Introduction to Probabilistic Machine Learning

Ralf Herbrich

Bayesian Classification

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

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

Overview

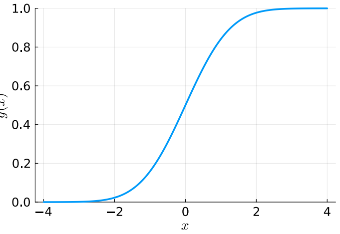
1. **Bayesian Classification Learning**
2. Bayesian Classification Learning via Approximate Message Passing
3. Appendix: Bayesian Classification via Optimization
   - Laplace Approximation
   - Bayesian Linear Logit Regression

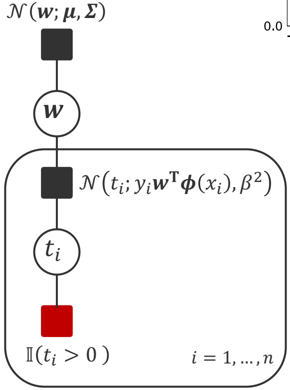
Bayesian Classification Learning Setting

- **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 $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^M$ and linear function model
     $$f(x;\boldsymbol{w}) := \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x)$$
     weight vector feature vector
  3. **Likelihood of functions**:
     $$p(D|f) = p(D|\boldsymbol{w}) = \prod_{i=1}^{n} \int_0^{+\infty} \mathcal{N}\left(t_i; y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x_i), \beta^2\right) dt_i$$
     $\mathrm{Ber}\left(1; g\left(y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x_i)\right)\right)$
  4. **Prior belief over functions**:
     $$p(f) = p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

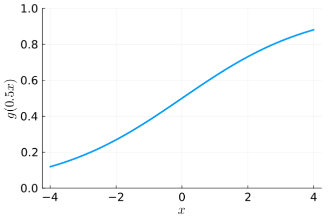
- **Bayesian Inference**:
  - **Posterior belief over functions**:
    $$p(f|D) = p(\boldsymbol{w}|D) = \frac{\prod_{i=1}^{n} \mathrm{Ber}\left(1; g\left(y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x_i)\right)\right) \cdot \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbb{R}^M} \prod_{i=1}^{n} \mathrm{Ber}\left(1; g\left(y_i \tilde{\boldsymbol{w}}^{\mathrm{T}} \boldsymbol{\phi}(x_i)\right)\right) \cdot \mathcal{N}(\tilde{\boldsymbol{w}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\tilde{\boldsymbol{w}}}$$

