Chapter 14
Bayesian Learning

supplementary slides to
Machine Learning Fundamentals

published by Cambridge University Press

August 2020

CAMBRIDGE UNIVERSITY PRESS

Outline

Bayesian Learning (I)

- *frequentist* vs. *Bayesian* views in machine learning
  - *frequentist*: model parameters as unknown but fixed quantities
  - *Bayesian*: model parameters as **random variables**
- Bayesians use probability distributions of model parameters
- Bayes' theorem:

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{x})} = \frac{p(\boldsymbol{\theta})\, p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}$$

  - $p(\boldsymbol{\theta})$: prior distribution of model parameters $\boldsymbol{\theta}$
  - $p(\boldsymbol{\theta}|\mathbf{x})$: the posterior distribution of $\boldsymbol{\theta}$ given data $\mathbf{x}$
  - $p(\mathbf{x}|\boldsymbol{\theta})$: the likelihood function of the model
- Bayesian learning rule: *posterior* $\propto$ *prior* $\times$ *likelihood*

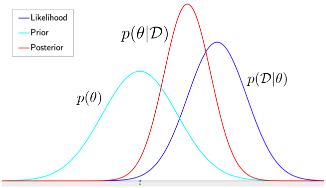
$$p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\boldsymbol{\theta})\, p(\mathbf{x}|\boldsymbol{\theta})$$

Bayesian Learning (II)

- prior specification: $p(\boldsymbol{\theta})$
  - use a prior distribution to describe prior knowledge on models
- Bayesian learning
  - optimally combine prior knowledge with data
  - given a training set: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$
  - Bayesian learning rule: $p(\boldsymbol{\theta}) \xrightarrow{\mathcal{D}} p(\boldsymbol{\theta}|\mathcal{D})$

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})\, p(\mathcal{D}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta})$$

- Bayesian inference
  - make a decision based on $p(\boldsymbol{\theta}|\mathcal{D})$



$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})$$

posterior $\propto$ prior $\times$ likelihood

Bayesian Inference for Classification

- given posterior $p(\boldsymbol{\theta}|\mathcal{D})$ and likelihood $p(\mathbf{x}\,|\,\boldsymbol{\theta})$
- define *predictive distribution* as

$$p(\mathbf{x}\,|\,\mathcal{D}) = \int_{\boldsymbol{\theta}} p(\mathbf{x}\,|\,\boldsymbol{\theta})\, p(\boldsymbol{\theta}\,|\,\mathcal{D})\, d\boldsymbol{\theta}$$

- Bayesian classification:
  - K classes: $\{\omega_1, \omega_2, \cdots, \omega_K\}$
  - choose prior $p(\theta_k)$ and a training set $\mathcal{D}_k$ for each class ω_k
  - Bayesian learning:

$$p(\theta_k\,|\,\mathcal{D}_k) = \frac{p(\theta_k)\, p(\mathcal{D}_k\,|\,\omega_k, \theta_k)}{p(\mathcal{D}_k)} \propto p(\theta_k)\, p(\mathcal{D}_k\,|\,\omega_k, \theta_k)$$

  - Bayesian inference:

$$\begin{aligned} g(\mathbf{x}) &= \arg\max_{k=1}^{K}\; p(\mathbf{x}\,|\,\mathcal{D}_k) \\ &= \arg\max_{k=1}^{K}\; \Pr(\omega_k) \int_{\theta_k} p(\mathbf{x}|\omega_k, \theta_k)\, p(\theta_k\,|\,\mathcal{D}_k)\, d\theta_k \end{aligned}$$

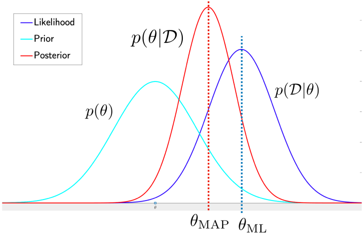
Maximum a Posteriori (MAP) Estimation

- not easy to use a distribution $p(\boldsymbol{\theta}|\mathcal{D})$ to describe models
- point estimation: only use a point to estimate a distribution $p(\boldsymbol{\theta}|\mathcal{D})$
- maximum a posteriori (MAP) estimation:

$$\begin{aligned}\boldsymbol{\theta}_{\mathsf{MAP}} &= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}\,|\,\mathcal{D}) \\ &= \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta})\,p(\mathcal{D}\,|\,\boldsymbol{\theta})\end{aligned}$$

- MAP estimation vs. ML estimation
  - ML solely relies on training data
  - MAP optimally combines prior knowledge with data

$$\boldsymbol{\theta}_{\mathsf{ML}} = \arg\max_{\boldsymbol{\theta}} \; p(\mathcal{D}\,|\,\boldsymbol{\theta})$$



$\boldsymbol{\theta}_{\mathsf{MAP}}$ vs. $\boldsymbol{\theta}_{\mathsf{ML}}$

Sequential Bayesian Learning

- Bayesian learning is an excellent tool for on-line learning, where training data come one by one
- sequential Bayesian learning
  - use the Bayesian learning to update models after each sample
  - track a slowly-changing environment

$$t=0 \qquad\qquad t=1 \qquad\qquad t=2 \qquad\qquad \cdots\cdots \qquad\qquad t=N$$

$$p(\theta) \xrightarrow{\mathbf{x}_1} p(\theta|\mathbf{x}_1) \xrightarrow{\mathbf{x}_2} p(\theta|\mathbf{x}_1,\mathbf{x}_2) \xrightarrow{\mathbf{x}_3} \cdots\cdots \; p(\theta|\mathcal{D}_N)$$

