

Introduction to Probabilistic Machine Learning

Bayesian Classification

Ralf Herbrich

Overview

1. Bayesian Classification Learning
2. Bayesian Classification Learning via Approximate Message Passing
3. Appendix: Bayesian Classification via Optimization
 - Laplace Approximation
 - Bayesian Linear Logit Regression

**Introduction to
Probabilistic Machine
Learning**

*Unit 9 – Bayesian
Classification*

1. **Bayesian Classification Learning**
2. Bayesian Classification Learning via Approximate Message Passing
3. Appendix: Bayesian Classification via Optimization
 - Laplace Approximation
 - Bayesian Linear Logit Regression

**Introduction to
Probabilistic Machine
Learning**

*Unit 9 – Bayesian
Classification*

Bayesian Classification Learning Setting

■ Given:

1. **Training Data:** $D \in (\mathcal{X} \times \{-1, +1\})^n$ of n (labelled) examples from input space \mathcal{X}

2. **Linear Basis Functions:** Basis function mapping $\phi: \mathcal{X} \rightarrow \mathbb{R}^M$ and linear function model

$$f(x; \mathbf{w}) := \mathbf{w}^T \phi(x)$$

3. **Likelihood of functions:** weight vector feature vector

$$p(D|f) = p(D|\mathbf{w}) = \prod_{i=1}^n \int_0^{+\infty} \mathcal{N}(t_i; y_i \mathbf{w}^T \phi(x_i), \beta^2) dt_i$$

4. **Prior belief over functions:**

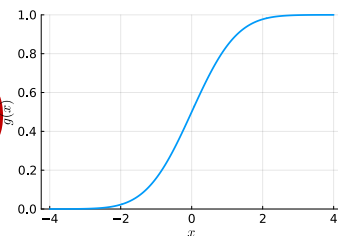
$$p(f) = p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

■ Bayesian Inference:

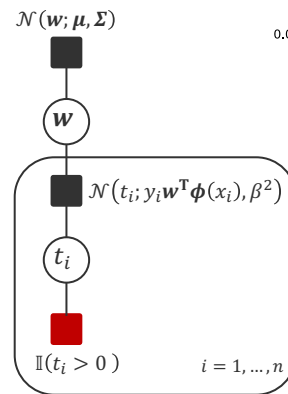
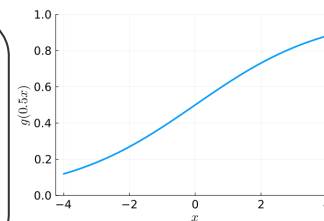
□ **Posterior belief over functions:**

$$p(f|D) = p(\mathbf{w}|D) = \frac{\prod_{i=1}^n \text{Ber}\left(1; g\left(y_i \mathbf{w}^T \phi(x_i)\right)\right) \cdot \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbb{R}^M} \prod_{i=1}^n \text{Ber}\left(1; g\left(y_i \tilde{\mathbf{w}}^T \phi(x_i)\right)\right) \cdot \mathcal{N}(\tilde{\mathbf{w}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\tilde{\mathbf{w}}}$$

$$g(t) = \int_0^{\infty} \mathcal{N}(z; t, 1) dz$$

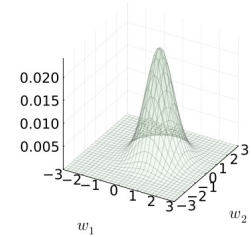
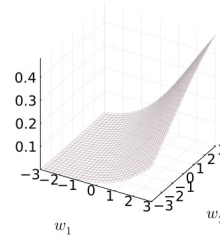
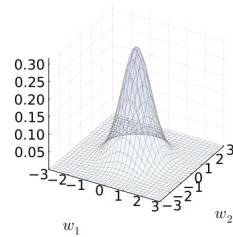
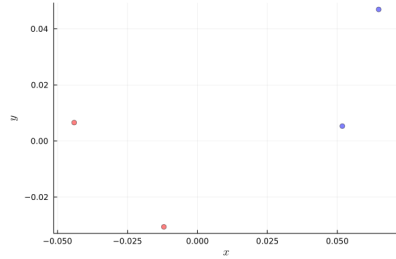


