Homework 1 Sample Solutions
COMP 766-001 – Machine Learning for Bioinformatics

[1] Assuming i.i.d. exponentially-distribution data, the likelihood of the data is

$$\begin{aligned} P(x_1, \ldots, x_N|\lambda) &= \Pi_{i=1}^N P(x_i|\lambda) && (1) \\ &= \Pi_{i=1}^N \lambda e^{-\lambda x_i} && (2) \end{aligned}$$

We want to choose λ to maximize the likelihood of the data. The "log trick" says that we can instead maximize the log-likelihood of the data, which is

$$\begin{aligned} l(x_1, \ldots, x_N|\lambda) &= \log(\Pi_{i=1}^N \lambda e^{-\lambda x_i}) && (3) \\ &= \sum_{i=1}^N \log(\lambda e^{-\lambda x_i}) && (4) \\ &= \sum_{i=1}^N \log \lambda - \lambda x_i && (5) \\ &= N \log \lambda - \lambda \sum_{i=1}^N x_i && (6) \end{aligned}$$

Technically, we should maximize the log-likelihood over the range $\lambda \in (0, +\infty)$—that is, we require that $\lambda > 0$. We take the derivative w.r.t. λ and set it equal to zero.

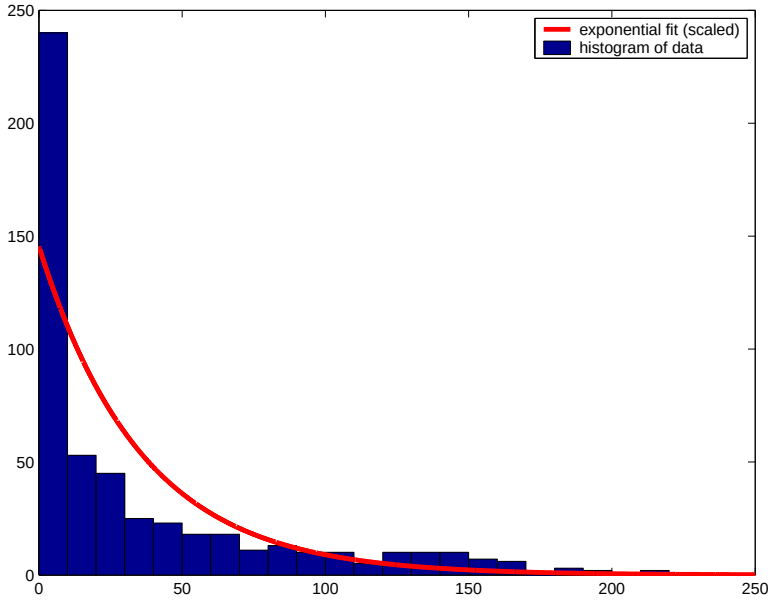
$$\frac{d}{d\lambda} l(x_1, \ldots, x_N|\lambda) = \frac{N}{\lambda} - \sum_{i=1}^N x_i = 0$$

This implies

$$\lambda = \frac{N}{\sum_{i=1}^N x_i} ,$$

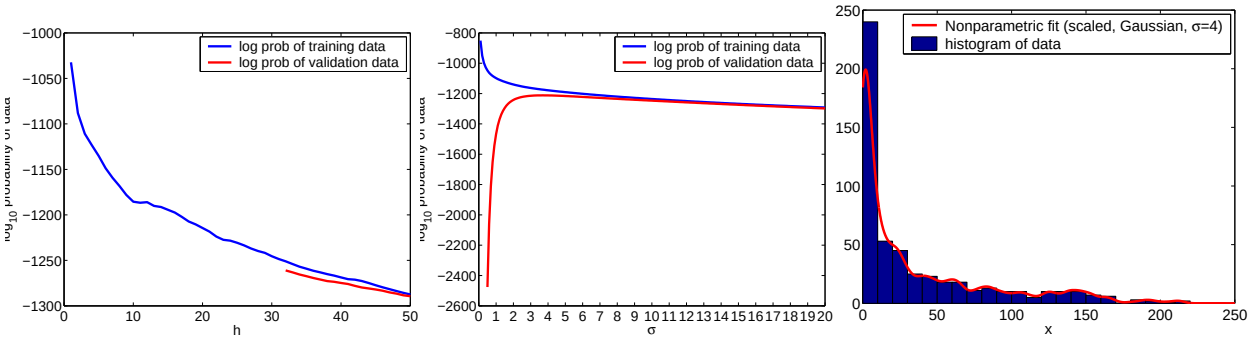
which is just one over the sample mean. If all the x_i are non-negative and at least one is positive, then this estimate for λ is positive, so the constraint $\lambda > 0$ is satisfied. It can be verified by taking the second erivative or be examining the limits $\lambda \to 0$ and $\lambda \to +\infty$ that this value for λ indeed maximizes the likelihood (rather than minimizing it).

[2] For the given data, the maximum likelihood estimate for λ is 0.02788668603064. (Of course, all these digits are not significant.) Below is a histogram of the data, with bins of width 10, and a red curve showing the exponential fit (multiplied by $522 \times 10 = 5220$ so that the area under both curves is the same).



