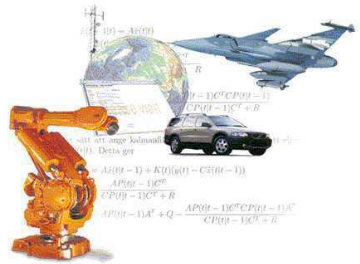
Machine Learning, Lecture 5
Support Vector Machines and Approximate Inference



Thomas Schön, Henrik Ohlsson and Umut Orguner

Division of Automatic Control
Linköping University
Linköping, Sweden.

Email: schon@isy.liu.se,
Phone: 013 - 281373,
`www.control.isy.liu.se/~schon/`

Outline Lecture 5

1. Summary of lecture 4
2. Support Vector Machines
3. Variational Bayesian Inference
   - General Derivation
   - Example – Identification of a Linear State-Space Model
   - Example – Gaussian Mixtures
4. Expectation Propagation
   - General Derivation
   - Example – State Estimation

(Chapter 7.1, 10)

Summary of Lecture 4 (I/III)

A **neural network** is a nonlinear function (as a function expansion) from a set of input variables $\{x_i\}$ to a set of output variables $\{y_k\}$ controlled by adjustable parameters w.

This function expansion is found by formulating the problem as usual, which results in a (non-convex) optimization problem. This problem is solved using numerical methods.

Backpropagation refers to a way of computing the gradients by making use of the chain rule, combined with clever reuse of information that is needed for more than one gradient.

Summary of Lecture 4 (II/III)

A kernel function $k(x,z)$ is defined as an inner product

$$k(x,z) = \phi(x)^T\phi(z),$$

where $\phi(x)$ is a fixed mapping.

Introduced the kernel trick (a.k.a. kernel substitution). In an algorithm where the input data x enters only in the form of scalar products we can replace this scalar product with another choice of kernel.

The use of kernels allows us to implicitly use basis functions of high, even infinite, dimensions ($M \to \infty$).

Summary of Lecture 4 (III/III)

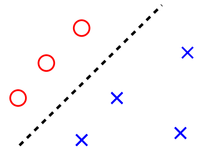
A **Gaussian process** is a collection of random variables, any finite number of which have a joint Gaussian distribution.

By assuming that the considered system is a Gaussian process, predictions can be made by computing the conditional distribution $p(y(x^*)|\text{all the observations})$, $y(x^*)$ being the output for which we seek a prediction. This regression approach is referred to as **Gaussian process regression**.

Support Vector Machines (SVM)

Very popular classifier.

- Non-probabilistic
- Discriminative
- Can also be used for regression (then called *support vector regression*, SVR).
- Convex optimization
- Sparse



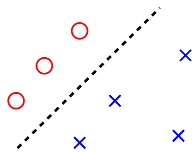
SVM for Classification

Assume: $\{(t_n, x_n)\}_{n=1}^N$, $x_n \in \mathcal{R}^{n_x}$ and $t_n \in \{-1, 1\}$, is a given training data set (linearly separable).

Task: Given x^*, what is the corresponding label?

SVM is a discriminative classifier, i.e. it provides a decision boundary. The decision boundary is given by $\{x | w^T\phi(x) + b = 0\}$.

Goal: Find the decision boundary that maximizes the margin! The *margin* is the distance to the closest point to the decision boundary.



SVM for Classification Cont'd

The decision boundary that maximizes the margin is given as the solution to the quadratic program (QP)

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$\text{s.t.} \quad t_n(w^T\phi(x_n) + b) - 1 \geq 0, \quad n = 1, \dots, N$$

To make it possible to let the dimension of the feature space (dim of $\phi(x_n)$) go to infinity, we have to derive the dual.