Learning Rule: posterior $\propto$ prior $\times$ likelihood

$$p(\boldsymbol{\theta}\,|\,\mathbf{x}_1) \propto p(\boldsymbol{\theta})p(\mathbf{x}_1\,|\,\boldsymbol{\theta})$$

$$p(\boldsymbol{\theta}\,|\,\mathbf{x}_1,\mathbf{x}_2) \propto p(\boldsymbol{\theta}\,|\,\mathbf{x}_1)\;p(\mathbf{x}_2\,|\,\boldsymbol{\theta})$$

Example: Sequential Bayesian Learning

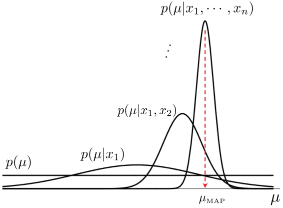
- a univariate Gaussian model with known variance:

$$p(x\,|\,\mu) = \mathcal{N}(x\,|\,\mu, \sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(x-\mu)^2}{2\sigma_0^2}}$$

- choose a prior distribution:

$$p(\mu) = \mathcal{N}(\mu\,|\,\nu_0, \tau_0^2) = \frac{1}{\sqrt{2\pi\tau_0^2}}\ e^{-\frac{(\mu-\nu_0)^2}{2\tau_0^2}}$$

- $p(\mu|x_1) \propto p(\mu)p(x_1|\mu) \implies p(\mu|x_1) = \mathcal{N}(\mu|\nu_1, \tau_1^2)$ with $\nu_1 = \frac{\sigma_0^2}{\tau_0^2+\sigma_0^2}\nu_0 + \frac{\tau_0^2}{\tau_0^2+\sigma_0^2}x_1$ and $\tau_1^2 = \frac{\tau_0^2\sigma_0^2}{\tau_0^2+\sigma_0^2}$
- $p(\mu\,|\,x_1,\cdots,x_n) = \mathcal{N}(\mu|\nu_n, \tau_n^2)$ with $\nu_n = \frac{n\tau_0^2}{n\tau_0^2+\sigma_0^2}\bar{x}_n + \frac{\sigma_0^2}{n\tau_0^2+\sigma_0^2}\nu_0$ and $\tau_n^2 = \frac{\tau_0^2\sigma_0^2}{n\tau_0^2+\sigma_0^2}$



as $n \to \infty$, we have

- $\tau_n \to 0$
- $\nu_n \to \bar{x}_n$
- $\mu_{\text{MAP}} \to \mu_{\text{ML}}$

Conjugate Priors

- *conjugate priors*: a prior is chosen to ensure its posterior has the same functional form as the prior
- conjugate to the likelihood function of the underlying model, i.e. both have the same function form
- choice of conjugate priors leads to computational convenience in Bayesian learning
- not every model has a conjugate prior, e.g. mixture models
- all e-family models have conjugate priors

Examples of Conjugate Priors

model $p(\mathbf{x}\|\boldsymbol{\theta})$	conjugate prior $p(\boldsymbol{\theta})$
1-D Gaussian (known variance) $\mathcal{N}(x \,\|\, \mu, \sigma_0^2)$	1-D Gaussian $\mathcal{N}(\mu \,\|\, \nu, \tau^2)$
1-D Gaussian (known mean) $\mathcal{N}(x \,\|\, \mu_0, \sigma^2)$	inverse-gamma gamma$^{-1}(\sigma^2 \,\|\, \alpha, \beta)$
Gaussian (known covariance) $\mathcal{N}(\mathbf{x} \,\|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}_0)$	Gaussian $\mathcal{N}(\boldsymbol{\mu} \,\|\, \boldsymbol{\nu}, \Phi)$
Gaussian (known mean) $\mathcal{N}(\mathbf{x} \,\|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma})$	inverse-Wishart $\mathcal{W}^{-1}(\boldsymbol{\Sigma} \,\|\, \Phi, \nu)$
multivariate Gaussian $\mathcal{N}(\mathbf{x} \,\|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian-inverse-Wishart GIW$(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,\|\, \boldsymbol{\nu}, \Phi, \lambda, \nu) = \mathcal{N}(\boldsymbol{\mu} \,\|\, \boldsymbol{\nu}, \frac{1}{\lambda}\boldsymbol{\Sigma})\, \mathcal{W}^{-1}(\boldsymbol{\Sigma} \,\|\, \Phi, \nu)$
multinomial Mult$(\mathbf{r} \,\|\, \mathbf{w}) = C(\mathbf{r}) \cdot \prod_{i=1}^{M} w_i^{r_i}$ with $C(\mathbf{r}) = \frac{(r_1+\cdots+r_M)!}{r_1!\cdots r_M!}$	Dirichlet Dir$(\mathbf{w} \,\|\, \boldsymbol{\alpha}) = B(\boldsymbol{\alpha}) \cdot \prod_{i=1}^{M} w_i^{\alpha_i - 1}$ with $B(\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1+\cdots+\alpha_M)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_M)}$

Conjugate Priors for Bayesian Learning: Multinomials

- a sample of some counts: $\mathbf{r} = \begin{bmatrix} r_1 \, r_2 \, \cdots r_M \end{bmatrix}$
- multinomial models: $p(\mathbf{r} \,|\, \mathbf{w}) = \text{Mult}\big(\mathbf{r} \,|\, \mathbf{w}\big) = C(\mathbf{r}) \cdot \prod_{i=1}^{M} w_i^{r_i}$
- the conjugate prior is Dirichlet:

$$p(\mathbf{w}) = \text{Dir}(\mathbf{w} \,|\, \boldsymbol{\alpha}^{(0)}) = B(\boldsymbol{\alpha}^{(0)}) \cdot \prod_{i=1}^{M} w_i^{\alpha_i^{(0)}-1}$$

