

Code

Pre-Process Data

```
1 % Sai Satya Charan Malladi
2 % AEROSP 567 Fall 21
3 % Final Project
4
5 % data_loader.m
6 % file to pre-process data
7
8 %% Begin
9
10 % load the walking data of subject 02
11 load('02_02_moc.mat')
12 data_dense = wsMoc.Dof;
13
14 % sample every 4 frames to generate sparse data
15 sample_rate = 4;
16 data_sparse = data_dense(:,1:sample_rate:length(data_dense));
17
18 % subtract mean from the sparse data
19 data_sparse = data_sparse - mean(data_sparse,2);
20 save('data_walking_sparse.mat','data_sparse')
21
22
23 % load the running data of subject 02
24 load('02_03_moc.mat')
25 data_dense = wsMoc.Dof;
26
27 % sample every 4 frames to generate sparse data
28 sample_rate = 4;
29 data_sparse = data_dense(:,1:sample_rate:length(data_dense));
30
31 % subtract mean from the sparse data
32 data_sparse = data_sparse - mean(data_sparse,2);
33 save('data_running_sparse.mat','data_sparse')
34
35
36 % load the jumping data of subject 02
37 load('02_04_moc.mat')
38 data_dense = wsMoc.Dof;
39
40 % sample every 4 frames to generate sparse data
41 sample_rate = 4;
42 data_sparse = data_dense(:,1:sample_rate:length(data_dense));
43
44 % subtract mean from the sparse data
45 data_sparse = data_sparse - mean(data_sparse,2);
46 save('data_jumping_sparse.mat','data_sparse')
```

RBF Kernel

```
1 function val = rbf_kernel(x, xp, beta1, beta2, beta3)
2 % rbf kernel
3
4     val = beta1*exp(-beta2/2*norm(x-xp)^2);
5     if x == xp
6         val = val + beta3;
7     end
8
9 end
```

Linear+RBF kernel

```
1 function val = linear_rbf_kernel(x, xp, alpha1, alpha2, alpha3, alpha4)
2 % linear + rbf kernel
3
4     val = alpha1*exp(-alpha2/2*norm(x-xp)^2) + alpha3*x*xp';
5     if x == xp
6         val = val + alpha4;
7     end
8
9 end
```

GPLVM

```
1 function cost = gplvm_objective(z, Y, Xdim, Ydim, W, betalen, kernel_Y)
2 %
3
4 % unpack z
5 X = reshape(z(1:end-betalen, 1), Xdim);
6 beta = z(end-betalen+1:end);
7
8 %
9 time_steps = Xdim(1);
10 obs_dim = Ydim(2);
11 K_Y = zeros(time_steps);
12
13 for ii = 1:time_steps
14     for jj = 1:time_steps
15         K_Y(ii, jj) = kernel_Y(X(ii, :), X(jj, :), beta(1), beta(2), beta(3));
16     end
17 end
18
19 % K_Y = K_Y + beta(3)*eye(time_steps);
20 det_term = obs_dim/2*logdet(K_Y);
21 trace_term = 1/2*trace(K_Y\Y*Y');
22 log_likelihood = det_term + trace_term + sum(log(beta));
23
24 cost = (log_likelihood);
25
```

26 end

GPDM

```
1 function cost = gpdm_objective(z,Y,Xdim,Ydim,W,alphalen,betalen,...
2                               kernel_X,kernel_Y)
3 %
4
5 % unpack z
6 X = reshape(z(1:end-(alphalen+betalen),1),Xdim);
7 alpha = z(end-(alphalen+betalen)+1: end-betalen);
8 beta = z(end-betalen+1:end);
9
10 %
11 Xout = X(2:end,:);
12 time_steps = Xdim(1);
13 lnt_dim = Xdim(2);
14 obs_dim = Ydim(2);
15
16 K_Y = zeros(time_steps);
17 K_X = zeros(time_steps-1);
18
19 for ii = 1:time_steps
20     for jj = 1:time_steps
21         K_Y(ii,jj) = kernel_Y(X(ii,:),X(jj,:),beta(1),beta(2),beta(3));
22         if ii ≠ time_steps && jj ≠ time_steps
23             K_X(ii,jj) = ...
24                 kernel_X(X(ii,:),X(jj,:),alpha(1),alpha(2),alpha(3),alpha(4));
25         end
26     end
27 end
28 % K_Y = K_Y + beta(3)*eye(time_steps);
29 det_term = obs_dim/2*logdet(K_Y);
30 trace_term = 1/2*trace(K_Y\Y*W^2*Y');
31 log_likelihood = det_term + trace_term;
32
33 % K_X = K_X + alpha(4)*eye(time_steps-1);
34 det_term = lnt_dim/2*logdet(K_X);
35 trace_term = 1/2*trace(K_X\Xout*Xout');
36 log_prior = det_term + trace_term + sum(log(alpha)) + sum(log(beta));
37
38 cost = (log_prior+log_likelihood);
39
40 end
```

