Machine Learning

Kernel Methods

[1]. Gaussian Processes (GP's)

[2]. Relevant Vector Machines (RVM's)

Lesson 7

Linear regression example

- Assume **M basis functions**:

$$\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_M(x))$$

- **Linear model**: $y(x) = w^T \phi(x) = \sum_{k=1}^{M} w_k \phi_k(x)$

  where ***w*** is a M-dimensional weight vector

- An **isotropic Gaussian prior** over ***w***

$$p(w \mid a) = N(w \mid 0, a^{-1} I)$$

- For any w obtain a particular function y(x).
- Intuitively, we take a **probability distribution over function y(x)**

$$y = \left(y(x_1), y(x_2), \ldots, y(x_N)\right)$$

- **Linear model**: $y = \Phi w$

  where $\Phi = [\ \Phi_{ik} = \phi_k(x_i)\ ]$ the **design matrix**.

➢ **Remark:** y is a linear combination of Gaussians

- **y** is also Gaussian $p(y) = N(y \mid 0, K)$

$$E[y] = E[\Phi w] = \Phi E[w] = 0$$

$$COV[y] = E\left[(y - E[y])(y - E[y])^T\right] = E[yy^T] =$$

$$= E\left[(\Phi w)(\Phi w)^T\right] = \Phi E[ww^T]\Phi^T = \frac{1}{a}\Phi\Phi^T = K$$

where K is a **kernel matrix (Gram matrix) :**

$$K_{ij} = k(x_i, x_j) = \frac{1}{a}\phi(x_i)^T \phi(x_j)$$

- ✓ This is a particular example **of Gaussian Process (GP) model**

- ✓ A **GP** provides a **probability distribution over functions y(x)**, such that **the set of values y(x)** evaluated at $\{x_1, \ldots, x_N\}$ jointly have a **Gaussian** distribution

- ✓ A **GP** provides a **stochastic process y(x)** that gives the joint distribution for any set of values $(y(x_1),\ldots,y(x_N))$

- ✓ In 2-dimensional input case GP is known as Gaussian Random Field

- In Gaussian Process the joint of y's is specified completely by **second-order statistics** (mean and variance)
- Usually, we don't have any knowledge of mean and thus we set it to **zero** (0) . (equivalent to zero-mean prior of w : $p(w|a)=N(w|0,a^{-1}I)$)
- Thus, GP is then defined by giving the **covariance** of y(x), **given by the kernel function**

$$E\left[y(x_i)y(x_j)\right]=k(x_i,x_j)$$

Gaussian Processes for Regression

- Training set: $D = \{(x_1, t_1), \ldots, (x_N, t_N)\}$

$$y(x) \sim GP$$

- **Generative model** for targets: $t_i = y_i + \varepsilon_i$

- where: $y_i = y(x_i)$

$\varepsilon_i \sim N(0, \beta^{-1})$ **random noise variable**

(β : **precision of noise**)

- Therefore: $p(t_i \mid y_i) = N(t_i \mid y_i, \beta^{-1})$

- Join distribution of N target values $\boldsymbol{T=(t_1,t_2,\ldots,t_N)}$

$$p(T \mid Y) = N(T \mid Y, \beta^{-1} I_N)$$

I_N : NxN unit matrix

where $\boldsymbol{Y=(y_1,y_2,\ldots,y_N)}$

- Assuming that *Y* is a GPs, *p(Y)* is a zero-mean Gaussian with kernel (covariance) matrix *K*:

$$p(Y) = N(Y \mid 0, K)$$

➢ Kernel function $K_{ij} = k(x_i, x_j)$ is chosen to express the correlation level of the corresponding values $y(x_i)$, $y(x_j)$

Marginal distribution of target values $T=(t_1,t_2,\ldots,t_N)$

$$p(T)=\int p(T\mid Y)p(Y)dY$$

Using the properties of Gaussians we take that

Marginal and Conditional Gaussians

Given a marginal Gaussian distribution for $\mathbf{x}$ and a conditional Gaussian distribution for $\mathbf{y}$ given $\mathbf{x}$ in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (2.113)$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}) \quad (2.114)$$

the marginal distribution of $\mathbf{y}$ and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b}, \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \quad (2.115)$$
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad (2.116)$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}. \quad (2.117)$$

Marginal distribution of target values $T=(t_1,t_2,\ldots,t_N)$

$$p(T)=\int p(T\mid Y)p(Y)dY$$

Using the properties of Gaussians we take that

$$p(Y)=N(Y\mid 0,K)$$

$$p(T\mid Y)=N(Y\mid Y,\beta^{-1}I_N)$$

$$p(T)=N(T\mid 0,C)$$

where

$$[C]_{ij}=C(x_i,x_j)=k(x_i,x_j)+\beta^{-1}\delta_{ij}$$

Marginal and Conditional Gaussians

Given a marginal Gaussian distribution for x and a conditional Gaussian distribution for y given x in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (2.113)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}) \quad (2.114)$$

the marginal distribution of y and the conditional distribution of x given y are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b}, \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \quad (2.115)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad (2.116)$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}. \quad (2.117)$$

Regression Problem: Make prediction

- Predict the **target t_{N+1}** for a **new input x_{N+1}** given the target values T

$$p(t_{N+1} \mid T) \quad ?$$

- **Joint distribution** of $T_{N+1} = (t_1, \ldots, t_N, t_{N+1})$

$$p(T_{N+1}) = N(T_{N+1} \mid 0, C_{N+1})$$

C_{N+1} covariance matrix is of size (N+1) x (N+1) with elements from kernel function on *N+1* inputs

- **Target:** find the conditional distribution $p(t_{N+1} \mid T)$

- **Solution:**

  Partition the covariance matrix $C_{N+1} = \begin{pmatrix} C & k \\ k^T & c \end{pmatrix}$

  where

  – **covariance matrix C** is of size N x N

  – **k is a vector** with elements $k(x_i, x_{N+1}),\ i=1, \ldots, N$

  – $c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$

Exploit the Gaussian properties

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}. \tag{2.65}$$

We also define corresponding partitions of the mean vector $\boldsymbol{\mu}$ given by

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \tag{2.66}$$

and of the covariance matrix $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}. \tag{2.67}$$

From these we obtain the following expressions for the mean and covariance of the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \tag{2.81}$$

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}. \tag{2.82}$$

Exploit the Gaussian properties

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}.$$

$$T_{N+1} = \begin{pmatrix} t_{N+1} \\ T \end{pmatrix}$$

We also define corresponding partitions of the mean vecto

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}$$

$$\mu_{N+1} = \begin{pmatrix} 0 \\ \mu_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

and of the covariance matrix $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}.$$

$$C_{N+1} = \begin{pmatrix} c & k^T \\ k & C \end{pmatrix}$$

From these we obtain the following expressions for the mean and covariance of the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$

$$\begin{aligned} \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}. \end{aligned}$$

$$m(x_{N+1}) = k^T C^{-1} T$$

$$\sigma^2(x_{N+1}) = c - k^T C^{-1} k$$

Gaussian Process regression model

$$p(t_{N+1} \mid T) = N(t_{N+1} \mid m(x_{N+1}), \sigma^2_{N+1}(x_{N+1}))$$

$$m(x_{N+1}) = k^T C^{-1} T$$

$$\sigma^2(x_{N+1}) = c - k^T C^{-1} k$$

$$c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$$

$$k = (k(x_1, x_{N+1}), \ldots, k(x_N, x_{N+1}))$$

[1]. Useful remark

- **A necessary constraint is that the covariance matrix**

$$[C]_{ij} = C(x_i, x_j) = k(x_i, x_j) + \beta^{-1}\delta_{ij}$$

must be **positive definite.**

- If λ_i is an eigenvalue of kernel matrix K, then matrix C will have as eigenvalue $\lambda_i + \beta^{-1}$.
- Thus, it is sufficient **K be positive semidefinite** so that $\lambda \geq 0$.
- Need for constructing **valid kernels**…

Techniques for Constructing New Kernels.