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





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

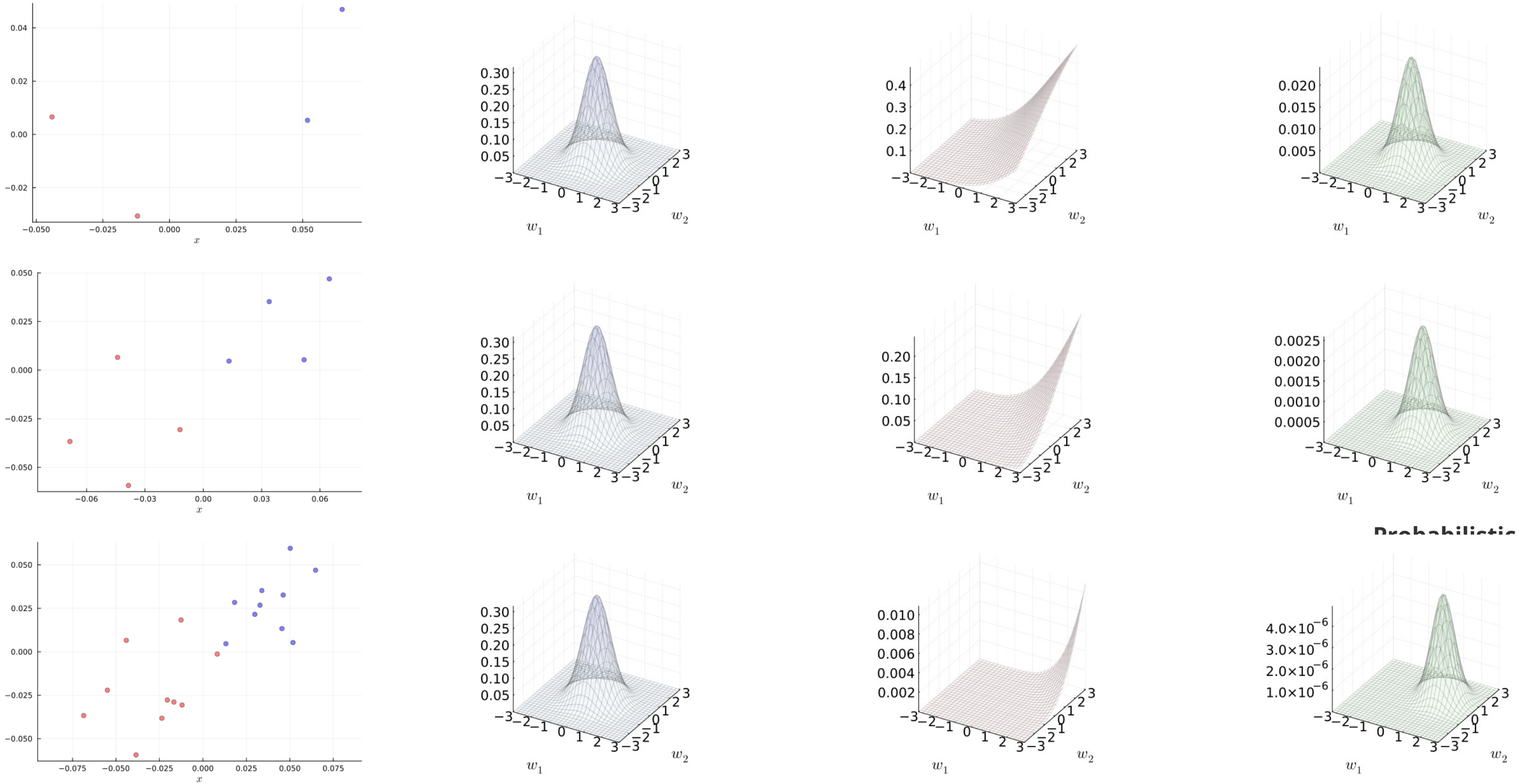


Bayesian Classification Learning in Pictures

$n = 4$

$n = 6$

$n = 20$

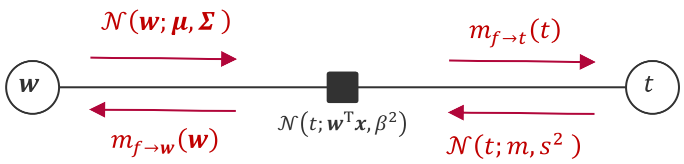


Overview

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

Update Equations for Combined Factors

Gaussian Projection-Noise Factor

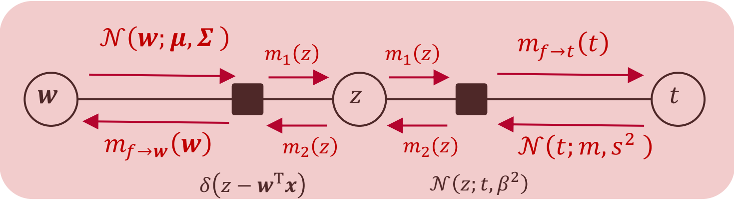


$$m_{f\to t}(t)=\mathcal{N}\left(t;\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x},\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{x}+\beta^2\right)$$

