Learning Models and Methods T-61.5040 (5 cr) L

Spring 2007

Computer Assignment

28.3.2007, Petteri Pajunen

General

To pass the course T-61.5040 requires passing the final exam and an accepted completion of this assignment. You must complete the assignment first, and take the exam afterwards. An exception is the exam in May 2007, which can be taken by everyone. The grade is given for this first exam only if the assignment is returned before deadline and eventually accepted. Regardless of the exam you are going to take, the deadline for the assignment is Wed 23rd May 2007.

Completing the Assignment

The assignment requires developing a solution to the problem described below and implementing it. It is recommended that MATLAB or R is used: the necessary data will be available for both.

The results are written as a report, where you answer *all questions* asked in this paper: these are the bold-faced lines beginning with Q.

Returning the Report

You should return the report to Ville Viitaniemi either by email (Ville.Viitaniemi@hut.fi) in *postscript or PDF format in a single file*, or as a printout to the box next to the bulletin board of CIS laboratory (3rd floor, Computer Science Building). If you return your report as an email attachment, prefix the file name with your student number. You *must* include your current email address and your student number on the report. When the report has been returned and accepted, you may take any exam of the course until January 2008. If you have already passed the May 2007 exam, you will get your grade soon after the assignment has been accepted.

Acceptance of the Work

When you return your report before deadline, it will be either accepted or you will be required to make corrections (a boomerang). In case of a boomerang, you will be given a new deadline for returning the corrected report.

Further Information

If this guide, or any other course material has shortcomings or there are any questions, please ask/comment by email (t615040@cis.hut.fi). Further information, hints, and corrections to these instructions will be updated to the course webpage as necessary. If you get stuck and cannot solve some part of this assignment, ask for advice from t615040@cis.hut.fi

Independence and Conditional Probabilities

The first problem illustrates a certain unintuitive property of conditional probabilities. Note that you need to understand this in order to solve another problem later.

Assume that the variables x_1 and x_2 are the result of tossing a coin. The result is either H or T, and the probability of H is θ (probability of T is therefore $1 - \theta$). Given θ, the results x_1 and x_2 are independent.

Q0a: What is the joint distribution $p(x_1, x_2|\theta)$?

Q0b: What is the joint distribution $p(x_1, x_2)$? Assume that the prior for θ is $p(\theta = 1/3) = 1/2, p(\theta = 2/3) = 1/2$.

Q0c: What is the product of distributions $p(x_1)$ and $p(x_2)$?
Hint: the answers to Q0b and Q0c are not the same.

Q0d: What can you say about the independence of x_1 and x_2 *when θ is not known*? Can you explain the result?

Reconstructing a High-Resolution Image

The main problem in this assignment is to reconstruct a high-resolution gray-scale image using several low-resolution digital photographs (LR-photos). The original image (HR-image) has some identifiable scene/object/etc., but each observed LR-photo is too blurred and low-resolution, so that it is impossible to directly identify what is in the original image. You should be able to reconstruct the image sufficiently accurately so that you can roughly identify what the image represents.

The LR-photos have been generated by simulating a digital camera by reducing the resolution, blurring, and translating the original high-resolution image. The original image has $4 \times 4 = 16$ pixels for every pixel in the LR-photos. Therefore one unit in the low-resolution coordinates corresponds to four units in the high-resolution coordinates.

The LR-photos can be found from

`http://www.cis.hut.fi/Opinnot/T-61.5040/photos.mat`

in MATLAB format and from

`http://www.cis.hut.fi/Opinnot/T-61.5040/photos.R`

in R format. The file contains a matrix of size $(K * px) \times py$, where K is the number of LR-photos There is also a matrix of size $2 \times K$ which contains the amount of translation for each LR-photo. The translation is defined as a two-dimensional vector for each LR-photo. The vector defines the position of each LR-photo relative to origin. You can think of the vector as the coordinates of the upper-left corner pixel of each LR-photo.

The photographs can be taken from the matrix im as follows (MATLAB-notation): the first photograph is

$$im[1 : px, 1 : py]$$

the second is

$$im[(px + 1) : (2 * px), 1 : py]$$

and so on.

The translation vectors are defined in an (x,y)-coordinate system where x increases as the row index of the photograph matrix increases, and y increases as the column index increases. The units of the translation vector are defined by the coordinate system of the original HR-image. Therefore to compute the high-resolution coordinates of pixel A[i,j], where A is a matrix containing one of the LR-photos, one has to multiply i and j by 4, and then add the translation vector components to the values $4i, 4j$. *Note that the translation vectors are already defined in the high-resolution coordinate system: do not multiply them by* 4.

