i(x_i = -1) = 1 - q_i(x_i = +1) = 1 - \sigma(2a_i) = \sigma(-2a_i)$, we can write the mean of our variational approximation $q_i(x_i)$ as follows.

$$\mu_i = E_{q_i}[x_i] = \sigma(2a_i) - \sigma(-2a_i) = \tanh(a_i) \tag{3.21}$$

In other words, our mean-field updates of the variational parameters μ_i at iteration k are computed as follows.

$$\begin{aligned} \mu_i^{(k)} &= \tanh\left(\sum_{j \in N(i)} w_{ij}\mu_j^{(k-1)} + \frac{1}{2}\left[\log\frac{N(x_i = +1, \sigma^2)}{N(x_i = -1, \sigma^2)}\right|\Bigg]\right) \\ &\qquad \times \lambda + (1 - \lambda) \times \mu_i^{(k-1)} \end{aligned} \tag{3.22}$$

Here, we added a learning rate parameter $\lambda \in (0, 1]$. We further note that we can simplify the computation of the ELBO term by term, as follows.

$$\begin{aligned} &\overbrace{\sum_{(s,t) \in E} E_{q(x)}[x_s w_{st} x_t]}^{\textbf{ELBO first term}} \\ &\quad = \frac{1}{2}\sum_{i=1}^{n}\sum_{j \in N(i)} \Big(\overbrace{\sum_{x_i \in \{-1,+1\}}\sum_{x_j \in \{-1,+1\}} q_i(x_i)q_j(x_j)x_iJx_j}^{\textbf{By definition of expectation}}\Big) \\ &\quad = \frac{1}{2}\sum_{i=1}^{n}\sum_{j \in N(i)} \underbrace{(q_i(x_i = +1)JE[x_j] - q_i(x_i = -1)JE[x_j])}_{\textbf{After substitution of values for xi and xj}} \\ &\quad = \frac{1}{2}\sum_{i=1}^{n}\sum_{j \in N(i)} \underbrace{E[x_i]JE[x_j]}_{\textbf{By definition of expectation}} \end{aligned} \tag{3.23}$$

Equation 3.24 is similar.

$$\overbrace{E_{q(x)}[\log N(x_i, \sigma^2)]}^{\textbf{ELBO second term}} = \sum_{i=1}^{n}\left[\overbrace{\sum_{x_i \in \{-1,+1\}} q_i(x_i)\log N(x_i, \sigma^2)}^{\textbf{By definition of expectation}}\right] = \sum_{i=1}^{n}\left[\underbrace{\sigma(2a_i)\log N(x_i = +1, \sigma^2) + \sigma(-2a_i)\log N(x_i = -1, \sigma^2)}_{\textbf{After expanding the summation}}\right] \tag{3.24}$$

To better understand the algorithm, let's examine the pseudo-code in figure 3.5.

```
class image_denoising
function mean_field($\sigma$, $y$, $w$, $\lambda$, max_iter):
logp1 = log $N(y; x_i = +1, \sigma^2)$
logm1 = log $N(y; x_i = -1, \sigma^2)$
logodds = logp1 − logm1
p1 = sigmoid(logodds) //init
$\mu^{(0)}$ = 2 × p1 −1 //init
for $k$ = 1 to max_iter:
    $\bar{S}_{ij} = \sum_{j \in N(i)} w_{ij} \mu_j^{(k-1)}$
    $\mu_i^{(k)} = \tanh(\bar{S}_{ij} + \frac{1}{2}\text{logodds}) \times \lambda + (1 - \lambda) \times \mu_i^{(k-1)}$
    ELBO[k] = ELBO[k] + $\frac{1}{2}\left(\bar{S}_{ij} \times \mu_i^{(k)}\right)$
    $a = \mu^{(k)} + \frac{1}{2}$ logodds
    qxp1 = sigmoid(+2$a$)
    qxm1 = sigmoid(−2$a$)
    Hx = −qxm1 × log(qxm1) − qxp1 × log(qxp1)
    ELBO[k] = ELBO[k]+ $\sum_{i=1}^{N}$ (qxp1[$i$]× logp1[$i$] + qxm1[$i$]× logm1[$i$])
    + $\sum_{i=1}^{N}$ (Hx[$i$])
end for
return $\mu^{(k)}$
```