Main

```
1 % Sai Satya Charan Malladi
```

```
2 % AEROSP 567 Fall 21
3 % Final Project
4
5 % main.m
6 % GPLVM and GDPM on the data
7
8 clc; clear all; close all;
9
10 %% Begin
11
12 X_pca = cell(3,1);
13 X_gpdm = cell(3,1);
14 alpha_gpdm = cell(3,1);
15 beta_gpdm = cell(3,1);
16 X_gplvm = cell(3,1);
17 beta_gplvm = cell(3,1);
18
19 % % kernels
20 kernel_X = @(x,xp,alpha1,alpha2,alpha3,alpha4) ...
    linear_rbf_kernel(x,xp,alpha1,alpha2,alpha3,alpha4);
21 kernel_Y = @(x,xp,beta1,beta2,beta3) rbf_kernel(x,xp,beta1,beta2,beta3);
22
23 for ii = 1:3
24     %%% load data
25     switch ii
26         case 1
27             load('data_walking_sparse');
28             motion = 'walking';
29         case 2
30             load('data_running_sparse');
31             motion = 'running';
32         case 3
33             load('data_jumping_sparse');
34             motion = 'jumping';
35     end
36
37     %%% PCA Initialization
38     latent_dim = 3;
39     coeff_pca = pca(data_sparse);
40     Y = data_sparse';
41     Ydim = size(Y);
42     vararr = var(Y);
43     vararr(vararr == 0) = 1e-15;
44     W = diag(1./sqrt(vararr));
45     % W = diag(sqrt(vararr));
46
47     X_pca{ii} = coeff_pca(:,1:3);
48     X0 = X_pca{ii};
49     Xdim = size(X0);
50
51     % initialize
52     alpha0 = [0.9; 1; 0.1; 1/exp(1)];
53     alphalen = length(alpha0);
54     beta0 = [1; 1; 1/exp(1)];
55     betalen = length(beta0);
56
```

```
57     %%%% GDPM
58     % initial guess
59     z0 = [X0(:); alpha0; beta0];
60     % test
61     test_gdpm = ...
        gpdm_objective(z0,Y,Xdim,Ydim,W,alphalen,betalen,kernel_X,kernel_Y);
62     options = ...
        optimoptions('fminunc','Algorithm','quasi-newton','Display','iter','MaxFunEvals',15e4);
63 %     options = optimoptions('fminunc','Display','iter','MaxFunEvals',15e4);
64     z_gdpm = fminunc(@(z) gpdm_objective(z,Y,Xdim,Ydim,W,alphalen,betalen,...
        kernel_X,kernel_Y),z0,options);
65
66     X_gdpm{ii} = reshape(z_gdpm(1:end-(alphalen+betalen),1),Xdim);
67     alpha_gdpm{ii} = z_gdpm(end-(alphalen+betalen)+1: end-betalen);
68     beta_gdpm{ii} = z_gdpm(end-betalen+1:end);
69
70     %%%% GPLVM
71     % initial guess
72     z0 = [X0(:); beta0];
73     % test
74     test_gplvm = gplvm_objective(z0,Y,Xdim,Ydim,W,betalen,kernel_Y);
75     options = ...
        optimoptions('fminunc','Algorithm','quasi-newton','Display','iter','MaxFunEvals',15e4);
76 %     options = optimoptions('fminunc','Display','iter','MaxFunEvals',15e4);
77     z_gplvm = fminunc(@(z) ...
        gplvm_objective(z,Y,Xdim,Ydim,W,betalen,kernel_Y),z0,options);
78     X_gplvm{ii} = reshape(z_gplvm(1:end-betalen,1),Xdim);
79     beta_gplvm{ii} = z_gplvm(end-betalen+1:end);
80
81 end
82
83
84 % %% plot setup
85 load('optim_result.mat')
86 figa = figure('Position', get(0, 'Screensize'));
87 figatile = tiledlayout(2,3,'TileSpacing','tight','Padding','tight');
88
89 % plot
90 for ii = 1:3
91     switch ii
92         case 1
93             motion = 'walking';
94         case 2
95             motion = 'running';
96         case 3
97             motion = 'jumping';
98     end
99
100     figure(figa)
101     nexttile(ii)
102     plot3(X_gdpm{ii}(:,1), X_gdpm{ii}(:,2), ...
        X_gdpm{ii}(:,3),'ro-','LineWidth',2,'DisplayName','GPDM')
103 %     hold on
104 %     plot3(X_gplvm{ii}(:,1), X_gplvm{ii}(:,2), ...
        X_gplvm{ii}(:,3),'bo-','LineWidth',2,'DisplayName','GPLVM')
105 %     plot3(X_pca{ii}(:,1), X_pca{ii}(:,2), ...
        X_pca{ii}(:,3),'ko-','LineWidth',2,'DisplayName','PCA')
```

```
106     set(gca,'FontSize',20)