[3] I first tried fitting Parzen windows to the data, however, splitting the data into a training set and a validation set did not give a satisfying estimate of a good width parameter, h. As shown below left, the probability of the validation set was zero when h was below about 32, because one of the validation set points was not sufficiently near any of the training set points. Above 32, the validation set probability monotonically declined. So, I could have gone with the value 32, but this is highly dependent on the particularly training/validation split of the data, and the slope of the curve there suggests a smaller value of h is really better. This all reveals a weakness in this approach to estimating h, which is that a single data point can have a large influence on our validation set probability.

I thus decided to fit a non-parametric density with Gaussian basis functions. The middle figure below shows training and validation set probability for a range of σ. The validation set likelihood is optimize somewhere in the range $\sigma \in [3, 4]$. The plot below right shows the density estimate for $\sigma = 4$, scaled so that the area under the red curve is the same as the area in the boxes of the histogram.



[4] The data was not generated from an exponential distribution. It's actually fruit fly expression data. The main problem with the exponential fit is that there isn't enough density near zero and there's too much density a medium value of x (around 25 to 50). That said, it's not a horrible fit. Depending on your purpose for fitting the data, it might be satisfactory.

My code:

```
%%%%%%%%%%%%%%%%%%
%%% Solutions! %%%
%%%%%%%%%%%%%%%%%%

% Question 2 %

if 0
    load Homework_01_data.txt
    lambda = length(Homework_01_data)/sum(Homework_01_data);
    BinWidth = 10;
    hist(Homework_01_data,5:BinWidth:245);
    set(gca,'xlim',[0 250]);
    xlim=get(gca,'xlim');
    hold on;
    x = xlim(1):0.01:xlim(2);
    y = lambda * exp(-lambda*x);
    h = plot(x,length(Homework_01_data)*BinWidth*y,'r-');
    set(h,'linewidth',3);
    hold off;
    legend('exponential fit (scaled)','histogram of data');
    print -depsc2 Two.eps
end

% Question 3 %

% Parzen windows, trying to optimize h

if 0
    if 0
        hrange = 1:1:50;
        PLPtrain = 0*hrange;
        PLPvalid = 0*hrange;
        Xtrain = Homework_01_data(1:261);
        Xvalid = Homework_01_data(262:end);
        for i=1:length(hrange)
            PLPtrain(i) = ParzenLogProb(Xtrain,Xtrain,hrange(i));
            PLPvalid(i) = ParzenLogProb(Xtrain,Xvalid,hrange(i));
        end
    end
    h1=plot(hrange,PLPtrain,'b-');
    hold on;
    h2=plot(hrange,PLPvalid,'r-');
    hold off;
    set([h1 h2],'linewidth',3);
```

```
    xlabel('h')
    ylabel('log_1_0 probability of data');
    legend('log prob of training data','log prob of validation data');
    print -depsc Three_Parzen_h.eps
end

% Gaussian basis functions, trying to optimize sigma

if 0
    if 1
        sigma_range = 0.1:0.1:20;
        LPtrain = 0*sigma_range;
        LPvalid = 0*sigma_range;
        Xtrain = Homework_01_data(1:261);
        Xvalid = Homework_01_data(262:end);
        for i=1:length(sigma_range)
            LPtrain(i) = GaussLogProb(Xtrain,Xtrain,sigma_range(i));
            LPvalid(i) = GaussLogProb(Xtrain,Xvalid,sigma_range(i));
        end
    end
    h1=plot(sigma_range,LPtrain,'b-');
    hold on;
    h2=plot(sigma_range,LPvalid,'r-');
    hold off;
    set([h1 h2],'linewidth',3);
    set(gca,'fontsize',18);
    xlabel('\sigma')
    ylabel('log_1_0 probability of data');
    legend('log prob of training data','log prob of validation data');
    set(gca,'xtick',0:20)
    print -depsc2 Three_Gauss_Sigma.eps
end

% Plotting the density estimate based on Gaussians

if 1
    x = 0:0.1:250;
    y = GaussCurve(Homework_01_data,x,4);
    hist(Homework_01_data,5:10:245);
    hold on;
    h=plot(x,length(Homework_01_data)*10*y,'r-');
    hold off;
    set(h,'linewidth',3);
    set(gca,'fontsize',18);
    xlabel('x');
    legend('Nonparametric fit (scaled, Gaussian, \sigma=4)','histogram of data');
    print -depsc2 Three_Fit.eps
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function PC = ParzenCurve(Data,Over,h);

PC = 0*Over;
Inc = 1/(h*length(Data));
for i=1:length(Over)
    for j=1:length(Data)
        if abs(Over(i)-Data(j))<=(h/2)
            PC(i) = PC(i)+Inc;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function PLP = ParzenLogProb(Data,Over,h);

PLP = sum(log(ParzenCurve(Data,Over,h)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function GC = GaussCurve(Data,Over,sigma);

GC = 0*Over;
N = length(Data);
A = 1/(sqrt(2*pi)*sigma*N);
for i=1:length(Over)
    GC(i) = sum(A*exp(((Over(i)-Data).^2)/(-2*sigma^2)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function GLP = GaussLogProb(Data,Over,sigma)

GLP = sum(log(GaussCurve(Data,Over,sigma)));
```