Statistical Machine Learning (BE4M33SSU)
Lecture 9: Bayesian inference and learning

Czech Technical University in Prague

- Bayesian inference
- Variational Bayesian inference
- Bayesian inference in Deep Learning

1. Motivation

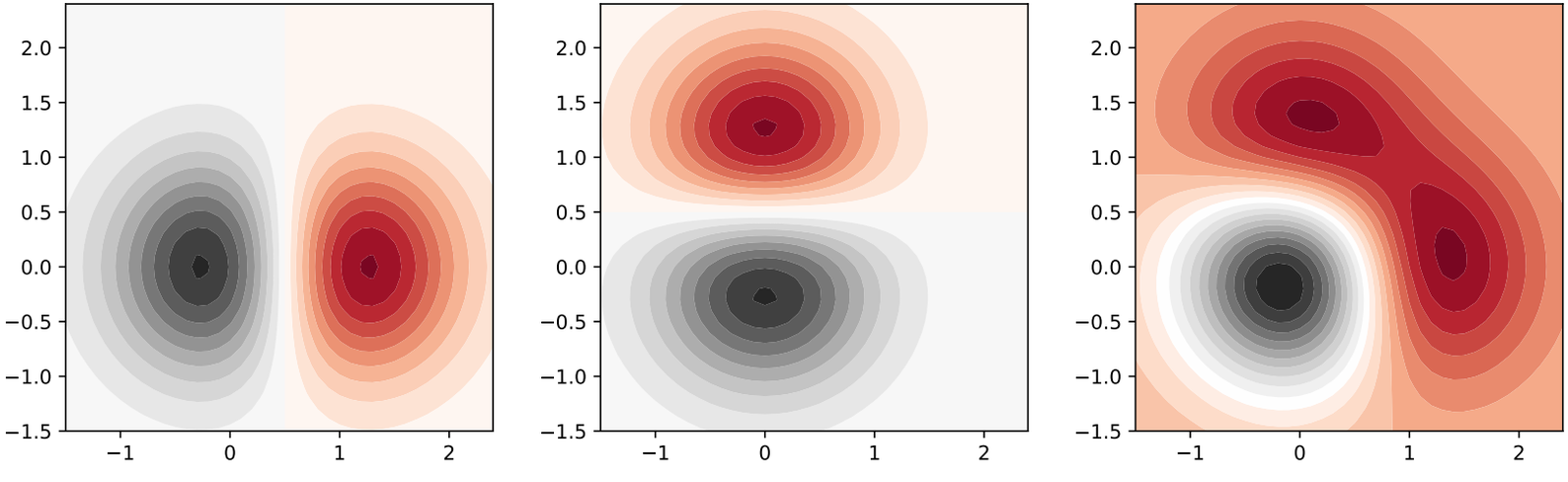
Empirical risk minimisation:

- The predictor h_m learned from $\mathcal{T}^m$ has risk $R(h_m)$ and estimation error $R(h_m) - R(h_{\mathcal{H}})$
- Size of $\mathcal{T}^m$ too small $\Rightarrow$ high estimation error

Maximum likelihood estimate:

- The parameter θ_m estimated from the training set $\mathcal{T}^m$ differs from the true parameter.
- Size of $\mathcal{T}^m$ too small $\Rightarrow$ high estimation error

Small amount of training data and/or weak model class: can we avoid to choose **one** h_m, or to decide for **one** θ_m? Yes, use model mixtures. This has the positive side-effect of more expressive power



Two Gaussian models and their mixture

2. Bayesian inference

Interpret the unknown parameter $\theta \in \Theta$ as a **random** variable

- Parametrised family of models $p(x, y \,|\, \theta)$, $\theta \in \Theta$
- Prior distribution $p(\theta)$ on Θ
- Prediction strategy $h\colon \mathcal{X} \to \mathcal{Y}$ and loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

Given training data $\mathcal{T}^m = \left\{(x^i, y^i) \;\middle|\; i = 1, \ldots, m\right\}$ compute the posterior probability to observe a pair (x, y) by marginalising over $\theta \in \Theta$:

$$p(x, y \,|\, \mathcal{T}^m) = \frac{1}{p(\mathcal{T}^m)} \int_\Theta p(\mathcal{T}^m \,|\, \theta)\, p(x, y \,|\, \theta)\, p(\theta)\, d\theta$$

Notice that a point estimate of θ is no longer needed! Instead we have the posterior parameter distribution

$$p(\theta \,|\, \mathcal{T}^m) = \frac{p(\theta) p(\mathcal{T}^m \,|\, \theta)}{p(\mathcal{T}^m)}$$

and use it as mixture weights:

Define the Bayes risk of a strategy h by

$$R(h, \mathcal{T}^m) \propto \sum_{x,y} \int_\Theta p(\mathcal{T}^m \,|\, \theta)\, p(x, y \,|\, \theta)\, p(\theta)\, \ell(y, h(x))\, d\theta$$

2. Bayesian inference

For 0-1 loss this leads to the predictor

$$h(x, \mathcal{T}^m) = \arg\max_{y \in \mathcal{Y}} \int_\Theta \underbrace{p(\theta)\, p(\mathcal{T}^m \,|\, \theta)}_{\alpha(\theta)}\, p(x, y \,|\, \theta)\, d\theta = \arg\max_{y \in \mathcal{Y}} \int_\Theta \alpha(\theta)\, p(x, y \,|\, \theta)\, d\theta$$

which means to find the optimal predictor for a **model mixture**.