$$\lim_{s \rightarrow \infty} g(s \cdot t) = \mathbb{I}(t > 0)$$

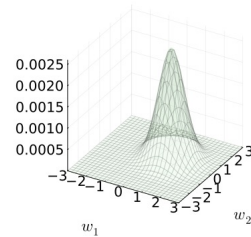
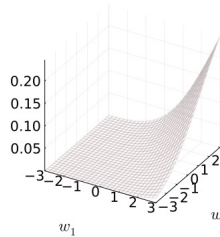
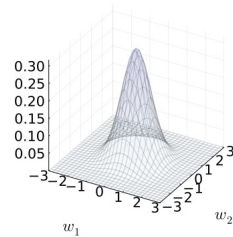
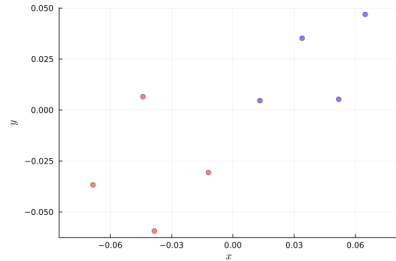


Bayesian Classification Learning in Pictures

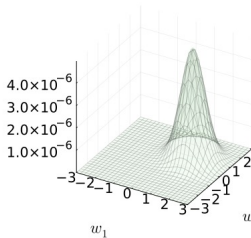
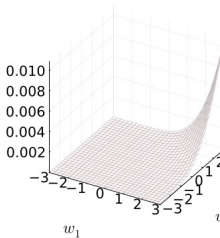
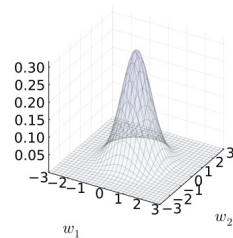
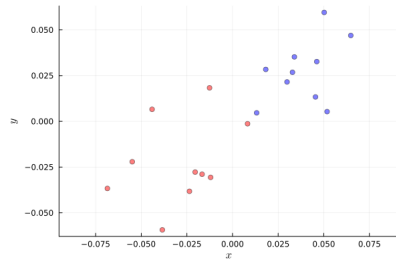
$n = 4$



$n = 6$



$n = 20$



0
Probabilistic Machine

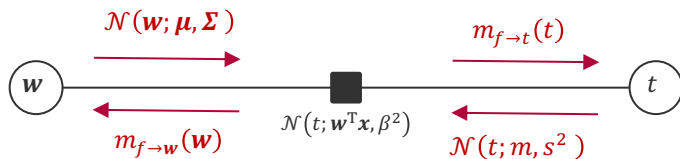
1. Bayesian Classification Learning
- 2. Bayesian Classification Learning via Approximate Message Passing**
3. Appendix: Bayesian Classification via Optimization
 - Laplace Approximation
 - Bayesian Linear Logit Regression

**Introduction to
Probabilistic Machine
Learning**

*Unit 9 – Bayesian
Classification*

Update Equations for Combined Factors

Gaussian Projection-Noise Factor

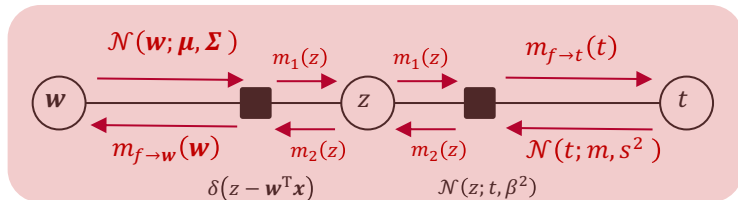


$$m_{f \rightarrow t}(t) = \mathcal{N}(t; \mu^T x, x^T \Sigma x + \beta^2)$$

$$m_{f \rightarrow w}(w) = \mathcal{G}\left(w; \frac{m}{s^2 + \beta^2} x, \frac{1}{s^2 + \beta^2} x x^T\right)$$

$$m_{f \rightarrow t}(t) = \int \mathcal{N}(t; w^T x, \beta^2) \cdot \mathcal{N}(w; \mu, \Sigma) dw = \int \left[\int \delta(z - w^T x) \cdot \mathcal{N}(w; \mu, \Sigma) dw \right] \mathcal{N}(t; z, \beta^2) dz$$

