Intro about Bayesian Machine Learning

- Setting: $D = \{(x_i, y_i)\}_{i=1}^N$
- Up to now, the ML methods we used can be seen as based on max likelihood

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(D \mid \theta)$$

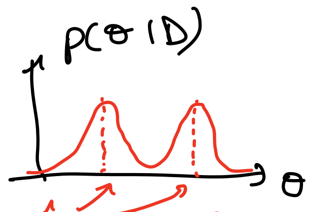
or max a posteriori

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta \mid D)$$

$$= \arg\max_{\theta} \underbrace{\log p(D \mid \theta)}_{\text{log likelihood}} + \underbrace{\log p(\theta)}_{\text{regularisation term}}$$

because $p(\theta \mid D) = \dfrac{p(D \mid \theta)\, p(\theta)}{p(D)} \leftarrow$ constant

- We were looking for a single model $\hat{\theta}$
- pb : - no level of confidence in our choice
  - maybe there is no good "single model"



$\hat{\theta}_{MAP}$ will arbitrarily choose one of those two modes

I) A simple coin estimation problem with the ML, MAP, and bayesian views.

↳ max likelihood.

A) Max likelihood approach

- We have a coin with $p(\text{heads}) = \theta$ unknown.
- We have drawn the coin N times, we got n_H heads and n_T tails.
- We want to estimate $p(\text{heads})$ based on n_H, n_T
- simplest approach is $\hat{\theta}_{ML} = \dfrac{n_H}{n_H + n_T}$ $\leftarrow$ max likelihood

- derivation of $\hat{\theta}_{ML}$:

- the likelihood is $P(\underbrace{n_H, n_T}_{D} | \theta) = \binom{n_H + n_T}{n_H} \theta^{n_H} (1-\theta)^{n_T}$

(binomial law)

- $\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}} \log P(n_H, n_T | \theta)$

$$= \underset{\theta}{\operatorname{argmax}} \underbrace{n_H \log \theta + n_T \log(1-\theta)}_{\text{derive and set derivative to zero}}$$

$$\frac{n_H}{\hat{\theta}} = \frac{n_T}{1-\hat{\theta}} \Rightarrow n_H - n_H \hat{\theta} = n_T \hat{\theta}$$

$$\Rightarrow \hat{\theta}_{ML} = \frac{n_H}{n_H + n_T}$$

B) Max à Posteriori approach

Note: D is the pair (n_H, n_T)

$$\hat{\theta}_{MAP} = \operatorname{argmax} \; P(\theta | D)$$

$$= \operatorname{argmax} \frac{P(D|\theta) \, P(\theta)}{P(D)}$$

$P(D|\theta)$ → likelihood

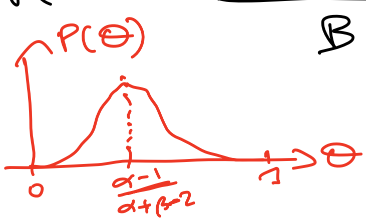
$P(\theta)$ → prior

$P(D)$ → evidence (cste)

We need a prior on θ.

Typically, we use a Beta distribution.

$$P(\theta) = \frac{\theta^{\alpha - 1} (1-\theta)^{\beta - 1}}{B(\alpha, \beta)}$$



Note:

- $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$
- $\mathbb{E}[\theta] = \frac{\alpha}{\alpha + \beta}$

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}} \log P(\theta|D) = \underset{\theta}{\operatorname{argmax}} \log P(D|\theta) + \log P(\theta)$$

$$= \underset{\theta}{\operatorname{argmax}} \; cste + n_H \log \theta + n_T \log(1-\theta) + (\alpha-1)\log\theta + (\beta-1)\log(1-\theta) + cte$$

$$= \underset{\theta}{\operatorname{argmax}} \underbrace{(n_H + \alpha - 1)\log\theta + (n_T + \beta - 1)\log(1-\theta)}_{= \log P(\theta|D) + cste \, = \, \log \text{ beta distribution w.r.t. } \theta}$$

$$\boxed{\hat{\theta}_{MAP} = \frac{n_H + \alpha - 1}{n_H + n_T + \alpha + \beta - 2}}$$

- The Beta prior is said to be the "conjugate" prior of the binomial because
  1) the posterior $P(\theta \mid D)$ is a binomial distribution and a beta distribution w.r.t. θ with parameters $Beta(\alpha + n_H, \beta + n_T)$
  2) adding the prior has the same effect as observing additional draws.
- Note that if $N \to \infty$ then $\hat{\theta}_{MAP} \to \hat{\theta}_{ML}$

c) Full Bayesian approach

- What is the probability that our next draw is a head?

