Probabilistic Machine Learning
Lecture 22
Summary and Cleanup

Philipp Hennig

17 July 2023

Eberhard Karls Universität Tübingen

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

The course so far

Intermediate summary, more to come in the final lectures

Probabilities: the language of reasoning under uncertainty

Lectures 1–3

- We can describe all inference tasks by assigning **probabilities**, (or, for continuous variables probability density functions) jointly to all variables in the problem.
- Probabilities and pdfs satisfy "the rules of probability":

$$\int_{\mathbb{R}^d} p(x)\,dx = 1$$

$$p_{X_1}(x_1) = \int_{\mathbb{R}} p_X(x_1, x_2)\,dx_2 \qquad \textbf{sum rule}$$

$$p(x_1 \mid x_2) = \frac{p(x_1, x_2)}{p(x_2)} \qquad \textbf{product rule}$$

$$p(x_1 \mid x_2) = \frac{p(x_1) \cdot p(x_2 \mid x_1)}{\int p(x_1) \cdot p(x_2 \mid x_1)\,dx_1} \qquad \textbf{Bayes' Theorem.}$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Definition (Exponential Family, simplified form)

Consider a random variable X taking values $x \in \mathbb{X} \subset \mathbb{R}^n$. A probability distribution for X with pdf of the functional form

$$p_w(x) = h(x) \exp\left[\phi(x)^\intercal w - \log Z(w)\right] = \frac{h(x)}{Z(w)} e^{\phi(x)^\intercal w} = p(x \mid w)$$

is called an **exponential family** of probability measures. The function $\phi : \mathbb{X} \rightharpoonup \mathbb{R}^d$ is called the **sufficient statistics**. The parameters $w \in \mathbb{R}^d$ are the **natural parameters** of p_w. The normalization constant $Z(w) : \mathbb{R}^d \rightharpoonup \mathbb{R}$ is the **partition function**. The function $h(x) : \mathbb{X} \rightharpoonup \mathbb{R}_+$ is the **base measure**. For notational convenience, it can be useful to re-parametrize the natural parameters w as $w := \eta(\theta)$ in terms of *canonical parameters* θ.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Exponential Families have Conjugate Priors

- Consider the exponential family $p_w(x \mid w) = h(x) \exp\left[\phi(x)^\intercal w - \log Z(w)\right]$
- its conjugate prior is the exponential family $F(\alpha, \nu) = \int \exp(\alpha^\intercal w - \nu \log Z(w))\, dw$

$$p_\alpha(w \mid \alpha, \nu) = \exp\left[\begin{pmatrix} w \\ -\log Z(w) \end{pmatrix}^\intercal \begin{pmatrix} \alpha \\ \nu \end{pmatrix} - \log F(\alpha, \nu)\right]$$

$$\text{because } p_\alpha(w \mid \alpha, \nu) \prod_{i=1}^{n} p_w(x_i \mid w) \propto p_\alpha\left(w \,\middle|\, \alpha + \sum_i \phi(x_i), \nu + n\right)$$

- and the predictive is

$$p(x) = \int p_w(x \mid w) p_\alpha(w \mid \alpha, \nu)\, dw = h(x) \int e^{(\phi(x)+\alpha)^\intercal w + (\nu+1) \log Z(w) - \log F(\alpha,\nu)}\, dw$$

$$= h(x) \frac{F(\phi(x) + \alpha, \nu + 1)}{F(\alpha, \nu)}$$

Computing $F(\alpha, \nu)$ can be tricky. In general, this is **the** challenge when constructing an EF.

Gaussians: Inference as Linear Algebra

Lectures 5–6

- products of Gaussians are Gaussians

$$\begin{aligned}&\mathcal{N}(x; a, A)\mathcal{N}(x; b, B)\\ &= \mathcal{N}(x; c, C)\mathcal{N}(a; b, A+B)\end{aligned}$$

$$C := (A^{-1} + B^{-1})^{-1} \qquad c := C(A^{-1}a + B^{-1}b)$$

- linear projections of Gaussians are Gaussians

$$\begin{aligned}&p(z) = \mathcal{N}(z; \mu, \Sigma)\\ \Rightarrow \quad &p(Az) = \mathcal{N}(Az, A\mu, A\Sigma A^\intercal)\end{aligned}$$