107     set(gca,'TickLabelInterpreter','latex');
108     xlabel('latent-dim-1','fontsize',20,'interpreter','latex')
109     ylabel('latent-dim-2','fontsize',20,'interpreter','latex')
110     zlabel('latent-dim-3','fontsize',20,'interpreter','latex')
111     title(motion,'fontsize',25,'interpreter','latex')
112     legend('location','best','fontsize',20,'interpreter','latex')
113     grid on
114
115     figure(figa)
116     nexttile(ii+3)
117     % plot3(X_gpdm{ii}(:,1), X_gpdm{ii}(:,2), ...
X_gpdm{ii}(:,3),'ro-','LineWidth',2,'DisplayName','GPDM')
118     % hold on
119     plot3(X_gplvm{ii}(:,1), X_gplvm{ii}(:,2), ...
X_gplvm{ii}(:,3),'bo-','LineWidth',2,'DisplayName','GPLVM')
120     % plot3(X_pca{ii}(:,1), X_pca{ii}(:,2), ...
X_pca{ii}(:,3),'ko-','LineWidth',2,'DisplayName','PCA')
121     set(gca,'FontSize',20)
122     set(gca,'TickLabelInterpreter','latex');
123     xlabel('latent-dim-1','fontsize',20,'interpreter','latex')
124     ylabel('latent-dim-2','fontsize',20,'interpreter','latex')
125     zlabel('latent-dim-3','fontsize',20,'interpreter','latex')
126     title(motion,'fontsize',25,'interpreter','latex')
127     legend('location','best','fontsize',20,'interpreter','latex')
128     grid on
129
130 end
131
132
133 %% mean prediction sequences
134 load('optim.result.mat')
135
136 % just for walking
137 X = X_gpdm{1};
138 alpha = alpha_gpdm{1};
139
140 %
141 Xout = X(2:end,:);
142 Xdim = size(X);
143 time_steps = Xdim(1);
144 lnt_dim = Xdim(2);
145
146 % compute K_X
147 K_X = zeros(time_steps-1);
148 for ii = 1:time_steps
149     for jj = 1:time_steps
150         if ii ≠ time_steps && jj ≠ time_steps
151             K_X(ii,jj) = ...
kernel_X(X(ii,:),X(jj,:),alpha(1),alpha(2),alpha(3),alpha(4));
152         end
153     end
154 end
155
156
157 % sample sequences
```

```
158 num_sequences = 25;
159 sample_sequence = zeros(time_steps, lnt_dim, num_sequences);
160
161
162 for ii = 1:num_sequences
163     %
164     x_tt = X(1,:);
165     sample_sequence(1,:,ii) = x_tt;
166
167     % get k_x
168     for tt = 2:time_steps
169         k_x = zeros(time_steps-1,1);
170         for jj = 1:time_steps-1
171             k_x(jj,1) = ...
172                 kernel_X(x_tt, X(jj,:), alpha(1), alpha(2), alpha(3), alpha(4));
173
174             % calculate mu and sigma
175             mu_tt = Xout'/K_X*k_x;
176             sigma2_tt = kernel_X(x_tt, x_tt, alpha(1), alpha(2), alpha(3), alpha(4)) ...
177                 - k_x'/K_X*k_x;
178
179             % next state
180             x_tt = (mu_tt + sqrt(sigma2_tt)*randn(lnt_dim,1))';
181             sample_sequence(tt,:,ii) = x_tt;
182         end
183     end
184
185     % plot
186     figb = figure('Position', get(0, 'Screensize'));
187
188     % plot
189     figure(figb)
190     plot3(X(:,1), X(:,2), X(:,3), 'ro-', 'LineWidth', 2, 'DisplayName', 'GPDM')
191     hold on
192     for ii = 1:num_sequences
193         plot3(sample_sequence(:,1,ii), sample_sequence(:,2,ii), ...
194             sample_sequence(:,3,ii), 'go-', 'LineWidth', 1, 'HandleVisibility', 'off')
195     end
196     set(gca, 'FontSize', 30)
197     set(gca, 'TickLabelInterpreter', 'latex');
198     xlabel('latent-dim-1', 'fontsize', 30, 'interpreter', 'latex')
199     ylabel('latent-dim-2', 'fontsize', 30, 'interpreter', 'latex')
200     zlabel('latent-dim-3', 'fontsize', 30, 'interpreter', 'latex')
201     title('25 sample trajectories using mean-prediction ...
202         [Walking]', 'fontsize', 30, 'interpreter', 'latex')
203     legend('location', 'best', 'fontsize', 30, 'interpreter', 'latex')
204     grid on
```