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following new kernels will also be valid:

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}') && (6.13)\\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') && (6.14)\\
k(\mathbf{x}, \mathbf{x}') &= q\left(k_1(\mathbf{x}, \mathbf{x}')\right) && (6.15)\\
k(\mathbf{x}, \mathbf{x}') &= \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) && (6.16)\\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') && (6.17)\\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}') && (6.18)\\
k(\mathbf{x}, \mathbf{x}') &= k_3\left(\phi(\mathbf{x}), \phi(\mathbf{x}')\right) && (6.19)\\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}' && (6.20)\\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) && (6.21)\\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) k_b(\mathbf{x}_b, \mathbf{x}'_b) && (6.22)
\end{aligned}
$$

where $c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with nonnegative coefficients, $\phi(\mathbf{x})$ is a function from $\mathbf{x}$ to $\mathbb{R}^M$, $k_3(\cdot, \cdot)$ is a valid kernel in $\mathbb{R}^M$, $\mathbf{A}$ is a symmetric positive semidefinite matrix, $\mathbf{x}_a$ and $\mathbf{x}_b$ are variables (not necessarily disjoint) with $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$, and k_a and k_b are valid kernel functions over their respective spaces.

Equipped with these properties, we can now embark on the construction of more complex kernels appropriate to specific applications. We require that the kernel $k(\mathbf{x}, \mathbf{x}')$ be symmetric and positive semidefinite and that it expresses the appropriate form of similarity between $\mathbf{x}$ and $\mathbf{x}'$ according to the intended application. Here we consider a few common examples of kernel functions. For a more extensive discussion of 'kernel engineering', see Shawe-Taylor and Cristianini (2004).

[2]. Useful remark

$$m(x_{N+1}) = k^T C^{-1} T = \sum_{i=1}^{N} a_i k(x_i, x_{N+1})$$

where $a_i = \left[C^{-1} T \right]_i$

- If kernel k(. , .) is Gaussian, then we obtain a radial basis function (RBF) network model.
- Note that inversion of matrix C requires $O(N^3)$ computational cost.

Learning the Gaussian process

- Use a **parametric (θ) kernel function** in the covariance matrix. An example:

$$k(x_i, x_j) = \theta_0 e^{-\frac{\theta_1}{2}\|x_i - x_j\|^2} + \theta_2 + \theta_3 x_i^T x_j$$

Θ={ θ_0 , θ_1 , θ_2 , θ_3 } is the set of the **unknown kernel parameters**.
These can be **estimated** using training examples.

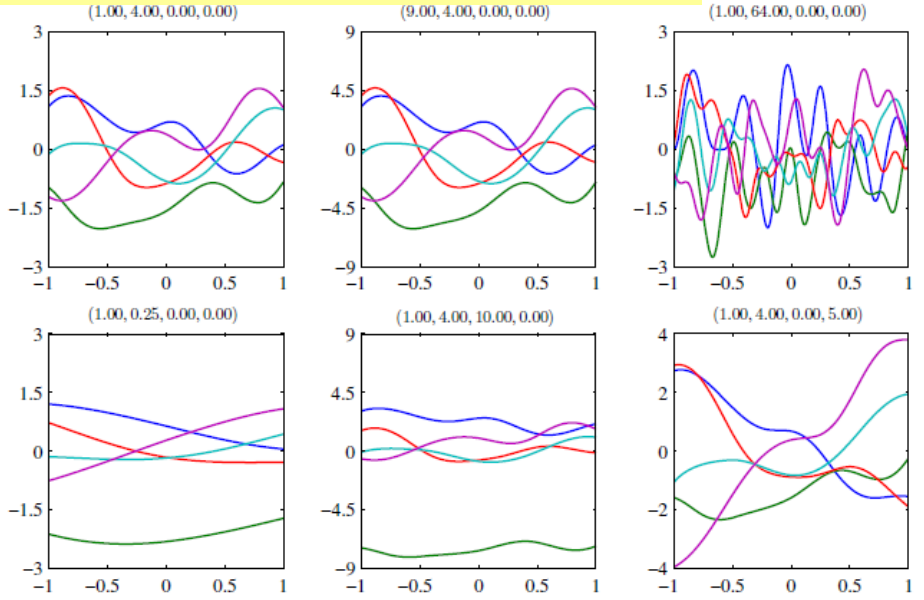


Figure 6.5 Samples from a Gaussian process prior defined by the covariance function (6.63). The title above each plot denotes $(\theta_0, \theta_1, \theta_2, \theta_3)$.

Learning the Gaussian process

- **Learning:** fit the Gaussian process to the data.
- Use the log likelihood function as a measure.

$$\ln p(T \mid \theta) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|C| - \frac{1}{2}T^T C^{-1} T$$

- **Non-convex** function => a lot of local maxima

$$\hat{\theta} = \arg\max_{\theta} \{\ln p(T \mid \theta)\}$$

Calculating the derivatives

$$\ln p(T \mid \theta) = -\frac{1}{2}\ln|C| - \frac{1}{2}T^T C^{-1} T$$

- Useful relations:

Similarly

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{AB}) = \frac{\partial \mathbf{A}}{\partial \mathbf{x}}\mathbf{B} + \mathbf{A}\frac{\partial \mathbf{B}}{\partial \mathbf{x}}. \tag{C.20}$$

The derivative of the inverse of a matrix can be expressed as

$$\frac{\partial}{\partial x}(\mathbf{A}^{-1}) = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{-1} \tag{C.21}$$

as can be shown by differentiating the equation $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ using (C.20) and then right multiplying by $\mathbf{A}^{-1}$. Also

$$\frac{\partial}{\partial x}\ln|\mathbf{A}| = \mathrm{Tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right) \tag{C.22}$$

- Let $\theta=(\theta_i)$ be the set of parameters. Then

Calculating the derivatives

$$\ln p(T \mid \theta) = -\frac{1}{2}\ln|C| - \frac{1}{2}T^T C^{-1} T$$

698 C. PROPERTIES OF MATRICES

- Useful relations:

Similarly

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{AB}) = \frac{\partial \mathbf{A}}{\partial \mathbf{x}}\mathbf{B} + \mathbf{A}\frac{\partial \mathbf{B}}{\partial \mathbf{x}}. \quad \text{(C.20)}$$

The derivative of the inverse of a matrix can be expressed as

$$\frac{\partial}{\partial x}(\mathbf{A}^{-1}) = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{-1} \quad \text{(C.21)}$$

as can be shown by differentiating the equation $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ using (C.20) and then right multiplying by $\mathbf{A}^{-1}$. Also

$$\frac{\partial}{\partial x}\ln|\mathbf{A}| = \mathrm{Tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right) \quad \text{(C.22)}$$

