Probabilistic Graphical Models

Christoph Lampert <chl@ist.ac.at>
TA: all of us <pgm_2016@lists.ist.ac.at>

Exercise Sheet 6 v1.0

1 Latent variables (Barber, Exercise 11.9)

A 2×2 table of probabilities, $p(x_1 = i, x_2 = j) = \theta_{ij}$, with (of course) $\theta_{ij} \in [0, 1]$ and $\sum_{ij} \theta_{ij} = 1$, should be learned. Assume that x_2 is never observed, so one aims at *maximizing the marginal likelihood* of x_1.

a) Show that if $\theta = \begin{pmatrix} 0.3 & 0.3 \\ 0.2 & 0.2 \end{pmatrix}$ is a global maximum of the marginal likelihood solution, then $\theta = \begin{pmatrix} 0.2 & 0.4 \\ 0.4 & 0.0 \end{pmatrix}$ is one as well.

b) Construct a 2-dimensional family of probability tables that are all global maxima.

c) Which of the distributions in the family has maximal entropy?

2 Loss Functions

For a probability distribution $p(y)$ and a loss function $\Delta(\bar{y}, y)$, the optimal prediction rule is

$$f(x) = \operatorname*{argmin}_{y \in \mathcal{Y}} \; \mathbb{E}_{\bar{y} \sim p(\bar{y})}[\Delta(\bar{y}, y)].$$

a) Construct a distribution such that the optimal predictions differ depending whether the loss function is the 0/1-loss, $\Delta_{0/1}(\bar{y}, y) = [\![\bar{y} \neq y]\!]$, or the Hamming loss, $\Delta_H(\bar{y}, y) = \sum_i [\![\bar{y}_i \neq y_i]\!]$.

b) Show that this cannot happen if the distribution is so peaked that $\max_{y \in \mathcal{Y}} p(y) > 0.5$.

3 CRFs with latent variables

In the lecture we saw two possibilities to train a CRF with latent variables: by 1) gradient descent on the non-convex conditional likelihood (*i.e.* marginalizing out the latent variables), or by 2) expectation maximization. This exercises analyzes the differences.

a) Look up (beware of my typos) or re-derive the expressions for both procedures from the lecture.

b) Which conditions does θ fulfill when either of the methods converged?

c) Someone suggests a new training procedure by alternating between the two methods: run gradient descent until convergence, then run EM until convergence, then gradient descent again, etc. What's your comment?

As part of EM, one has to solve the subproblems of minimizing the variational upper bound. Imagine you do so by gradient descent.

d) What will be the practical difference between the two procedures now?

e) Imagine you want to save time on the subproblem and don't run it to global optimality, but stop the gradient descent procedure after making just one gradient update step. What will change?

f) Having answered a)–e): in which situations would you prefer one of the procedures over the other?

4 Restricted Boltzmann Machines

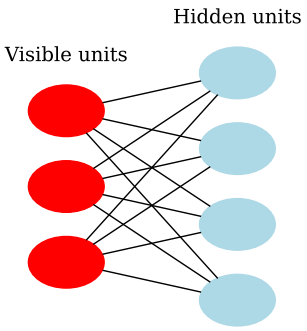

Image: Wikipedia (Qwertyus CC-SA 3.0)

Restricted Boltzmann Machines (RBFs) are a special family probabilistic graphical models in which the underlying graph is *bipartite* between m visible (observable) v and n hidden (latent) variables h, all of them binary, i.e. $\{0, 1\}$-valued.

a) What's the most general form of the corresponding distribution and energy function that is compatible with the graph structure?

Make each entries in the energy a parameter and give it a reasonable names, e.g. θ with suitable indices.

b) Derive the simplest expressions for $p(h|v;\theta)$ and $p(v|h;\theta)$.

c) What is the computational complexity of evaluating these probabilities for fixed h and v?

d) Given a fixed h, how would you *sample* a vector $v \sim p(v|h;\theta)$?

Given observations, $v^1, \ldots, v^N$, we want to train the RBM. As h is not observed, we will maximize the marginal likelihood.

e) Write down the expression for $p(v)$ and compute its gradient with respect to the energy parameters.

f) Suggest a strategy to compute the gradient. As usual, there will be an easy part and a hard part (which ones are these?)

If you're feeling adventurous, or as final course project:

g*) download the USPS dataset, e.g. from

`http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/zip.train.gz`

Ignore the labels (first column in the matrix) and threshold the remaining entries to be $\{0, 1\}$-valued.

Train an RBM with $m = 256$ (16×16 images) and $n = 10$ using stochastic gradient descent.

After training, how would you find out if the RBM did a good job learning the distribution?

Imagine it didn't. How can you tell if the model itself or the learning procedure are to blame?