Constructing the Model: Coordinates, Blurring, Translation, and Scaling

The problem seems somewhat complicated but the solution outlined below gives the result in closed form using Normal distributions. The first step is to decide a high-resolution coordinate system and choose the parameters describing the model.

The unknown interesting quantities are naturally the pixel intensities of the original image. It is possible to reconstruct the original image at a resolution of your choice. It is best to develop the solution without fixing the resolution until making experiments. For computational reasons it is recommended that you reconstruct the image at the same resolution as the original image.

Whatever the resolution, you can choose the size of the high-resolution pixel to be 1 in both dimensions. Then the low-resolution pixels will have size D in both dimensions assuming you want to increase resolution by a factor of D. *Remember that the translation vectors are in units corresponding to $D = 4$.*

You need to assume something about the low-resolution photographs. Assume that all the photos are taken so that the x and y axis in the photos are perfectly aligned with the HR coordinate system. In other words, there is no rotation in the photos. However, there is translation: without any movement the LR photos would be identical except for additive noise. It is the different translations that make the reconstruction possible.

In addition to translation, there is blurring. This means that a single pixel in the low-resolution photograph receives some light intensity from the HR image pixels that are close to the LR photograph pixel. The intensity decreases rapidly when the distance between pixels grows.

Intensity scaling can be decided freely because images are typically processed to enhance their visual properties anyway, and such scaling does not change what the image represents. In the solution sketched below, you can assume that the pixel intensities in the HR-image are in the interval $[-0.5, 0.5]$. You will need this when solving the problem Q1 below.

Finally, assume that the LR-photos are corrupted by additive, Normally distributed zero-mean noise. You can assume that the variance of this noise is 0.01.

The Image Prior

To start constructing the Bayesian model, we need to decide on a prior distribution for unknown quantities. The most important unknown quantity is the set of pixel intensities of the unknown HR image. It turns out that using a reasonable prior on the unknown image allows us to reconstruct the image surprisingly well.

In the following, images will be represented as vectors for simplicity of notation. Denote the unknown high-resolution image as a vector s, where each component of s is a pixel intensity. You should choose a prior for s that makes 'natural scenes' most probable. To make the further development of the solution tractable, you should choose a Normal distribution with zero mean. Then the specification of the prior is completed by defining the covariance matrix. Consider carefully what kind of covariances would you expect in an image representing a 'natural scene', as opposed to e.g. noise or an image containing very little detail. Don't choose the prior without thinking: your choice of prior must be reasonable, considering the information given above. Also, scale the covariance matrix properly: this requires careful consideration of your model, especially the intensity scale.

Q1: What is your prior for s? Explain why you chose the covariance matrix as you did.

Other unknowns in the model are parameters describing the process generating the LR photographs from the image. These include the translations $z(k)$ for each LR photo $k \in \{1, 2, \ldots, K\}$ and a parameter β describing the 'blurriness' of the camera lens. You may assume that these have a constant prior for simplicity.

The Likelihood

Next we must construct the likelihood. Denote the LR photographs as vectors $y(k)$, where $k = 1, \ldots, K$ is the number of the photograph. Denote all LR photographs collected in a single vector as y. Construct the likelihood by using the following information:

1. each LR-photo pixel is independent of all other LR-photo pixels, given the HR-image s and all other unknowns
2. the intensity of each LR-photo pixel with coordinates a (in units such that a LR-photo pixel has size 4x4) is obtained as a weighted average of the HR-image pixels with

coordinates b, weighted by the blurring function $0.25\exp(-\beta\|a-b\|^2)$. Zero-mean Normal noise is added to the intensity. You can use a blurriness constant $\beta = 1$.

3. remember to use the translation vectors when computing the coordinates of LR-photo pixels.

To construct the likelihood, it is useful to write $y(k)$ as a linear function of s, i.e.$y(k) = W(k)s$, and add noise to this representation.

Q2: What is the likelihood $p(y|s, z, \beta)$, and what is the matrix $W(k)$?. Hint: it is recommended that you use the blurring function $0.25\exp(-\beta\|a-b\|^2)$.

The HR Image Posterior

Your likelihood should be a Normal distribution, and the prior is also a Normal distribution. Since we know the translations z and we know β, we are interested in the distribution of s conditional to y, z, β.

Q3: Compute the posterior $p(s|y, z, \beta)$. Hint: this posterior is a Normal distribution of s, so it must proportional to an exponential function with an exponent $-\frac{1}{2}(s-m)^T G^{-1}(s-m)$. It is enough to find m and G.