- Let $\theta=(\theta_i)$ be the set of parameters. Then

$$\frac{\partial \ln p(T \mid \theta)}{\partial \theta_i} = -\frac{1}{2}Tr\left(C^{-1}\frac{\partial C}{\partial \theta_i}\right) + \frac{1}{2}T^T C^{-1}\frac{\partial C}{\partial \theta_i}C^{-1}T$$

Gaussian processes for Classification

- **Bayesian estimation**: Model the posterior probability

- Adapt Gaussian Process model to classification problems

- Transform the output of GPs using an nonlinear activation function

Problem formulation

- Consider a binary classification problem $t \in \{0,1\}$
- Define a **GP over a *function a(x)***
- Use a ***logistic sigmoid function*** so as to obtain ***y ∈ [0, 1]***

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

- Probability distribution over target is **Bernoulli**

$$p(t \mid a) = \sigma(a)^t \left(1-\sigma(a)\right)^{1-t}$$

Prediction with Gaussian processes

- **Training set** $D = \{(x_1, t_1), \ldots, (x_N, t_N)\}$ *(input, output pairs)*

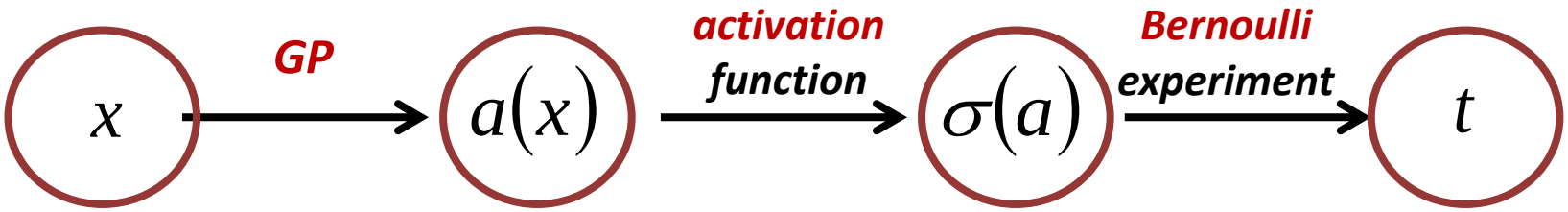
➢ **Introduce a GP over the a's** $a = \{a(x_1), \ldots, a(x_N)\}$

$$p(a) = N(a \mid 0, C)$$

$$C = K + \beta^{-1} I$$

$$K_{ij} = k(x_i, x_j)$$

➢ **Generative (or transformation) scheme**



Prediction with Gaussian processes

- Test point x_{N+1} with unknown class t_{N+1}
- **Goal is** to determine the **posterior**

$$p(t_{N+1} \mid T) \qquad T = (t_1, \ldots, t_N)$$

➢ **Joint distribution** of $a_{N+1} = (a_1, \ldots, a_N, a_{N+1})$

$$a_{N+1} = \{a(x_1), \ldots, a(x_N), a(x_{N+1})\}$$

$$p(a_{N+1}) = N(a_{N+1} \mid 0, C_{N+1})$$

Prediction with Gaussian processes

- For 2-class problems we want to predict the

$$p(t_{N+1}=1|T) \quad p(t_{N+1}=0|T_N)=1-p(t_{N+1}=1|T_N)$$

$$p(t_{N+1}=1|T)=$$

$$=\int p(t_{N+1}=1|a(x_{N+1}))p(a(x_{N+1})|T)da(x_{N+1})$$

where $p(t_{N+1}=1|a(x_{N+1}))=\sigma(a(x_{N+1}))$

$$p(a(x_{N+1})|T)=N(a(x_{N+1})|m(x_{N+1},\sigma^2_{N+1}(x_{N+1})))$$

Obtaining from GP regression

Remember GP regression

$$p(t_{N+1} \mid T) = N(t_{N+1} \mid m(x_{N+1}), \sigma^2_{N+1}(x_{N+1}))$$

$$m(x_{N+1}) = k^T C^{-1} T$$

$$\sigma^2(x_{N+1}) = c - k^T C^{-1} k$$

$$c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$$

$$k = (k(x_1, x_{N+1}), \ldots, k(x_N, x_{N+1}))$$

Approximation methods for integral calculation

$$p(t_{N+1}=1|T)=\int \sigma(a(x_{N+1}))N(a(x_{N+1})|\, m_{N+1},\sigma^2_{N+1})da(x_{N+1})=$$
$$=E_{N(m_{N+1},\sigma^2_{N+1})}[\sigma(a(x_{N+1}))]$$

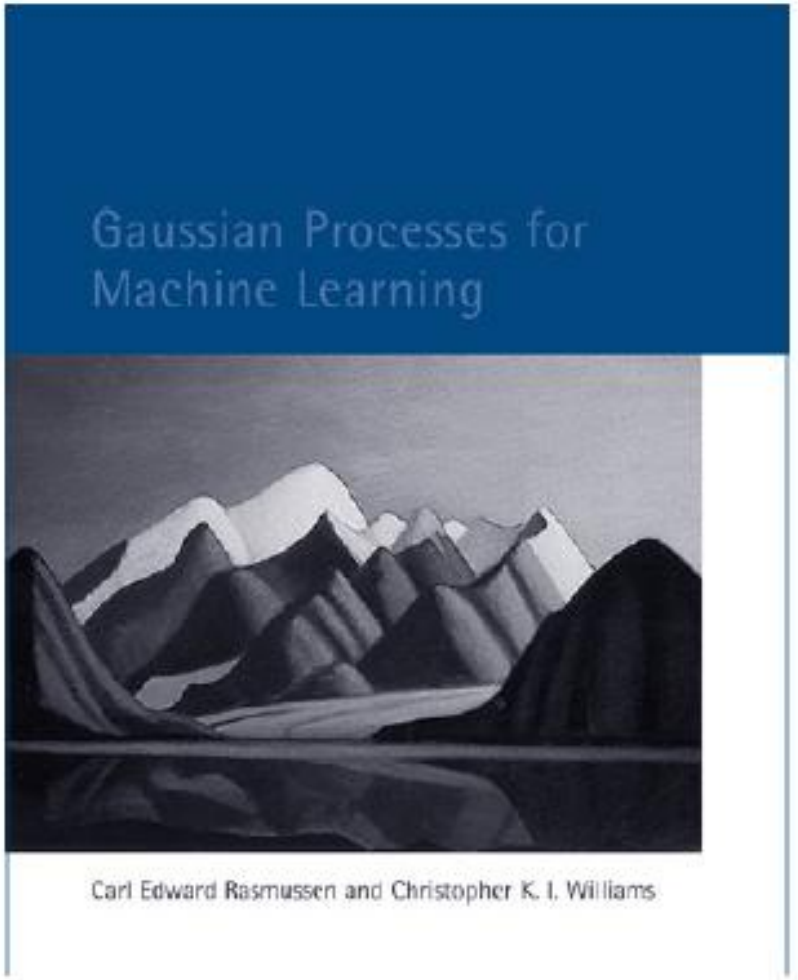
- **Laplace approximation** [Barber & Williams]
- **Variational methods** [Gibbs & MacKay]
- **Expectation-Propagation** [Minka & Ghahramani]
- **MCMC sampling** [Neal]

Summary

- GP's provide a **structured method of model** and parameter selection.
- The key ingredient of a GP is the **covariance function**; a recipe to construct covariance matrices.
- Learning takes the form of setting the hyper-parameters, using the **marginal likelihood**.
- GP's can be used for **regression** or **classification.** However require approximate inference techniques.