- Bayesian learning:

$$p(\mathbf{w} \,|\, \mathbf{r}) \propto p(\mathbf{w})\, p(\mathbf{r} \,|\, \mathbf{w}) \propto \prod_{i=1}^{M} w_i^{\alpha_i^{(0)}+r_i-1}$$

- the posterior is also Dirichlet:

$$p(\mathbf{w} \,|\, \mathbf{r}) = \text{Dir}(\mathbf{w} \,|\, \boldsymbol{\alpha}^{(1)}) = B(\boldsymbol{\alpha}^{(1)}) \cdot \prod_{i=1}^{M} w_i^{\alpha_i^{(1)}-1}$$

- MAP estimation:

$$\mathbf{w}^{(\text{MAP})} = \arg\max_{\mathbf{w}} \;\; p(\mathbf{w} \,|\, \mathbf{r}) \qquad \text{subject to} \quad \sum_{i=1}^{M} w_i = 1$$

$$\Longrightarrow \; w_i^{(\text{MAP})} = \frac{\alpha_i^{(1)}-1}{\sum_{i=1}^{M} \alpha_i^{(1)} - M} = \frac{r_i + \alpha_i^{(0)} - 1}{\sum_{i=1}^{M} \big(r_i + \alpha_i^{(0)}\big) - M} \quad \forall i = 1, 2, \cdots, M$$

Conjugate Priors for Bayesian Learning: Gaussians (1)

- Gaussian models: $p(\mathbf{x}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma})$
- the conjugate prior is a Gaussian-inverse-Wishart (GIW) distribution:
$$p(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathrm{GIW}\big(\boldsymbol{\mu},\boldsymbol{\Sigma}\big|\boldsymbol{\nu}_0,\Phi_0,\lambda_0,\nu_0\big) = \mathcal{N}\Big(\boldsymbol{\mu}\Big|\boldsymbol{\nu}_0,\tfrac{1}{\lambda_0}\boldsymbol{\Sigma}\Big)\mathcal{W}^{-1}\Big(\boldsymbol{\Sigma}\Big|\Phi_0,\nu_0\Big)$$
$$= c_0\,\big|\boldsymbol{\Sigma}^{-1}\big|^{\frac{\nu_0+d+2}{2}}\exp\Big[-\tfrac{1}{2}\lambda_0(\boldsymbol{\mu}-\boldsymbol{\nu}_0)^\intercal\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\nu}_0)-\tfrac{1}{2}\mathrm{tr}\big(\Phi_0\boldsymbol{\Sigma}^{-1}\big)\Big]$$
- the likelihood function of a training set $\mathcal{D}=\big\{\mathbf{x}_1,\mathbf{x}_2,\cdots\mathbf{x}_N\big\}$:
$$p(\mathcal{D}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma}) = \prod_{i=1}^{N} p(\mathbf{x}_i\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma})$$
$$= \frac{\big|\boldsymbol{\Sigma}^{-1}\big|^{\frac{N}{2}}}{(2\pi)^{Nd/2}}\exp\Big[-\frac{1}{2}\mathrm{tr}\big(N\mathbf{S}\,\boldsymbol{\Sigma}^{-1}\big)-\frac{N}{2}(\boldsymbol{\mu}-\bar{\mathbf{x}})^\intercal\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\bar{\mathbf{x}})\Big]$$
- Bayesian learning:
$$p(\boldsymbol{\mu},\boldsymbol{\Sigma}\,|\,\mathcal{D}) \propto \mathrm{GIW}\big(\boldsymbol{\mu},\boldsymbol{\Sigma}\,|\,\boldsymbol{\nu}_0,\Phi_0,\lambda_0,\nu_0\big)\cdot p\big(\mathcal{D}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma}\big)$$

Conjugate Priors for Bayesian Learning: Gaussians (2)

- the posterior is another GIW distribution:

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathcal{D}_N) = \mathsf{GIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \boldsymbol{\nu}_1, \Phi_1, \lambda_1, \nu_1)$$
$$= \quad c_1 \left|\boldsymbol{\Sigma}^{-1}\right|^{\frac{\nu_1+d+2}{2}} \exp\left[-\frac{1}{2}\lambda_1(\boldsymbol{\mu}-\boldsymbol{\nu}_1)^\intercal\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\nu}_1) - \frac{1}{2}\mathrm{tr}(\Phi_1\boldsymbol{\Sigma}^{-1})\right]$$

  - $\lambda_1 = \lambda_0 + N$ and $\nu_1 = \nu_0 + N$
  - $\boldsymbol{\nu}_1 = \frac{\lambda_0\boldsymbol{\nu}_0 + N\bar{\mathbf{x}}}{\lambda_0+N}$
  - $\Phi_1 = \Phi_0 + N\mathbf{S} + \frac{\lambda_0 N}{\lambda_0+N}(\bar{\mathbf{x}}-\boldsymbol{\nu}_0)(\bar{\mathbf{x}}-\boldsymbol{\nu}_0)^\intercal$

- MAP estimation: $\{\boldsymbol{\mu}_{\mathsf{MAP}}, \boldsymbol{\Sigma}_{\mathsf{MAP}}\} = \arg\max_{\boldsymbol{\mu},\boldsymbol{\Sigma}}\ p(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathcal{D}_N)$

$$\boldsymbol{\mu}_{\mathsf{MAP}} = \boldsymbol{\nu}_1 = \frac{\lambda_0\boldsymbol{\nu}_0 + N\bar{\mathbf{x}}}{\lambda_0 + N}$$

$$\boldsymbol{\Sigma}_{\mathsf{MAP}} = \frac{\Phi_1}{\nu_1 + d + 1} = \frac{\Phi_0 + N\mathbf{S} + \frac{\lambda_0 N}{\lambda_0+N}(\bar{\mathbf{x}}-\boldsymbol{\nu}_0)(\bar{\mathbf{x}}-\boldsymbol{\nu}_0)^\intercal}{\nu_0 + N + d + 1}$$