$$p(Head \mid D) = \int_\theta P(Head, \theta \mid D)\, d\theta$$

(law of total proba)

$$= \int_\theta P(Head \mid \theta, D) \times P(\theta \mid D)\, d\theta$$

$$= \int_\theta \underbrace{P(Head \mid \theta)}_{\theta} \times P(\theta \mid D)\, d\theta$$

$$= \int_\theta \theta \cdot P(\theta \mid D)\, d\theta$$

$$= \mathbb{E}_{\underbrace{\theta \sim P(\theta \mid D)}_{Beta(\alpha + n_H, \beta + n_T)}} [\theta \mid D]$$

$$P(Heads \mid D) = \frac{n_H + \alpha}{n_H + n_T + \alpha + \beta}$$

II) Bayesian view of Machine learning

- Assume $D = \{(x_i, y_i)\}_{i=1}^{N}$ (classification or regression)
- Let $\mathbb{X} = (x_1 \dots x_N)$ $\mathbb{Y} = (y_1 \dots y_N)$
- Assume $\mathbb{X}$ is fixed
- Assume the "nature" chose θ from $p(\theta)$
- $\mathbb{Y}$ is drawn from $p(\mathbb{Y} \mid \mathbb{X}, \theta) = p(\mathbb{Y} \mid \theta)$

Let x_{N+1} be a new data point on which a prediction has to be made.

In max likelihood: $\hat{\theta}_{ML} = \operatorname{argmax} p(\mathbb{Y} \mid \theta)$

In the Bayesian setting, we don't want to choose a specific $\hat{\theta}$, we only want to predict y_{N+1}

$$\boxed{p(y_{N+1} \mid \mathbb{X}, \mathbb{Y}, x_{N+1})}$$

(predictive distribution)

$$= p(y_{N+1} \mid \mathbb{Y}) = \int p(y_{N+1}, \theta \mid \mathbb{Y})\, d\theta$$

total law of prob.

$$= \int p(y_{N+1} \mid \theta, \mathbb{Y})\, p(\theta \mid \mathbb{Y})\, d\theta$$

$$= \int p(y_{N+1} \mid \theta)\, p(\theta \mid \mathbb{Y})\, d\theta$$

$$\propto \int \underbrace{p(y_{N+1} \mid \theta)}_{\text{prediction for } \theta}\, \underbrace{p(\mathbb{Y} \mid \theta)}_{\text{likelihood}}\, \underbrace{p(\theta)}_{\text{prior}}\, d\theta$$

Bayesian Model Averaging

III Approximation of the BAM integral.

- x_{N+1} is constant as well as $\mathbb{X}$, so no need to put them in the conditional.

$$\underbrace{p(y_{N+1} \mid \mathbb{Y})}_{\substack{\text{predictive}\\ \text{proba distribution}}} \propto \int p(y_{N+1} \mid \theta)\, p(\mathbb{Y} \mid \theta)\, p(\theta)\, d\theta$$

Naive idea: draw randomly $\theta_1 .. \theta_M$ from $p(\theta)$
e.g. $p(\theta) = \mathcal{N}(0, I_d)$

$$p(y_{N+1} \mid \mathbb{Y}) \approx \frac{1}{Z} \sum_{k=1}^{M} p(y_{N+1} \mid \theta_k) \cdot \underbrace{p(\mathbb{Y} \mid \theta_k)}_{\text{likelihood}}$$

$$Z = \sum_{k=1}^{M} p(\mathbb{Y} \mid \theta_k)$$

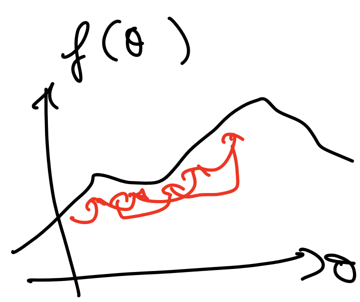
Can't work because M would need to be huge for the sum to approximate the integral.

Better Idea: draw $\theta_1 .. \theta_M$ from $p(\theta \mid \mathbb{Y})$ the posterior.

$$p(y_{N+1} \mid \mathbb{Y}) \approx \frac{1}{M} \sum_{k=1}^{M} p(y_{N+1} \mid \theta_k)$$

Much better estimator!
sampling from $p(\theta \mid \mathbb{Y})$ is hard.