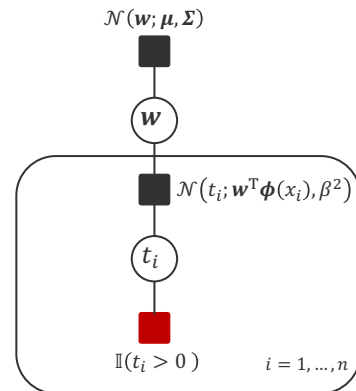
$$m_{f \rightarrow w}(w) = \int \mathcal{N}(t; w^T x, \beta^2) \cdot \mathcal{N}(t; m, s^2) dt = \int \left[\int \mathcal{N}(t; z, \beta^2) \cdot \mathcal{N}(t; m, s^2) dt \right] \delta(z - w^T x) dz$$



$$m_1(z) = \mathcal{N}(z; \mu^T x, x^T \Sigma x)$$

$$m_2(z) = \mathcal{N}(z; m, s^2 + \beta^2)$$

Helper factor tree to derive closed form message equations



Introduction to Probabilistic Machine Learning

Unit 9 – Bayesian Classification

Bayesian Linear Classification by Message Passing

- **Factor Tree:** Each training example is summarized in an M -dimensional message

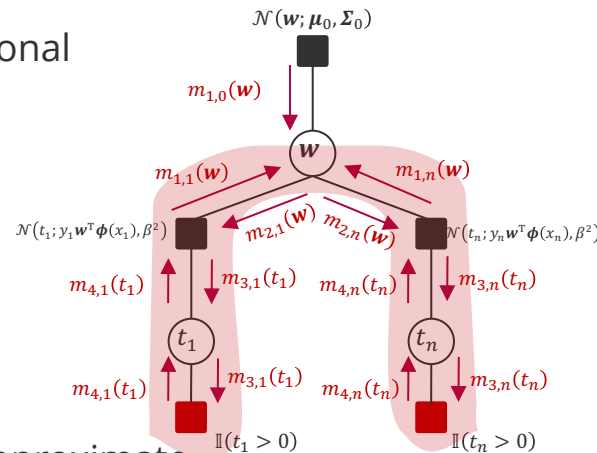
- Prior Message $m_{1,0}(\mathbf{w}) = \mathcal{G}(\mathbf{w}; \Sigma_0^{-1} \boldsymbol{\mu}_0, \Sigma_0^{-1}) = p(\mathbf{w})$
- Data Message $m_{1,i}(\mathbf{w}) = \mathcal{G}(\mathbf{w}; a_i y_i \boldsymbol{\phi}(x_i), b_i y_i^2 \boldsymbol{\phi}(x_i) \boldsymbol{\phi}^T(x_i)) = p(y_i | \mathbf{w})$

- **Posterior:** Multiplying prior and data messages we have

$$p(\mathbf{w} | D) = \mathcal{G}\left(\mathbf{w}; \Sigma_0^{-1} \boldsymbol{\mu}_0 + \sum_{i=1}^n a_i y_i \boldsymbol{\phi}(x_i), \Sigma_0^{-1} + \sum_{i=1}^n b_i \boldsymbol{\phi}(x_i) \boldsymbol{\phi}^T(x_i)\right)$$

- **Iterative Approximations:** Since each data message involves an approximate factor, we need to iterate message passing until convergence!

- We maintain the (approximate) posterior $p(\mathbf{w} | D) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the (a_i, b_i)
- 1. Compute $m_{2,i}(\mathbf{w})$ in scale-location parameters (using Sherman-Morrison formula)
- 2. Compute $m_{3,i}(t_i)$ in scale-location parameters by Gaussian projection-noise factor
- 3. Compute $m_{4,i}(t_i)$ in natural parameters by matching moments
- 4. Update $p(\mathbf{w} | D) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in scale-location parameters (using Sherman-Morrison)



Introduction to
Probabilistic Machine
Learning

Unit 9 – Bayesian
Classification

Bayesian Linear Classification: Update of i^{th} Data Message

- **Sherman-Morrison Formula:** For any invertible matrix \mathbf{A} it holds that

