**Machine Intelligence II** **Exercise Sheet 1**
SS 10, Prof. Obermayer due to 29.4.2010

## Bayesian Networks & Inference

In this exercise we will create a Bayesian network and answer a probabilistic query about network variables applying belief propagation (*message passing*). The second block of this exercise consists of a Bayesian linear regression task. Use Matlab or Python for the programming exercises.

***Additional Material:*** You can find the links to download the *BayesNetToolbox* for Matlab and the *OpenBayes* package for Python as well as template code on the ISIS platform.

### 1.1 Bayesian Network Example

(6 points)

- Create a *DAG* using the node variables $B$ (Burglary), $E$ (Earthquake), $A$ (Alarm), $R$ (Radio broadcast) and the following (conditional) probabilities:

  $$P(B) = 0.01, P(E) = 10^{-6}, P(R|E = false) = 0, P(R|E = true) = 1$$

  | $B$ | $E$ | $P(A)$ |
  |---|---|---|
  | f | f | 0.001 |
  | f | t | 0.41 |
  | t | f | 0.95 |
  | t | t | 0.98 |

  Our question is: What is the probability for a burglary, given the alarm goes off and the radio broadcasts an earthquake warning? Formally: $P(B|A = true, R = true)$

  In order to answer this question construct a *junction tree* based on the *DAG*, initialize *clique* and *separator* potentials, introduce the evidence, perform *message passing* and calculate the marginal probability. (4 points)

- Verify your calculations using the *BayesNetToolbox* for Matlab or the *OpenBayes* package for Python. When using Matlab you can implement the network by applying the functions `mk_bnet`, `draw_graph` (to plot the *DAG*), `tabular_CPD`, `jtree_inf_engine`, `enter_evidence`, and `marginal_nodes`. (2 points)

### 1.2 Bayesian Regression

(4 points)

Apply sequential Bayesian regression to a linear model with polynomial basis functions. Matlab template code in the `bayes_regression.zip` archive is available for download. You can either complete the template program `bayes_polyfit.m` or write your own code in Matlab or Python.

*Josef Ladenbauer:* `jl@cs.tu-berlin.de`

- Generate a dataset containing $N = 50$ data points ($x$,$t$), where $x$ is drawn uniformly and independently from the interval [-1, 1] and $t = h(x) + \epsilon$ is given by the deterministic function $h(x) = -0.25x^2 + 0.75x^3$ and a zero mean Gaussian random variable $\epsilon$ with precision (inverse variance) $\beta = 25$.
  Use the $M = 2$ basis functions $\phi_1(x) = x^2$, $\phi_2(x) = x^3$ to transform your input data. (1 point)

- Implement sequential Bayesian learning. We assume a Gaussian prior distribution for the weights of the linear model $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$, with mean $\mathbf{m}_0 = \mathbf{0}$ and covariance (inverse precision) matrix $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$ where $\mathbf{I}$ denotes the $M \times M$ identity matrix. For the likelihood function we obtain the following expression:

  $$p(\mathbf{t}|\mathbf{w}) \sim \prod_i \mathcal{N}(t_i|\mathbf{w}^T\phi(x_i), \beta^{-1})$$

  Derive the updates $\mathbf{m}_n, \mathbf{S}_n$ of posterior distribution using the following result for Gaussians:

  Given

  $$p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|\mu, \mathbf{\Lambda}^{-1}), \qquad p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1})$$

  we obtain for the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$

  $$p(\mathbf{x}|\mathbf{y}) \sim \mathcal{N}(\mathbf{x}|\mathbf{\Gamma}\{\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \mathbf{\Lambda}\mu\}, \mathbf{\Gamma})$$

  where $\mathbf{\Gamma} = (\mathbf{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}$.

  For sequentially arriving data points, the posterior can act as the prior for subsequent data points. (1 point)

- Apply Bayesian sequential learning using the dataset. During learning the posterior distribution $p(\mathbf{w}|\mathbf{t})$ changes after the presentation of each training data point. Draw 5 weight vectors randomly from the posterior distributions corresponding to the $n$-th learning sequence for $n = 1, 5, 25$ and visualize the model output. Investigate how the model output changes in dependence of n, i.e. as more and more data is used to determine $p(\mathbf{w}|\mathbf{t})$, and show the output for $n = N$ using the *maximum a posteriori* weight vector $\mathbf{w}_{MAP}$. Carry out this analysis for $\alpha = 0.01, 1, 100$. Interpret the results. (2 points)

**Total points: 10**