Expectation Propagation for Approximate Bayesian Inference

José Miguel Hernández Lobato

Universidad Autónoma de Madrid, Computer Science Department

February 5, 2007

Bayesian Inference

Inference

- Given some data we want to assign probabilities to a set of hypothesis.
- Uncertainty is involved in the whole process.
- The tool to reason under uncertainty is Probability Theory.

Bayes Theorem

- $\mathcal{P}(\theta|\mathcal{D},\mathcal{M}) = \frac{\mathcal{P}(\mathcal{D}|\theta,\mathcal{M})\mathcal{P}(\theta|\mathcal{M})}{\mathcal{P}(\mathcal{D}|\mathcal{M})}$.
- $\mathcal{M}$ represents our assumptions (model) for the problem.
- θ is a hypothesis.

Main difficulties in the Bayesian framework

We have to work with complex integrals

- To normalize distributions:

$$\mathcal{P}(\mathcal{D}|\mathcal{M}) = \int \mathcal{P}(\mathcal{D}|\theta, \mathcal{M})\mathcal{P}(\theta|\mathcal{M})\,. \tag{1}$$

- To make predictions:

$$\mathcal{P}(y|\mathcal{D}, \mathcal{M}) = \int \mathcal{P}(y|\theta)\mathcal{P}(\theta|\mathcal{D}, \mathcal{M})d\theta\,. \tag{2}$$

Approximate Solutions

- Laplace's method.
- Monte Carlo methods.
- Variational Inference.
- Expectation Propagation.

Kullback-Leibler Divergence

- $D_{KL}(p\|q) = \int p(x) log \frac{p(x)}{q(x)} dx$
- Is a distance measure from a true density p to an another density q.
- $D_{KL}(p\|q) = 0 \Leftrightarrow p = q$, otherwise $D_{KL}(p\|q) > 0$.
- It is not symmetric $D_{KL}(p\|q) \neq D_{KL}(q\|p)$.

- We can approximate p with a simpler density q by minimizing $D_{KL}(p\|q)$ (direct) or $D_{KL}(q\|p)$ (inverse).

Non-symmetry of the KL divergence I

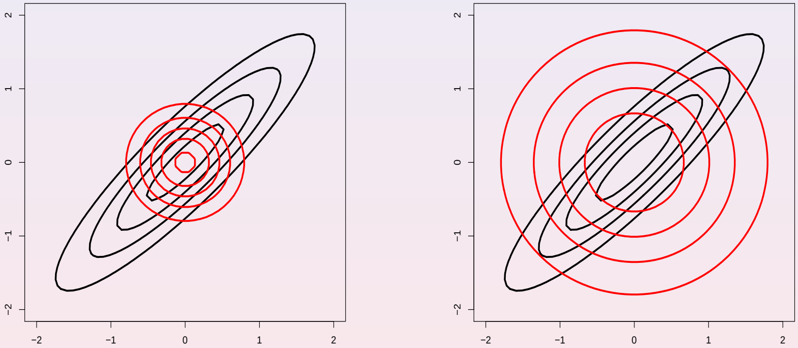

Figure: Inverse solution (left) and direct solution (right) for an approximation of a bivariate Gaussian with two independent Gaussian components.

Non-symmetry of the KL divergence II

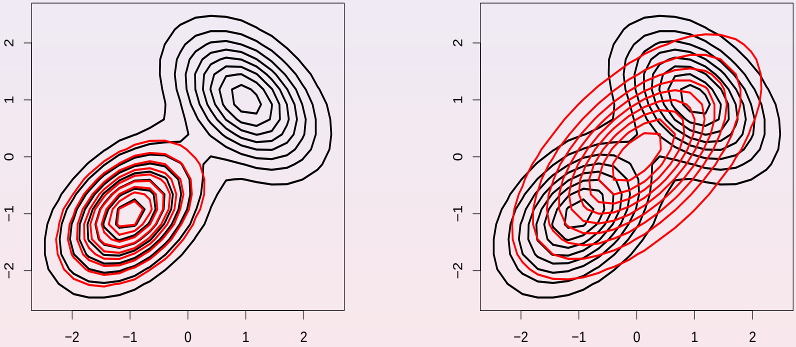

Figure: Inverse solution (left) and direct solution (right) for an approximation of a mixture of two Gaussians with a Bivariate Gaussian.