Approximate Inference

- when conjugate priors do not exist, Bayesian learning may lead to very complicated posterior distributions

- *approximate inference*: approximate the true posterior distribution with a simple distribution for Bayesian inference

- popular approximate inference methods:
  1. Laplace's method
  2. variational Bayesian (VB) method

Laplace's Method

- use a Gaussian centered at $\boldsymbol{\theta}_{\text{MAP}}$ to approximate the true posterior $p(\boldsymbol{\theta} \,|\, \mathcal{D})$
- Taylor's expansion of $f(\boldsymbol{\theta}) = \ln p(\boldsymbol{\theta} \,|\, \mathcal{D})$ at $\boldsymbol{\theta}_{\text{MAP}}$:

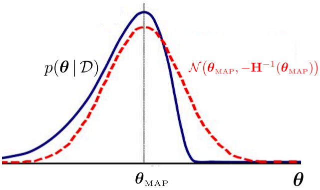
$$f(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_{\text{MAP}}) + \nabla(\boldsymbol{\theta}_{\text{MAP}})\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big) +$$

$$\frac{1}{2!}\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big)^{\intercal}\mathbf{H}(\boldsymbol{\theta}_{\text{MAP}})\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big) + \cdots$$

- 2nd-order approximation:

$$f(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}_{\text{MAP}}) + \frac{1}{2}\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big)^{\intercal}\mathbf{H}(\boldsymbol{\theta}_{\text{MAP}})\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big)$$

$$p(\boldsymbol{\theta} \,|\, \mathcal{D}) \approx \underbrace{C \cdot \exp\left(\frac{1}{2}\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big)^{\intercal}\mathbf{H}(\boldsymbol{\theta}_{\text{MAP}})\big(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}\big)\right)}_{\mathcal{N}\left(\boldsymbol{\theta}_{\text{MAP}},\, -\mathbf{H}^{-1}(\boldsymbol{\theta}_{\text{MAP}})\right)}$$



Bayesian Learning of Logistic Regression

- a training set $\mathcal{D} = \big\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\big\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$
- likelihood function of logistic regression:

$$p(\mathcal{D} \,|\, \mathbf{w}) = \prod_{i=1}^{N} \Big(l(\mathbf{w}^\intercal \mathbf{x}_i)\Big)^{y_i} \Big(1 - l(\mathbf{w}^\intercal \mathbf{x}_i)\Big)^{1-y_i}$$

- choose a Gaussian prior: $p(\mathbf{w}) = \mathcal{N}\big(\mathbf{w} \,|\mathbf{w}_0, \boldsymbol{\Sigma}_0\big)$
- Bayesian learning: $p(\mathbf{w} \,|\, \mathcal{D}) \propto p(\mathbf{w})\, p(\mathcal{D} \,|\, \mathbf{w})$
- the posterior $p(\mathbf{w} \,|\, \mathcal{D})$ is not Gaussian anymore
- use **Laplace's method** to approximate the true posterior
  - use a gradient descent to find $\mathbf{w}_{\text{MAP}}$

    $\nabla(\mathbf{w}) = \nabla \ln p(\mathbf{w} \,|\, \mathcal{D}) = -\boldsymbol{\Sigma}_0^{-1}(\mathbf{w} - \mathbf{w}_0) + \sum_{i=1}^{N} \big(y_i - l(\mathbf{w}^\intercal \mathbf{x}_i)\big)\mathbf{x}_i$
  - use a Gaussian approximation:

$$p(\mathbf{w} \,|\, \mathcal{D}) \approx \mathcal{N}\Big(\mathbf{w} \,\Big|\, \mathbf{w}_{\text{MAP}}, -\mathbf{H}^{-1}(\mathbf{w}_{\text{MAP}})\Big)$$

    with $\mathbf{H}(\mathbf{w}) = -\boldsymbol{\Sigma}_0^{-1} - \sum_{i=1}^{N} l(\mathbf{w}^\intercal \mathbf{x}_i)\big(1 - l(\mathbf{w}^\intercal \mathbf{x}_i)\big)\, \mathbf{x}_i \mathbf{x}_i^\intercal$

Variational Bayesian Methods (I)

- variational Bayesian (VB): use a simpler *variational distribution* $q(\boldsymbol{\theta})$ to approximate the true posterior $p(\boldsymbol{\theta} \,|\, \mathcal{D})$:

$$q^*(\boldsymbol{\theta}) = \arg\min_q \ \mathsf{KL}\Big(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \,|\, \mathcal{D})\Big)$$

- $\mathsf{KL}\Big(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \,|\, \mathcal{D})\Big) = \ln p(\mathcal{D}) - \underbrace{\int_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \ln \frac{p(\mathcal{D}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \, d\boldsymbol{\theta}}_{L(q)}$
- $\min_q \ \mathsf{KL}\Big(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \,|\, \mathcal{D})\Big) \iff \max_q \ L(q)$
- assume $q(\boldsymbol{\theta}) = q_1(\boldsymbol{\theta}_1)\, q_2(\boldsymbol{\theta}_2) \, \cdots \, q_I(\boldsymbol{\theta}_I)$ can be factorized over some disjoint subsets $\boldsymbol{\theta} = \boldsymbol{\theta}_1 \cup \boldsymbol{\theta}_2 \cup \cdots \cup \boldsymbol{\theta}_I$
- $L(q) = \int_{\boldsymbol{\theta}} \ \prod_{i=1}^I \ q_i(\boldsymbol{\theta}_i) \ln p(\mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{\theta} - \sum_{i=1}^I \ \int_{\boldsymbol{\theta}_i} q_i(\boldsymbol{\theta}_i) \ln q_i(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i$