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

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

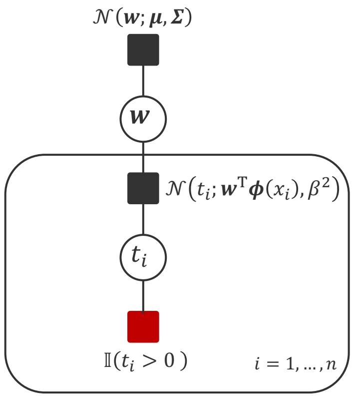
$$m_{f\to\boldsymbol{w}}(\boldsymbol{w})=\int\mathcal{N}\left(t;\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x},\beta^2\right)\cdot\mathcal{N}(t;m,s^2)\,dt=\int\left[\int\mathcal{N}\left(t;z,\beta^2\right)\cdot\mathcal{N}(t;m,s^2)\,dt\right]\cdot\delta\left(z-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)dz$$



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

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

Helper factor tree to derive closed form message equations



Bayesian Linear Classification by Message Passing

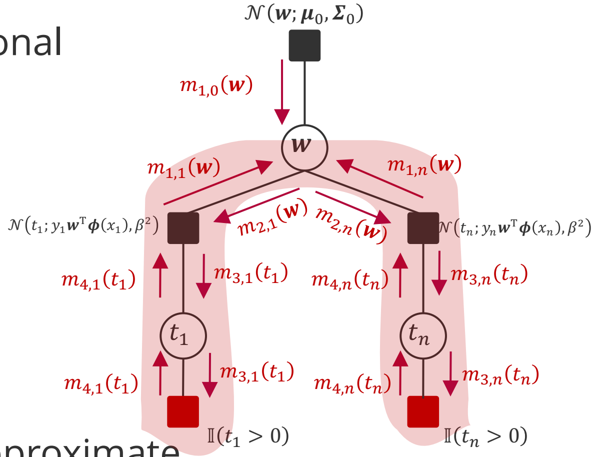
- **Factor Tree**: Each training example is summarized in an M-dimensional message
  - Prior Message $m_{1,0}(\boldsymbol{w}) = \mathcal{G}(\boldsymbol{w}; \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0^{-1}) = p(\boldsymbol{w})$

    $y_i^2 = 1$
  - Data Message $m_{1,i}(\boldsymbol{w}) = \mathcal{G}\left(\boldsymbol{w}; a_i y_i \boldsymbol{\phi}(x_i), b_i y_i^2 \boldsymbol{\phi}(x_i)\boldsymbol{\phi}^{\mathrm{T}}(x_i)\right) = p(y_i|\boldsymbol{w})$
- **Posterior**: Multiplying prior and data messages we have

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

- **Iterative Approximations**: Since each data message involves an approximate factor, we need to iterate message passing until convergence!
  - We maintain the (approximate) posterior $p(\boldsymbol{w}|D) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the (a_i, b_i)

1. Compute $m_{2,i}(\boldsymbol{w})$ 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 $p(\boldsymbol{w}|D) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 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 $\boldsymbol{A}$ it holds that

$$\left(\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right)^{-1}=\boldsymbol{A}^{-1}-\frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}}{1+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{u}}$$

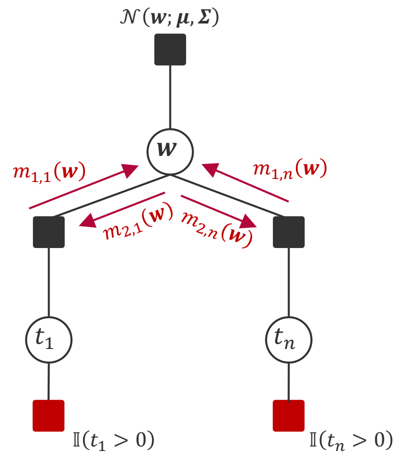
- **Rank-1 Updates** of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ with addition of $\Delta_i^a \cdot y_i \boldsymbol{\phi}_i$ and $\Delta_i^b \cdot \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}}$

$$\boldsymbol{\Sigma}_{\text{new}} = \left(\boldsymbol{\Sigma}^{-1}+\Delta_i^b \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}}\right)^{-1} = \boldsymbol{\Sigma}-\frac{\Delta_i^b}{1+\Delta_i^b \cdot \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}_i} \cdot (\boldsymbol{\Sigma}\boldsymbol{\phi}_i)(\boldsymbol{\Sigma}\boldsymbol{\phi}_i)^{\mathrm{T}}$$