The HR-image can be estimated by simply choosing the mean m of the posterior $p(s|y, z\beta)$ as the MAP-estimate.

Unknown Translations and Blurriness

The problem you are required to solve has been simplified, since you are given the values z and β. In reality, these parameters would be unknown and they should be treated as uncertain quantities. Maximizing the posterior by choosing maximizing values simultaneously for s, z and β leads to overfitting. This can be alleviated by first estimating z and β so that s is marginalized out. Then, using estimated values for z and β, one can proceed as if using known values for these parameters.

In question Q4, you are required to perform the marginalization of s formally, but it is not necessary to use the result to estimate z and β (recall that z and β are given, so you can directly use the values in $p(s|y, z, \beta)$).

The priors of the parameters z, β are proportional to a constant, so their posterior is proportional to the likelihood. We want to find the marginal likelihood

$$\begin{aligned} p(y|z,\beta) &= \int p(y,s|z,\beta)ds \\ &= \int p(y|s,z,\beta)p(s|z,\beta)ds \\ &= \int p(y|s,z,\beta)p(s)ds \end{aligned}$$

The last integral has familiar terms in the integrand. The first is the likelihood computed before, and the second is the image prior. Both are Normal distributions and their product is a Normal distribution of y, s. The integral is a marginal distribution of the integrand, so it is also Normal.

Q4: Compute the marginal likelihood $p(y|z,\beta)$. Hint: iteration formulas are an easy way to find the mean and covariance of the Normal likelihood.

Now you should have a likelihood which is a function of the weights W, noise variance, and the prior covariance. The estimates of z, β can be found by maximizing this likelihood. Since these parameters affect the weights W through the nonlinear blurring function, numerical methods are required for estimating z and β. This requires heavy computations, so you are not required to do this in practice. The data you downloaded includes the correct translations z, and the value of β was given earlier. You may complete the assignment by using the given values.

Reconstructing the Image

Now you should have developed all the necessary results. The final part of the assignment is to apply the results to the data described earlier. In other words, find an estimate for the HR-image using the downloaded LR-photos. The requirement is that you should reconstruct the image so that one can roughly identify what the image represents. Note that the reconstructed image will not be a very sharp detailed image. At best, you can obtain an image that resembles something recognizable. So you should not necessarily think that your solution is wrong if the resulting image is not crystal clear.

You should use the given parameter values z, β to obtain the reconstructed image.

It is important that you decide clearly how to position everything in a coordinate system. The problem is translation-invariant since the results would be the same if all photographs and the unknown image are translated by the same distance and direction. An easy way of fixing the HR coordinate system is to define the coordinates of the upper-left corner pixel in each LR-photo to be the translation vector $z(k)$.

Q5: Implement the reconstruction developed in Q1-Q4 and apply it to the given data. Print the reconstructed gray-scale image on your report. The image depicts a part of a license plate of a vehicle. Can you make the characters

recognisable? You *must* include a listing of your code in the report.

How To Compute the Solution Faster

The problem has been constructed so that it should be possible to solve in 10-20 minutes on a modern PC with no special optimization tricks. However, decently non-wasteful coding practices should be used. You should use caution when implementing some key computations. The solution makes it necessary to perform computations on large matrices. To avoid unnecessary memory use, clear temporary matrices as soon as they are not needed any further.

With enough memory available, it may be faster to perform computations directly on matrices instead of writing them as nested loops.

Low Memory Environments

Large matrices consume lots of memory and may cause swapping. You can try to avoid constructing full matrices by performing certaing calculations row by row (this contradicts the above advice on speeding up computations: the best compromise depends on your computing environment).

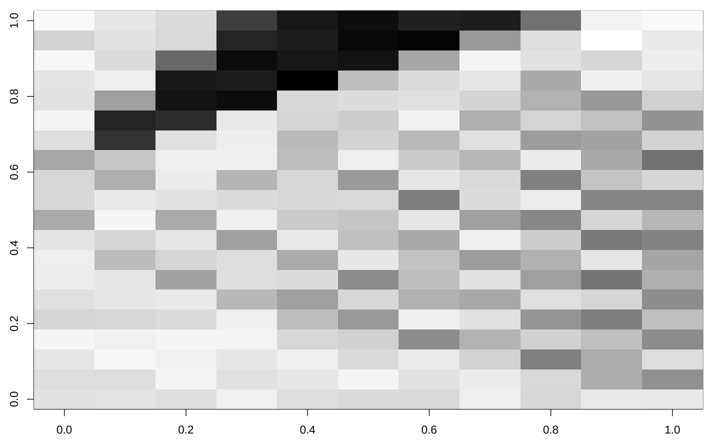
A more elaborate solution is to use sparse matrices. Some large matrices in the solution contain a large number of elements with values very close to zero. By setting these values to zero, it is possible to represent the matrix as a sparse matrix. The sparse matrix data type ignores the zeros and the memory usage is proportional to the number of non-zero elements. Also most matrix operations are faster because they can be computed on the nonzero elements only.