SVM for Classification Cont'd

First, the Lagrangian is

$$L(w,b,\mathbf{a}) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n \Big(t_n(w^T\phi(x_n) + b) - 1 \Big)$$

and minimizing wrt w, b we obtain the dual $g(\mathbf{a})$. Taking the derivative wrt w, b and set them to zero,

$$\frac{dL(w,b,\mathbf{a})}{db} = \sum_{n=1}^{N} a_n t_n = 0, \quad \frac{dL(w,b,\mathbf{a})}{dw} = w - \sum_{n=1}^{N} a_n t_n \phi(x_n) = 0$$

This gives

$$g(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} a_n a_m t_n t_m \phi(x_m)^T\phi(x_n)$$

SVM for Classification Cont'd

Let $k(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. The dual objective then becomes

$$g(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} a_n a_m t_n t_m k(x_m, x_n)$$

which we can maximize w.r.t. $\mathbf{a}$ and subject to

$$a_n \geq 0, \quad \sum_{n=1}^{N} a_n t_n = 0.$$

The maximizing $\mathbf{a}$, let say $\hat{\mathbf{a}}$, gives using $w^T\phi(x^*) = (\sum_{n=1}^{N} a_n t_n \phi(x_n))^T\phi(x^*)$ that

$$y(x^*) = \sum_{n=1}^{N} \hat{a}_n t_n k(x^*, x_n) + b.$$

Many $\hat{a}$'s will be zero $\Rightarrow$ computational remedy.

SVM for Classification – Non-Separable Classes

If points are on the right side of the decision boundary, then $t_n(w^T\phi(x_n) + b) \geq 1$. To allow for some violations, we introduce slack variables ζ_n, $n = 1, \ldots, N$. The modified optimization problem becomes

$$\min_{w,b,\zeta} \frac{1}{2}\|w\|^2 + C\sum_{n}^{N}\zeta_n$$

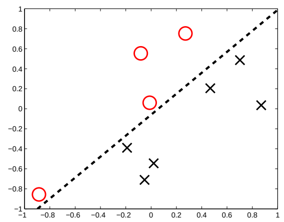
$$\text{s.t.} \quad t_n(w^T\phi(x_n) + b) + \zeta_n - 1 \geq 0, \quad n = 1, \ldots, N,$$

$$\zeta_n \geq 0, \quad n = 1, \ldots, N.$$

Example – CVX to Compute SVM

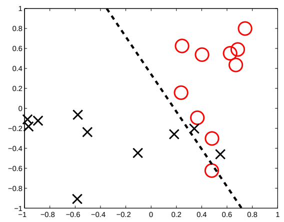
Linearly separable data:

```
cvx_begin
variables w(nx,1) b
minimize (0.5*w'*w)
  subject to
   y.*(w'*x+b*ones(1,N))-ones(1,N) >= 0
cvx_end
```



Non-separable data:

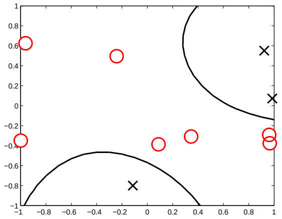
```
cvx_begin
variables w(nx,1) b zeta(1,N)
minimize (0.5*w'*w + C*ones(1,N)*zeta')
  subject to
   y.*(w'*x+b*ones(1,N))-ones(1,N)+zeta >= 0
   zeta >= 0
cvx_end
```



Example – CVX to Compute SVM

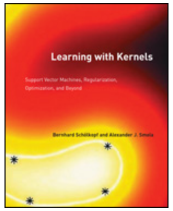
SVM – Solving the dual:

```
k=@(x1,x2) exp(-sum((x1*ones(1,size(x2,2))-x2).^2)/0.5)'
for t=1:N;for s=t:N
K(t,s)=k(x(:,t),x(:,s));K(s,t)=K(t,s);
end;end
cvx_begin
variables a(N,1)
minimize( 1/2*(a.*y')'*K*(a.*y') - ones(1,N)*a)
  subject to
   ones(1,N)*(a.*y') == 0
   a >= 0
cvx_end
ind=find(a>0.01);
wphi = @(xstar) ones(1,N)*(a.*y'.*k(xstar,x))
b=0;
for i=1:length(ind)
b=b+1/y(ind(i))-wphi(x(:,ind(i)));
end
b=b/length(ind);
ystar = @(xstar) wphi(xstar)+b
```



Further Reading and Code

- Bernhard Schölkopf and Alex Smola. Learning with Kernels. MIT Press, Cambridge, MA, 2002.
- Yalmip can be downloaded from `http://users.isy.liu.se/johanl/yalmip/`
- CVX can be downloaded from `http://cvxr.com/cvx/`



Bayesian Framework Reminder

- Let $X = x_1, \dots, x_N$ be the measurements.
- Let $Z = z_1, \dots, z_N$ be the latent variables as in the EM framework.
- Then, the Bayesian framework is interested in the posterior density $p(Z|X)$ given by Bayes rule as

$$p(Z|X) = \frac{p(X|Z)p(Z)}{p(X)}$$

- For quite many instances, the posterior can be found exactly using the concept of *conjugate pairs*.
  - Gaussian case
  - More generally the exponential family.
- What happens when there is no exact solution?

