Machine Learning II – Assignment 4

Stefan Roth, Uwe Schmidt, Tobias Plötz

December 10, 2014

This homework is due January 21, 2015 at 20:00.
Please read the instructions carefully!

Important: For the programming tasks in this assignment, you may use a programming language of your choice. In addition to submitting your solution in Moodle as usual, you will have to demonstrate and explain the code in person; details about this will be announced via Moodle. If your program does not work on the machines in our computer pool, you will have to use your own computer for demonstration. You may not work any further on your code after the submission deadline.

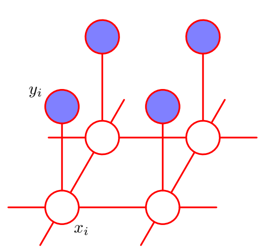
Please note that this assignment draws on many of the concepts and techniques that have been presented throughout the semester, and as such might be more difficult than previous ones.

1 Binary Image Denoising – The Ising Model (5 Points)

You have seen binary image denoising with a Markov Random Field in the lecture, using the so-called Ising model. In this problem, we are given an observed noisy image $\mathbf{y}$ and want to recover the original uncorrupted image $\mathbf{x}$ as best as possible. All pixels $x_i, y_i \in \{-1, +1\}, i = 1, \ldots, N$ are binary random variables and are arranged in a grid in the undirected graphical model. The joint distribution $p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp(-E(\mathbf{x}, \mathbf{y}))$ is defined by its energy

$$E(\mathbf{x}, \mathbf{y}) = h \sum_{i=1}^{N} x_i - \beta \sum_{i \sim j} x_i x_j - \eta \sum_{i=1}^{N} x_i y_i, \tag{1}$$

where $\beta, \eta \geq 0$ and h are the model parameters, and $i \sim j$ denotes the index pairs of all horizontally and vertically neighboring pixels.



1. Explain the role of the three factor types hx_i, $-\beta x_i x_j$, and $-\eta x_i y_i$. For each of the parameters, explain what happens when they are set to 0.

2. What is the effect of setting β much larger than η?

3. Why does it not make sense to allow $\beta, \eta < 0$?

2 MAP Estimation (10 Points)

Download the test pair $(\mathbf{x}_{\text{GT}}, \mathbf{y})$ of original and noisy binary image from here[1]. We first infer a less noisy image by approximately computing the MAP estimate

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \tag{2}$$

We will use *Iterated Conditional Modes* (ICM) to iteratively optimize the energy $p(\mathbf{x}|\mathbf{y})$ of the Ising model. For evaluation, use as loss function the fraction of incorrectly recovered pixels

$$L(\mathbf{u}, \mathbf{v}) = \frac{1}{N} \sum_{i=1}^{N} [u_i - v_i \neq 0], \tag{3}$$

where $[P]$ evaluates to 1 if P is true, and to 0 otherwise.

1. Implement ICM for the Ising model and run it until convergence to get an estimate of $\mathbf{x}^*$. Start from the noisy image $\mathbf{y}$. Your program must show the current state of the restored image after every ICM iteration (update of all pixels). Try to recover $\mathbf{x}_{\text{GT}}$ as best as possible, i.e. try choosing model parameters such that the error (loss) $L(\mathbf{x}_{\text{GT}}, \mathbf{x}^*)$ becomes as small as possible.

2. What are your chosen parameters? Is it easy to find model parameters that achieve good results (low error)?

3. Which is the least important model parameter (i.e. can be set to 0 with little effect)?

4. Try and document different update orderings of the pixels. How much does the update order influence the results?

5. Report your best result (lowest error) in percent and show the denoised image $\mathbf{x}^*$.

6. How may inference with ICM be improved to find $\mathbf{x}^*$ with even higher probability?

[1] http://www.gris.tu-darmstadt.de/teaching/courses/ws1415/ml2/assignments/assignment4_test.zip

3 Sampling (10 Points)

We want to improve on the MAP estimate. Consider the same image pair and the same loss function as in the previous task.

1. What is the Bayes optimal estimator for the given loss function?
2. Why can't we compute the Bayes optimal estimator in closed form?
3. Implement a *Gibbs sampler* to draw samples from $p(\mathbf{x}|\mathbf{y})$; it should update one pixel at a time, while holding all other pixels fixed. Note that you should be able to reuse code that you have already written for the previous task. Your program must show the current state of the sampler each time after updating all pixels.
4. Try different initializations for the sampler (noisy image, random image); do you eventually get similar looking samples? If not, why might this be the case? Plot the energy values of the samples after each iteration through all pixels.
5. Draw a set of samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)} \sim p(\mathbf{x}|\mathbf{y})$ and use all of them to estimate the denoised image with the Bayes optimal estimator.
6. How else could you use the sampler to estimate the restored image?
7. Did you use the same model parameters as in the previous task, or did they lead to poor results? Why might you need different parameters?
8. Again, report your best result (smallest loss in percent) and show the final denoised image.

4 Parameter learning (10 Points)

Up to now, you have tweaked the parameters of the Ising model by hand. This is cumbersome and does not scale with an increasing number of parameters. Therefore, we now want to learn all the parameters by (approximate) maximum likelihood training of $p(\mathbf{x}|\mathbf{y})$. Download training data from here[2].

1. Explain two main difficulties in computing the maximum likelihood estimate of the parameters?
2. Implement (approximate) maximum likelihood learning of all parameters and use gradient descent for optimization.
3. Plot the evolution of all model parameters during learning.
4. Are your final learned model parameters different from your manually chosen ones in the previous tasks?
5. Does inference (ICM or sampling) with the learned parameters lead to better results for the test image?

[2] http://www.gris.tu-darmstadt.de/teaching/courses/ws1415/ml2/assignments/assignment4_train.zip