A) Stochastic Langevin gradient for sampling from $p(\theta \mid \mathbb{Y})$



- let $f(\theta)$ be a density over θ.
- assume we want to sample $\theta \sim f$

- One way is to use Langevin stochastic gradient:

Repeat:

$$\theta \leftarrow \theta + \frac{\alpha_t}{2} \nabla \log f + \varepsilon_t$$

$$\boxed{\begin{array}{l} \sum_{t=0}^{\infty} \alpha_t = \infty \\ \sum_{t=0}^{\infty} \alpha_t^2 < \infty \\ \varepsilon_t \sim \mathcal{N}(0, \alpha_t \cdot Id) \end{array}}$$

After "some" step of the algorithm, θ is distributed according to $f(\theta)$

- In our machine learning setting, we do a gradient descent on $-\log p(\theta \mid Y)$ and we add noise on θ at each step.
- the successive θs are kept as samples of $p(\theta \mid Y)$

B) Variational approximation

We will learn $q(\theta) = \mathcal{N}(\theta \mid \mu, \Sigma)$ which approximate $p(\theta \mid Y)$ as well as possible.

I want to minimize

$$KL(q(\theta) \,\|\, p(\theta \mid Y)) = \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid Y)} \, d\theta$$

$$\frac{p(\theta \mid Y)}{q(\theta)} = \frac{P(Y \mid \theta)\, P(\theta)}{q(\theta) \cdot P(Y)}$$

$$KL(q(\theta) \,\|\, p(\theta|Y)) = -\int q(\theta) \log \frac{P(\theta|Y)}{q(\theta)} \, d\theta$$

$$= -\int q(\theta) \cdot \left[\log P(Y|\theta) + \log\left(\frac{P(\theta)}{q(\theta)}\right) - \log P(Y)\right]$$

$$= \underbrace{\underbrace{-\mathbb{E}_{\theta \sim q}\left[\log P(Y|\theta)\right]}_{\text{negative log likelihood}} + \underbrace{KL(q(\theta), p(\theta))}_{\text{complexity term}}}_{\text{ELBO}} + \underbrace{\log P(Y)}_{\text{cste evidence}}$$

To minimize $KL(q(\theta) \,\|\, p(\theta \| Y))$, I will use gradient descent with the reparametrization trick:

let $\Sigma = \begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_d^2 \end{pmatrix}$ recall $q(\theta) = \mathcal{N}(\theta | \mu, \Sigma)$

$$\mathbb{E}_{\theta \sim q}[f(\theta)] = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, I_d)}\left[f(\mu + \sqrt{\Sigma} \cdot \varepsilon)\right]$$

$$KL(q(\theta) \,\|\, p(\theta \| Y)) = -\mathbb{E}_{\varepsilon \sim \mathcal{N}(0, I_d)}\left[\log P(Y | \mu + \sqrt{\Sigma} \cdot \varepsilon)\right] + KL(q(\theta), P(\theta)) + \text{cste}.$$

Learning algorithm:

Repeat

$\quad \varepsilon \sim \mathcal{N}(0, I_d)$

$$\begin{pmatrix} \mu \\ \Sigma \end{pmatrix} \longleftarrow \begin{pmatrix} \mu \\ \Sigma \end{pmatrix} - \nabla_{\mu, \Sigma} \left\{ -\log(P(Y | \mu + \sqrt{\Sigma} \cdot \varepsilon) + KL(\mathcal{N}(\mu, \Sigma), P(\theta)) \right\}$$

To simplify, assume we set $\Sigma = Id$, and we only learn μ.
and $p(\theta) = \mathcal{N}(0, Id)$. The algo becomes:

$$\left[\begin{array}{l} \text{Repeat} \\ \quad \varepsilon \sim \mathcal{N}(0, Id) \\ \quad \mu \leftarrow \mu - \nabla_\mu \left\{ -\log\left(P(\mathcal{Y} \mid \mu + \varepsilon)\right) + \lambda \|\mu\|^2 \right\} \end{array}\right.$$

to make a prediction

$$p(y_{N+1} \mid \mathcal{Y}) = \int p(y_{N+1} \mid \theta)\, \mathcal{N}(\theta \mid \mu, \Sigma)\, d\theta$$

$$\approx \frac{1}{M} \sum_{k=1}^{M} p(y_{N+1} \mid \theta_k)$$

with $\theta_k \sim \mathcal{N}(\mu, \Sigma)$

Note: Langevin SGD requires us to store M models $\theta_1 \dots \theta_M$,
but Variational methods only require us to store μ, Σ which is lighter.

