Advanced probabilistic methods

Lecture 4: ML-II, Laplace approximation, and Gaussian mixtures

Pekka Marttinen

Aalto University

February, 2021

Lecture 4 overview

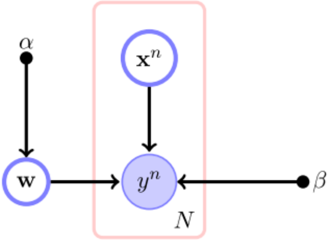
- Bayesian Linear Parameter Models (LPMs), continued
  - Lecture 3: Posterior computation given fixed hyperparameters
  - ML-II: Determining hyperparameters
  - Example using radial basis functions
- Logistic regression for classification
  - Laplace approximation
- Gaussian mixture models (GMMs)
- Suggested reading:
  - Barber, Ch. 18
  - Bishop, *Pattern Recognition and Machine Learning*, p. 110-113 (2.3.9): Mixtures of Gaussians

Recap: Bayesian linear regression

- **Data:** $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$
- **Model:**

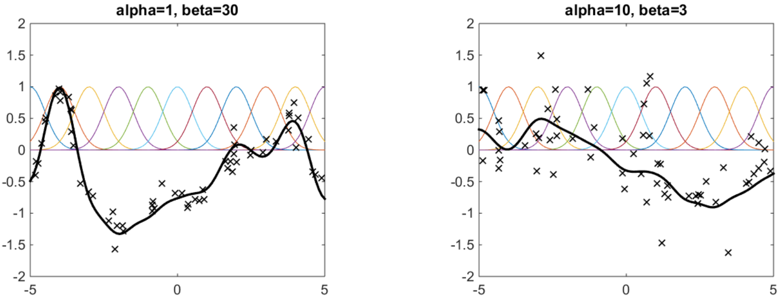
$$y_i = \mathbf{w}^T \mathbf{x}_i + \eta_i, \quad i = 1, \ldots, N$$
$$\eta_i \sim N(0, \beta^{-1}), \quad \mathbf{w} \sim N(\mathbf{0}, \alpha^{-1}\mathbf{I})$$

- **Parameters:** $\mathbf{w}$ called *weights* or *regression coefficients*
- **Hyperparameters:** $\Gamma = (\alpha, \beta)$



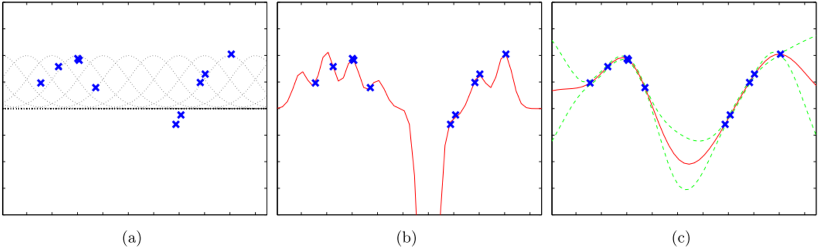
Non-linear transformation of the inputs

- Assume model $y_i = \mathbf{w}^T \phi(\mathbf{x}_i) + \eta_i$
- $\phi(\mathbf{x}_i)$ represent some transformation of $\mathbf{x}_i$ and are called *basis functions*
- Example
  - weights drawn from $N(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$; β is the noise precision.
  - $\mathbf{w} = (-0.7, 1.1, -0.8, -1.1, -0.8, -0.6, -0.6, 0.2, -0.2, 0.6, -0.9)$ for **radial basis functions** ordered from left to right (left panel)



Importance of learning hyperparameters

- (a): raw data and 15 radial basis functions $\phi_i(x) = \exp\left(-0.5(x - c_i)^2/\lambda^2\right)$ with $\lambda = 0.03^2$ and c_i spread evenly over the input space
- (b): predictions with $\beta = 100$ and $\alpha = 1$ (severe overfitting)
- (c): predictions with ML-II fitted hyperparameter values



Determining hyperparameters

- The hyperparameter posterior distribution is

$$p(\Gamma|\mathcal{D}) \propto p(\mathcal{D}|\Gamma)p(\Gamma)$$

- If $p(\Gamma) \approx \mathit{const}$ the optimal hyperparameter Γ^* is given by

$$\Gamma^* = \arg\max_{\Gamma} p(\mathcal{D}|\Gamma),$$

  where the **marginal likelihood**

$$p(\mathcal{D}|\Gamma) = \int p(\mathcal{D}|\Gamma, \mathbf{w})p(\mathbf{w}|\Gamma)d\mathbf{w}$$