- marginals of Gaussians are Gaussians

$$\int \mathcal{N}\left[\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}\right] dy = \mathcal{N}(x; \mu_x, \Sigma_{xx})$$

- (linear) conditionals of Gaussians are Gaussians

$$\begin{aligned}p(x \mid y) &= \frac{p(x, y)}{p(y)}\\ &= \mathcal{N}\left(x; \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\right)\end{aligned}$$

Bayesian inference becomes linear algebra

If $p(x) = \mathcal{N}(x; \mu, \Sigma)$ and $p(y \mid x) = \mathcal{N}(y; A^\intercal x + b, \Lambda)$, then

$$p(B^\intercal x + c \mid y) = \mathcal{N}[B^\intercal x + c; B^\intercal \mu + c + B^\intercal \Sigma A(A^\intercal \Sigma A + \Lambda)^{-1}(y - A^\intercal \mu - b), B^\intercal \Sigma B - B^\intercal \Sigma A(A^\intercal \Sigma A + \Lambda)^{-1} A^\intercal \Sigma B]$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\text{prior} \quad p(w) = \mathcal{N}(w; \mu, \Sigma) \quad \Rightarrow \quad p(f) = \mathcal{N}(f_x; \phi_x^\intercal \mu, \phi_x \Sigma \phi_x)$$

$$\text{likelihood} \quad p(y \mid w, \phi_X) = \mathcal{N}(y; \phi_X^\intercal w, \sigma^2 I) = \mathcal{N}(y; f_X, \sigma^2 I)$$

$$\begin{aligned}
\text{posterior on } \boldsymbol{w} \quad p(w \mid \boldsymbol{y}, \phi_X) &= \mathcal{N}(w; \mu + \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal \mu), \\
&\qquad \Sigma - \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}\phi_X^\intercal \Sigma) \\
&= \mathcal{N}\left(w; (\Sigma^{-1} + \sigma^{-2}\phi_X\phi_X^\intercal)^{-1}\left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right),\right. \\
&\qquad \left.(\Sigma^{-1} + \sigma^{-2}\phi_X\phi_X^\intercal)^{-1}\right)
\end{aligned}$$

$$\begin{aligned}
\text{posterior on } f \quad p(f_x \mid y, \phi_X) &= \mathcal{N}(f_x; \phi_x^\intercal \mu + \phi_x^\intercal \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal \mu), \\
&\qquad \phi_x^\intercal \Sigma \phi_x - \phi_x^\intercal \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}\phi_X^\intercal \Sigma \phi_x) \\
&\qquad \mathcal{N}\left(f_x; \phi_x(\Sigma^{-1} + \sigma^{-2}\phi_X\phi_X^\intercal)^{-1}\left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right),\right. \\
&\qquad \left.\phi_x(\Sigma^{-1} + \sigma^{-2}\phi_X\phi_X^\intercal)^{-1}\phi_x^\intercal\right)
\end{aligned}$$

Gaussian Processes: Models with "infinite freedom"

Lectures 8-11

$$
\begin{aligned}
p(f(\bullet) \mid \boldsymbol{w}) &= \mathcal{N}(f(\bullet); \phi(\bullet)^\intercal \boldsymbol{w}, \sigma I) \\
p(f(\bullet)) &= \int p(f(\bullet) \mid \boldsymbol{w}) p(\boldsymbol{w}) \, \mathrm{d}\boldsymbol{w} \\
&= \mathcal{N}(f(\bullet); \phi(\bullet)^\intercal \mu, \phi(\bullet)^\intercal \Sigma \phi(\circ) + \sigma I)
\end{aligned}
$$

using the abstraction / encapsulation

$m(\bullet) := \phi(\bullet)^\intercal \mu$	$m : \mathbb{X} \to \mathbb{R}$	mean function
$k(\bullet, \circ) := \phi(\bullet)^\intercal \Sigma \phi(\circ)$	$k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$	covariance function, aka. **kernel**

Algorithm 1 Linear Algebra for/as Gaussian (process) inference – efficient data-loading and book-keeping