No inverse needed but just matrix multiplications!

$$\boldsymbol{\mu}_{\text{new}} = \left(\boldsymbol{\Sigma}^{-1}+\Delta_i^b \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}}\right)^{-1}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\Delta_i^a y_i \boldsymbol{\phi}_i\right) = \boldsymbol{\mu}+\boldsymbol{\Sigma}\boldsymbol{\phi}_i\left(\frac{\Delta_i^a \cdot y_i-\Delta_i^b \cdot \boldsymbol{\phi}_i^{\mathrm{T}}\boldsymbol{\mu}}{1+\Delta_i^b \cdot \boldsymbol{\phi}_i^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\phi}_i}\right)$$

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



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

$$m_i = \frac{y_i \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\mu} - a_i \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}_i}{1 - b_i \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}_i} \qquad s_i^2 = \frac{\boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}_i}{1 - b_i \cdot \boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}_i} + \beta^2$$

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

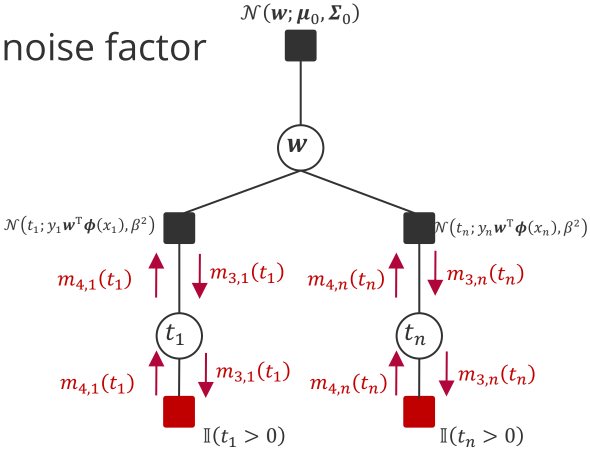
Mean correction of truncated Gaussian

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

Variance correction of truncated Gaussian

- **Final Update**

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



Bayesian Linear Classification: Putting It all Together

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

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

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

   Compute $\boldsymbol{z} = \boldsymbol{\Sigma}\boldsymbol{\phi}_i$, $e = \boldsymbol{\phi}_i^{\mathrm{T}}\boldsymbol{z}$, $f = \boldsymbol{\phi}_i^{\mathrm{T}}\boldsymbol{\mu}$, $c = 1 - b_i \cdot e$

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

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

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

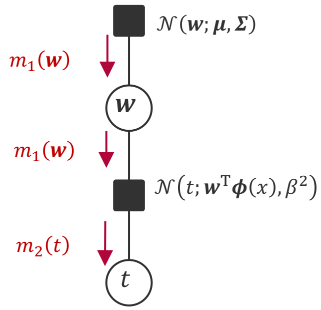
   End For

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

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

Predictions

- **Predicition Tree**: Simple factor chain given posterior $p(\boldsymbol{w}|x,D) = \mathcal{N}(\boldsymbol{w};\boldsymbol{\mu},\boldsymbol{\Sigma})$
  - Posterior Message $m_1(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w};\boldsymbol{\mu},\boldsymbol{\Sigma}) = p(\boldsymbol{w}|x,D)$
  - Projection-Noise Message $m_2(t) = \mathcal{N}\left(t;\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\phi}(x),\boldsymbol{\phi}^{\mathrm{T}}(x)\boldsymbol{\Sigma}\boldsymbol{\phi}(x)+\beta^2\right) = p(t|x,D)$
  - Outcome Probability $\int_0^\infty m_2(t)\,dt = p(y=+1|x,D)$



