

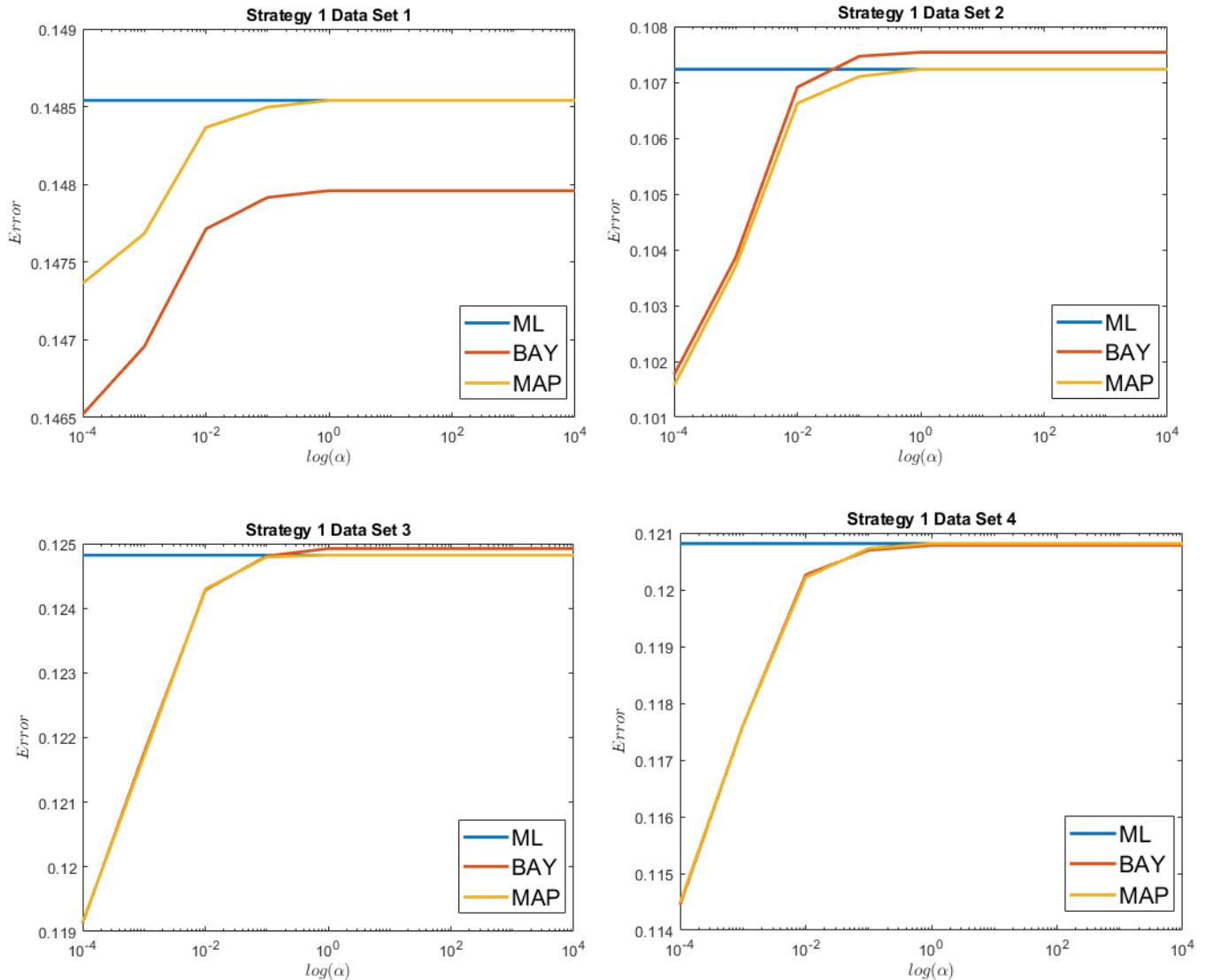
Kalvin Goode

“ML” means maximum likelihood

“Bay” means Bayesian estimation, predictive equation

“MAP” means Maximum A Posteriori

1. Strategy 1

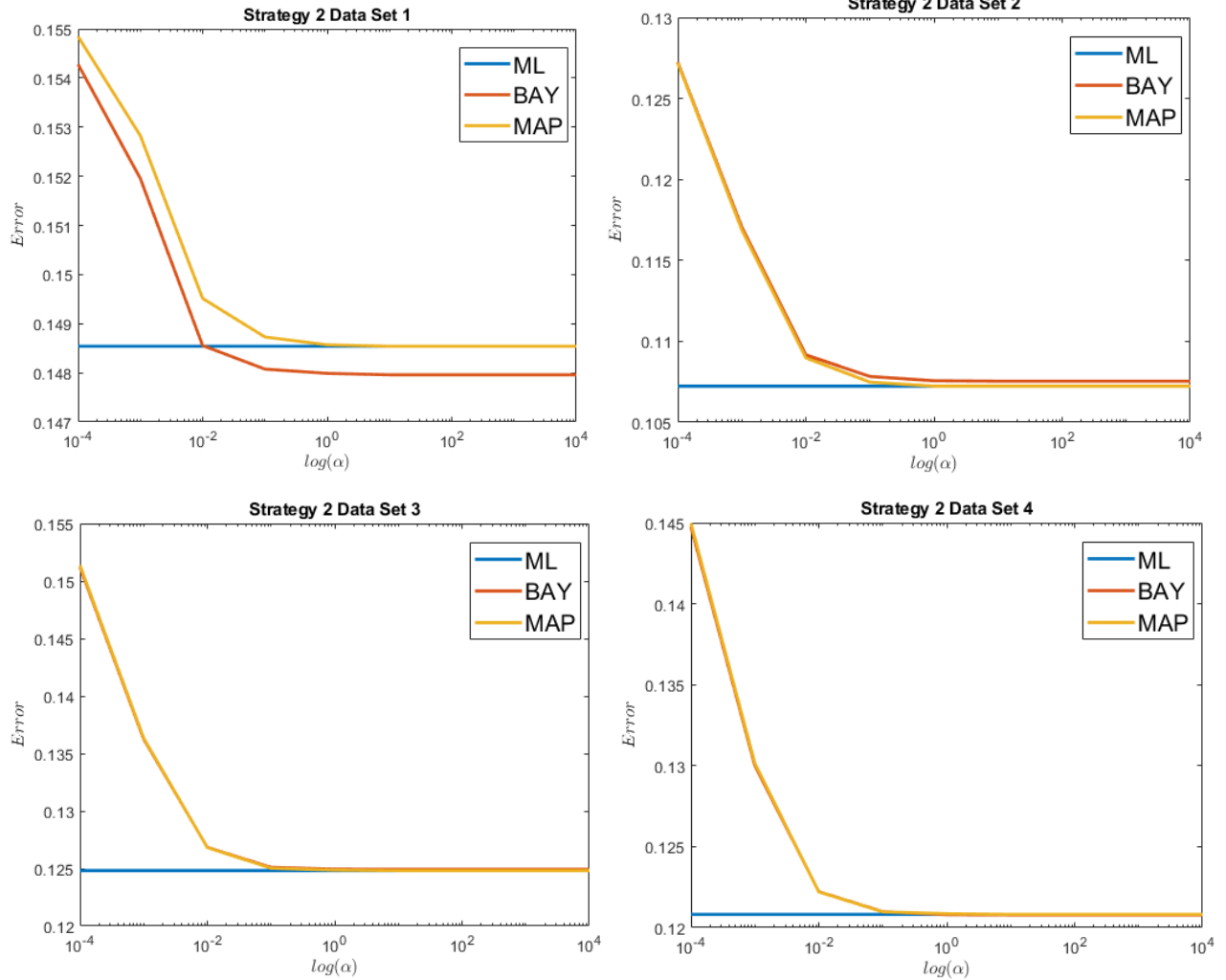


We see that the probability of error has the pattern, in most of the case, $ML \geq BAY \geq MAP$, and error increases as **alpha** increases. This shows that giving the prior mean for strategy 1, smaller **alpha** gives better prediction. The reason is that as **alpha** is larger, Σ_0 increases, but the weight of Σ_1 decreases, which resulted in high error because Σ_1 is the true covariance while Σ_0 has assign different weight to features in the data. Notice that Data 1 has higher error level compare with others since we have less prior data than other data set. Data 2 has the lowest error level compare with others

indicating the train data include features of grass and cheetah.

2. Strategy 2

i. Dataset 1