Input: sufficient statistics $K = k_{XX} + \sigma^2 I, \bar{y} = y - \mu_X$, initial guesses α_0, C_0

Output: defragmented statistics S, C_i, α_i

procedure TRAIN($K, y, C_0 = 0, \alpha_0 = 0$)
    for $i \in \{1, \dots, n\}$ **do**
        $s_i \leftarrow \text{POLICY}(S = [s_{j<i}], Z = [z_{j<i}])$ // Action – `load`, $s_i \in \mathbb{R}^{N \times k_i}$
        $z_i \leftarrow K s_i$ // Observation – `compute`, $z_i \in \mathbb{R}^{N \times k_i}$
        $d_i \leftarrow (I - C_{i-1} K) s_i = s_i - C_{i-1} z_i$ // low-rank update, $d_i \in \mathbb{R}^{N \times k_i}$
        $H_i \leftarrow s_i^\top K d_i = z_i^\intercal d_i$ // Schur complement, $H_i \in \mathbb{R}^{k_i \times k_i}$
        $C_i \leftarrow C_{i-1} + d_i H_i^{-1} d_i^\top$ // Inverse estimate
        $\alpha_i \leftarrow C_i y = \alpha_{i-1} + d_i H_i^{-1} d_i^\intercal \bar{y}$ // Solution Estimate
    end for
    return $S = [s_j]_{j \leq i}, \alpha_i, C_i$
end procedure
procedure PREDICT(x, S, α, C)
    $k_{xS} \leftarrow k[x, S]$ // Covariance to Observations
    $\mu_x \leftarrow \mu_x + k_{xS} \alpha$ // Point estimate
    $v_{xx} \leftarrow k_{xx} - k_{xS} C k_{Sx}$ // Uncertainty
end procedure

Classification: approximate inference with non-Gaussian likelihoods

$$p(f) = \mathcal{GP}(f; m, k) \quad p(y \mid f_x) = \sigma(yf_x) = \begin{cases} \sigma(f) & \text{if } y = 1 \\ 1 - \sigma(f) & \text{if } y = -1 \end{cases} \qquad \text{using } \sigma(x) = 1 - \sigma(-x).$$

- Find maximum posterior probability for **latent** f at **training points**

$$\hat{\boldsymbol{f}} = \arg\max \log p(\boldsymbol{f}_X \mid y)$$

- Assign approximate Gaussian posterior at training points

$$q(\boldsymbol{f}_X) = \mathcal{N}(\boldsymbol{f}_X; \hat{\boldsymbol{f}}, -(\nabla\nabla^\intercal \log p(\boldsymbol{f}_X \mid y)|_{\boldsymbol{f}_X = \hat{\boldsymbol{f}}})^{-1}) =: \mathcal{N}(\boldsymbol{f}_X; \hat{\boldsymbol{f}}, \hat{\Sigma})$$

- approximate posterior **predictions** at f_x for **latent function**

$$q(f_x \mid y) = \int p(f_x \mid \boldsymbol{f}_X) q(\boldsymbol{f}_X)\, d\boldsymbol{f}_X = \int \mathcal{N}(f_x; m_x + k_{xX} K_{XX}^{-1}(\boldsymbol{f}_X - m_X), k_{xx} - k_{xX} K_{XX}^{-1} k_{Xx}) q(\boldsymbol{f}_X)\, d\boldsymbol{f}_X$$

$$= \mathcal{N}(f_x;\ m_x + k_{xX} K_{XX}^{-1}(\hat{\boldsymbol{f}} - m_X),\ k_{xx} - k_{xX} K_{XX}^{-1} k_{Xx} + k_{xX} K_{XX}^{-1} \hat{\Sigma} K_{XX}^{-1} k_{Xx})$$

- compute predictions for **label probabilities**:

$$\mathbb{E}_{p(f \mid y)}[\pi_x] \approx \mathbb{E}_q[\pi_x] = \int \sigma(f_x) q(f_x \mid y)\, df_x$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