Further reading on Gaussian Processes

Carl Edward Rasmussen and Chris Williams, MIT Press, 2006



Many more topics and code:

http://www.gaussianprocess.org/

Relevance Vector Machine – RVM

Michael E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", JMLR, 2001

- The relevance vector machine (RVM) is a Bayesian **sparse kernel method** for regression and classification.
- It covers **many applications**
- **Solves problems** with the SVM

Support Vector Machines (SVM)

- A non-probabilistic decision machine: Returns **point estimate** for decision
- Makes decisions based on the function:

$$y(x, w) = \sum_{i=1}^{N} w_i K(x, x_i) + w_0$$

where K is the kernel function

- Attempts to minimize the error while simultaneously maximize the margin between the two classes

SVM "Problems"

- Non-probabilistic predictions.
- Requires estimation of error/margin trade-off parameters
- The kernel function $K(x, x_i)$ must satisfy **mercer's condition:**

  K must be a positive definite function

$$\int\int f(x)f(y)K(x,y)dxdy > 0 \qquad (\forall f \in L_2)$$

$$\int_{-\infty}^{+\infty} |f(x)|^2 dx < \infty$$

Relevance Vector Machines – RVM's

- Apply **Bayesian treatment** to SVM
- The kernel functions in RVM are treated simply as a set of basis functions **without many restrictions** imposed on SVM kernels
- **Sparseness:** Posterior distributions of the majority of weights are peaked around zero. Training vectors associated with the non-zero weights are the '**relevant vectors**'.
- Uses significantly **fewer kernel functions** than SVM.

RVM for regression

- Training set $D = \{(x_1, t_1), \ldots, (x_N, t_N)\}$
- **Generative model**

$$t = y(x) + \varepsilon = w^T \phi(x) + \varepsilon$$

- Assuming zero mean Gaussian noise

$$\varepsilon \sim N(0, \beta^{-1})$$

- We take the conditional distribution of target

$$p(t \mid x, w, \beta) = N(t \mid w^T \phi(x), \beta^{-1})$$

RVM for regression

- "**Classical linear regression**" assumes a linear combination with M nonlinear basis functions

$$y(x) = w^T \phi(x) = \sum_{i=1}^{M} w_i \phi_i(x)$$

- **RVM assumes** N kernels, one for each training example:

$$y(x) = \sum_{i=1}^{N} w_i k(x, x_i) + w_0$$

- It has the **same structure of SVM**, except that coefficients $\boldsymbol{a_i}$ are now denoted as $\boldsymbol{w_i}$.

- **Likelihood function** for the set of *N* target values

$$p(T \mid X, w, \beta) = \prod_{i=1}^{N} p(t_i \mid x_i, w, \beta) = N(T \mid \Phi w, \beta^{-1} I)$$

$$p(T \mid X, w, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp\left\{-\frac{\beta}{2}\|T - \Phi w\|^2\right\}$$

where $T = (t_1, t_2, \ldots, t_N)$ $\quad w = (w_0, w_1, \ldots, w_N)$

$$\Phi = [\phi(x_1)\phi(x_2)\ldots\phi(x_N)]$$

$$\phi(x_i) = [k(x_1, x_i)k(x_2, x_i)\ldots k(x_N, x_i)]$$

- **ARD Prior:** Gaussian prior for weights defining a separate hyperparameter $a_i \; \forall \; w_i$:

$$p(w \mid a) = \prod_{i=0}^{N} N\left(w_i \mid 0, a_i^{-1}\right) = N\left(w \mid 0, A^{-1}\right)$$

$$A = diag(a_1, \ldots, a_N)$$

- a: is a vector of N+1 **hyperparameters**

- Introduce an hyperprior over a and precision β

$$p(a) = \prod_{i=0}^{N} Gamma(a_i \mid \alpha, b)$$

$$p(\beta) = Gamma(\beta \mid c, d)$$

- **Introduce sparse weights:** integrate out the precision a and take the marginal weight prior

$$p(w_i) = \int p(w_i \mid a_i) p(a_i) da_i =$$

$$= \int N\left(w_i \mid 0, \alpha_i^{-1}\right) Gamma(\alpha_i \mid a, b) d\alpha_i =$$

$$= \frac{b^a \Gamma\left(a + \frac{1}{2}\right)}{(2\pi)^{\frac{1}{2}} \Gamma(a)} \left(b + \frac{w_i^2}{2}\right)^{-\left(a+\frac{1}{2}\right)}$$

Student-t distribution

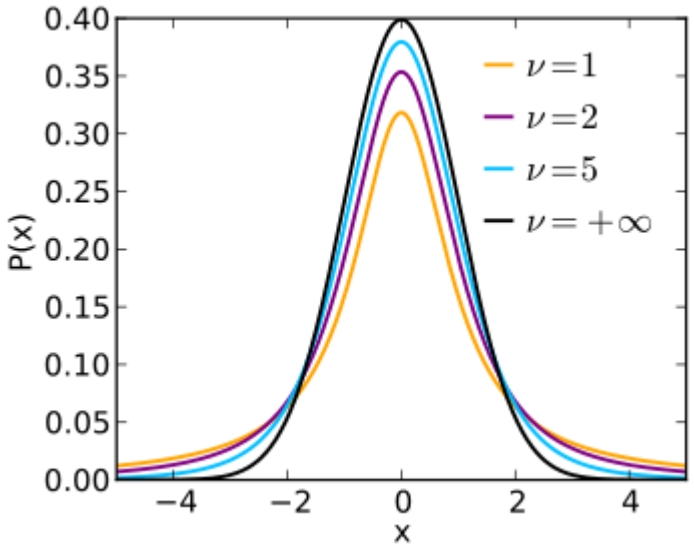
- Special case where *a=b=0* =>

Uninformative priors

$$p(w_i) \propto \frac{1}{|w_i|}$$

Student-t distribution *(Gosset 1908)* $T \sim t_n$

- . $E(T) = 0$ $VAR(T) = \frac{n}{n-2}$
- . $\lim_{n\to\infty} f_n(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$ ***N(0,1)***



pdf:

$$p_n(t) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{\pi}\sqrt{n}\Gamma\left(\frac{n}{2}\right)}\left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}} = \frac{1}{\sqrt{\pi}\mathrm{B}\left(\frac{1}{2},\frac{n}{2}\right)}\left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}}$$