Exponential Family

- A density q is in the exponential family if

$$q(x) = exp\{\sum_i g_i(x)\nu_i\}\,. \tag{3}$$

- ν_i are the natural parameters.
- g_i are the sufficient statistics: $(1, x, x^2)$ for a Gaussian.
- Exponential families are closed under multiplication.

- We can minimize $D_{KL}(p\|q)$ if q is in an exponential family just by

$$\forall i, \int g_i q(x)dx = \int g_i p(x)dx\,. \tag{4}$$

Assumed Density Filtering

- We write $p(x)$ as a product of terms $p(x) = \prod_{i=1}^{n} t_i(x)$.
- We approximate p term by term with q.

Algorithm

```
1 $q_0(x) \leftarrow constant$
2 for $i \leftarrow 1$ to $n$
    1 $Z_i \leftarrow \int q_{i-1}(x) t_i(x) dx$ .
    2 $q_i \leftarrow min_q \, D_{KL}(\frac{1}{Z_i} q_{i-1} t_i \| q)$ .
3 $Z \leftarrow \prod_{i=1}^{n} Z_i$
4 Return $q_n$ and $Z$
```

- q_n approximates $\frac{p(x)}{\int p(x)dx}$ and Z approximates $\int p(x)dx$.

Main disadvantage of ADF

Problem

- The solution depends on the processing order of the $t_i(x)$.

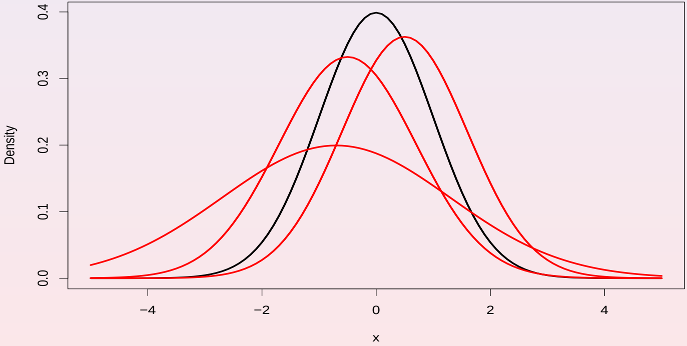

Figure: $p(x)$ is shown in black and the approximations obtained by ADF using different orderings are shown in red.

Expectation Propagation

- Solves the ordering dependence of ADF.
- Approximates $p(x) = \prod_{i=1}^{n} t_i(x)$ by $q(x) = \prod_{i=1}^{n} \hat{t}_i(x)$ where all $\hat{t}_i$ are in the same exponential family and so is q.
- Each $\hat{t}_j$ term is chosen so that

$$q(x) = \hat{t}_j(x) \prod_{i \neq j} \hat{t}_i(x) \tag{5}$$

is as close as possible to

$$t_j(x) \prod_{i \neq j} \hat{t}_i(x)\,. \tag{6}$$

- The distance measure used is the direct K-L divergence.
- It is easy to work with q and it can be integrated automatically.

Pseudocode of Expectation Propagation

Algorithm

```
1. Initialize all $\hat{t}_i$ and $q$ to constant densities.
2. Until all $\hat{t}_i$ converge:
    1. Choose a $\hat{t}_i$ to update.
    2. $q_{old} \leftarrow \frac{q}{\hat{t}_i}$ .
    3. $q \leftarrow min_{q'} D_{KL}(q_{old} t_i \| q')$ .
    4. $\hat{t}_i \leftarrow \frac{q}{q_{old}}$ .
3. Return $q$ .
```

- $\int p(x)dx$ is approximated by $\int q(x)dx$
- $\frac{p(x)}{\int p(x)dx}$ is approximated by $\frac{q(x)}{\int q(x)dx}$.

Final Notes on Expectation Propagation

- The $\hat{t}_i$ terms and therefore q could also be factorized densities.
- In this case we only have to perform some marginalizations in the algorithm.