Variational Methods (I/II)

- Classic calculus involves functions and defines *derivatives* to optimize them.
- The so-called *calculus of variations* investigates functions of functions which are called *functional*s.

Example: Entropy $\mathcal{H}[p(\cdot)] = -\int p(x)\log(p(x))\,\mathrm{d}x$

- The derivatives of functionals are called *variations*.
- Calculus of variations has its origins in the 18th century and the most important result is probably the so-called Euler-lagrange equation

$$C(q) \triangleq \int \underbrace{L(t, q(t), q'(t))}_{\triangleq L(t,x,v)} dt \quad : \quad L_x(t, q_*, q'_*) + \frac{d}{dt} L_v(t, q_*, q'_*) = 0$$

which is the core of Optimal Control Theory.

Variational Methods (II/II)

- In general variational methods, one generally assumes a predetermined form of the argument function, possibly parametric.

$$\text{Quadratic:} \quad q(x) = x^{\mathrm{T}}Ax + b^{\mathrm{T}}x + c$$

or

$$\text{Basis functions:} \quad q(x) = \sum_{i=1}^{N_\phi} w_i \phi(x)$$

Variational Inference

In the case of probabilistic inference, the variational approximation takes the form:

$$q(Z) = \prod_{i=1}^{M} q_i(Z_i)$$

where $Z = \{Z_1, \ldots, Z_M\}$ is a partitioning of the unknown variables.

Variational Inference

Algorithm (Variational Iteration)

Solve the problem iteratively:

1. For $j = 1, \ldots, M$
   - Fix $\{q_i(Z_i)\}_{\substack{i=1\\i\neq j}}^{M}$ to their last estimated values $\{\hat{q}_i(Z_i)\}_{\substack{i=1\\i\neq j}}^{M}$.
   - Find the solution of

$$\hat{q}_j(Z_j) = \arg\max_{q_j} \mathcal{L}(q)$$

2. Repeat 1 until convergence.

VB Example 1 – Linear System Identification

Consider the following linear scalar state-space model

$$x_{k+1} = \theta x_k + v_k, \qquad y_k = \frac{1}{2}x_k + e_k, \qquad \begin{pmatrix} v_k \\ e_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_e^2 \end{pmatrix} \right).$$

- The initial state: $x_0 \sim \mathcal{N}(x_0; \bar{x}_0, \Sigma_0)$.
- θ with prior distribution $\theta \sim \mathcal{N}(\theta; 0, \sigma_\theta^2)$
- The identification problem is now to determine the posterior $p(\theta|y_{0:N})$ using the VB framework.
- We still have some latent variables $x_{0:N} \triangleq \{x_0, \ldots, x_N\}$.
- Note the difference in notation compared to Bishop! The observations are denoted y and the latent variables are given by x.

VB Example 1 – Linear System Identification

With latent variables

$$p(\theta|y_{0:N}) = \int p(\theta, x_{0:N}|y_{0:N})\, \mathrm{d}x_{0:N}$$

There is still no exact form for the joint density $p(\theta, x_{0:N}|y_{0:N})$.

Variational Approximation

- Approximate the posterior $p(\theta, x_{0:N}|y_{0:N})$ as

$$p(\theta, x_{0:N}|y_{0:N}) \approx q_\theta(\theta) q_x(x_{0:N})$$

- Find $q_\theta(\theta)$ and $q_x(x_{0:N})$ using

$$\begin{aligned} \log q_\theta(\theta) &= E_{q_x}\left[\log p(y_{0:N}, x_{0:N}, \theta)\right] + \text{const.} \\ \log q_x(x_{0:N}) &= E_{q_\theta}\left[\log p(y_{0:N}, x_{0:N}, \theta)\right] + \text{const.} \end{aligned}$$

VB Example 1 – Linear System Identification

Variational Bayes formulas are

$$\log q_\theta(\theta) = E_{q_x}\left[\log p(y_{0:N}, x_{0:N}, \theta)\right] + \text{const.}$$
$$\log q_x(x_{0:N}) = E_{q_\theta}\left[\log p(y_{0:N}, x_{0:N}, \theta)\right] + \text{const.}$$