1. Realise that the loss is a **negative log-posterior**

$$\mathcal{L}(\boldsymbol{\theta}) = \left(\frac{1}{N} \sum_{i=1}^{N} \underbrace{\ell(y_i; \overbrace{f(x_i, \boldsymbol{\theta})}^{\text{deep net}})}_{\text{empirical risk}} + \underbrace{r(\boldsymbol{\theta})}_{\text{regularizer}} \right) = -\sum_{i=1}^{N} \log p(\boldsymbol{y} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta} \mid \boldsymbol{y}) + \text{const.}$$

2. **Train the deep net *as usual*** to find $\boldsymbol{\theta}_* = \arg\max_{\boldsymbol{\theta} \in \mathbb{R}^D} p(\boldsymbol{\theta} \mid \boldsymbol{y})$
3. At $\boldsymbol{\theta}_*$, compute a **Laplace approximation** of the log-posterior, with $\Psi := \nabla\nabla^\intercal \log p(\boldsymbol{\theta}_* \mid \boldsymbol{y})$

$$\log p(\boldsymbol{\theta} \mid \boldsymbol{y}) + \text{const.} = \mathcal{L}(\boldsymbol{\theta}) \approx \mathcal{L}(\boldsymbol{\theta}_*) + {}^1\!/_2(\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\intercal \Psi (\boldsymbol{\theta} - \boldsymbol{\theta}_*) = \log \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_*, -\Psi^{-1})$$

4. **Linearize** $f(\boldsymbol{x}, \boldsymbol{\theta})$ around θ_*, with $[J(\boldsymbol{x})]_{ij} = \frac{\partial f_i(\boldsymbol{x}, \boldsymbol{\theta}_*)}{\partial \theta_j}$ as $f(\boldsymbol{x}, \boldsymbol{\theta}) \approx f(\boldsymbol{x}, \boldsymbol{\theta}_*) + J(\boldsymbol{x}, \boldsymbol{\theta}_*)(\boldsymbol{\theta} - \boldsymbol{\theta}_*)$

$$\text{thus } p(f(\bullet) \mid \mathcal{D}) = \int p(f \mid w)\, dp(w) \approx \mathcal{GP}(f(\bullet); f(\bullet, \boldsymbol{\theta}_*), -J(\bullet)\Psi^{-1}J(\circ)) \quad \text{with}$$

$\mathbb{E}(f(\bullet)) = f(\bullet, \boldsymbol{\theta}_*)$ — the **trained net** as the **mean function**

$\operatorname{cov}(f(\bullet), f(\circ)) = -J(\bullet)\Psi^{-1}J(\circ)^\intercal$ — the **Laplace tangent kernel** as the **covariance function**

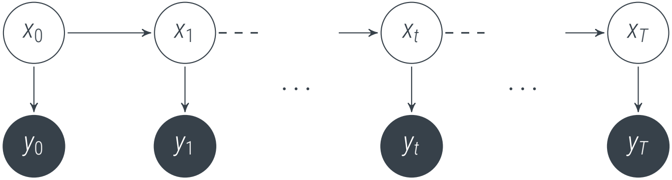
Markov Chains: $\mathcal{O}(T)$ inference in time series

Lectures 20/21

Assume:

$$p(x_t \mid X_{0:t-1}) = p(x_t \mid x_{t-1})$$

and $\quad p(y_t \mid X) = p(y_t \mid x_t)$



$$p(x_t|Y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|Y_{1:t-1})dx_{t-1}$$

$$p(x_t \mid Y_{0:t}) = \frac{p(y_t \mid x_t)p(x_t \mid Y_{0:t-1})}{\int p(y_t \mid x_t)p(x_t \mid Y_{0:t-1})\, dx_t}$$

$$p(x_t \mid Y) = p(x_t \mid Y_{0:t}) \int p(x_{t+1} \mid x_t) \frac{p(x_{t+1} \mid Y)}{p(x_{t+1} \mid Y_{1:t})}\, dx_{t+1}$$

The Toolbox

Framework:

$$\int p(x_1, x_2)\, dx_2 = p(x_1) \qquad p(x_1, x_2) = p(x_1 \mid x_2) p(x_2) \qquad p(x \mid y) = \frac{p(y \mid x) p(x)}{p(y)}$$

Modelling:

- Directed Graphical Models
- Exponential Families (also as likelihoods)
- Gaussian Distributions
- Kernels
- Markov Chains
- Deep Networks