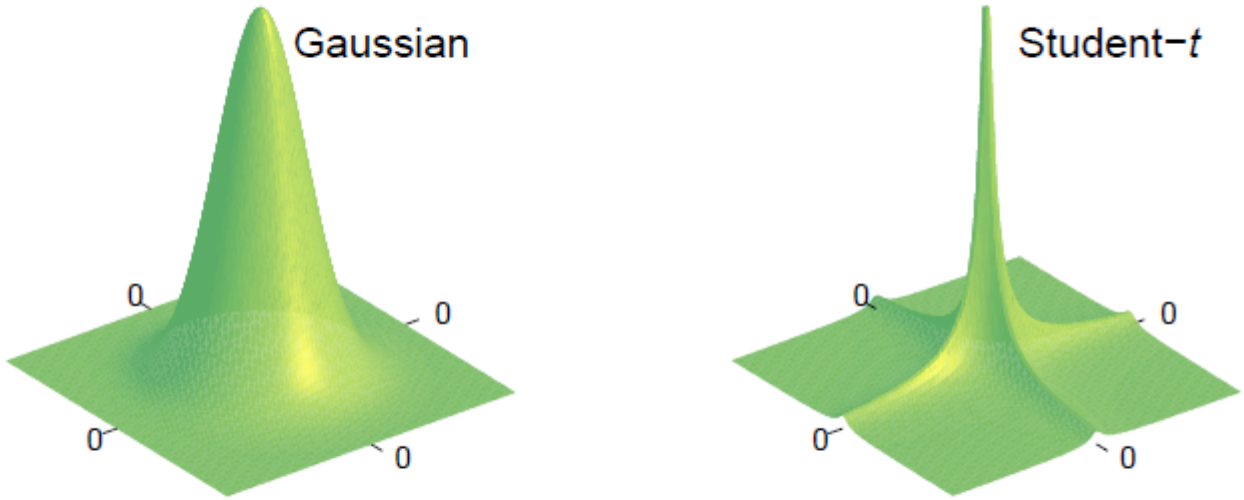
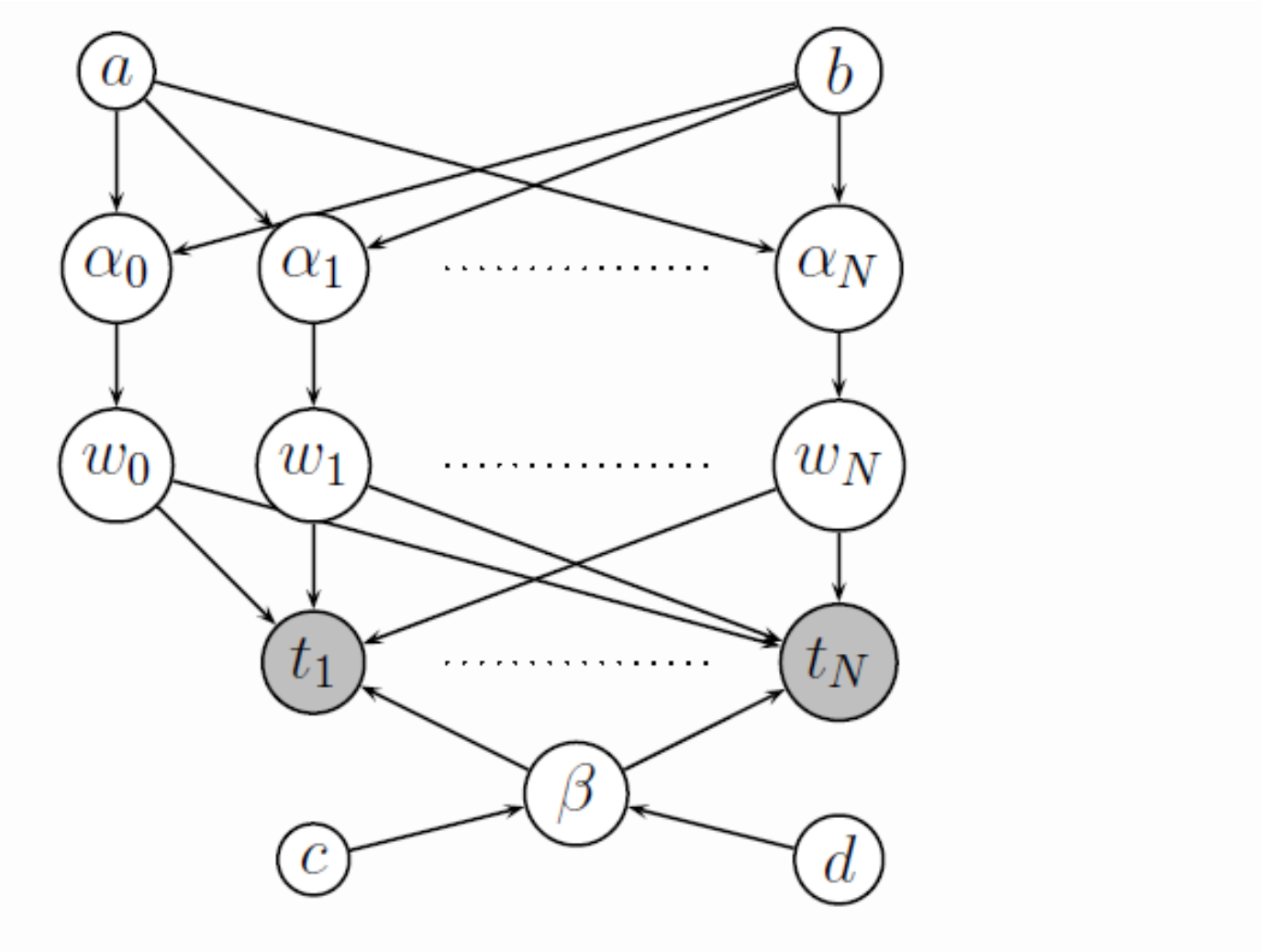


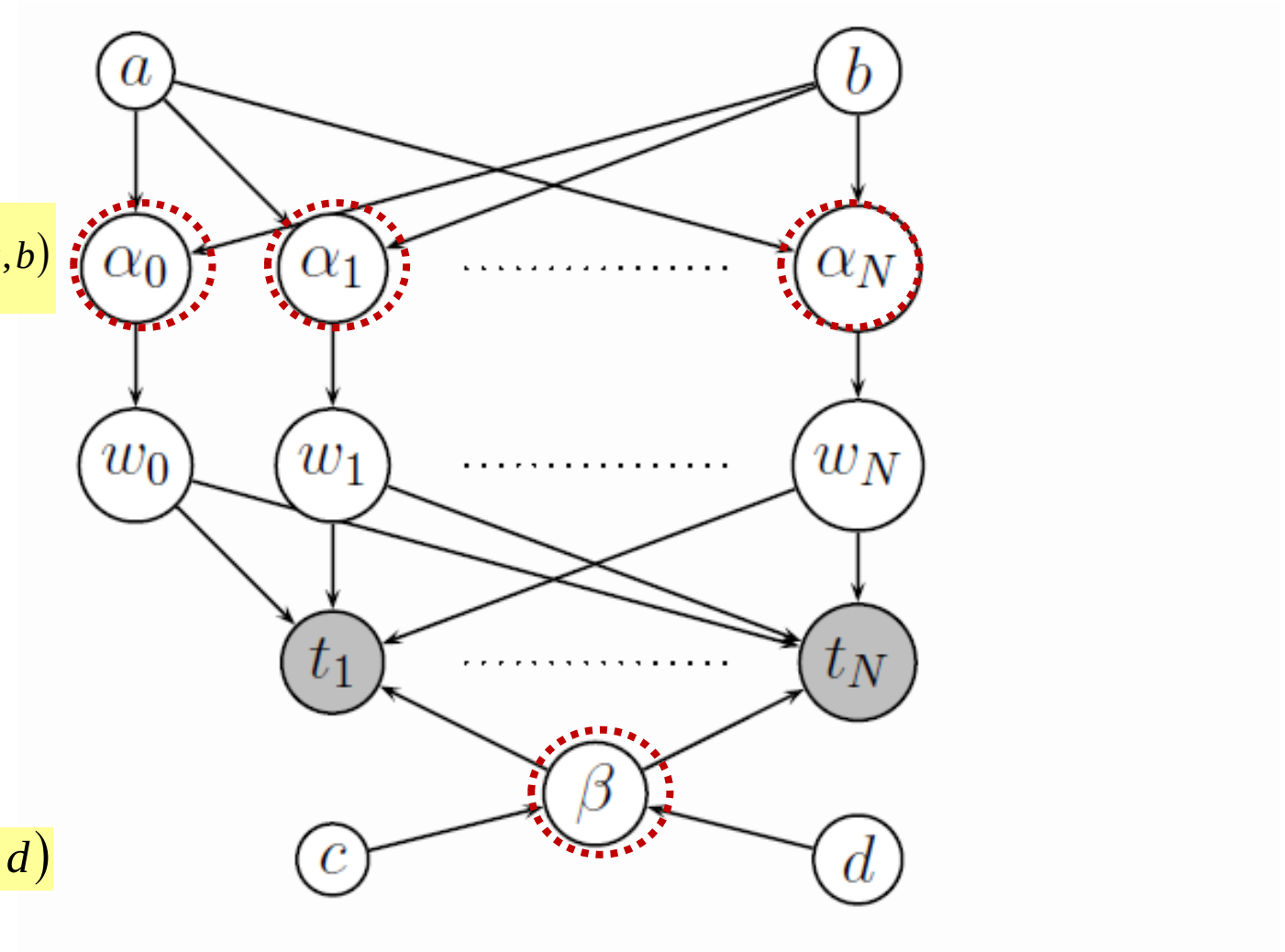
Figure 6: LEFT: an example Gaussian prior $p(\mathbf{w}|\boldsymbol{\alpha})$ in two dimensions. RIGHT: the prior $p(\mathbf{w})$, where the hyperparameters have been integrated out to give a product of Student-t distributions. Note that the probability mass is concentrated both at the origin and along 'spines' where one of the two weights is zero.