We have the joint density $p(y_{0:N}, x_{0:N}, \theta)$ as

$$p(y_{0:N}, x_{0:N}, \theta) = p(y_{0:N}|x_{0:N})p(x_{1:N}|x_{0:N-1}, \theta)p(x_0)p(\theta)$$
$$= \prod_{i=0}^{N} p(y_i|x_i) \prod_{i=1}^{N} p(x_i|x_{i-1}, \theta)p(x_0)p(\theta)$$

Taking the logarithm and separating the constant terms

$$\log p(y_{0:N}, x_{0:N}, \theta) = -\sum_{i=0}^{N} \frac{0.5}{\sigma_e^2}(y_i - 0.5x_i)^2 - \sum_{i=1}^{N} \frac{0.5}{\sigma_v^2}(x_i - \theta x_{i-1})^2$$
$$- 0.5/\sigma_0^2(x_0 - \bar{x}_0)^2 - 0.5/\sigma_\theta^2\theta^2 + \text{const.}$$

VB Example 2 – Gaussian Mixture Identification

Back to the Bishop's notation: x now denotes a measurement.

- Suppose we have $x_{1:N}$ i.i.d. and distributed as

$$x_i \sim p(x|\pi_{1:K}, \mu_{1:K}, \Lambda_{1:K}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(x; \mu_k, \Lambda_k^{-1}\right)$$

- In the Bayesian framework, all the unknowns $\{\pi_{1:K}, \mu_{1:K}, \Lambda_{1:K}\}$ are random.

$$\pi_{1:K} \sim \text{Dir}(\pi_{1:K}|\alpha_0) \overset{\triangle}{\propto} \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1}$$
$$\mu_{1:K}, \Lambda_{1:K} \sim p(\mu_{1:K}, \Lambda_{1:K}) \triangleq \prod_{k=1}^{K} \mathcal{N}(\mu_k; m_0, (\beta_0\Lambda_k)^{-1})\mathcal{W}(\Lambda_k|W_0, \nu_0)$$

VB Example 2 – Gaussian Mixture Identification

- Define the latent variables $z_i \triangleq [z_{i1}, \cdots, z_{iK}]^\mathrm{T}$ as in EM. Then

$$p(x_{1:N}, z_{1:N}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \pi_k^{z_{ik}} \mathcal{N}\left(x; \mu_k, \Lambda_k^{-1}\right)^{z_{ik}}$$

- The Bayesian framework then asks for the posterior density $p(z_{1:N}, \pi_{1:K}, \mu_{1:K}, \Lambda_{1:K}|x_{1:N})$.

Variational Approximation

- Approximate the posterior as

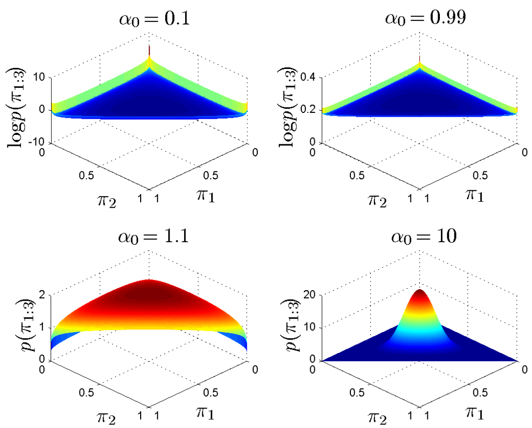
$$p(z_{1:N}, \pi_{1:K}, \mu_{1:K}, \Lambda_{1:K}|x_{1:N}) \approx q_z(z_{1:N})q_{\pi,\mu,\Lambda}(\pi_{1:K}, \mu_{1:K}, \Lambda_{1:K})$$

- Find $q_z(z_{1:N})$ and $q_{\pi,\mu,\Lambda}(\pi_{1:K}, \mu_{1:K}, \Lambda_{1:K})$ iteratively.