Variational Bayesian Methods (II)

- maximize $L(q)$ w.r.t. each $q_i(\boldsymbol{\theta}_i)$ separately

$$\max_{q_i} \int_{\boldsymbol{\theta}_i} q_i(\boldsymbol{\theta}_i) \underbrace{\Big[\int_{\boldsymbol{\theta}_{j\neq i}} \prod_{j\neq i} q_j(\boldsymbol{\theta}_j) \ln p(\mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{\theta}_{j\neq i} \Big]}_{\mathbb{E}_{j\neq i}\big[\ln p(\mathcal{D},\boldsymbol{\theta})\big]} d\boldsymbol{\theta}_i - \int_{\boldsymbol{\theta}_i} q_i(\boldsymbol{\theta}_i) \ln q_i(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i$$

- define a new distribution: $\widetilde{p}(\boldsymbol{\theta}_i; \mathcal{D}) \propto \exp\Big(\mathbb{E}_{j\neq i}\big[\ln p(\mathcal{D}, \boldsymbol{\theta})\big]\Big)$
- we have $q_i^*(\boldsymbol{\theta}_i) = \arg\max_{q_i} \int_{\boldsymbol{\theta}_i} q_i(\boldsymbol{\theta}_i) \ln \frac{\widetilde{p}(\boldsymbol{\theta}_i;\mathcal{D})}{q_i(\boldsymbol{\theta}_i)} d\boldsymbol{\theta}_i$

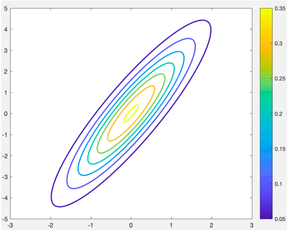
$$\Longrightarrow \; q_i^*(\boldsymbol{\theta}_i) = \arg\min_{q_i} \; \mathsf{KL}\Big(q_i(\boldsymbol{\theta}_i) \,\|\, \widetilde{p}(\boldsymbol{\theta}_i; \mathcal{D})\Big)$$

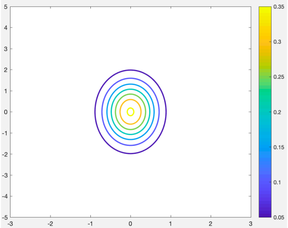
- derive $q_i^*(\boldsymbol{\theta}_i) = \widetilde{p}(\boldsymbol{\theta}_i; \mathcal{D}) \propto \exp\Big(\mathbb{E}_{j\neq i}\big[\ln p(\mathcal{D}, \boldsymbol{\theta})\big]\Big)$ or

$$\ln q_i^*(\boldsymbol{\theta}_i) = \mathbb{E}_{j\neq i}\big[\ln p(\mathcal{D}, \boldsymbol{\theta})\big] + C$$

Variational Bayesian Methods (III)

- mean field theory: use a factorizable variational distribution to approximate a true posterior distribution
- a 2-D Gaussian with $\mathbf{\Sigma} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$
- approximate with a variational distribution $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$
- best-fit is found by minimizing the KL-divergence





Variational Bayesian Learning of GMMs (I)

- a Gaussian mixture model (GMM):

$$p(\mathbf{x}\,|\,\boldsymbol{\theta}) = \textstyle\sum_{m=1}^{M} w_m \cdot \mathcal{N}\big(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\big)$$

where model parameters $\boldsymbol{\theta} = \big\{w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m \,|\, m = 1, 2, \cdots, M\big\}$

- no conjugate prior exists for GMMs
- choose a prior distribution as

$$p(\boldsymbol{\theta}) = p(w_1, \cdots, w_M) \prod_{m=1}^{M} p(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

with

$$p(w_1, \cdots, w_M) = \mathsf{Dir}(w_1, \cdots, w_M \,|\, \alpha_1^{(0)}, \cdots, \alpha_M^{(0)})$$

$$p(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \mathsf{GIW}\big(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m \,|\, \boldsymbol{\nu}_m^{(0)}, \Phi_m^{(0)}, \lambda_m^{(0)}, \nu_m^{(0)}\big)$$

Variational Bayesian Learning of GMMs (II)

- introduce 1-of-M latent variable $\mathbf{z} = \begin{bmatrix} z_1 \, z_2 \, \cdots z_M \end{bmatrix}$ for GMMs:

$$p(\mathbf{x}, \mathbf{z} \,|\, \boldsymbol{\theta}) = \prod_{m=1}^{M} \left(w_m\right)^{z_m} \Big(\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\Big)^{z_m}$$

- use the variational Bayesian method to approximate the posterior distribution $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{x})$
- introduce a variational distribution factorized as:

$$q(\mathbf{z}, \boldsymbol{\theta}) = q(\mathbf{z})q(\boldsymbol{\theta}) = q(\mathbf{z})\, q(w_1, \cdots, w_M) \prod_{m=1}^{M} q(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

- derive the best-fit variational distribution $q^*(\mathbf{z}, \boldsymbol{\theta})$

Variational Bayesian Learning of GMMs (III)

1. $\ln q^*(\mathbf{z}) = \mathbb{E}_{\boldsymbol{\theta}}\big[\ln p(\mathbf{x},\mathbf{z},\boldsymbol{\theta})\big] + C = \mathbb{E}_{\boldsymbol{\theta}}\big[\ln p(\boldsymbol{\theta}) + \ln p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})\big] + C$