Notice how the posterior distribution

$$\alpha(\theta) = p(\mathcal{T}^m \,|\, \theta)\, p(\theta) \propto p(\theta \,|\, \mathcal{T}^m)$$

interpolates between the situation without any training data, i.e. $m = 0$ and the likelihood of training data for $m \to \infty$.

3. Variational Bayesian inference

- Computing integrals like

$$\int_{\Theta} p(\mathcal{T}^m \,|\, \theta)\, p(\theta)\, d\theta$$

  is in most cases not tractable.
- Approximate $p(\theta \,|\, \mathcal{T}^m)$ by some simple distribution $q_\beta(\theta)$ and find the optimal parameter β by minimising the Kullback-Leibler divergence

$$D_{KL}(q_\beta(\theta) \,\|\, p(\theta \,|\, \mathcal{T}^m)) = D_{KL}(q_\beta(\theta) \,\|\, p(\theta)) - \int_{\Theta} q_\beta(\theta) \log p(\mathcal{T}^m \,|\, \theta)\, d\theta + c \to \min_\beta$$

- use $q_\beta(\theta)$ with optimal β for prediction

$$h(x) = \arg\max_y \sum_{y'} \int_{\Theta} q_\beta(\theta)\, p(x, y \,|\, \theta)\, \ell(y', y)\, d\theta$$

The integrals over θ can be further simplified by sampling from $q_\beta(\theta)$

$$\int_{\Theta} q_\beta(\theta) f(\theta)\, d\theta \approx \frac{1}{m} \sum_{i=1}^{n} f(\theta_i)$$

3. Variational Bayesian inference

Example 1. Consider the optimisation task

$$\int_{\Theta} q_\beta(\theta) \log p(\mathcal{T}^m \,|\, \theta)\, d\theta - D_{KL}(q_\beta(\theta) \,\|\, p(\theta)) \to \max_\beta$$

for the following examples

(1) $p(\theta)$ - uniform, $q_{\theta_m}(\theta) = \delta(\theta - \theta_m)$, i.e. point estimate $\Rightarrow$

$$\theta_m = \arg\max_\theta \log p(\mathcal{T}^m \,|\, \theta)$$

(2) $p(\theta)$ - $\mathcal{N}(0, \sigma_0^2)$, $q_{\theta_m}(\theta) = \delta(\theta - \theta_m)$, i.e. point estimate $\Rightarrow$

$$\theta_m = \arg\max_\theta \Big[\log p(\mathcal{T}^m \,|\, \theta) + \lambda \|\theta\|^2\Big]$$

(3) $p(\theta)$ - $\mathcal{N}(0, \sigma_0^2)$, $q_\beta(\theta)$ - $\mathcal{N}(\mu, \sigma^2)$

$$\frac{1}{\sqrt{2\pi\sigma^2}} \int_{\Theta} e^{-\frac{1}{2\sigma^2}(\theta-\mu)^2} \log p(\mathcal{T}^m \,|\, \theta)\, d\theta - \frac{1}{2}\Big[\frac{\sigma^2 + \mu^2}{\sigma_0^2} - \ln \sigma\Big] \to \max_{\mu,\sigma}$$

4. Bayesian inference in Deep Learning

Dropout (Hinton et al., 2012): randomly switch off neurons with fixed probability p (during training).

- Dropout is equivalent to independent multiplicative binary noise in front of the activation functions of the neurons.
- Dropout implements a probability distribution over network architectures.
- Training: SDG w.r.t. mini-batches + sampled network architecture.
- Test time: sample outputs for given input, or, even simpler, weight node outputs by $(1-p)$.

4. Bayesian inference in Deep Learning

Bayesian inference for deep networks: Model assumptions

- all network weights are random variables
- prior distribution $p(w) = \prod_{ij} p(w_{ij})$ with $w_{ij} \sim \mathcal{N}(0, \sigma_0^2)$
- approximated posterior distribution $q(w) = \prod_{ij} q(w_{ij})$ with $w_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$

Training:

$$\mathbb{E}_{q(w;\mu,\sigma)}\big[\log p(\mathcal{T}^m \,|\, w)\big] - \sum_{ij} \frac{1}{2}\Big[\frac{\sigma_{ij}^2 + \mu_{ij}^2}{\sigma_0^2} - \ln \sigma_{ij}\Big] \to \max_{\mu,\sigma}$$

approximate the integral by sampling and use re-parametrisation

$$w \sim \mathcal{N}(\mu, \sigma^2) \Leftrightarrow w = \sigma(\epsilon + \mu) \text{ with } \epsilon \sim \mathcal{N}(0, 1)$$

Apply SDG w.r.t. mini-batches + sampled network weights.