- Selecting hyperparameters that maximize the marginal likelihood is called *ML-II* (a.k.a. *evidence maximization, empirical Bayes, maximum marginal likelihood*)

ML vs. ML-II

- In **maximum likelihood**, we select parameter values **w** that maximize the log-likelihood

$$\log p(y|\mathbf{w},\mathbf{x}) = \sum_{i=1}^{N} \log N(y_i|\mathbf{w}^T\phi(\mathbf{x}_i),\beta^{-1})$$

$$\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}}\{\log p(y|\mathbf{w},\mathbf{x})\} \quad \text{(does not depend on } \beta\text{)}$$

- In **ML-II**, we select hyperparameter values α and β that *maximize the (log-)marginal likelihood* (parameters **w** integrated out)

$$p(y|\Gamma,\mathbf{x}) = \int p(y|\Gamma,\mathbf{w},\mathbf{x})p(\mathbf{w}|\Gamma)d\mathbf{w}$$

$$\Gamma^* = \arg\max_{\Gamma}\{\log p(y|\Gamma,x)\}$$

Hyperparameter optimization in practice

- EM-algorithm
- using the gradient
- compute log-marginal likelihood over a grid of values and choose the best value
- use some standard optimization routine

Alternative to ML-II: validation data (1/2)*

- Set the hyperparameters Γ to the value that minimizes the prediction error in the validation data

$$\{\mathcal{X}_{val}, \mathcal{Y}_{val}\} = \left\{(\mathbf{x}_j^{val}, y_j^{val}), j = 1, \ldots, M\right\}.$$

- Mean squared error (MSE)

$$\mathrm{MSE}(\Gamma) = \frac{1}{M}\sum_{j=1}^{M}(y_j^{val} - \widetilde{y}_j^{val})^2,$$

where

$$\widetilde{y}_j^{val} = \mathbf{m}^T\phi(\mathbf{x}_j^{val}), \qquad \mathbf{m} = E(\mathbf{w}|\Gamma, \mathcal{X}_{train}, \mathcal{Y}_{train})$$

Alternative to ML-II: validation data (2/2)*

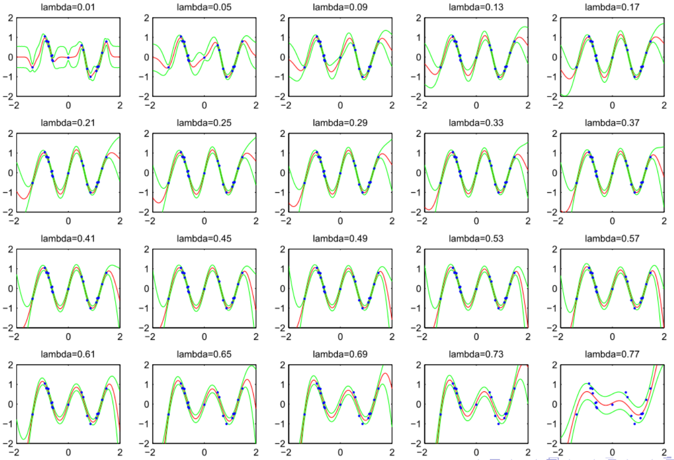
- Or by maximizing the validation data marginal likelihood

$$p(\mathcal{Y}_{val}|\Gamma, \mathcal{D}_{train}, \mathcal{X}_{val}) = \int_{\mathbf{w}} p(\mathcal{Y}_{val}|\mathbf{w},\mathcal{X}_{val}, \Gamma)p(\mathbf{w}|\Gamma, \mathcal{X}_{train}, \mathcal{Y}_{train})d\mathbf{w}$$

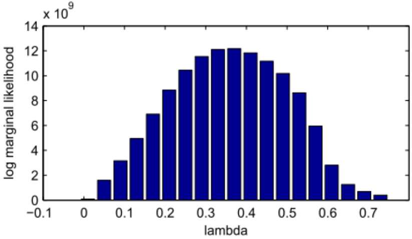
- Possible extension: *cross-validation*

Learning radial basis function width (1/2)

- A set of 10 evenly spaced radial basis functions is used $\phi_i(x) = \exp\left(-0.5(x - c_i)^2/\lambda^2\right)$
- $\Gamma = (\alpha, \beta)$ optimized for different width parameters λ



Learning radial basis function width (2/2)



- The log marginal likelihood

$$\log p(\mathcal{D}|\lambda, \alpha^*(\lambda), \beta^*(\lambda))$$

  having optimized α and β using ML-II. These values depend on λ.
- The best model corresponds to $\lambda = 0.37$.