Computation:

- autodiff
- MAP with Laplace approximations
- Linear algebra as a computational primitive
-
-

Bayesian Hierarchical Learning

beyond analytical inference

Bayesian Hierarchical Learning

the formalism for training hyperparameters / fit architectures & models

$$p(f \mid \mathbf{y}, x, \boldsymbol{\theta}) = \frac{p(\mathbf{y} \mid f, x, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{\int p(\mathbf{y} \mid f, x, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})\, df} = \frac{p(\mathbf{y} \mid f, x, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{p(\mathbf{y} \mid x, \boldsymbol{\theta})}$$

- **Model parameters** like θ are also known as **hyper-parameters**.
- This is largely a computational, practical distinction:

data	are observed	$\rightarrow$ **condition**
variables	are the things we care about	$\rightarrow$ **full probabilistic treatment**
parameters	are the things we have to deal with to get the model right	$\rightarrow$ **integrate out**
hyper-parameters	are the top-level, too expensive to properly infer	$\rightarrow$ **fit**

The **model evidence** in Bayes' Theorem is the (marginal) **likelihood** for the model. So we would like

$$p(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\mathbf{y} \mid \boldsymbol{\theta}') p(\boldsymbol{\theta}')\, d\boldsymbol{\theta}'}$$

The Gaussian Base Case

Bayesian Hierarchical Inference for GP regression, from Lecture 10

$$p(f \mid \theta) = \mathcal{GP}(f; m_\theta, k_\theta) \qquad \text{e.g.} \quad m_\theta(\bullet) = \phi(\bullet)^\intercal \boldsymbol{\theta}, \text{ or } k_\theta(\bullet, \circ) = \theta_1 \exp\left(-\frac{(\bullet - \circ)^2}{2\theta_2^2}\right).$$

- The **evidence** in Bayes' theorem is the **marginal likelihood for the model**

$$p(f \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{y} \mid f, \boldsymbol{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{\int p(\boldsymbol{y} \mid f, \boldsymbol{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})\, df} = \frac{p(\boldsymbol{y} \mid f, \boldsymbol{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})}$$

- For Gaussians *and Gaussian processes*, die evidence has **analytic form**:

$$\underbrace{\mathcal{N}(\boldsymbol{y}; \phi_X^{\boldsymbol{\theta}\intercal} w + \boldsymbol{b}, \Lambda)}_{p(\boldsymbol{y}|f,\boldsymbol{x},\boldsymbol{\theta})} \cdot \underbrace{\mathcal{N}(w, \mu, \Sigma)}_{p(f)} = \underbrace{\mathcal{N}(w; m_{\text{post}}^{\boldsymbol{\theta}}, V_{\text{post}}^{\boldsymbol{\theta}})}_{p(f|\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})} \cdot \underbrace{\mathcal{N}(\boldsymbol{y}; \phi_X^{\boldsymbol{\theta}\intercal} \mu + b, \phi_X^{\boldsymbol{\theta}\intercal} \Sigma \phi_X^{\boldsymbol{\theta}} + \Lambda)}_{p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})}$$

$$\mathcal{N}(\boldsymbol{y}; f^{\boldsymbol{\theta}}(X), \Lambda^{\boldsymbol{\theta}}) \cdot \mathcal{GP}(f, \mu^{\boldsymbol{\theta}}, k^{\boldsymbol{\theta}}) = \mathcal{GP}(f; m_{\text{post}}^{\boldsymbol{\theta}}, V_{\text{post}}^{\boldsymbol{\theta}}) \cdot \mathcal{N}(\boldsymbol{y}; \mu^{\boldsymbol{\theta}}(X), \Lambda^{\boldsymbol{\theta}} + k^{\boldsymbol{\theta}}(X, X))$$

ML / MAP in Practice

Finding the "best fit" θ in Gaussian models

[e.g. DJC MacKay, *The evidence framework applied to classification networks*, 1992]