Advantages of Expectation Propagation

- No local minima minimizing the K-L divergence.
- Applicable to high dimensional densities.
- Usually faster than other approaches.

Disadvantages of Expectation Propagation

- It is not guaranteed to converge.
- q_{old} might not be a proper density.
- The t_i terms have to be simple.

Bayes Machine I

- It is a Bayesian single layer perceptron.

- $\mathbf{w}$ determines the hyperplane of a perceptron.
- Given a data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, $y \in \{-1, 1\}$, the likelihood for $\mathbf{w}$ is

$$\mathcal{P}(\mathcal{D}|\mathbf{w}) = \prod_i \mathcal{P}(y_i|\mathbf{w}) = \prod_i \Theta(y_i \mathbf{w}^T \mathbf{x}_i)\,, \tag{7}$$

  where Θ is the step function.
- We can take into account a labeling error rate ϵ

$$\mathcal{P}(y_i|\mathbf{w}) = \epsilon + (1 - 2\epsilon)\Theta(y_i \mathbf{w}^T \mathbf{x}_i)\,. \tag{8}$$

- The likelihood only depends on the number of errors.

Bayes Machine II

- The prior for $\mathbf{w}$ is $\mathcal{P}(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, a spherical Gaussian.
- The posterior for $\mathbf{w}$ is $\mathcal{P}(\mathbf{w}|\mathcal{D}) \propto \prod_i \mathcal{P}(y_i|\mathbf{w})\mathcal{P}(\mathbf{w})$.
- The predictive distribution for a point $\mathbf{x}$ is

$$\mathcal{P}(y|\mathbf{x}, \mathcal{D}) = \int_{\mathbf{w}} \mathcal{P}(y|\mathbf{x}, \mathbf{w})\mathcal{P}(\mathbf{w}|\mathcal{D})\,. \tag{9}$$

- The model evidence is

$$\mathcal{P}(\mathcal{D}|\mathcal{M}) = \int_{\mathbf{w}} \prod_i \mathcal{P}(y_i|\mathbf{w})\mathcal{P}(\mathbf{w})\,. \tag{10}$$

- We approximate the posterior $\mathcal{P}(\mathbf{w}|\mathcal{D})$ with a multivariate Gaussian $\mathcal{N}(\mu, \Sigma)$ by means of EP.
- EP also approximates the evidence.

Example of a Bayes Machine

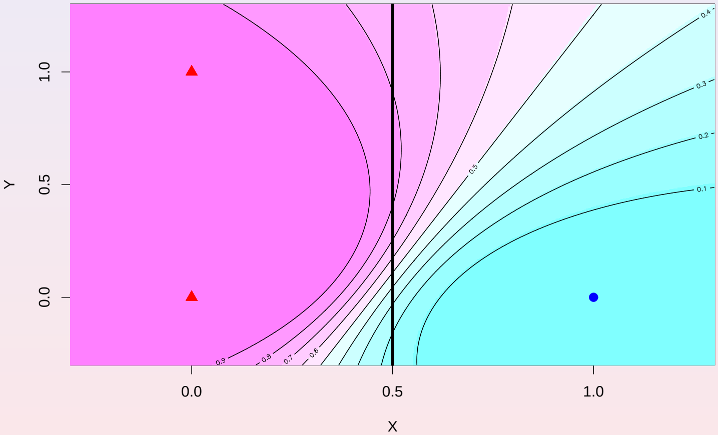

Figure: Contour plot of the decision surface obtained by the Bayes machine and maximum margin classifier in black. The Bayes machine approximates a vote between all possible linear separators. In this example $\epsilon = 0$.

Non-linear Bayes Machine

- It is possible to rewrite the whole EP algorithm for the Bayes Machine in terms of inner products.
- The *kernel trick* allows us to work with the data projected on an infinite dimension space where it can be linearly separated.

- We can fix the kernel and its parameters just by maximizing the approximation for the evidence.
- The same procedure can be used to perform selection of attributes.