VB Example 2 – Sparsity with Bayesian Methods

Symmetric Dirichlet distribution for $K = 3$.

$$\pi_{1:3} \sim \text{Dir}(\pi_{1:3}|\alpha_0)$$
$$\overset{\triangle}{\propto} \prod_{k=1}^{3} \pi_k^{\alpha_0 - 1}$$
$$= (\pi_1\pi_2(1 - \pi_1 - \pi_2))^{\alpha_0 - 1}$$



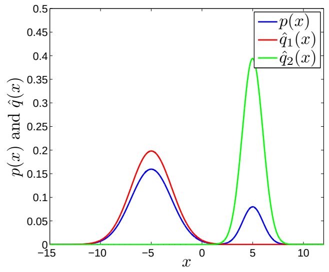
Minimization of KL-divergence (I/III)

Suppose we have

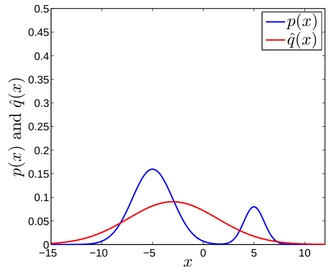
$$p(x) = 0.2\mathcal{N}(x;5,1) + 0.8\mathcal{N}(x,-5,2^2)$$

Let $q_{\mu,\sigma}(x) \triangleq \mathcal{N}(x;\mu,\sigma^2)$

Find $\min_{\mu,\sigma} \mathrm{KL}(q_{\mu,\sigma}||p)$

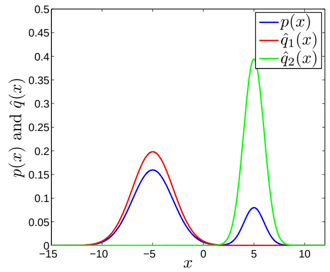


Find $\min_{\mu,\sigma} \mathrm{KL}(p||q_{\mu,\sigma})$



Minimization of KL-divergence (II/III)

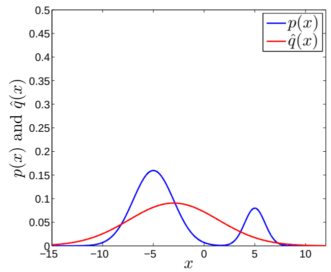
Find $\min_{\mu,\sigma} \mathrm{KL}(q_{\mu,\sigma}||p)$



$$\mathrm{KL}(q_{\mu,\sigma}||p) \triangleq \int q_{\mu,\sigma}(x) \log \frac{q_{\mu,\sigma}}{p(x)}\, \mathrm{d}x$$

zero–forcing

Find $\min_{\mu,\sigma} \mathrm{KL}(p||q_{\mu,\sigma})$



$$\mathrm{KL}(p||q_{\mu,\sigma}) \triangleq \int p(x) \log \frac{p(x)}{q_{\mu,\sigma}}\, \mathrm{d}x$$

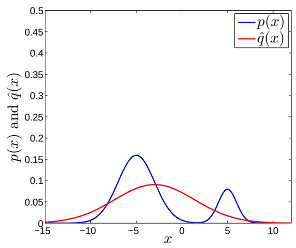
non-zero-forcing

Minimization of KL-divergence (III/III)

This second form of optimization

$$\mathrm{KL}(p||q_{\mu,\sigma}) \triangleq \int p(x) \log \frac{p(x)}{q_{\mu,\sigma}}\, \mathrm{d}x$$

has the following attractive property.



$$\hat{\mu} = E_{\hat{q}}(x) = E_p(x)$$
$$\hat{\sigma}^2 = E_{\hat{q}}\left[(x - E_{\hat{q}}(x))^2\right] = E_p\left[(x - E_p(xx^{\mathrm{T}}))^2\right]$$

- Similar properties hold for the entire exponential family.
- A variational method using this type of KL-divergence minimization and hence the expectation equations above is *Expectation Propagation*.

Expectation Propagation (I/II)

- Suppose we have a posterior distribution in the form of

$$p(X|Y) \propto \prod_{i=1}^{I} f_i(X)$$

  which is intractable or too computationally costly to compute.
- Then EP approximates the posterior as

$$p(X|Y) \approx q(X) \triangleq \prod_{i=1}^{I} q_i(X) = \prod_{i=1}^{I} \mathcal{N}(X;\mu_i,\Sigma_i)$$

- Ideally we want to minimize the KL divergence between the true posterior and the approximation,

$$\hat{q}(X) = \arg\min_{q} \mathrm{KL}\left(\frac{1}{Z}\prod_{i=1}^{I} f_i(X) || \prod_{i=1}^{I} q_i(X)\right)$$

Expectation Propagation (II/II)

Solving this is intractable, make the approximation that we minimize the KL divergence between pairs of factors $f_i(X)$ and $q_i(X)$.