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \int \mathcal{N}(\boldsymbol{y}; f(X), \Lambda_\theta) \cdot \mathcal{N}(f_X, \mu_X, k_{XX})\, df_X \\
&= \arg\max_{\boldsymbol{\theta}} \mathcal{N}(\boldsymbol{y}; \mu_X; k_{XX} + \Lambda_\theta) \int \mathcal{N}(f_X; \mu_{y,X}, v_{y,XX})\, df_X \\
&= \arg\max_{\boldsymbol{\theta}} \mathcal{N}(\boldsymbol{y}; \quad \mu_X^\theta, \quad k_{XX}^\theta + \Lambda^\theta) \\
&= \arg\max_{\boldsymbol{\theta}} \log \mathcal{N}(\boldsymbol{y}; \quad \mu_X^\theta, \quad k_{XX}^\theta + \Lambda^\theta) \\
&= \arg\min_{\boldsymbol{\theta}} -\log \mathcal{N}(\boldsymbol{y}; \quad \mu_X^\theta, \quad k_{XX}^\theta + \Lambda^\theta) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \left(\underbrace{(\boldsymbol{y} - \mu_X^\theta)^\intercal \left(k_{XX}^\theta + \Lambda^\theta\right)^{-1} (\boldsymbol{y} - \phi_X^{\boldsymbol{\theta}\,\intercal} \mu)}_{\text{square error}} + \underbrace{\log \left| k_{XX}^\theta + \Lambda^\theta \right|}_{\text{model complexity / \textbf{Occam factor}}} \right) + \frac{N}{2} \log 2\pi
\end{aligned}
$$

Numquam ponenda est pluralitas sine necessitate.
Plurality must never be posited without necessity.

William of Occam
(1285 (Occam, Surrey)–1349 (Munich, Bavaria))
stained-glass window by Lawrence Lee

The Evidence framework beyond GP regression

Laplace approximations yield approximate evidences.

- For general likelihoods, the evidence $p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \int p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{\theta}) p(\boldsymbol{f} \mid \boldsymbol{\theta})\, d\boldsymbol{f}$ will be intractable
- If we approximate it with a Gaussian by a Laplace approximation, we need to be careful with the constant term. Say we have found $\boldsymbol{f}^* = \arg\max \log p(\boldsymbol{f} \mid \boldsymbol{y})$ and $\Psi = -\nabla\nabla \log p(\boldsymbol{f} \mid \boldsymbol{y})$. Then

$$\begin{aligned}
p(\boldsymbol{y} \mid \boldsymbol{\theta}) &= \int p(\boldsymbol{y} \mid \boldsymbol{f}) p(\boldsymbol{f})\, d\boldsymbol{f} \\
&\approx \int \exp\left(\log p(\boldsymbol{y} \mid \boldsymbol{f}^*) + \log p(\boldsymbol{f}^*) - \frac{1}{2}(\boldsymbol{f} - \boldsymbol{f}^*)\Psi(\boldsymbol{f} - \boldsymbol{f}^*)\right) d\boldsymbol{f} \\
&= p(\boldsymbol{y} \mid \boldsymbol{f}^*) p(\boldsymbol{f}^*) \cdot (2\pi)^{D/2} |\Psi|^{1/2}
\end{aligned}$$

- If the prior happens to be Gaussian $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}; \boldsymbol{m}, K)$, we get

$$\log p(\boldsymbol{y} \mid \boldsymbol{\theta}) \approx \log p(\boldsymbol{y} \mid \boldsymbol{f}^*) - \frac{1}{2}(\boldsymbol{f}^* - \boldsymbol{m}) K^{-1} (\boldsymbol{f}^* - \boldsymbol{m}) - \frac{1}{2} \log |B|$$

with $|B| = |K| \cdot |K^{-1} + \underbrace{\nabla\nabla^\intercal \log p(\boldsymbol{y} \mid \boldsymbol{f}^*)}_{=:W}|^{-1} = |I + K \cdot W|$.

