Machine Learning - Supplement 1

Dom Hutchinson

January 14, 2020

1 Variational Bayes

Variational Bayes is a method for approximating an intractable integral, P, with a tractable one, Q. This is useful as *Posterior*s are integrals and generally intractable.

$$P(X|Y) \approx Q(X)$$

From the derivation below we get that

$$\ln P(Y) = KL(Q\|P) - \mathbb{E}_X[\ln Q(X) - \ln P(X,Y)]$$

since $\ln P(Y)$ is fixed wrt Q we can derive a likelihood function

$$\mathcal{L}(Q) = -\mathbb{E}_X[\ln Q(X) - \ln P(X,Y)]$$

If we maximise this likelihood function, $\mathcal{L}$, then we are minimising $KL(Q||P)$ which means Q & P are becoming similar.
We have reduced the problem of approximation to just maximising $\mathcal{L}(Q)$.
By choosing a good form for Q means $\mathcal{L}(Q)$ becomes tractable.

1.1 Kullback-Leibler Divergence

Kullback-Leibler Divergence is a similarity measure for two distributions, Q & P.

$$KL(Q||P) := \int Q(X) \ln\left(\frac{Q(X)}{P(X|Y)}\right) dX$$

The lower the value of $KL(Q||P)$ the greater the similarity of Q & P.
N.B. $KL(\cdot||\cdot) \geq 0$ and $KL(Q||Q) = 0$.

1.2 Intractibiltiy

Here I show why the *Posterior* is intractrable

$$\underbrace{P(X|Y)}_{\text{Intractable}} = \frac{\overbrace{P(Y|X)P(X)}^{\text{Tractable}}}{\underbrace{P(Y)}_{\text{Intractable}}} = \underbrace{\frac{P(Y|X)P(X)}{\int P(X,Y)dX}}_{\text{Intractable}}$$

$\int P(X,Y)dX$ is intractable since the space X is intractablly large.
This makes the evidence, and thus posterior, intractable.

1.3 Derivation

$$\begin{aligned}
KL(Q||P) &\approx \sum_X Q(X)\ln\left[\frac{Q(X)}{P(X|Y)}\right] \\
&= \sum_X Q(X)\ln\left[\frac{Q(X)P(X)}{P(X,Y)}\right] \text{ by product rule} \\
&= \sum_X Q(X)[\ln Q(X) + \ln P(Y) - \ln P(X,Y)] \text{ by log rules} \\
&= \mathbb{E}_X[\ln Q(X) - \ln P(X,Y)] + \ln P(Y) \text{ since } P(Y) \text{ is independent of X} \\
\implies \ln P(Y) &= KL(Q||P) - \underbrace{\mathbb{E}_X[\ln Q(X) - \ln P(X,Y)]}_{\mathcal{L}(Q)} \text{ by rearrangement}
\end{aligned}$$

1.4 Mean Field Variational Bayes - Ising Model

Here we make the *Mean Field Approximation*, that

$$q(\mathbf{X}) = \prod_i q_i(x_i, \mu_i) \text{ where } \mu_i := \mathbb{E}_{q_i}(x_i)$$

Since this is an *Ising Model* we have posterior

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z_1}\prod_i e^{L_i(x_i)}$$

where $L_i(x_i)$ is a predefined function which gives a large value if it is likely x_i generated y_i.
By derivations, shown in *Problem Sheet 9*, we produce the following process

i) Randomly initialise all μ_i.

ii) For each x_i:

   (a) Set $\mu_i = \tanh\left(m_i + \frac{1}{2}[L_i(1), L_i(-1)]\right)$
   where $m_i := \sum_{j\in\mathcal{N}_i} w_{ij}\mu_j^T$ and $\{1, -1\}$ are the two values the x_i can take (Ising Model).

iii) Repeat until time is up

2 Predicitive Gaussian Processes

Gaussian Processes are the class of *Stochastic Processes* st every finite linear combination of random variables is normally distributed.
Here I describe the process of make predictions for the value at a set of points, given training data (X, y).

i) Observed data points, (X, y).

ii) Define the set of points we wish to predict values at, (X^*).

iii) Define a kernel function to use as the covaraince funtion, $k(\cdot,\cdot)$.

iv) Calculate $\boldsymbol{\mu}^*, \Sigma^*$ for the points X^*, using the equations below.

v) Draw samples from Normal$(\boldsymbol{\mu}^*, \Sigma^*)$. Each sample can be used to infer a function.