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

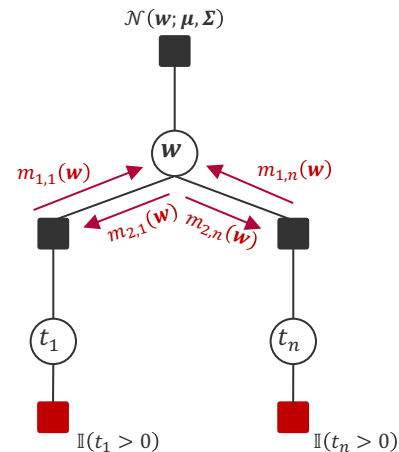
- **Rank-1 Updates** of Σ and μ with addition of $\Delta_i^a \cdot y_i \phi_i$ and $\Delta_i^b \cdot \phi_i \phi_i^T$

$$\Sigma_{\text{new}} = (\Sigma^{-1} + \Delta_i^b \phi_i \phi_i^T)^{-1} = \Sigma - \frac{\Delta_i^b}{1 + \Delta_i^b \cdot \phi_i^T \Sigma \phi_i} \cdot (\Sigma \phi_i)(\Sigma \phi_i)^T$$

No inverse needed but just matrix multiplications!

$$\mu_{\text{new}} = (\Sigma^{-1} + \Delta_i^b \phi_i \phi_i^T)^{-1} (\Sigma^{-1} \mu + \Delta_i^a y_i \phi_i) = \mu + \Sigma \phi_i \left(\frac{\Delta_i^a \cdot y_i - \Delta_i^b \cdot \phi_i^T \mu}{1 + \Delta_i^b \cdot \phi_i^T \Sigma \phi_i} \right)$$

- **Step 1:** Compute mean and covariance of $m_{2,i}(\mathbf{w})$ using $\Delta_i^a = -a_i$ and $\Delta_i^b = -b_i$
- **Step 4:** Update μ and Σ of $p(\mathbf{w}|D)$ using $\Delta_i^a = a_i^{\text{new}} - a_i$ and $\Delta_i^b = b_i^{\text{new}} - b_i$



Introduction to
Probabilistic Machine
Learning

Unit 9 – Bayesian
Classification

Bayesian Linear Classification: Update of i^{th} Data Message

- **Step 2:** $m_{3,i}(t_i) = \mathcal{N}(t_i; m_i, s_i^2)$ follows from the Gaussian projection-noise factor

$$m_i = \frac{y_i \phi_i^T \mu - a_i \phi_i^T \Sigma \phi_i}{1 - b_i \phi_i^T \Sigma \phi_i} \quad s_i^2 = \frac{\phi_i^T \Sigma \phi_i}{1 - b_i \cdot \phi_i^T \Sigma \phi_i} + \beta^2$$

- **Step 3:** $m_{4,i}(t_i) = \mathcal{G}(t_i; \tau_i, \rho_i)$ follows from the greater-than factor

Mean correction
of truncated Gaussian

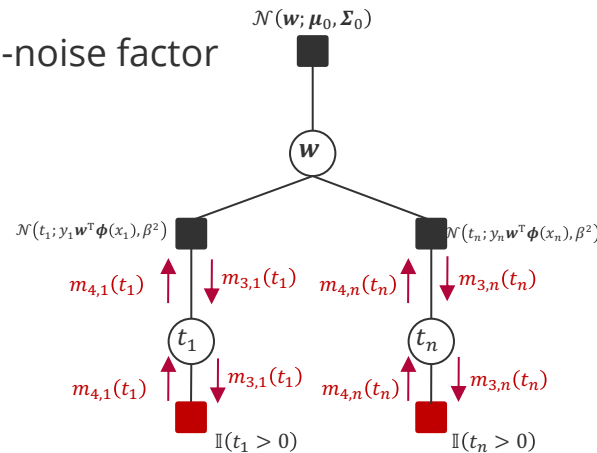
$$\tau_i = \frac{s_i \cdot v\left(\frac{m_i}{s_i}\right) + m_i \cdot w\left(\frac{m_i}{s_i}\right)}{s_i^2 \cdot \left(1 - w\left(\frac{m_i}{s_i}\right)\right)}$$