Figure 3.5 Mean field VI for Ising model pseudo-code

In the `image_denoising` class, we have a single method called `mean_field`, which takes as input the noise level sigma, noisy binary image `y`, coupling strength `w=J`, learning rate lambda, and max number of iterations. We start by computing log-odds ratio (i.e., the probability of observing image pixel `y` under a Gaussian random variable with the means `+1` and `-1`). We then compute the sigmoid function of the log-odds ratio and use the result to initialize the mean variable. Next, we iterate until we have the max number of iterations, and in each iteration, we compute the influence of the neighbors `Sij`, which we include in the mean-field update equation. We then compute our objective function `ELBO` and mean entropy `Hx` to monitor the convergence of the algorithm.

We now have all the tools we need to implement the mean-field variational inference for the Ising model in application to image denoising! In the following listing, we will read in a noisy image and execute mean-field variational inference on a grid of pixels to denoise it.

Listing 3.1 Mean-field variational inference in an Ising model

```
import numpy as np
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt

from PIL import Image
from tqdm import tqdm
from scipy.special import expit as sigmoid
from scipy.stats import multivariate_normal

np.random.seed(42)
sns.set_style('whitegrid')

class image_denoising:

    def __init__(self, img_binary, sigma=2, J=1):

        #mean-field parameters
        self.sigma  = sigma
        self.y = img_binary + self.sigma*np.random
        ➥ .randn(M, N)
        self.J = J
        self.rate = 0.5
        self.max_iter = 15
        self.ELBO = np.zeros(self.max_iter)
        self.Hx_mean = np.zeros(self.max_iter)

    def mean_field(self):

        #Mean-Field VI
        print("running mean-field variational inference...")
        logodds = multivariate_normal.logpdf(self.y.flatten(), mean=+1,
        ➥ cov=self.sigma**2) - \
                  multivariate_normal.logpdf(self.y.flatten(), mean=-1,
                  ➥ cov=self.sigma**2)
        logodds = np.reshape(logodds, (M, N))

        #init
        p1 = sigmoid(logodds)
        mu = 2*p1-1

        a = mu + 0.5 * logodds
        qxp1 = sigmoid(+2*a)   #q_i(x_i=+1)
        qxm1 = sigmoid(-2*a)   #q_i(x_i=-1)

        logp1 = np.reshape(multivariate_normal.logpdf(self.y.flatten(),
        ➥ mean=+1, cov=self.sigma**2), (M, N))
        logm1 = np.reshape(multivariate_normal.logpdf(self.y.flatten(),
```

Noise level

Coupling strength (wij)

y_i ~ N(x_i; sigma ^ 2);

Smoothing rate update

Initial value of mu

```
        ➥ mean=-1, cov=self.sigma**2), (M, N))

        for i in tqdm(range(self.max_iter)):
            muNew = mu
            for ix in range(N):
                for iy in range(M):
                    pos = iy + M*ix
                    neighborhood = pos + np.array([-1,1,-M,M])
                    boundary_idx = [iy!=0,iy!=M-1,ix!=0,ix!=N-1]
                    neighborhood = neighborhood[np.where(boundary_idx)[0]]
                    ➥ xx, yy = np.unravel_index(pos, (M,N), order='F')
                    nx, ny = np.unravel_index(neighborhood, (M,N), order='F')

                    Sbar = self.J*np.sum(mu[nx,ny])
                    muNew[xx,yy] = (1-self.rate)*muNew[xx,yy] +
                    ➥ self.rate*np.tanh(Sbar + 0.5*logodds[xx,yy])
                    self.ELBO[i] = self.ELBO[i] + 0.5*(Sbar * muNew[xx,yy])
                #end for
            #end for
            mu = muNew

            a = mu + 0.5 * logodds
            qxp1 = sigmoid(+2*a) #q_i(x_i=+1)
            qxm1 = sigmoid(-2*a) #q_i(x_i=-1)
            Hx = -qxm1*np.log(qxm1+1e-10) - qxp1*np.log(qxp1+1e-10) #entropy

            self.ELBO[i] = self.ELBO[i] + np.sum(qxp1*logp1 + qxm1*logm1)
            ➥ + np.sum(Hx)
            self.Hx_mean[i] = np.mean(Hx)
        #end for
        return mu

if __name__ == "__main__":

    #load data
    print("loading data...")
    data = Image.open('./figures/bayes.bmp')
    img = np.double(data)
    img_mean = np.mean(img)
    img_binary = +1*(img>img_mean) + -1*(img<img_mean)
    [M, N] = img_binary.shape

    mrf = image_denoising(img_binary, sigma=2, J=1)
    mu = mrf.mean_field()

    #generate plots
    plt.figure()
    plt.imshow(mrf.y)
    plt.title("observed noisy image")
    plt.show()

    plt.figure()
    plt.imshow(mu)
    plt.title("after %d mean-field iterations" %mrf.max_iter)
    plt.show()
plt.figure()
plt.plot(mrf.Hx_mean, color='b', lw=2.0, label='Avg Entropy')
plt.title('Variational Inference for Ising Model')
plt.xlabel('iterations'); plt.ylabel('average entropy')
plt.legend(loc='upper right')
plt.show()

plt.figure()
plt.plot(mrf.ELBO, color='b', lw=2.0, label='ELBO')
plt.title('Variational Inference for Ising Model')
plt.xlabel('iterations'); plt.ylabel('ELBO objective')
plt.legend(loc='upper left')
plt.show()
```