We see that the probability of error has the pattern, in most of the case, $ML \leq BAY \leq MAP$, and error decreases as **alpha** increases. This shows that giving the prior mean for strategy 2 is inaccurate to the testing sample. The reason is that when **alpha** is small, the prior is far from the true probability. So as **alpha** increases, Σ_0 become more and more weighted and the posterior is closer to the true probability set. Similarly to strategy 1, Data 1 has higher error level compare with others while Data 2 has lowest error level compare with others.

Appendix:

```
load('TrainingSamplesDCT_8.mat');
load('TrainingSamplesDCT_subsets_8.mat');
load('Alpha.mat');
zig=load('Zig-Zag Pattern.txt')+1;
truth=imread('cheetah_mask.bmp');
truth=im2double (truth);
pri_f=size(TrainsampleDCT_FG,1)...
    /(size(TrainsampleDCT_BG,1)+size(TrainsampleDCT_FG,1));
pri_b=1-pri_f;
%cheetah= im2double(imread('cheetah.bmp'));
cheetah= im2double(imread('cheetah.bmp'));
cheetah_p=padarray(cheetah,[4,3],0,'pre');
cheetah_p=padarray(cheetah_p,[3,4],0,'post');
n=1;
cheetah_dct=zeros(68850,64);

for i=1:size(cheetah_p,1)-7
    for j=1:size(cheetah_p,2)-7
        temp=dct2(cheetah_p(i:i+7, j:j+7));
        for k=1:8
            for m=1:8
