Expectation Maximization
Mathematics of Imaging
Spring 2008

February 14, 2008
Deconvolution without Noisy Data

- original
- observed data
- not smoothed

General Implementation
Deconvolution
Smoothness

What about noise?
Deconvolution of noisy data
Improvements

MATLAB fun
tools for image reconstruction

1D EM code
Gibbs ringing is what occurs when we try to approximate a discontinuity (as given by our original square wave image) with a finite sum of continuous functions.
Having a smooth original ‘image’ allows for more accurate deconvolution in fewer iterations $i$. Here, $i = 5$. 
For the next few slides, we will see what happens to our image estimates once we add noise to our observations. We will look at the differences based on the variance of the Gaussian used to convolve the image prior to adding noise; the number of iterations run; and the amplitude of the original square wave ($\alpha$).
General Implementation
Deconvolution
Smoothness
What about noise?
Deconvolution of noisy data
Improvements
MATLAB fun
tools for image reconstruction
1D EM code

Deconvolution with Poisson noise

Specs: $\sigma^2 = 1; \ i = 50; \ \alpha = 500$
Deconvolution with Poisson noise

Specs: $\sigma^2 = 4; \ i = 50; \ \alpha = 500$
Deconvolution with Poisson noise

Original data
Observed data
Deconvolution

Specs: $\sigma^2 = 4$; $i = 50$; $\alpha = 100$
Deconvolution with Poisson noise

Specs: $\sigma^2 = 4; \ i = 50; \ \alpha = 300$
Deconvolution with Poisson noise

Specs: $\sigma^2 = 4; \ i = 100; \ \alpha = 300$
Deconvolution with Poisson noise

Specs: \( \sigma^2 = 4; \ i = 200; \ \alpha = 300 \)
Deconvolution with noise: conclusions

- $\sigma^2$
  Increasing the variance seems to have the effect of reducing the frequency of the Gibbs ringing in the examples presented.

- number of iterations
  Increasing the number of iterations increases the noise in the deconvolution but also seems to capture the discontinuity better.

- amplitude of squarewave
  Increases in the amplitude of the square wave leads to increases in the noise amplitude.
Can it be better?

So far, we’ve looked at deconvolution under various circumstances. The question now is to ask whether there are improvements that can be made to reduce the noise in a reconstructed image when the data we’re given is noisy. In the following slides, we will observe the effects of smoothing at each iteration on the reconstructed image.
Deconvolution with Poisson noise & smoothing

- Original data
- Observed data
- Deconvolution
- Smoothed
Deconvolution with Poisson noise & smoothing

- **original**
- **observed data**
- **deconvolution**
- **smoothed**
Deconvolution with Poisson noise & smoothing

- **original**
- **observed data**
- **deconvolution**
- **smoothed**
Deconvolution with Poisson noise & smoothing

- original
- observed data
- deconvolution
- smoothed

General Implementation
Deconvolution
Smoothness

What about noise?
Deconvolution of noisy data

Improvements

MATLAB fun
Tools for image reconstruction

1D EM code
Deconvolution with Poisson noise & smoothing

- Original
- Observed data
- Deconvolution
- Smoothed
Using a Gaussian \((\mu = 0, \sigma^2 \text{ varying})\) to smooth our estimate at each iteration has the effect of reducing the noise in the deconvolution, both in frequency and in magnitude. The trade-off for reductions in noise are the return of Gibbs ringing artifacts and the less accurate capture of discontinuities in the original image. There seems to be a threshold value for the variance of this method, in the sense that for variances below \(\sigma^2 \approx .01\) we begin to see “smoothed” estimates that resemble their noisy counterparts.
The following slides are examples of image reconstruction using various MATLAB tools. Although the very basics were used in the implementation of these, there are many different options that are aimed to produce a better reconstructed image. Take your pick!
Blind Deconvolution

- General Implementation
- Deconvolution
- Smoothness
- What about noise?
- Deconvolution of noisy data
- Improvements

MATLAB fun
- tools for image reconstruction

1D EM code
Regularized deconvolution
Wiener deconvolution
Back to EM (Lucy-Richardson)
clear all
clc
clf

x=zeros(128,1);
alpha = 300;
x(44:100) = alpha;

z=linspace(-5,5,41);
sigma=2;
g = exp(-(z.*z)/(2*sigma^2));
g=g./sum(g);

z1=linspace(-5,5,41); % to be used for smoothing

mu = conv(g,x);
mu = mu(21:128+20);

% Now we want to generate our data using a Poisson distribution on mu
M = random('poiss',mu);
mu=M;
figure(1)
hold on
plot(x,'g','LineWidth',2)
plot(mu,'c','LineWidth',2)

% Using M, we want to estimate our original lambda
maxit = 500; % maximum number of iterations;
L = ones(128,1); % this will be our initial guess for L(x)
Lsmooth = L;

for k=1:maxit
    tmp=conv(g,L);
    % take principal part of convolution only
    tmp1 = tmp(21:128+20);
General Implementation
Deconvolution Smoothness
What about noise?
Deconvolution of noisy data
Improvements
MATLAB fun
tools for image reconstruction

1D EM code

tmp = mu./tmp1;
tmp = conv(tmp,g);
tmp1 = tmp(21:128+20);
L = L.*tmp1;

tmpb = conv(g,Lsmooth);
tmpb1 = tmpb(21:128+20);
tmpb = mu./tmpb1;
tmpb = conv(tmpb,g);
tmpb1 = tmpb(21:128+20);
Lsmooth = Lsmooth.*tmpb1;

%regularization via convolution with Gaussian
tmpb = conv(Lsmooth,g1);
Lsmooth = tmpb(21:128+20);

end
plot(L,'r','LineWidth',2)
plot(Lsmooth,'LineWidth',2)
set(gcf,'Color',[1 1 1])