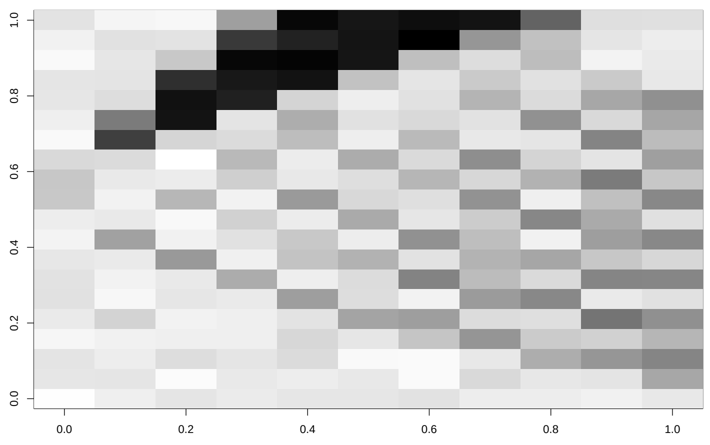
In R, some support for sparse matrices is implemented in the library `Matrix`. The library is not very easy to use, so its use is recommended only if you run into problems with memory usage that you can't otherwise avoid. More information can be found at
`http://cran.r-project.org/src/contrib/Descriptions/Matrix.html`

In Matlab, matrices can be declared sparse with command sparse(). Then the matrices are stored as sparse and arithmetic operations automatically use sparse algorithms. Before declaring a matrix sparse, you need to set all its elements below some threshold to zero. Some operations on matrices return full matrices, so you may have to re-sparsify your matrices between the steps of the computation.

Example

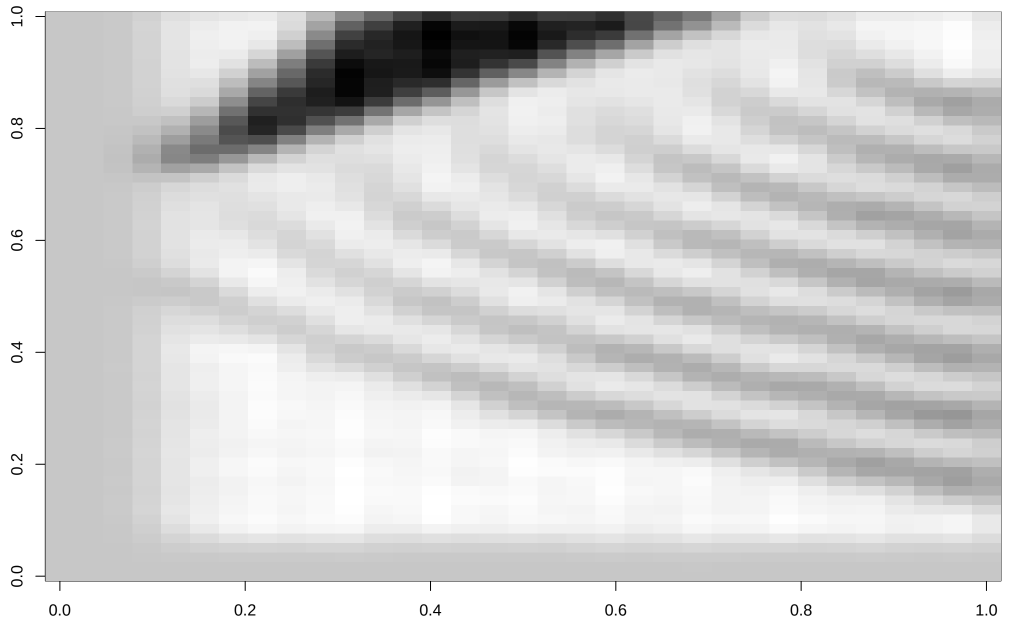
Below is an example illustrating the results obtainable by the approach in the computer assignment. This is not the data you are using, but the problem size is roughly the same: there were 16 low-resolution blurred photographs of the HUT logo. The reconstructed HUT logo has a triple resolution compared to the observed photos, i.e. it has 3×3 pixels for each low-resolution pixel.





Blurred HUT logos in low resolution

The reconstructed image below gives you an idea of how much you can improve resolution by combining the low-resolution images.



Reconstructed image at triple resolution