- The terms $q_j(x_j)$ are estimated iteratively as in VB by keeping the last estimates of $\{\hat{q}_i\}_{\substack{i=1\\i\neq j}}^{I}$.

$$\hat{q}_j(X) = \arg\min_{q_j} \mathrm{KL}\left(f_j(X)\prod_{i\neq j}\hat{q}_i(X)\middle\|q_j(X)\prod_{i\neq j}\hat{q}_i(X)\right)$$

- This is in the Gaussian case obtained by solving the equations

$$\begin{aligned}E_{q_j\prod_{i\neq j}\hat{q}_i}(X) &= E_{f_j\prod_{i\neq j}\hat{q}_i}(X)\\ E_{q_j\prod_{i\neq j}\hat{q}_i}(XX^{\mathrm{T}}) &= E_{f_j\prod_{i\neq j}\hat{q}_i}(XX^{\mathrm{T}})\end{aligned}$$

for the mean μ_i and the covariance Σ_i of $\hat{q}_i(\cdot)$.

EP Example – Smoothing under GM noise

Consider the following linear scalar state-space model

$$\begin{aligned}x_{k+1} &= x_k + v_k, & x_0 &= 0 \text{ is known}\\ y_k &= x_k + e_k, & v_k &\sim \mathcal{N}(v_k; 0, \sigma_v^2)\end{aligned}$$

$$e_k \sim p_e(e_k) \triangleq 0.9\mathcal{N}(e_k; 0, \sigma_e^2) + 0.1\mathcal{N}(e_k; 0, (10\sigma_e)^2)$$

- The problem is to obtain the posterior density $p(x_{1:N}|y_{1:N})$.
- The true posterior factorizes as

$$p(x_{1:N}|y_{1:N}) \propto \prod_{i=1}^{N} p(y_i|x_i)p(x_i|x_{i-1})$$

- The true posterior in this case is a Gaussian mixture with 2^N components which is not feasible to compute.

EP Example – Smoothing under GM noise

- Make the variational approximation

$$p(x_{1:N}|y_{1:N}) \approx q(x_{1:N}) \triangleq \prod_{i=1}^{N}\mathcal{N}(x_i; \mu_i, \sigma_i^2)$$

- Consider the density for x_j given as

$$\begin{aligned}\bar{p}(x_j) \propto \int\int & p(y_j|x_j)p(x_{j+1}|x_j)p(x_j|x_{j-1})\\ &\times \mathcal{N}(x_{j+1}; \mu_{j+1}, \sigma_{j+1}^2)\mathcal{N}(x_{j-1}; \mu_{j-1}, \sigma_{j-1}^2)\, \mathrm{d}x_{j+1}\, \mathrm{d}x_{j-1}\end{aligned}$$

which can be calculated as

$$\begin{aligned}\bar{p}(x_j) =& w_1(\mu_{j\pm1}, \sigma_{j\pm1})\mathcal{N}\left(x_j; \eta_1(\mu_{j\pm1}, \sigma_{j\pm1}), \rho_1^2(\mu_{j\pm1}, \sigma_{j\pm1})\right)\\ &+ w_2(\mu_{j\pm1}, \sigma_{j\pm1})\mathcal{N}\left(x_j; \eta_2(\mu_{j\pm1}, \sigma_{j\pm1}), \rho_2^2(\mu_{j\pm1}, \sigma_{j\pm1})\right)\end{aligned}$$

EP Example – Smoothing under GM noise

$$\begin{aligned}\bar{p}(x_j) =& w_1(\mu_{j\pm1}, \sigma_{j\pm1})\mathcal{N}\left(x_j; \eta_1(\mu_{j\pm1}, \sigma_{j\pm1}), \rho_1^2(\mu_{j\pm1}, \sigma_{j\pm1})\right)\\ &+ w_2(\mu_{j\pm1}, \sigma_{j\pm1})\mathcal{N}\left(x_j; \eta_2(\mu_{j\pm1}, \sigma_{j\pm1}), \rho_2^2(\mu_{j\pm1}, \sigma_{j\pm1})\right)\end{aligned}$$