Priors - graphical model



Priors - graphical model

$$p(a) = \prod_{i=0}^{N} Gamma(a_i \mid \alpha, b)$$



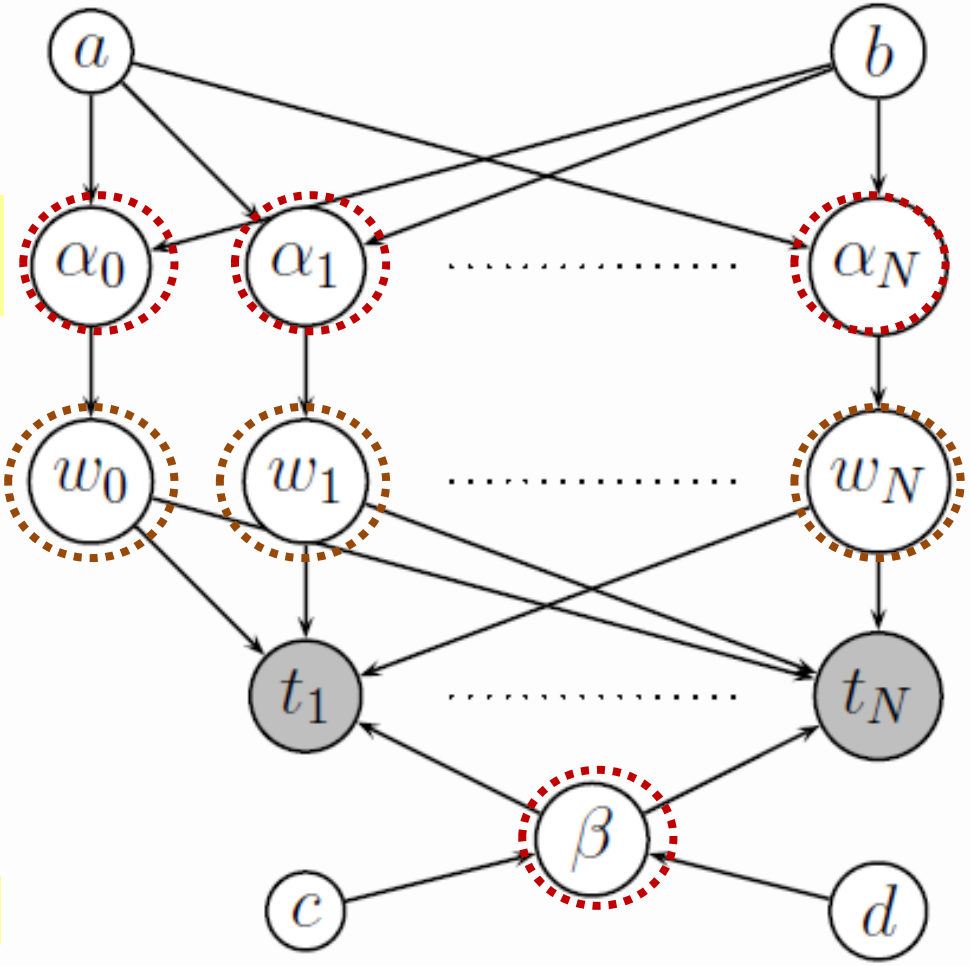
$$p(\beta) = Gamma(\beta \mid c, d)$$

Priors - graphical model

$$p(a) = \prod_{i=0}^{N} Gamma(a_i \mid \alpha, b)$$

$$p(w \mid a) = \prod_{i=0}^{N} N(w_i \mid 0, a_i^{-1})$$

$$p(\beta) = Gamma(\beta \mid c, d)$$



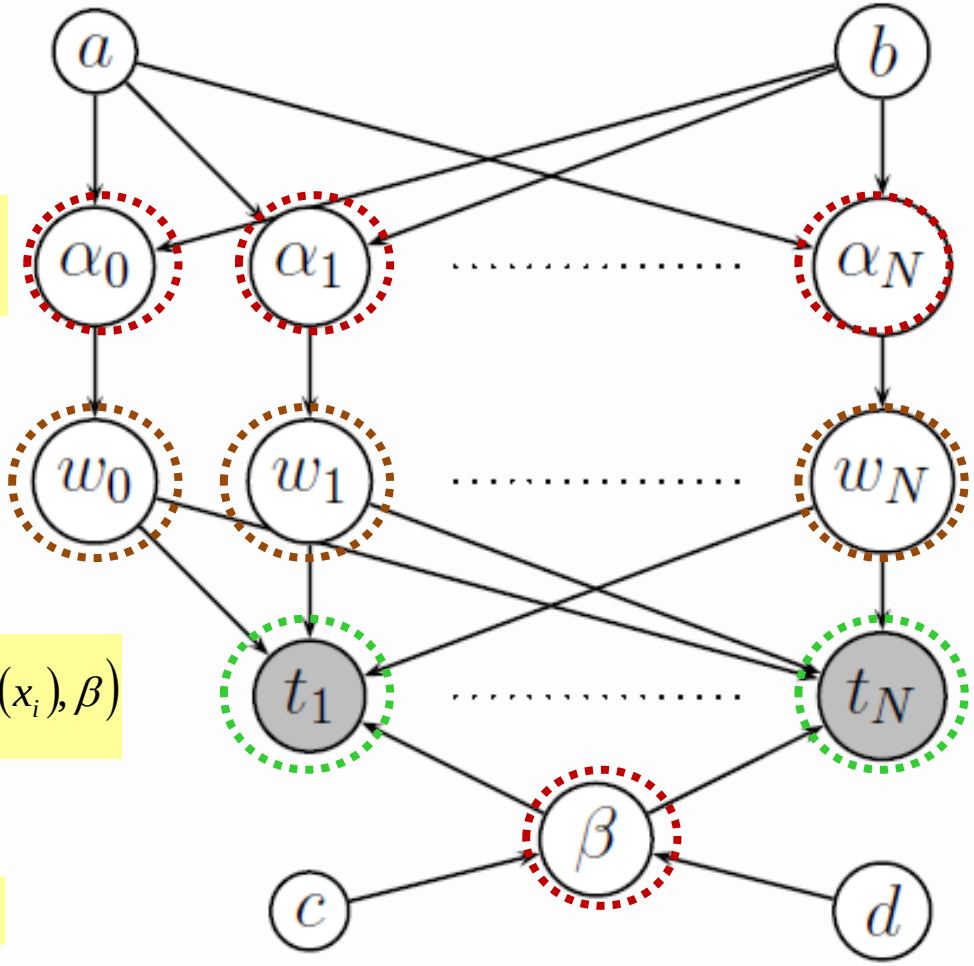
Priors - graphical model

$$p(a) = \prod_{i=0}^{N} Gamma(a_i \mid \alpha, b)$$

$$p(w \mid a) = \prod_{i=0}^{N} N\left(w_i \mid 0, a_i^{-1}\right)$$

$$p(T \mid w, \beta) = \prod_{i=1}^{N} N\left(t_i \mid w^T \phi(x_i), \beta\right)$$

$$p(\beta) = Gamma(\beta \mid c, d)$$



observations

Inference – posterior over w

- The **posterior over the weights w**

$$p(w \mid T, a, \beta) = \frac{p(T \mid w, \beta)p(w \mid \alpha)}{\int p(T \mid w, \beta)p(w \mid \alpha)dw}$$

- Since all distributions are Gaussian, we can obtain analytical expression for the posterior pdf:

- product of two Gaussians

$$p(w|T,\alpha,\beta) \propto p(T|w,\beta)p(w|\alpha) = \\ = N(T|\Phi w, \beta^{-1}I)N(w|0, A^{-1})$$

If we have a marginal Gaussian distribution for x and a conditional Gaussian distribution for y given x in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad \text{(B.42)}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad \text{(B.43)}$$

then the marginal distribution of y, and the conditional distribution of x given y, are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \quad \text{(B.44)}$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad \text{(B.45)}$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}. \quad \text{(B.46)}$$

C. M. Bishop, "*Pattern Recognition and Machine Learning*", page 689

- Posterior distribution: product of two Gaussians

$$p(w|T,\alpha,\beta) \propto p(T|w,\beta)p(w|\alpha) = N(T|\Phi w, \beta^{-1}I)N(w|0, A^{-1})$$

If we have a marginal Gaussian distribution for x and a conditional Gaussian distribution for y given x in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad \text{(B.42)}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}) \quad \text{(B.43)}$$

$N(w|0, A^{-1})$

$N(T|\Phi w, \beta^{-1}I)$

then the marginal distribution of y, and the conditional distribution of x given y, are given by

$p(w|T,X) = N(w|\mu,\Sigma)$

$$\Lambda^{-1}\mathbf{A}^{\mathrm{T}}) \quad \text{(B.44)}$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad \text{(B.45)}$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}. \quad \text{(B.46)}$$

$\mu = \beta \Sigma \Phi^T T$

$\Sigma = \left(A + \beta \Phi^T \Phi\right)^{-1}$

C. M. Bishop, "*Pattern Recognition and Machine Learning*", page 689

- **Posterior distribution** of weights

$$p(w|T,a,\beta)=N(w|\mu,\Sigma)$$

$$\mu=\beta\Sigma\Phi^{T}T$$

$$\Sigma=\left(A+\beta\Phi^{T}\Phi\right)^{-1}$$

Estimation of model parameters