2.1 Equations

Without Noise

$$\begin{array}{rcl}\boldsymbol{\mu}^* & = & k(X^*,X^*)k(X,X)^{-1}y \\ \Sigma^* & = & k(X^*,X^*)-k(X^*,X)k(X,X)^{-1}k(X^*,X)^T\end{array}$$

With Noise

$$\begin{array}{rcl}\boldsymbol{\mu}^* & = & k(X^*,X^*)k(X,X\boldsymbol{+c})^{-1}y \\ \Sigma^* & = & k(X^*,X^*)-k(X^*,X)k(X,X\boldsymbol{+c})^{-1}k(X^*,X)^T\end{array}$$

The only difference is the $+c$ in the $K(X,X)$ terms.
N.B. $\boldsymbol{\mu}^*\in\mathbb{R}^N$ & $\Sigma^*\in\mathbb{R}^{N\times N}$ where $N:=|X^*|$.

2.2 Kernels

Linear	$k(\mathbf{x},\mathbf{y})=\sigma^2\times\mathbf{x}^T\mathbf{y}$
White	$k(\mathbf{x},\mathbf{y})=\sigma^2 I$
Periodic	$k(\mathbf{x},\mathbf{y})=\sigma^2\exp\left\{-\frac{2}{\ell^2}\sin^2\left(\frac{\pi}{p}\|\mathbf{x}-\mathbf{y}\|\right)\right\}$ where ℓ =length scale p =period
Radial Basis Function	$k(\mathbf{x},\mathbf{y})=\exp\left\{-\frac{1}{\ell^2}\|\mathbf{x}-\mathbf{y}\|^2\right\}$ where ℓ =length scale

Vary σ^2 depending on noise in readings.
N.B. $\|\mathbf{x}\|:=\sqrt{\sum x_i^2}\implies\|\mathbf{x}-\mathbf{y}\|=\sqrt{\sum(x_i-y_i)^2}$. This is the *Euclidean Distance.*

3 Dirichlet Processes

Dirichlet Processes are the class of *Stochastic Processes* whose realisations are probability distributions.
Dirichlet Processes take a base distribution, $f(\cdot)$, and a concentration parameter, $\alpha\in\mathbb{R}$. Realisations become more continuous the greater the value $\lim\alpha$ tends to.

The following algorithm is used to construct a realisation

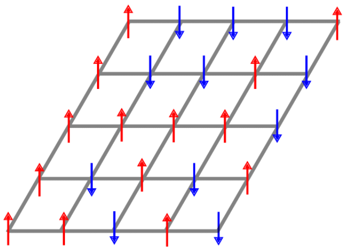
i) With probability $\dfrac{\alpha}{\alpha+n-1}$ draw X_n from $f(\cdot)$.

ii) With probability $\dfrac{n_x}{\alpha+n-1}$ set $X_n=x$
where $n_x:=|\{j<n:X_j=x\}|$ (*i.e.* the number of previous observations of x).

These $X_1,X_2,\dots$ represent the relative frequencies with which each value should occur in the distribution. In practice we cannot produce full distributions as that would require infinite iterations of the algorithm, instead we are just approximating it.
N.B. $X_1,X_2,\dots$ are not independent since they depend on the previously generated results.
N.B. The base distribution, $f(\cdot)$, is the expected result.

4 Ising Model

An *Ising Model* is one where each latent variable takes one of two states and only has dependency on neighbours in such a way that a grid is formed. Thus dependencies are undirected, forming a *Markov Random Field.*



4.1 Ising Prior

$$p(\mathbf{x})=\frac{1}{Z_0}e^{\sum_{i\in\mathbf{x}}\sum_{j\in N_i}w_{ij}x_ix_j}$$

where Z_0 is a normalising term, N_i is the neighbourhood of x_i & w_{ij} is the weighting of the relationship between variables x_i & x_j.
N.B. $x_ix_j=1$ iff $x_i=x_j$, otherwise $x_i,x_j=-1$. Thus this term only increases when x_i has the same value as many of its neighbours.