Logistic regression for classification

- Binary classification problem: $\mathcal{D} = \{(\mathbf{x}_i, c_i), i = 1, \ldots, N\}$, where the output $c \in \{0, 1\}$.
- Let p denote the probability that $p(c = 1|\mathbf{x})$
- Logistic (linear) regression

$$\log \frac{p}{1-p} = \mathbf{w}^T \mathbf{x}$$

- Or, equivalently

$$p(c = 1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}),$$

where $\sigma(\cdot)$ is the so-called *logistic sigmoid*

$$\sigma(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$

Logistic regression for classification

- When used for classification, the decision boundary is defined by $p(c = 1|\mathbf{x}) = p(c = 0|\mathbf{x}) = 0.5$. This corresponds to a hyperplane

$$\mathbf{w}^T \mathbf{x} = 0.$$

Classification rule

$$\mathbf{w}^T \mathbf{x} > 0 \rightarrow c = 1$$
$$\mathbf{w}^T \mathbf{x} < 0 \rightarrow c = 0$$

- Note: $\mathbf{x}$ can include a constant term, $\mathbf{x} = (1, x_1, \ldots, x_D)$, such that the *intercept* is automatically included

$$\mathbf{w}^T \mathbf{x} = w_0 + w_1 x_1 + \ldots + w_D x_D$$

Logistic regression, interpretation of parameters*

$$\log\left(\frac{p}{1-p}\right) = w_0 + w_1 x$$
$$\Leftrightarrow \frac{p}{1-p} = \exp(w_0 + w_1 x)$$

- Interpretation: when x increases by one unit, the **odds** $\frac{p}{1-p}$ of belonging in class 1 increases by a factor equal to e^{w_1}.
- If x is binary itself, $x \in \{0, 1\}$, then e^{w_1} is the **odds ratio** between classes $x = 1$ and $x = 0$.
  - a common term in medical literature, e.g., X='smoking', C='cancer'.

Prior for logistic regression

- Gaussian prior

$$p(\mathbf{w}|\alpha) = N_D(\mathbf{w}|\mathbf{0},\alpha^{-1}\mathbf{I}) = \alpha^{\frac{D}{2}}(2\pi)^{-\frac{D}{2}}e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}}$$

  where α is the precision.

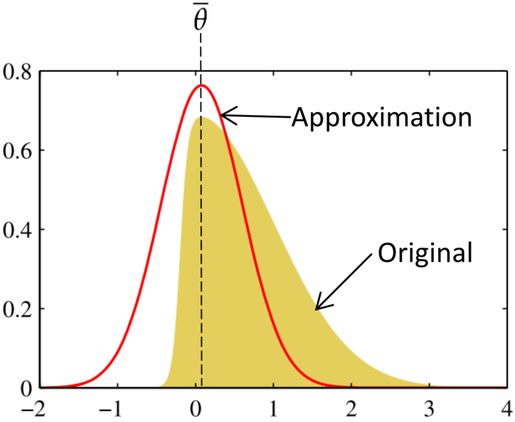
- Given $\mathcal{D} = \{(\mathbf{x}_i, c_i), i = 1, \ldots, N\}$ the posterior equals

$$p(\mathbf{w}|\alpha, \mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w}, \alpha)p(\mathbf{w}|\alpha)}{p(\mathcal{D}|\alpha)} = \frac{1}{p(\mathcal{D}|\alpha)}p(\mathbf{w}|\alpha)\prod_{i=1}^{N}p(c_i|\mathbf{x}_i, \mathbf{w})$$

  (not of standard form, Laplace approximation is feasible to compute).

Laplace approximation

- Gaussian approximation at the mode



modified from Bishop, Fig. 4.14

Laplace approximation of posterior distribution

- In general, for any posterior $p(\mathbf{w}|\alpha, \mathcal{D})$ it holds that

$$p(\mathbf{w}|\alpha, \mathcal{D}) \propto \exp(-E(\mathbf{w})), \quad E(\mathbf{w}) = -\log p(\mathbf{w}|\alpha, \mathcal{D}).$$

1. Approximate $E(\mathbf{w})$ by a 2nd order Taylor polynomial $\widetilde{E}(\mathbf{w})$ at the minimum $\overline{\mathbf{w}}$

$$\widetilde{E}(\mathbf{w}) = E(\overline{\mathbf{w}}) + \frac{1}{2}(\mathbf{w}-\overline{\mathbf{w}})^T H_{\overline{\mathbf{w}}}(\mathbf{w}-\overline{\mathbf{w}})$$