   $\Longrightarrow \quad \ln q^*(\mathbf{z}) = C' +$
   $$\sum_{m=1}^{M} z_m \Bigg(\underbrace{\mathbb{E}\big[\ln w_m\big] - \mathbb{E}\Big[\frac{\ln|\boldsymbol{\Sigma}_m|}{2}\Big] - \mathbb{E}\Big[\frac{(\mathbf{x}-\boldsymbol{\mu}_m)^\intercal {\boldsymbol{\Sigma}_m}^{-1}(\mathbf{x}-\boldsymbol{\mu}_m)}{2}\Big]}_{\ln \rho_m} \Bigg)$$

   - $q^*(\mathbf{z})$ is a multinomial: $q^*(\mathbf{z}) \propto \prod_{m=1}^{M} \big(\rho_m\big)^{z_m} \propto \prod_{m=1}^{M} \big(r_m\big)^{z_m}$, where $r_m = \frac{\rho_m}{\sum_{m=1}^{M}\rho_m}$ for all m

2. $\ln q^*(w_1,\cdots,w_M) = \mathbb{E}_{\mathbf{z},\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m}\Big[\ln p(\boldsymbol{\theta}) + \ln p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})\Big]$

   $= \sum_{m=1}^{M}(\alpha_m^{(0)} - 1)\ln w_m + \sum_{m=1}^{M} r_m \ln w_m + C$

   - $q^*(w_1,\cdots,w_M)$ is a Dirichlet distribution:
   $$q^*(w_1,\cdots,w_M) = \mathrm{Dir}\big(w_1,\cdots,w_M \,|\, \alpha_1^{(1)},\cdots,\alpha_M^{(1)}\big)$$
   where $\alpha_m^{(1)} = \alpha_m^{(0)} + r_m$ for all $m = 1,2,\cdots,M$

Variational Bayesian Learning of GMMs (IV)

3. $$\ln q^*(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \mathbb{E}_{\mathbf{z},w_m}\Big[\ln p(\boldsymbol{\theta}) + \ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})\Big] + C$$
$$= \ln p(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) + \mathbb{E}\big[z_m\big] \ln \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) + C'$$

   - $q^*(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ is also a GIW distribution:

$$q^*(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \text{GIW}\big(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m \,|\, \boldsymbol{\nu}_m^{(1)}, \Phi_m^{(1)}, \lambda_m^{(1)}, \nu_m^{(1)}\big)$$

where

$$\lambda_m^{(1)} = \lambda_m^{(0)} + r_m$$
$$\nu_m^{(1)} = \nu_m^{(0)} + r_m$$
$$\boldsymbol{\nu}_m^{(1)} = \frac{\lambda_m^{(0)}\boldsymbol{\nu}_m^{(0)} + r_m\mathbf{x}}{\lambda_m^{(0)} + r_m}$$
$$\Phi_m^{(1)} = \Phi_m^{(0)} + \frac{\lambda^{(0)} r_m}{\lambda_m^{(0)} + r_m}\big(\mathbf{x} - \boldsymbol{\nu}_m^{(0)}\big)\big(\mathbf{x} - \boldsymbol{\nu}_m^{(0)}\big)^\intercal$$

Variational Bayesian Learning of GMMs (V)

- based on the above distributions, we have

$$\ln \pi_m \triangleq \mathbb{E}\big[\ln w_k\big] = \psi\big(\alpha_m^{(1)}\big) - \psi\Big(\sum_{m=1}^{M} \alpha_m^{(1)}\Big)$$

$$\ln B_m \triangleq \mathbb{E}\big[\ln \big|\boldsymbol{\Sigma}_m\big|\big] = \sum_{i=1}^{d} \psi\Big(\frac{\lambda_m + 1 - i}{2}\Big) - \ln \big|\Phi_m^{(1)}\big|$$

$$\mathbb{E}\Big[(\mathbf{x}-\boldsymbol{\mu}_m)^\intercal \boldsymbol{\Sigma}_m{}^{-1}(\mathbf{x}-\boldsymbol{\mu}_m)\Big] = \frac{d}{\nu_m^{(1)}} + \lambda_m^{(1)}(\mathbf{x}-\boldsymbol{\nu}_m^{(1)})^\intercal \big(\Phi_m^{(1)}\big)^{-1}(\mathbf{x}-\boldsymbol{\nu}_m^{(1)})$$

  to compute ρ_m as well as r_m $(\forall m = 1, 2, \cdots, M)$
- derive an EM-like algorithm to solve mutual dependency

Variational Bayesian Learning of GMMs (VI)

Variational Bayesian GMMs

Input: $\left\{\alpha_m^{(0)}, \boldsymbol{\nu}_m^{(0)}, \Phi_m^{(0)}, \lambda_m^{(0)}, \nu_m^{(0)} \,|\, m = 1, 2, \cdots, M\right\}$

set $n = 0$

while not converge **do**

E-step: collect statistics:

$$\left\{\alpha_m^{(n)}, \boldsymbol{\nu}_m^{(n)}, \Phi_m^{(n)}, \lambda_m^{(n)}, \nu_m^{(n)}\right\} + \mathbf{x} \longrightarrow \left\{r_m\right\}$$

M-step: update all hyperparameters:

$$\left\{\alpha_m^{(n)}, \boldsymbol{\nu}_m^{(n)}, \Phi_m^{(n)}, \lambda_m^{(n)}, \nu_m^{(n)}\right\} + \left\{r_m\right\} + \mathbf{x}$$

$$\longrightarrow \left\{\alpha_m^{(n+1)}, \boldsymbol{\nu}_m^{(n+1)}, \Phi_m^{(n+1)}, \lambda_m^{(n+1)}, \nu_m^{(n+1)}\right\}$$