Variance correction
of truncated Gaussian

$$\rho_i = \frac{w\left(\frac{m_i}{s_i}\right)}{s_i^2 \cdot \left(1 - w\left(\frac{m_i}{s_i}\right)\right)}$$

- **Final Update**

$$a_i^{\text{new}} = \frac{\tau_i}{1 + \rho_i \cdot \beta^2} \quad b_i^{\text{new}} = \frac{\rho_i}{1 + \rho_i \cdot \beta^2}$$



Introduction to
Probabilistic Machine
Learning

Unit 9 – Bayesian
Classification

Bayesian Linear Classification: Putting It all Together

Given: n training examples (x_i, y_i) where $y_i \in \{-1, +1\}$ and feature map $\phi: \mathcal{X} \rightarrow \mathbb{R}^M$

1. **Initialize** $\phi_i := \phi(x_i)$, $\Sigma = \tau^2 I$, $\mu = \mathbf{0}$ and $a_i = b_i = 0$
2. **Repeat until** maximum change componentwise in \mathbf{a} and \mathbf{b} is small (e.g., $\leq 10^{-6}$)

For all $i = 1, \dots, n$

Compute $\mathbf{z} = \Sigma \phi_i$, $e = \phi_i^T \mathbf{z}$, $f = \phi_i^T \mu$, $c = 1 - b_i \cdot e$

Compute $m = \frac{y_i \cdot f - a_i \cdot e}{c}$ and $s^2 = \frac{e}{c} + \beta^2$

Compute $\tau = \frac{s \cdot V + m \cdot W}{s^2 \cdot (1 - W)}$ and $\rho = \frac{W}{s^2 \cdot (1 - W)}$ for $V = v\left(\frac{m}{s}\right)$ and $W = w\left(\frac{m}{s}\right)$

Compute $a_i^{\text{new}} = \frac{\tau}{1 + \rho \cdot \beta^2}$ and $b_i^{\text{new}} = \frac{\rho}{1 + \rho \cdot \beta^2}$

Compute $d = 1 + (b_i^{\text{new}} - b_i) \cdot e$

Update $\Sigma \leftarrow \Sigma - \frac{b_i^{\text{new}} - b_i}{d} \cdot \mathbf{z} \mathbf{z}^T$ and $\mu \leftarrow \mu + \frac{y_i \cdot (a_i^{\text{new}} - a_i) - f \cdot (b_i^{\text{new}} - b_i)}{d} \cdot \mathbf{z}$

Update $a_i \leftarrow a_i^{\text{new}}$ and $b_i \leftarrow b_i^{\text{new}}$

End For

Majority of algorithm is additions, multiplication, and division of scalars!

Algorithm requires **no** matrix inverses – just matrix products!

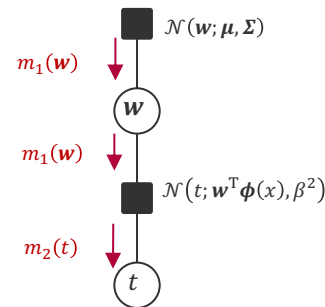
Introduction to Probabilistic Machine Learning

Unit 9 – Bayesian Classification

Predictions

■ Prediction Tree: Simple factor chain given posterior $p(\mathbf{w}|x, D) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$

- Posterior Message $m_1(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\mathbf{w}|x, D)$
- Projection-Noise Message $m_2(t) = \mathcal{N}(t; \boldsymbol{\mu}^T \boldsymbol{\phi}(x), \boldsymbol{\phi}^T(x) \boldsymbol{\Sigma} \boldsymbol{\phi}(x) + \beta^2) = p(t|x, D)$
- Outcome Probability $\int_0^\infty m_2(t) dt = p(y = +1|x, D)$