   (Note, this is quadratic in **w**.**)**

2. Obtain a Gaussian approximation $q(\mathbf{w}|\alpha, \mathcal{D})$:

$$p(\mathbf{w}|\alpha, \mathcal{D}) \approx q(\mathbf{w}|\alpha, \mathcal{D}) \propto \exp(-\widetilde{E}(\mathbf{w}))$$

- For logistic regression,

$$E(\mathbf{w}) = \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \log\sigma(\mathbf{w}^T\mathbf{h}_i), \quad \mathbf{h}_i \equiv (2c_i - 1)\mathbf{x}_i.$$

Laplace approximation in practice

- In practice:
  - Find the minimum $\overline{\mathbf{w}}$ of $E(\mathbf{w})$ analytically (root of the derivative) or by numerical optimization, e.g. Newton's method:

$$\mathbf{w}^{new} = \mathbf{w} - \mathbf{H}_w^{-1} \nabla E$$

  - When converged, compute the Hessian $H_{\overline{\mathbf{w}}}$ of $E(\mathbf{w})$ at $\overline{\mathbf{w}}$.
  - The posterior approximation is

$$q(\mathbf{w}|\alpha, \mathcal{D}) = N(\mathbf{w}|\mathbf{m}, \mathbf{S}), \quad \mathbf{m} = \overline{\mathbf{w}}, \quad \mathbf{S} = \mathbf{H}_{\overline{\mathbf{w}}}^{-1}.$$

- Reminder: if $f \equiv f(x_1, \dots, x_n)$

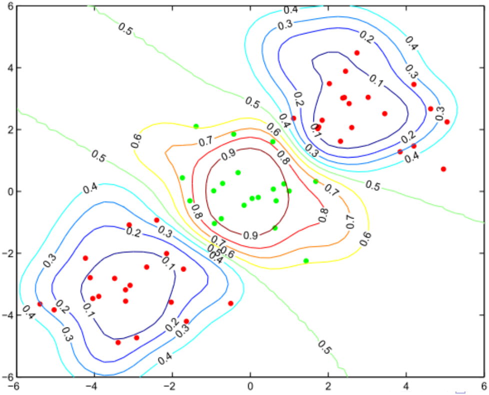
$$H_f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Laplace approximation for a univariate posterior distribution

- For some univariate parameter θ, you are given a prior $p(\theta)$ and the likelihood $p(\mathbf{x}|\theta)$.
- How do you calculate the Laplace approximation $q(\theta|\mathbf{x})$ of the posterior $p(\theta|\mathbf{x})$?

Laplace approximation for logistic regression

- Bayesian logistic regression with RBF functions $\phi_i(\mathbf{x}) = \exp(-\lambda(\mathbf{x} - \mathbf{m}_i)^2)$.
- $\mathbf{m}_i$ placed on a subset of training points, λ set to 2
- Hyperparameter α optimized as with the Bayesian linear regression by maximizing the approximated marginal likelihood ($\rightarrow \alpha = 0.45$).

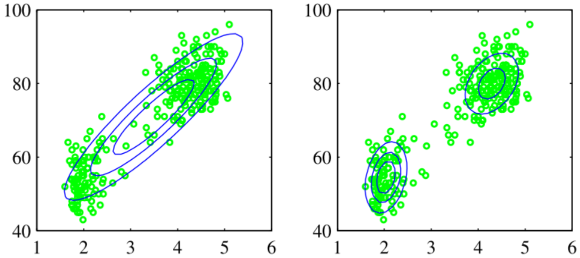


General comments on usage*

- Curse of dimensionality limits the use of RBFs to low-dimensional cases
  - Number of required basis functions grows exponentially w.r.t. the dimension D
  - Possible remedy: place basis functions on observations
  - Alternatives: kernel methods, Gaussian processes
- With sparse priors, standard linear models can be used with very large D
  - $y = \sum_{i=1}^{D} w_i x_i + \epsilon$

Gaussian mixture models (motivation)

- Standard Gaussian model (left) gives bad fit to data with clusters
- Combination of two Gaussians (right) is much better

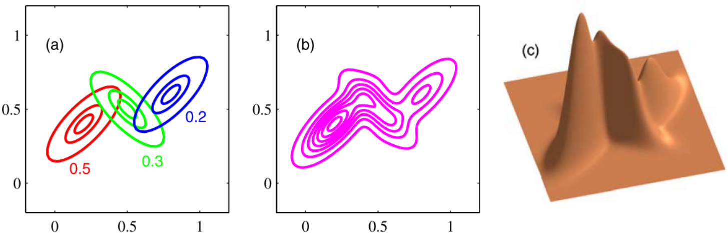


Gaussian mixture models

- Gaussian mixture model with K components has density

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k).$$

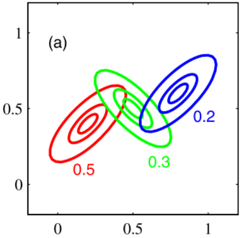
- $N(x|\mu_k, \Sigma_k)$ is a **component** with its own mean μ_k and covariance Σ_k.
- π_k are the **mixing coefficients**, which satisfy $\sum_k \pi_k = 1$, $0 \leq \pi_k \leq 1$.