$n = n + 1$

end while

Non-Parametric Bayesian Methods

- Bayesian learning of parametric models: rely on prior/posterior distributions of model *parameters*
- how about Bayesian learning of non-parametric models?
- non-parametric Bayesian methods: use stochastic processes as priors for non-parametric models
  - **Gaussian processes**
  - Dirichlet processes

Gaussian Processes: Concepts (I)

- given an arbitrary function $f(\mathbf{x})$
- for any set of N points in $\mathbb{R}^d$, i.e. $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$
- function values form an N-dimensional real-valued vector

$$\mathbf{f} = \big[f(\mathbf{x}_1)\, f(\mathbf{x}_2)\, \cdots\, f(\mathbf{x}_N)\big]^\intercal$$

- assume $\mathbf{f}$ follows a multivariate Gaussian distribution

$$\mathbf{f} = \big[f(\mathbf{x}_1)\, f(\mathbf{x}_2)\, \cdots\, f(\mathbf{x}_N)\big]^\intercal \sim \mathcal{N}\big(\boldsymbol{\mu}_\mathcal{D}, \boldsymbol{\Sigma}_\mathcal{D}\big)$$

  where $\boldsymbol{\mu}_\mathcal{D}$ and $\boldsymbol{\Sigma}_\mathcal{D}$ depends on N data points in $\mathcal{D}$
- it holds for any $\mathcal{D}$, $f(\mathbf{x})$ is a sample from a **Gaussian process**:

$$f(\mathbf{x}) \sim \mathrm{GP}\Big(\mathbf{m}(\mathbf{x}), \Phi(\mathbf{x}, \mathbf{x}')\Big)$$

  - $\mathbf{m}(\mathbf{x})$: *mean function* $\Longrightarrow \boldsymbol{\mu}_\mathcal{D}$
  - $\Phi(\mathbf{x}, \mathbf{x}')$: *covariance function* $\Longrightarrow \boldsymbol{\Sigma}_\mathcal{D}$

Gaussian Processes: Concepts (II)

- how to specify a Gaussian process?
- mean function $\mathbf{m}(\mathbf{x}) = 0$
- covariance function: Mercer's condition

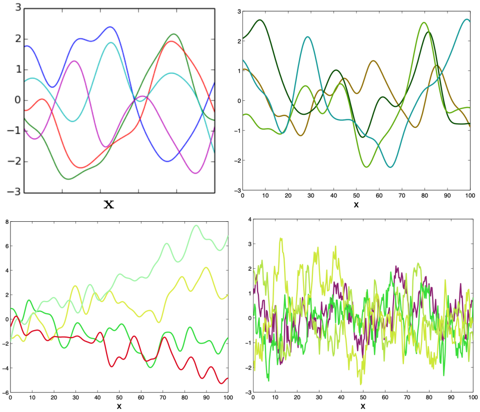
$$\Phi(\mathbf{x}, \mathbf{x}') = \mathrm{cov}\big(f(\mathbf{x}_i), f(\mathbf{x}_j)\big)$$

  - $\mathbf{\Sigma}_{\mathcal{D}} = \Big[\ \Phi(\mathbf{x}_i, \mathbf{x}_j)\ \Big]_{N\times N}$

- RBF kernel function

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2\, e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2l^2}}$$

  - σ: vertical scale
  - l: horizontal scale



Gaussian Processes for Non-Parametric Bayesian Learning

- Gaussian processes as a non-parametric prior
  - randomly sample a function $f(\cdot)$ from a Gaussian process
  - a prior can be implicitly computed with a data set

$$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$$

  - function values $\mathbf{f}$ follow a multivariate Gaussian distribution
  - non-parametric prior:

$$p(f \mid \mathcal{D}) = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \boldsymbol{\Sigma}_{\mathcal{D}})$$

- Gaussian processes for regression or classification
  - input-*output* pairs yield likelihood function
  - apply Bayesian learning rule:

$$\textit{posterior} \propto \textit{prior} \times \textit{likelihood}$$

Gaussian Processes for Regression (I)

basic setting for regression:

- $f(\mathbf{x}) \sim \mathsf{GP}\Big(\mathbf{0}, \Phi(\mathbf{x}, \mathbf{x}')\Big)$
- $y = f(\mathbf{x}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_0^2)$

- given a training set: $\mathcal{D} = \big\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\big\}$
- the corresponding outputs: $\mathbf{y} = \big[y_1 \, y_2 \, \cdots \, y_N\big]^{\mathsf{T}}$
- a non-parametric prior:

$$p\big(f \,|\, \mathcal{D}\big) = \mathcal{N}\big(\mathbf{f} \,|\, \mathbf{0}, \mathbf{\Sigma}_{\mathcal{D}}\big)$$

- the likelihood function due to the residual Gaussian noise ϵ:

$$p\big(\mathbf{y} \,|\, f, \mathcal{D}\big) = \mathcal{N}\big(\mathbf{y} \,|\, \mathbf{f}, \sigma_0^2 \mathbf{I}\big)$$

Gaussian Processes for Regression (II)