Figure 3.6 shows experimental results for binary image denoising via mean-field variational inference.

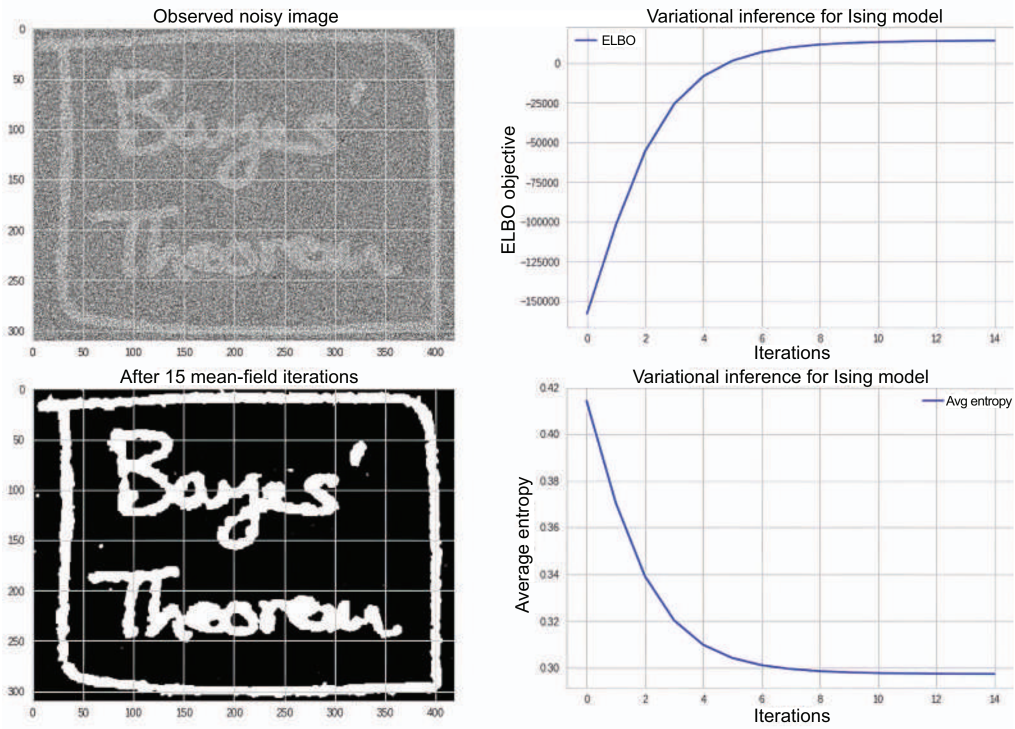


Figure 3.6 Mean-field variational inference for image denoising in an Ising model

The noisy observed image is shown in the top left and is obtained by adding Gaussian noise to each pixel and binarizing the image based on a mean threshold. We then set the variational inference parameters, such as the coupling strength $J=1$, noise level $\sigma = 2$, smoothing rate $\lambda = 0.5$, and max number of iterations of 15. The resulting denoised binary image is shown in the bottom-left corner of the figure. We can also see an increase in the ELBO objective (top right) and a decrease in the average entropy of our binary random variables $q_i(x_i)$ representing the value of each pixel (bottom right) as the number of mean-field iterations increases. The 2D Ising model can be extended in multiple ways (e.g., via 3D grids and K-states per node [aka a Potts model]).