4.2 Iterative Conditional Modes

Iterative Conditional Modes is a tecnique for inferring latent variable values in an *Ising Model.*

i) Randomly initalise $\mathbf{x}$.

ii) For each $x_i\in\mathbf{x}$.

   (a) Assume all latent values are fixed except for x_i.

   (b) Assign x_i to the most likely value given the other values.

iii) Repeat ii) until time is up

4.3 Gibbs Sampling

Gibbs Sampling is an implementation of *Markov Chain Monte Carlo.*

The general idea is that given a distribution $p(\mathbf{x})$ which we wish to sample from we shall draw samples from one dimension at a time, using the other dimensions as conditions, $p(x_i|\mathbf{x}_{\neg i})$.

If we now just consider the *Ising Model* we can perform some derivations

$$\begin{aligned}p(x_i|\mathbf{x}_{\neg i},\mathbf{y}) &= \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x}_{\neg i},\mathbf{y})}\\ &= \frac{p(\mathbf{x},\mathbf{y})}{\displaystyle\int p(\mathbf{x},\mathbf{y})dx_i}\\ &= \frac{p(\mathbf{x},\mathbf{y})}{\displaystyle\sum_{x_i\in\{1,-1\}}p(\mathbf{x},\mathbf{y})}\\ &= \frac{p(\mathbf{x},\mathbf{y})}{p(x_i=1,\mathbf{x}_{\neg i},\mathbf{y})+p(x_i=-1,\mathbf{x}_{\neg i},\mathbf{y})}\\ \implies p(x_i=1|\mathbf{x}_{\neg i},\mathbf{y}) &= \frac{p(x_i=1\mathbf{x}_{\neg i},\mathbf{y})}{p(x_i=1,\mathbf{x}_{\neg i},\mathbf{y})+p(x_i=-1,\mathbf{x}_{\neg i},\mathbf{y})}\end{aligned}$$

This is a tractable expression.

Here is an algorithm for *Gibbs Sampling*

i) Randomly initalise $\mathbf{x}$.

ii) For each $x_i \in \mathbf{x}$.

   (a) Calculate $p(x_i = 1|\mathbf{x}_{\neg i}, \mathbf{y})$.

   (b) Draw u from Uniform$[0, 1]$.

   (c) If $p(x_i = 1|\mathbf{x}_{\neg i}, \mathbf{y}) > u$ set $x_i = 1$, otherwise set $x_i = -1$.

iii) Repeat ii) until time is up

5 Gradient Descent

Gradient Descent is an iterative *optimisaiton algorithm* which aims to find the local minimum of a function, and thus the parameters which produce it. *Gradient Descent* can be used in scenarios where a function is intractable, where *Least Squares* cannot.

Here is an algorithm for *Gradient Descent*

i) Set $t = 0$.

ii) Randomly initalise parameter values, θ_t.

iii) Repeat until convergence of θ_t.

   (a) Evaluate the performance of this state using a *Loss Function*, $L(\mathbf{x}, \theta_t)$.

   (b) Find the derivative of the *Loss Function* wrt the parameters, $\frac{d}{d\theta}L(\mathbf{x}, \theta)$.

   (c) Evaluate the derivative for the parmater values we just tested, z.

   (d) Calculate *Step Size* $s := z \times \alpha$, where α is the *Learning Rate* (≈ 0.1).

   (e) Set $\theta_{t+1} = \theta_t - s$.

N.B. Sum of the Squared Residuals is a good loss function, $L(\mathbf{x}, \theta) := \sum_i [o_i - e_i]^2$ where o_i is the observed value & e_i is the predict value at a given point.

6 Maximum a Posteriori Estimates

Maximum a Posteriori are point estimates which maximise the *Posterior* distribution.

$$\hat{\theta}_{\text{MAP}} := \text{argmax}_\theta P(\theta|\mathcal{D})$$

MAP Estimates are used to avoid overfitting but are not invariant reparameterisation.
N.B. $\hat{\theta}_{\text{MAP}}$ is not necessarily unique, similar to *MLE*.
N.B. $\hat{\theta}_{\text{MLE}} := \text{argmax}_\theta L(\theta|\mathcal{D}) = \text{argmax}_\theta f(\mathcal{D}|\theta)$