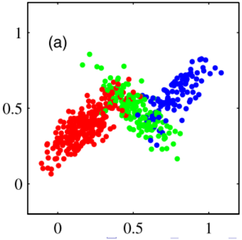


GMMs, latent variable representation (1/2)

- Equivalent formulation is obtained by defining **latent variables** $\mathbf{z}_n = (z_{n1}, \ldots, z_{nK})$ which tell the component for observation $\mathbf{x}_n$
- In detail $\mathbf{z}_n$ is a vector with exactly one element equal to 1 and other elements equal to 0. $z_{nk} = 1$ means that the observation $\mathbf{x}_n$ belongs to component k.

$$\mathbf{z}_n = (0, \ldots, 0, \underbrace{1}_{k^{th} \text{ elem.}}, 0, \ldots, 0)^T$$





GMMs, latent variable representation (2/2)

- Define

$$p(z_{nk}=1)=\pi_k \qquad \text{and} \qquad p(\mathbf{x}_n|z_{nk}=1)=N(\mathbf{x}_n|\mu_k,\Sigma_k),$$

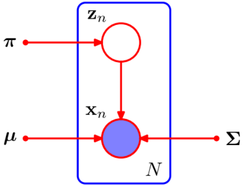
  or equivalently

$$p(\mathbf{z}_n)=\prod_{k=1}^{K}\pi_k^{z_{nk}} \quad \text{and} \quad p(\mathbf{x}_n|\mathbf{z}_n)=\prod_{k=1}^{K}N(\mathbf{x}_n|\mu_k,\Sigma_k)^{z_{nk}}$$

- Then

$$p(\mathbf{x}_n)=\sum_{z_n}p(\mathbf{z}_n)p(\mathbf{x}_n|\mathbf{z}_n)=\sum_k\pi_kN(\mathbf{x}_n|\mu_k,\Sigma_k)$$

→ $\mathbf{x}_n$ has marginally the Gaussian mixture model distribution.



GMM: responsibilities (1/2)

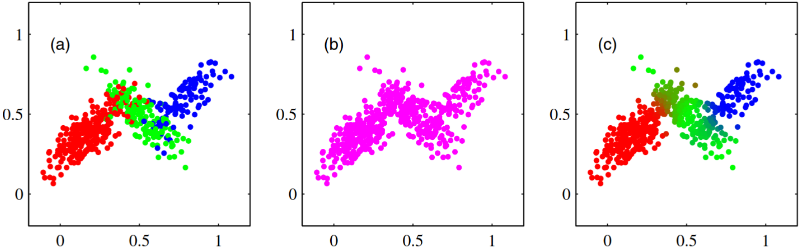
- Posterior probability $p(z_{nk} = 1|\mathbf{x}_n)$ that observation $\mathbf{x}_n$ was generated by component k

$$\gamma(z_{nk}) \equiv p(z_{nk} = 1|\mathbf{x}_n) = \frac{p(z_{nk} = 1)p(\mathbf{x}_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\mathbf{x}_n|z_{nj} = 1)}$$
$$= \frac{\pi_k N(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}_n|\mu_j, \Sigma_j)}$$

- $\gamma(z_{nk})$ can be viewed as the **responsibility** that component k takes for explaining the observation $\mathbf{x}_n$

GMM: responsibilities (2/2)

- (left) samples from a joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$, showing both cluster labels $\mathbf{z}$ and observations $\mathbf{x}$ (**complete** data)
- (center) samples from the marginal distribution $p(\mathbf{x})$ (**incomplete** data)
- (right) **responsibilities** of the data points, computed using *known* parameters $\pi = (\pi_1, \ldots, \pi_K)$, $\mu = \mu_1, \ldots, \mu_K$, $\Sigma = (\Sigma_1, \ldots, \Sigma_K)$.
- Problem: in practice π, μ, and Σ are usually *unknown*.



Important points

- In classification, no closed form solution is available for logistic regression and approximations, e.g., the Laplace approximation, are needed.
- Hyperparameters can be set by maximizing the marginal likelihood (either exact or approximate).
- Definition of the Gaussian mixture model.
- Representing the GMM using discrete latent variables, which specify the components (or clusters) of the observations.