CS 5135/6035 Learning Probabilistic Models

Exercise Questions for Lecture 12: Factor Analysis

Gowtham Atluri

2/25/2020

Questions

1. An environmental science graduate student collected airquality measures that include carbon monoxide (denoted as x_c), ground-level ozone (x_o), sulfur dioxide (x_s), and nitrogen dioxide (x_n) at 50 different sites around Cincinnati. The student beleives that there are three major sources of air pollution: diesel emissions (denoted using f_v), coal-fired power plants located along the Ohio River valley (denoted using f_p), and industrial emissions from paper mills (denoted using f_m).
   a. Write the factor analysis model for this data for explaining the air quality measures at one site. **[3 points]**
   b. Write the matrix notation for the factor analysis model for all the samples. Indicate the dimensions for each of the matrices clearly. **[3 points]**
   c. From the matrix notation, what are the known terms and what terms are to be estimated? **[3 points]**
   d. If you were to estimate these terms using probabilistic modelling and MLE, what assumptions would you make about the probability distribution of the observed and latent variables? **[3 points]**
   e. Using Julia, generate a dataset with 50 samples using the parameters **[3 points]**

$$\mathbf{\Lambda} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \\ 0.5 & 0.5 & 0 \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix} \mathbf{\Psi} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.4 \end{bmatrix}$$

   f. Using Julia, use EM algorithm code below to estimate the parameters $\mathbf{\Lambda}, \boldsymbol{\mu}, \mathbf{\Psi}$. Use $k = 3$. Did you results match with the parameters you used to generate the data? If not, why not? **[3 points]**
   g. Using Julia, plot the log-likelihood values. Did your EM approach converge to the final estimates? **[3 points]**
   h. How does your estimates for $\mathbf{\Lambda}, \boldsymbol{\mu}, \mathbf{\Psi}$ change when you use $k = 2$? Compare the convergence of your likelihood for $k = 2$ and $k = 3$. **[4 points]**

Bonus question

1. For the M-Step of the EM algorithm for factor analysis,
   a. Write the equation for $\mathcal{L}(q, \boldsymbol{\theta})$.
   b. Compute the gradient w.r.t $\boldsymbol{\mu}$ and solve for $\boldsymbol{\mu}$
   c. Compute the gradient of $\mathcal{L}(q, \boldsymbol{\theta})$ w.r.t $\mathbf{\Psi}$ and solve for $\mathbf{\Psi}$

Sample code

1. For generating data

```
mu = [0 0 0 0]';
Lambda =
[1.0  0
 1.0  0
 0    0.99
 0    1.0];
Psi =  diagm([0.1, 0.1, 0.1, 0.1]);
d1 = MvNormal([0,0],ones(2));
X = zeros(10000,4);
for i=1:10000
    f = rand(d1,1);
    d2 = MvNormal(vec(mu+ Lambda*f),Psi);
    x = rand(d2,1);
    X[i,:] = x';
end
```

2. EM algorithm

```
function E_Step(X,mu,Lambda,Psi,k)
    mu_f_by_x = (X - repmat(mu',size(X,1),1))*(Lambda'*inv(Lambda*Lambda' + Psi))';
    Sig_f_by_x = eye(k) - Lambda'*inv(Lambda*Lambda' + Psi)*Lambda;
    return mu_f_by_x,Sig_f_by_x;
end

function M_Step(X,mu_f_by_x,Sig_f_by_x,k)
    nrows, ncols = size(X);
    #Computing mu
    mu = mean(X,1)';
    #Computing Lambda
    Lambda_term1 = zeros(ncols,k);
    Lambda_term2 = zeros(k,k);
    for i=1:nrows
        Lambda_term1 = Lambda_term1 + ((X[i,:] - mu)*mu_f_by_x[i,:]');
        Lambda_term2 = Lambda_term2 + (mu_f_by_x[i,:]*mu_f_by_x[i,:]')+Sig_f_by_x;
    end
    Lambda = Lambda_term1*inv(Lambda_term2);
    #Computing Psi
    Phi = zeros(ncols,ncols);
    for i=1:nrows
        Phi = Phi + (X[i,:]*X[i,:]' - X[i,:]*mu_f_by_x[i,:]'*Lambda' - Lambda*mu_f_by_x[i,:]*X[i,:]' +
    end
    Psi = diagm(diag(Phi./nrows));
    return mu, Lambda, Psi
end

function compute_llh(X,mu,Lambda,Psi)
    llh = 0;
    for i=1:size(X,1)
        llh = llh + log(pdf(MvNormal(vec(mu),(Lambda*Lambda')+Psi),X[i,:]));
    end
    return llh;
```

```
end


function fa_em(X,k)
    max_Iter = 100;
    eps = 0.0001;
    llh = -Inf*ones(max_Iter+1);
    mu = mean(X,1)';
    Lambda = rand(size(X,2),k);
    Psi = diagm(rand(size(X,2)));
    print(mu,"\n",Lambda,"\n",Psi,"\n");
    llh[1] = compute_llh(X,mu,Lambda,Psi);
    print(llh[1],"\n")
    for i=1:max_Iter
        print(i,"\n");
        mu_f_by_x,Sig_f_by_x = E_Step(X,mu,Lambda,Psi,k);
        mu_new, Lambda_new, Psi_new = M_Step(X,mu_f_by_x,Sig_f_by_x,k);
        print(mu_new,"\n",Lambda_new,"\n",Psi_new,"\n");
        llh[i+1] = compute_llh(X,mu_new,Lambda_new,Psi_new);
        print(llh[i+1],"\n");
        if(sum(abs.(mu_new-mu))<eps && sum(abs.(Lambda_new-Lambda))<eps && sum(abs.(Psi_new-Psi))<eps)
            break;
        end
        mu = mu_new;
        Lambda = Lambda_new;
        Psi = Psi_new;
    end
    mu_f_by_x,Sig_f_by_x = E_Step(X,mu,Lambda,Psi,k);
    return mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh;
end

#Calling the EM approach for dataset X and 2 factors
mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh = fa_em(X,2)

#plot the log-likelihood
plot(x=collect(1:1:101), y=llh,Geom.line)
```