Example: the Spiral Dataset with 100 points

- We fix a Gaussian kernel $exp(-\frac{1}{2\sigma^2}(\mathbf{x_i}-\mathbf{x_j})^T(\mathbf{x_i}-\mathbf{x_j}))$ and maximize $log(\mathcal{P}(\mathcal{D}|\mathcal{M}))$ with respect to σ.

σ	$log(\mathcal{P}(\mathcal{D}\|\mathcal{M}))$
1	-33
0.5	-25
0.25	-22.6
0.1	-34
0.35	-22.7
0.3	-22.4

The Non-linear Bayes Machine on the Spiral Data Set I

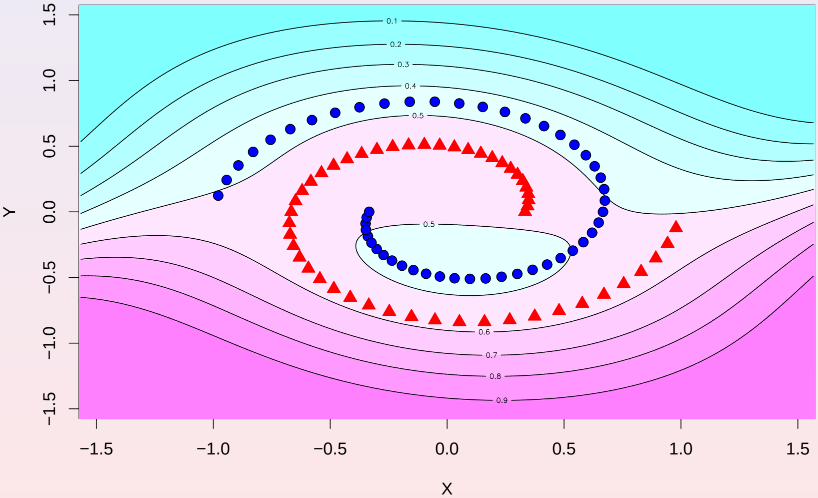

Figure: Contour plot of the decision surface obtained by the non-linear Bayes machine. In this example $\sigma^2 = 1$ and $\epsilon = 0$.

The Non-linear Bayes Machine on the Spiral Data Set II

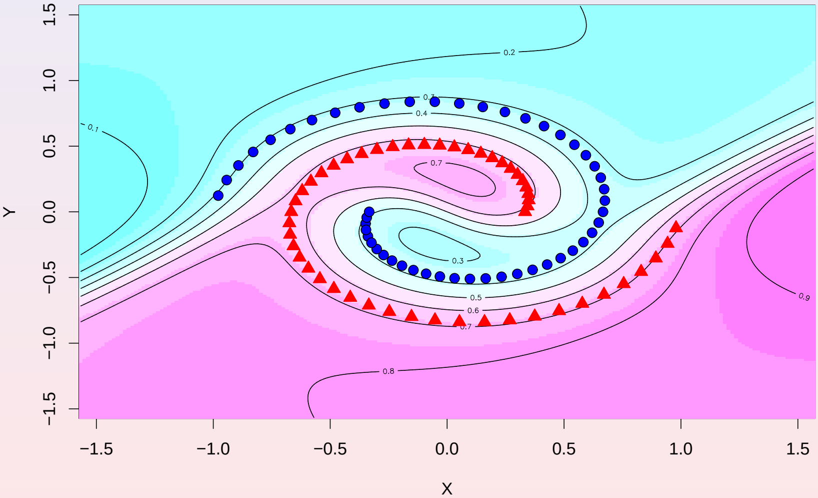

Figure: Contour plot of the decision surface obtained by the non-linear Bayes machine. In this example $\sigma^2 = 0.3$ and $\epsilon = 0$.