2 approaches

1st approach

- **Maximum A-Posteriori (MAP) estimation problem**

$$\ln p(w,\alpha,\beta \mid T) \propto$$
$$\propto \ln p(T \mid w,\beta)p(w \mid \alpha)p(\alpha \mid a,b)p(\beta \mid c,d)$$

- Maximizing the log-likelihood (e.g. using **EM algorithm**)

$$\{\theta = \{w,\alpha,\beta\}\} = \max_{\theta} \ln p(w,\alpha,\beta \mid T)$$

Using **EM algorithm** for Maximizing the MAP log-likelihood

- Treat weights as hidden variables and maximize

$$E_{w|T,\alpha,\beta}\left[\ln p(T \mid w,\beta)p(w \mid \alpha)p(\alpha)p(\beta)\right]$$

- Update rules:

$$\alpha_i = \frac{1+2a}{E_{w|T,\alpha,\beta}\left[w_i^2\right]+2b} = \frac{1+2a}{\Sigma_{ii}+\mu_i^2+2b}$$

$$\left(\beta^{-1}\right)^{new} = \frac{\left\|T-\Phi\mu\right\|^2 + \left(\beta^{-1}\right)^{old}\sum_i \gamma_i + 2d}{N+2c}$$

$$\gamma_i \equiv 1 - a_i\Sigma_{ii}$$

Estimation of model parameters

2nd approach

- **Obtaining the marginal likelihood of targets**

$$p(T \mid X, \alpha, \beta) = \int p(T \mid X, w, \beta) p(w \mid \alpha) dw$$

- **Maximum likelihood estimation problem**

$$\{\theta = \{\alpha, \beta\}\} = \max_{\theta} \ln p(T \mid X, \alpha, \beta)$$

- **Marginal likelihood**

$$p(T \mid \alpha, \beta) = \int p(T \mid w, \beta)\, p(w \mid \alpha)\, dw$$

If we have a marginal Gaussian distribution for x and a conditional Gaussian distribution for y given x in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad \text{(B.42)}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad \text{(B.43)}$$

then the marginal distribution of y, and the conditional distribution of x given y, are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \quad \text{(B.44)}$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad \text{(B.45)}$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}. \quad \text{(B.46)}$$

C. M. Bishop, "*Pattern Recognition and Machine Learning*", page 689

- **Marginal likelihood**

$$p(T \mid \alpha, \beta) = \int p(T \mid w, \beta)\, p(w \mid \alpha)\, dw$$

If we have a marginal Gaussian distribution for x and a conditional Gaussian distribution for y given x in the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \qquad N\left(w \mid 0, A^{-1}\right) \tag{B.42}$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid \mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \qquad N\left(T \mid \Phi w, \beta^{-1} I\right) \tag{B.43}$$

then the marginal distribution of y, and the conditional distribution of x given y, are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \tag{B.44}$$

$$p \qquad p(T \mid \alpha, \beta) = N(T \mid 0, S) \qquad) \tag{B.45}$$

where

$$S = \beta^{-1} I + \Phi A^{-1} \Phi^{T} \tag{B.46}$$

C. M. Bishop, "*Pattern Recogni*

Maximize marginal log-likelihood

$$L = -\frac{1}{2}\ln|S| - \frac{1}{2}T^T S^{-1} T + \sum_{i=0}^{N} \ln Gamma(\alpha_i \mid a,b) + \ln p(\beta)$$