where the parameters $w_{1,2}, \eta_{1,2}$ and $\rho_{1,2}$ are

$$\begin{aligned}\eta_1 &= \rho_1^2\left(\frac{\bar{\eta}}{\bar{\rho}^2} + \frac{y_j}{\sigma_e^2}\right) & \eta_2 &= \rho_2^2\left(\frac{\bar{\eta}}{\bar{\rho}^2} + \frac{y_j}{(10\sigma_e)^2}\right)\\ \rho_1^2 &= \left(\frac{1}{\bar{\rho}^2} + \frac{1}{\sigma_e^2}\right)^{-1} & \rho_2^2 &= \left(\frac{1}{\bar{\rho}^2} + \frac{1}{(10\sigma_e)^2}\right)^{-1}\\ w_1 &\propto 0.9\mathcal{N}\left(y_j; \bar{\eta}, \bar{\rho}^2 + \sigma_e^2\right) & w_2 &\propto 0.1\mathcal{N}\left(y_j; \bar{\eta}, \bar{\rho}^2 + (10\sigma_e)^2\right)\\ \bar{\eta} &= \bar{\rho}^2\left(\frac{\mu_{j-1}}{\sigma_{j-1}^2 + \sigma_v^2} + \frac{\mu_{j+1}}{\sigma_{j+1}^2 + \sigma_v^2}\right) & \bar{\rho}^2 &= \left(\frac{1}{\sigma_{j-1}^2 + \sigma_v^2} + \frac{1}{\sigma_{j+1}^2 + \sigma_v^2}\right)^{-1}\end{aligned}$$

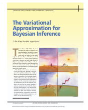
EP Example – Smoothing under GM noise

$$\bar{p}(x_j) = w_1(\mu_{j\pm 1}, \sigma_{j\pm 1})\mathcal{N}\left(x_j; \eta_1(\mu_{j\pm 1}, \sigma_{j\pm 1}), \rho_1^2(\mu_{j\pm 1}, \sigma_{j\pm 1})\right)$$
$$+ w_2(\mu_{j\pm 1}, \sigma_{j\pm 1})\mathcal{N}\left(x_j; \eta_2(\mu_{j\pm 1}, \sigma_{j\pm 1}), \rho_2^2(\mu_{j\pm 1}, \sigma_{j\pm 1})\right)$$

The EP solution for $q_j(x_j) = \mathcal{N}(x_j; \mu_j, \sigma_j^2)$ is obtained by matching (propagating) expectations between $q_j(\cdot)$ and $\bar{p}(x_j)$.

$$\mu_j = w_1\eta_1 + w_2\eta_2$$
$$\sigma_j^2 = w_1\left(\rho_1^2 + (\eta_1 - \mu_j)^2\right) + w_2\left(\rho_2^2 + (\eta_2 - \mu_j)^2\right)$$

References




- Tzikas, D.G.; Likas, A.C.; Galatsanos, N.P.; , "The variational approximation for Bayesian inference," IEEE Signal Processing Magazine, vol.25, no.6, pp.131-146, November 2008. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4644060&isnumber=4644043
- Seeger, M.W.; Wipf, D.P.; , "Variational Bayesian Inference Techniques," IEEE Signal Processing Magazine, vol.27, no.6, pp.81-91, Nov. 2010. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5563102&isnumber=5563096
- Beal, M.J.; Variational Algorithms for Approximate Bayesian Inference, PhD Thesis, University College London, UK, 2003. http://www.cse.buffalo.edu/faculty/mbeal/papers/beal03.pdf
- Minka, T.; , A Family of Algorithms for Approximate Bayesian Inference, PhD Thesis, Massachusetts Institute of Technology, 2001. http://research.microsoft.com/en-us/um/people/minka/papers/ep/minka-thesis.pdf

A Few Concepts to Summarize Lecture 7

Support vector machines: A discriminative classifier that gives the maximum margin decision boundary.

Variational Inference: Approximate Bayesian inference where factorial approximations are made on the form of the posteriors.

Kullback-Leibler (KL) Divergence: A cost function to find optimal approximations for the posteriors in two different forms.

Variational Bayes: A form of variational inference where $\mathrm{KL}(q||p)$ is used for the optimization.

Expectation Propagation: A form of variational inference where $\mathrm{KL}(p||q)$ is used for the optimization.