Results of a SVM on the Spiral Data Set

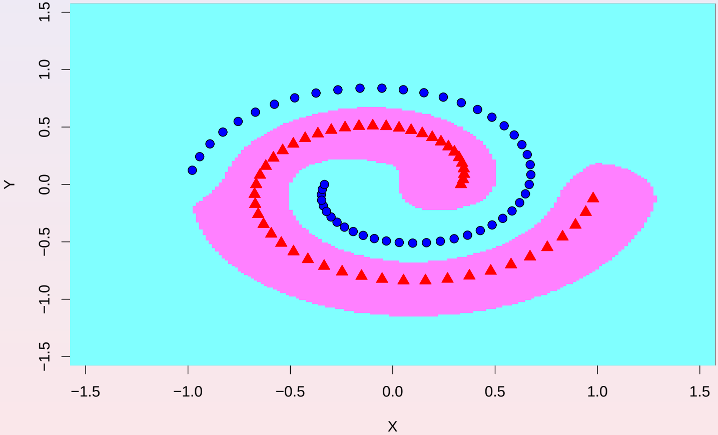

Figure: Decision surface obtained by a support vector machine. We used a Gaussian kernel and $\epsilon = 0$. The with of the kernel was selected by cross validation.

Results of a SVM on another Data Set

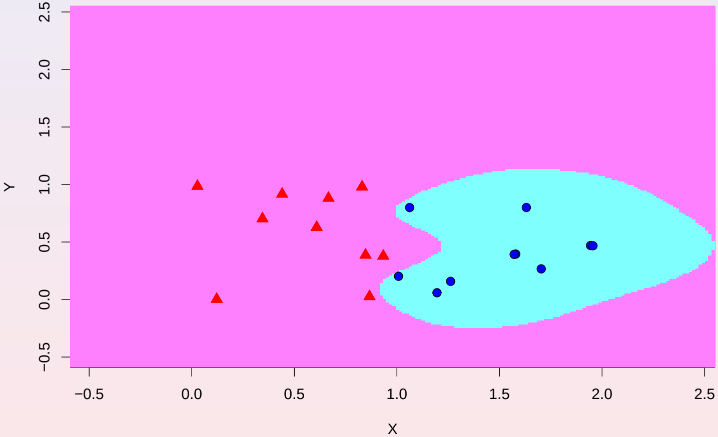

Figure: Decision surface obtained by a support vector machine. We used a Gaussian kernel and $\epsilon = 0$. The with of the kernel was selected by cross validation.

Results of the Bayes Machine on another Data Set

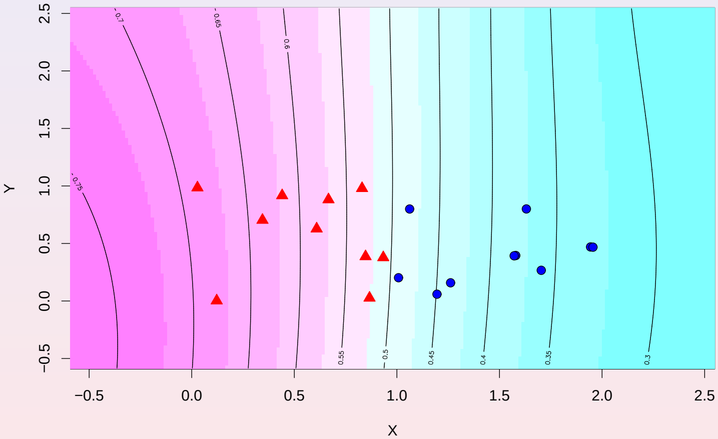

Figure: Decision surface obtained by the non-linear Bayes machine. We used a Gaussian kernel and $\epsilon = 0$. The with of the kernel was selected by maximizing the approximation for the evidence.

Bayes Machines vs SVMs

SVMs

- Generally lead to less smooth decision borders.
- To tune their parameters you have to use CV and discard data.
- Training process is faster.
- Prediction is faster because they only use the support vectors.
- They do not output probabilities directly.

Bayes Machines

- Seem to generalize better (see ref 1).
- The kernel parameters can be fixed using all data.
- Training is slow, $O(n^3)$.
- Prediction is slow.
- They output a predictive distribution.

Bibliography

1. T. Minka. A family of algorithms for approximate Bayesian inference. PhD thesis, MIT Media Lab, 2001.
2. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer (2006).
3. David MacKay. Information Theory, Inference, and Learning Algorithms. (Available on the web).