- Bayesian learning for the predictive distribution:

$$
\begin{aligned}
p(\mathbf{y} \mid \mathcal{D}) &= \int_f p(\mathbf{y}, f \mid \mathcal{D})\, df = \int_f p(\mathbf{y} \mid f, \mathcal{D}) p(f \mid \mathcal{D})\, df \\
&= \int_{\mathbf{f}} \mathcal{N}\big(\mathbf{y} \mid \mathbf{f}, \sigma_0^2 \mathbf{I}\big) \mathcal{N}\big(\mathbf{f} \mid \mathbf{0}, \mathbf{\Sigma}_{\mathcal{D}}\big)\, d\mathbf{f} \\
&= \mathcal{N}\big(\mathbf{y} \mid \mathbf{0}, \mathbf{\Sigma}_{\mathcal{D}} + \sigma_0^2 \mathbf{I}\big) = \mathcal{N}\big(\mathbf{y} \mid \mathbf{0}, \mathbf{C}_N\big)
\end{aligned}
$$

- hyper-parameter learning:

$$
\{\sigma^*, l^*, \sigma_0^*\} = \arg\max_{\sigma, l, \sigma_0} \; p\big(\mathbf{y} \mid \mathcal{D}, \sigma, l, \sigma_0\big) = \arg\max_{\sigma, l, \sigma_0} \; \ln \mathcal{N}\big(\mathbf{y} \mid \mathbf{0}, \mathbf{C}_N\big)
$$

  - may use a gradient descent method

Gaussian Processes for Regression (III)

- predict output $\tilde{y}$ for a new input $\tilde{\mathbf{x}}$:

$$p(\mathbf{y}, \tilde{y} \mid \mathcal{D}, \mathbf{x}) = \mathcal{N}(\mathbf{y}, \tilde{y} \mid \mathbf{0}, \mathbf{C}_{N+1})$$

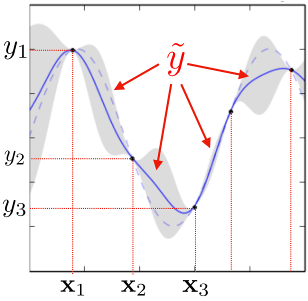
with

$$\mathbf{C}_{N+1} = \left[\begin{array}{c|c} \mathbf{C}_N & \mathbf{k} \\ \hline \mathbf{k}^\intercal & \kappa^2 \end{array} \right]$$

where $\kappa^2 = \Phi(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) + \sigma_0^2$ and $\mathbf{k}_i = \Phi(\mathbf{x}_i, \tilde{\mathbf{x}})$

- the predictive distribution:

$$\begin{aligned} p(\tilde{y} \mid \mathcal{D}, \mathbf{y}, \tilde{\mathbf{x}}) &= \frac{p(\mathbf{y}, \tilde{y} \mid \mathcal{D}, \tilde{\mathbf{x}})}{p(\mathbf{y} \mid \mathcal{D})} \\ &= \mathcal{N}\Big(\tilde{y} \,\Big|\, \mathbf{k}^\intercal \mathbf{C}_N^{-1} \mathbf{y}, \kappa^2 - \mathbf{k}^\intercal \mathbf{C}_N^{-1} \mathbf{k}\Big) \end{aligned}$$



point estimation (MAP or mean):

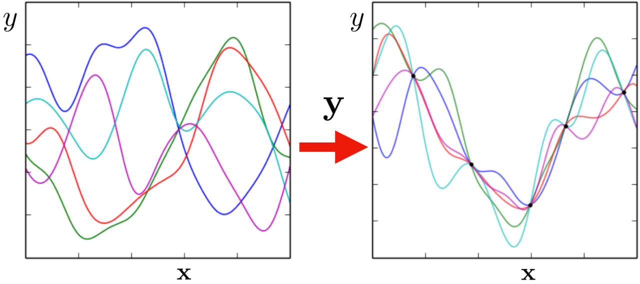
$$\mathbb{E}\big[\tilde{y} \mid \mathcal{D}, \mathbf{y}, \tilde{\mathbf{x}}\big] = \tilde{y}_{\text{MAP}}$$

$$= \mathbf{k}^\intercal \mathbf{C}_N^{-1} \mathbf{y}$$

Gaussian Processes for Regression (IV)

- derive a non-parametric prior from $\mathcal{D}$ and $\Phi(\mathbf{x}, \mathbf{x}')$
- non-parametric Bayesian learning based on $\mathbf{y}$:

non-parametric prior $p(f \mid \mathcal{D})$ **non-parametric posterior** $p(f \mid \mathcal{D}, \mathbf{y})$



Gaussian Processes for Classification

basic setting for binary classification $y \in \{0, 1\}$:

- $f(\mathbf{x}) \sim \text{GP}\Big(\mathbf{0}, \Phi(\mathbf{x}, \mathbf{x}')\Big)$
- $\Pr\big(y = 1 \,|\, \mathbf{x}\big) = l\big(f(\mathbf{x})\big) = \frac{1}{1+e^{-f(\mathbf{x})}}$

  - given a training set $\mathcal{D} = \Big\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\Big\}$ and the corresponding outputs $\mathbf{y} = \big[y_1 \; y_2 \; \cdots \; y_N\big]^\intercal$
  - non-parametric prior: $p\big(f \,|\, \mathcal{D}\big) = \mathcal{N}\big(\mathbf{f} \,|\, \mathbf{0}, \mathbf{\Sigma}_{\mathcal{D}}\big)$
  - likelihood: $p(\mathbf{y} \,|\, f, \mathcal{D}) = \prod_{i=1}^{N} \Big(l\big(f(\mathbf{x}_i)\big)\Big)^{y_i} \Big(1 - l\big(f(\mathbf{x}_i)\big)\Big)^{1-y_i}$
  - no closed-form solution to derive the marginal and predictive distributions, i.e. $p(\mathbf{y} \,|\, \mathcal{D})$ and $p\big(\tilde{y} \,|\, \mathcal{D}, \mathbf{y}, \tilde{\mathbf{x}}\big)$
  - require approximate inference, such as Laplace's method