IV Bayesian linear regression

A) $D = \{(x_i, y_i)\}_{i=1}^{N}$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

Assume there exists θ^* such that

$y_i = \theta^{*T} x_i + \varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0, 1)$

so $y_i \sim \mathcal{N}(\cdot \mid \theta^{*T} x_i, 1)$

$$p(\mathbb{Y} \mid \theta) = \prod_{i=1}^{N} p(y_i \mid \theta) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \theta^T x_i, 1)$$

- I define the prior $p(\theta) = \mathcal{N}(0, \sigma^2 \cdot I_d)$

$$P(\theta \mid \mathbb{Y}) = \frac{P(\mathbb{Y} \mid \theta) \cdot P(\theta)}{P(\mathbb{Y})} \quad \text{is gaussian.}$$

$$\begin{aligned}
\log P(\theta \mid \mathbb{Y}) &= \log P(\mathbb{Y} \mid \theta) + \log P(\theta) + \text{cste} \\
&= -\frac{1}{2} \| \mathbb{Y} - \mathbb{X}\theta \|^2 - \frac{1}{2\sigma^2} \|\theta\|^2 + \text{cste} \\
&= -\frac{1}{2} \left\{ \mathbb{Y}^T \mathbb{Y} - 2\theta^T \mathbb{X}^T \mathbb{Y} + \theta^T \mathbb{X}^T \mathbb{X} \theta + \frac{1}{\sigma^2} \|\theta\|^2 \right\} + \text{cste} \\
&= -\frac{1}{2} \theta^T \underbrace{\left(\mathbb{X}^T \mathbb{X} + \frac{1}{\sigma^2} I_d \right)}_{\Sigma^{-1}} \theta + \theta^T \underbrace{\mathbb{X}^T \mathbb{Y}}_{\Sigma^{-1}\mu} + \text{cste}
\end{aligned}$$

I know $P(\theta \mid \mathbb{Y})$ is gaussian.

let $P(\theta \mid \mathbb{Y}) = \mathcal{N}(\mu, \Sigma)$.

Now we have to identify μ and Σ.

$$\begin{aligned}
\log P(\theta \mid \mathbb{Y}) &= -\frac{1}{2} (\theta - \mu)^T \Sigma^{-1} (\theta - \mu) + \text{cste} \\
&= -\frac{1}{2} \theta^T \Sigma^{-1} \theta + \theta^T \Sigma^{-1} \mu + \text{cste}
\end{aligned}$$

$$\Sigma = \left(\mathbb{X}^T \mathbb{X} + \frac{1}{\sigma^2} I_d \right)^{-1}$$

$$\mu = \Sigma \mathbb{X}^T \mathbb{Y} = \underbrace{\left(\mathbb{X}^T \mathbb{X} + \frac{1}{\sigma^2} I_d \right)^{-1} \mathbb{X}^T \mathbb{Y}}_{\text{recall that this coincide with } \hat{\theta}_{MAP}}$$

$P(\theta \mid \mathbb{Y})$ is centered on $\hat{\theta}_{MAP}$

$$p(y_{N+1} \mid \mathbb{Y}) = \int_\Theta p(y_{N+1} \mid \theta)\, p(\theta \mid \mathbb{Y})\, d\theta$$

$$= \int_\theta \mathcal{N}(y_{N+1} \mid \theta^T x_{N+1}, 1)\, \mathcal{N}(\theta \mid \mu, \Sigma)\, d\theta$$

$$\vdots$$

$$= \mathcal{N}(\cdot \mid \mu^T x_{N+1}, 1 + x_{N+1}^T \Sigma x_{N+1})$$

B) Non linear bayesian regression

Like for the kernel trick,

we use a mapping $\phi : \mathbb{R}^d \to \mathbb{R}^D$

we will learn a predictor $\underbrace{f_\theta(x) = \theta^T \phi(x)}_{\text{non linear in } x}$

Assume $\theta \sim p(\theta) = \mathcal{N}(0, Id)$

then $\mathbb{E}_\theta[f_\theta(x)] = 0$

$$\mathrm{Cov}_\theta[f_\theta(x), f_\theta(x')] = \mathbb{E}_\theta[\phi(x)^T \theta\, \theta^T \phi(x')]$$

$$= \phi(x)^T \underbrace{\mathbb{E}_\theta[\theta\theta^T]}_{Id} \phi(x')$$

$$= \phi(x)^T \phi(x')$$

$$= k(x, x')$$

If $\theta \sim \mathcal{N}(0, Id)$

then $f_\theta(x)$ is gaussian

and the vector

$$(f_\theta(x_1), \dots, f_\theta(x_N)) \sim \mathcal{N}(0, K)$$

where $K_{ij} = k(i, j)$

This is a gaussian process.