- **Bayesian Linear Probit Regression in Matrix Notation**

$$p(y=+1|x,D) = \int_{0}^{+\infty} \mathcal{N}\left(y;\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\phi}(x),\beta^2+\boldsymbol{\phi}^{\mathrm{T}}(x)\boldsymbol{\Sigma}\boldsymbol{\phi}(x)\right)dt$$

data uncertainty model uncertainty

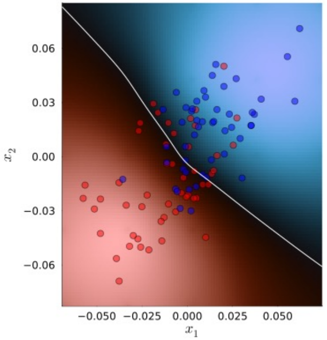
Introduction to Probabilistic Machine Learning

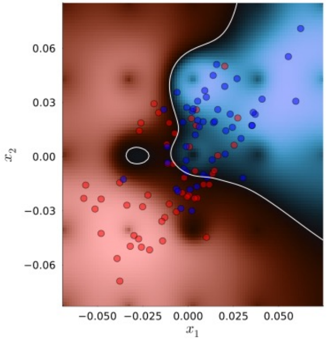
Unit 9 – Bayesian Classification

Bayesian Linear Classification in Pictures

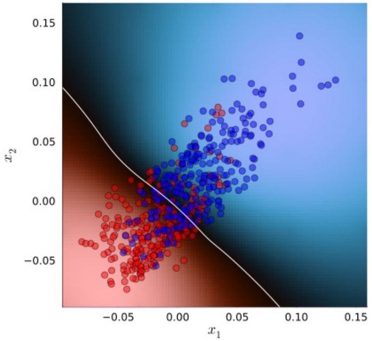
$\tau = 1$ $\tau = 10$

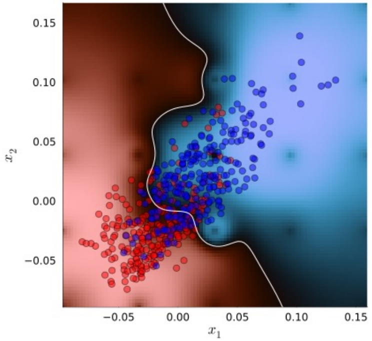
$n = 100$





$n = 500$



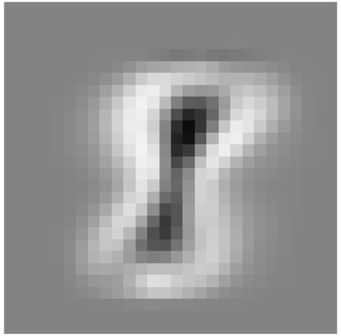


225 features of
value of 225 Gaussian
density in 2D on a 5×5 grid
at 9 resolutions

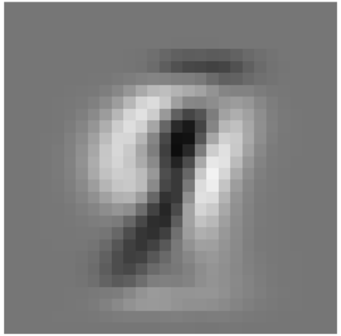
Bayesian Linear Classification for MNIST

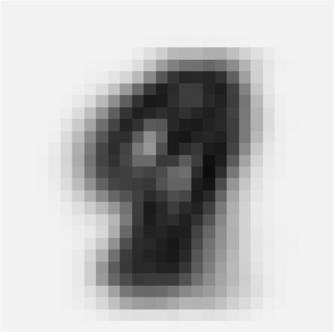
$\boldsymbol{\mu}$ | $\mathbf{diag}(\boldsymbol{\Sigma})$

"1" vs. "8"



"1" vs. "9"





Summary

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)

See you next week!