- Maximizing wrt ln(α) and log(β) and using $p(\ln a) = a p(a)$

$$L = -\frac{1}{2}\left\{\ln|S| + T^T S^{-1} T\right\} + \sum_{i=0}^{N} \left(a \ln \alpha_i - b\alpha_i\right) + c \ln \beta - d\beta$$

where

$$S = \beta^{-1} I + \Phi A^{-1} \Phi^T$$

- Taking derivatives equal to zero:

$$\frac{\partial L}{\partial \ln \alpha_i} = \frac{1}{2}\left[1-\alpha_i\left(\mu_i^2+\Sigma_{ii}\right)\right]+a-b\alpha_i = 0 \Rightarrow$$

$$\alpha_i^{new} = \frac{1+2a}{\mu_i^2+\Sigma_{ii}+2b}$$

$$\Sigma = \left(A+\beta\Phi^T\Phi\right)^{-1}$$

- Alternative rule using the quantities: $\gamma_i \equiv 1-a_\iota\Sigma_{ii}$

MacKay 1992

$$\alpha_i^{new} = \frac{\gamma_i+2a}{\mu_i^2+2b}$$

faster convergence

- Alternative rule using the quantities: $\gamma_i \equiv 1 - a_i \Sigma_{ii}$

$$\alpha_i^{new} = \frac{\gamma_i + 2a}{\mu_i^2 + 2b}$$

$$\Sigma = \left(A + \beta \Phi^T \Phi\right)^{-1}$$

- Quantities $\gamma_i \in [0,1]$ show **how well-determined** parameter *w* is by the data (***MacKay, 1992***).
  - If a_i is large (***w_i doesn't fits the data***) => $\Sigma_{ii} \approx \alpha_i^{-1}$ and thus $\gamma_i \approx 0$
  - If a_i is small (***w_i fits the data***) => $\gamma_i \approx 1$

- Taking derivatives equal to zero:

$$\frac{\partial L}{\partial \ln \beta} = \frac{1}{2}\left[\frac{N}{\beta} - \left\|T - \Phi\mu\right\|^2 - Tr\left(\Sigma\Phi^T\Phi\right)\right] + c - d\beta = 0 \Rightarrow$$

- Using the fact that $Tr\left(\Sigma\Phi^T\Phi\right) = \beta^{-1}\sum_i \gamma_i$

$$\gamma_i \equiv 1 - a_i\Sigma_{ii}$$

$$\Sigma = \left(A + \beta\Phi^T\Phi\right)^{-1}$$

then:

$$\left(\beta^{-1}\right)^{new} = \frac{\left\|T - \Phi\mu\right\|^2 + 2d}{N - \sum_i \gamma_i + 2c}$$

Inference – Making predictions

- Given a new input x^* make a prediction of its target t^*. This is equivalent on **estimating posterior distribution**

$$p\left(t^* \mid T=\{t_1,\ldots,t_N\}\right)=p\left(t^* \mid x^*,T,\hat{\alpha},\hat{\beta}\right)$$

- Marginalizing we take:

$$p\left(t^* \mid x^*,T,\hat{\alpha},\hat{\beta}\right)=\int p\left(t^* \mid x^*,w,\hat{\beta}\right)p\left(w \mid T,\hat{\alpha},\hat{\beta}\right)dw$$

Integration of two Gaussians

$$p\left(t^* \mid x^*, T, a, \beta\right) = \int p\left(t^* \mid x^*, w, \hat{\beta}\right) p\left(w \mid T, \hat{a}, \hat{\beta}\right) dw$$

where

$$p\left(t^* \mid x^*, w, \beta\right) = N\left(t^* \mid w^T \phi\left(x^*\right), \beta^{-1}\right)$$

$$p\left(w \mid T, \hat{\alpha}, \hat{\beta}\right) = N\left(w \mid \mu, \Sigma\right)$$

$$\mu = \hat{\beta} \Sigma \Phi^T T \qquad \Sigma = \left(\hat{A} + \hat{\beta} \Phi^T \Phi\right)^{-1}$$

Predictive distribution (cont.)

$$p\left(t^* \mid x^*, T, a, \beta\right) = \int p\left(t^* \mid x^*, w, \hat{\beta}\right) p\left(w \mid T, \hat{a}, \hat{\beta}\right) dw$$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad \text{(B.42)}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad \text{(B.43)}$$

then the marginal distribution of $\mathbf{y}$, and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$, are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \quad \text{(B.44)}$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad \text{(B.45)}$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}. \quad \text{(B.46)}$$

where:

Predictive distribution (cont.)

$$p(t^* \mid x^*, T, a, \beta) = \int p(t^* \mid x^*, w, \hat{\beta}) p(w \mid T, \hat{a}, \hat{\beta}) dw$$

$$p(w \mid T, \hat{\alpha}, \hat{\beta}) = N(w \mid \mu, \Sigma)$$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad \text{(B.43)}$$

$$p(t^* \mid x^*, w) = N(t^* \mid w^T \phi(x^*), \beta^{-1})$$

then the marginal distribution of y, and the conditio
given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}) \quad \text{(B.44)}$$

where $p(t^* \mid x^*, T) = N(t^* \mid \mu^T \phi(x^*), \sigma^2(x^*))$

$$\Sigma = (\Lambda + A^T L A)^{-1}. \quad \text{(B.46)}$$

$$\sigma^2(x^*) = \hat{\beta}^{-1} + \phi^T(x^*) \Sigma \phi(x^*)$$

where:

- **Predictive distribution:**

$$p\left(t^* \mid x^*, T\right) = N\left(t^* \mid \mu^T \phi\left(x^*\right), \sigma^2\left(x^*\right)\right)$$

- **Prediction:**

$$y\left(x^*; \mu\right) = \mu^T \phi\left(x^*\right) = \sum_{i=1}^{N} \mu_i k\left(x_i, x^*\right)$$

- **Variance:**

$$\sigma^2\left(x^*\right) = \hat{\beta}^{-1} + \phi^T\left(x^*\right) \Sigma \phi\left(x^*\right)$$

noise variance + uncertainty on the prediction of the weights

RVM for classification

- Consider a binary classification problem $t \in \{0,1\}$

- Output is calculated using a logistic sigmoid function

$$y(x) = \sigma\left(w^T \phi(x)\right)$$

- **ARD prior** over the weights

$$p(w \mid a) = \prod_{i=1}^{N} N\left(w_i \mid 0, a_i^{-1}\right) = N\left(w \mid 0, A^{-1}\right)$$

- Obtain the posterior distribution of weights

RVM for classification

- **Posterior distribution** over w:

$$p(w \mid a) = N(w \mid 0, A^{-1})$$

$$p(w \mid T, a) = \frac{p(T \mid w)\, p(w \mid a)}{p(T \mid a)}$$

- **Class conditional distribution** of inputs

$$p(T \mid w) = \prod_{i=1}^{N} y_i^{t_i} (1 - y_i)^{1 - t_i}$$

$$y_i = \sigma\left(w^T \phi(x_i)\right)$$

- Log-likelihood

$$\ln p(w \mid T, a) \propto \sum_{i=1}^{N} \{t_i \ln y_i + (1 - t_i) \ln(1 - y_i)\} - \frac{1}{2} w^T A w$$

- **Maximization problem** $y_i = \sigma\left(w^T \phi(x_i)\right)$

$$\max_w \left\{ \sum_{i=1}^{N} \left\{ t_i \ln y_i + (1 - t_i) \ln(1 - y_i) \right\} - \frac{1}{2} w^T A w \right\}$$

- Use ***Newton-Raphson*** **optimization** scheme

$$w^{(new)} = w^{(old)} - H^{-1} \nabla E(w) \qquad H = \nabla \nabla E(w)$$

- We obtain:

$$\nabla E(w) = \Phi^T (T - Y) - A w$$

$$\nabla \nabla E(w) = -\left(\Phi^T B \Phi + A\right) \qquad B = \left[b_i = y_i (1 - y_i) \right]$$