A special option: The EM algorithm

Maximizing model Evidence

<u>The general recipe</u> for hyperparameter inference:
Consider a model with **parameters** $\boldsymbol{\theta}$, **observed data** $\boldsymbol{y}$ and **latent variables** z

- Ideally, we would like to maximize the **marginal (log-) likelihood (evidence)**

$$\log p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \log \left(\int p(\boldsymbol{y}, z \mid \boldsymbol{\theta})\, dz \right) \qquad (\star)$$

- if we can not do this integral, we can try **Laplace** (as above). This is nearly always *possible* (if $\log p(\boldsymbol{y} \mid \boldsymbol{z})$ is twice differentiable), bu it is fundamentally an approximation.
- however, *in some cases*, we may be able to compute the Expectation **of the "complete data" log likelihood** (for a fixed value $\boldsymbol{\theta}_*$)

$$q(\boldsymbol{\theta}, \boldsymbol{\theta}_*) = \int p(z \mid \boldsymbol{y}, \boldsymbol{\theta}_*) \log p(\boldsymbol{y}, z \mid \boldsymbol{\theta})\, dz$$

and then Maximize $q(\boldsymbol{\theta}, \boldsymbol{\theta}_*)$ **with respect to** $\boldsymbol{\theta}$. This can be easier than $(\star)$ because the log "simplifies things" (e.g. turns products into sums, thus factors into components).

The EM Algorithm

Dempster, Laird, Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, RSS B **39** (1) 1977, 1–38

Definition: The **Expectation Maximizatiion (EM)** algorithm:
Consider a model with **parameters** $\boldsymbol{\theta}$, **observed data** $\boldsymbol{y}$ and **latent variables** $\boldsymbol{z}$.

```
while not converged, do:
```

E compute the **Expected complete data log-likelihood**

$$q(\boldsymbol{\theta}, \boldsymbol{\theta}_t) = \int p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\theta}_t) \log p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{\theta}) \, dz$$

M Set θ_{t+1} to **Maximize** $\boldsymbol{\theta}_{t+1} = \arg\max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}, \boldsymbol{\theta}_{t+1})$.

We will see on Thursday why this is a meaningful thing to do.

Example: EM for Gauss-Markov Models

completed in this week's exercise

Consider the Gauss-Markov Model with $\boldsymbol{z} := [\boldsymbol{x}_t]_{t=0,\dots,T}$ and $\boldsymbol{\theta} := (A, Q, H, R)$.

$$p(\boldsymbol{z}, \boldsymbol{y} \mid \boldsymbol{\theta}) = p(\boldsymbol{x}_0) \prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{\theta}) p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_0; m_0, P_0) \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{x}_t; A\boldsymbol{x}_{t-1}, Q) \mathcal{N}(\boldsymbol{y}_t, H\boldsymbol{x}_t, R)$$

Here it is actually possible to compute $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ (see last lectures), but that term can only be maximized numerically. Instead, notice that the complete-data log-likelihood neatly separates into local terms

$$\log p(\boldsymbol{z}, \boldsymbol{y} \mid \boldsymbol{\theta}) = \log \mathcal{N}(\boldsymbol{x}_0) + \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{x}_t; A\boldsymbol{x}_{t-1}, Q) + \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{y}_t; H\boldsymbol{x}_t, R)$$

And the expectation of a log-Gaussian is easy to compute, because

$$\int (W\boldsymbol{x} - \boldsymbol{\mu})^\intercal \Sigma^{-1} (W\boldsymbol{x} - \boldsymbol{\mu}) \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{m}, V)\, d\boldsymbol{x} = (\boldsymbol{\mu} - W\boldsymbol{m})^\intercal \Sigma^{-1} (\boldsymbol{\mu} - W\boldsymbol{m}) + \operatorname{tr}(W^\intercal \Sigma^{-1} W V)$$

Rest: Homework.

Summary:

- The *Evidence* in Bayes' theorem is the *marginal likelihood* of the model

$$p(\mathbf{y} \mid \theta) = \int p(\mathbf{y}, \mathbf{z} \mid \theta)\, d\mathbf{z}$$

- In principle it provides the "next level" for Bayesian inference on θ. But it is often intractable.
- Laplace approximations provide a *general, approximate* way to approximate the Evidence
- in *some* models, the **EM** algorithm offers a tractable iterative solution.

Please cite this course, as

```
@techreport{Tuebingen_ProbML23 ,
    title =
    {Probabilistic Machine Learning},
    author = {Hennig , Philipp},
    series = {Lecture Notes
        in Machine Learning},
    year = {2023},
    institution = {Tübingen AI Center}}
```