3.4 MI maximization

In this section, we look at mutual information (MI) maximization, which commonly occurs in information planning and data communications settings. Consider a wireless communications scenario, in which we transmit a signal $x \sim p_X(x)$, it passes through a multiple-input, multiple-output (MIMO) channel H. At the output, we receive our signal, $y = Hx + n$, where $n \sim N(0, \sigma^2 I)$ is an additive Gaussian noise. We would like to maximize the amount of information transmitted over the wireless channel. In other words, we would like to maximize the capacity or mutual information between the transmitted signal X and the received signal Y: $C = \max I(X;Y)$, where the maximization is taken over $p(x)$. To compute channel capacity, we discuss a general procedure based on KL divergence to approximately maximize mutual information.

$$I(X;Y) = H(X) - H(X|Y) = -E_{p(x)}\left[\log p(x)\right] - E_{p(x,y)}\left[\log p(x|y)\right] \tag{3.25}$$

Let $q(x)$ be an approximating distribution to $p(x)$, and then consider KL divergence between the posterior distributions of p and q, as shown in equation 3.26.

$$D_{KL}(p(x|y)||q(x|y)) \geq 0 \tag{3.26}$$

We would like to derive a lower bound on mutual information (MI). Expanding the expression in equation 3.26, we can proceed as shown in equation 3.27.

$$\sum_x p(x|y)\log p(x|y) - \sum_x p(x|y)\log q(x|y) \geq 0 \tag{3.27}$$

Multiplying both sides by $p(y)$, we get the following.

$$\sum_{x,y} p(y)p(x|y)\log p(x|y) \geq \sum_{x,y} p(x,y)\log q(x|y) \tag{3.28}$$

Recognizing the left-hand side as $-H(X|Y)$, we obtain the following MI lower bound.

$$I(X;Y) = H(X) - H(X|Y) \geq H(X) - E_{p(x,y)}\left[\log q(x|y)\right] = \tilde{I}(X;Y) \tag{3.29}$$

Using the preceding lower bound, we can describe MI maximization algorithm (see figure 3.7).

```
Choose approximating distribution family $Q(x;\theta)$
Initialize $\theta$
repeat
    for a fixed $q(x|y;\theta)$, find
        $\theta^{new} = \text{argmax}_{\theta}\tilde{I}(X;Y)$
    for a fixed $\theta$, find
        $q_{new}(x|y;\theta) = \text{argmax}_{q(x|y)\in Q}\tilde{I}(X;Y)$
until convergence
```

Figure 3.7 Mutual information maximization pseudo-code

The preceding algorithm alternates between finding a set of parameters that maximize the MI lower bound and finding the approximate distribution.

In this section, we saw how we can use the definitions of entropy, mutual information, and KL divergence to derive a lower bound that could then be iteratively maximized by updating our approximation distribution *q*. In the following chapter, we will look at ML from a computer science perspective and explore useful data structures and algorithmic paradigms.

3.5 Exercises

3.1 Compute KL divergence between two univariate Gaussians: $q(x) \sim N(\mu_1, {\sigma_1}^2)$ and $q(x) \sim N(\mu_2, {\sigma_2}^2)$.

3.2 Compute $E[X]$, $\text{Var}(X)$, and $\text{H}(X)$ for a Bernoulli distribution.

3.3 Derive the mean, mode, and variance of a $\text{Beta}(a, b)$ distribution.

Summary

- The main idea of variational inference is to choose an approximate distribution $q(x)$ from a family of tractable distributions and then make this approximation as close as possible to the true posterior distribution $p(x)$.
- An evidence lower bound is an objective function that we seek to maximize to learn the variational parameters of our model.
- In mean-field approximation, we assume the approximate distribution $q(x)$ is fully factorized.
- Mutual information maximization can be carried out by deriving and maximizing the MI lower bound.