■ Bayesian Linear Probit Regression in Matrix Notation

$$p(y = +1|x, D) = \int_0^{+\infty} \mathcal{N}(y; \boldsymbol{\mu}^T \boldsymbol{\phi}(x), \beta^2 + \boldsymbol{\phi}^T(x) \boldsymbol{\Sigma} \boldsymbol{\phi}(x)) dt$$

data uncertainty

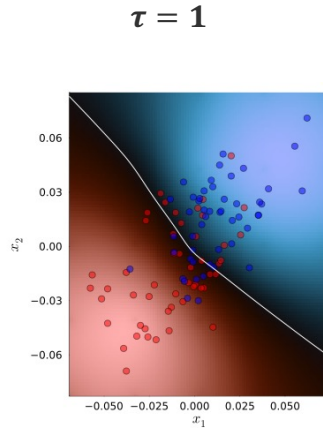
model uncertainty

Introduction to
Probabilistic Machine
Learning

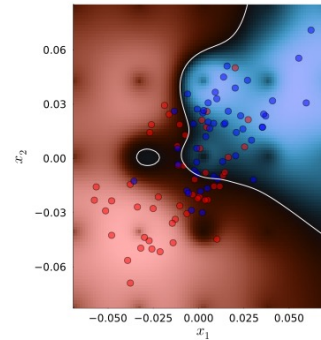
Unit 9 – Bayesian
Classification

Bayesian Linear Classification in Pictures

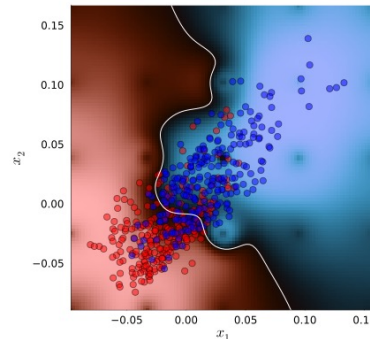
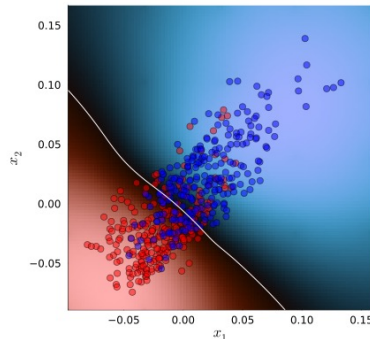
$n = 100$



$\tau = 10$



$n = 500$



225 features of
value of 225 Gaussian
density in 2D on a 5x5 grid
at 9 resolutions

**Introduction to
Probabilistic Machine
Learning**

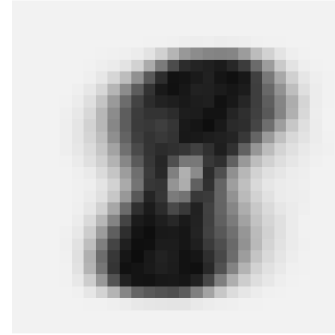
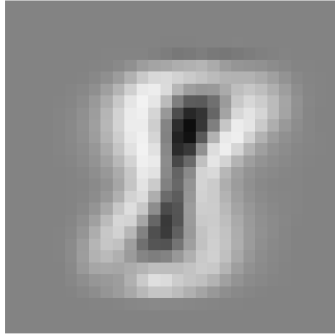
*Unit 9 – Bayesian
Classification*

Bayesian Linear Classification for MNIST

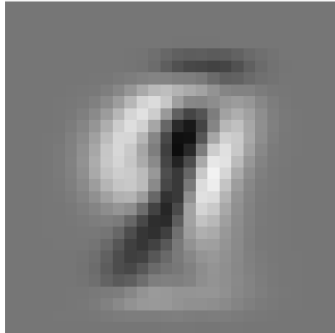
μ

$\text{diag}(\Sigma)$

“1” vs. “8”



“1” vs. “9”



**Introduction to
Probabilistic Machine
Learning**

*Unit 9 – Bayesian
Classification*

1. Bayesian Classification

- The likelihood model is no longer conjugate to the prior over the weight vector
- Approximate inference methods can be used (similar to Bayesian ranking) as the approximation is on a 1D-projection of each feature vector at each training example
- The message passing algorithm is highly efficient and requires no matrix inversion!

2. Laplace Approximation

- Matches the mode of the true likelihood/posterior to that of a Gaussian
- Usually easy to compute when we have a posterior that can be differentiated
- Allows to derive Bayesian linear classification for the logistic likelihood function (though this requires matrix inversion)

**Introduction to
Probabilistic Machine
Learning**

*Unit 9 – Bayesian
Classification*

See you next week!