                cheetah_dct(n,zig(k,m))=temp(k,m);
            end
        end
        n=n+1;
    end
end

im_ml=zeros(2,4,255,270);
err_ml=zeros(2,4,1);
im_bay_set=zeros(2,4,size(alpha,2),255,270);
err_bay_set=zeros(2,4,size(alpha,2));
im_map_set=zeros(2,4,size(alpha,2),255,270);
err_map_set=zeros(2,4,size(alpha,2));

for s=1:2
```

```

if s==1
    load('Prior_1.mat');
else
    load('Prior_2.mat');
end

bg_set={D1_BG,D2_BG,D3_BG,D4_BG};
fg_set={D1_FG,D2_FG,D3_FG,D4_FG};

for k=1:4

    %ML
    mean_ml_bg=take_mean(bg_set{k});
    cov_ml_bg=take_cov(bg_set{k});
    mean_ml_fg=take_mean(fg_set{k});
    cov_ml_fg=take_cov(fg_set{k});

    [im_ml(s,k,:,:), err_ml(s,k,:)] = take_im(cheetah_dct,...
        cheetah,pri_b,pri_f,truth,...
        mean_ml_bg,cov_ml_bg,...
        mean_ml_fg,cov_ml_fg);

    %Bayes
    mean_bay_bg=take_mean(bg_set{k});
    cov_bay_bg=take_cov(bg_set{k});
    mean_bay_fg=take_mean(fg_set{k});
    cov_bay_fg=take_cov(fg_set{k});
    n_bay_bg=size(bg_set{k},1);
    n_bay_fg=size(fg_set{k},1);
    for alp=1:size(alpha,2)
        cov0_bay=take_cov0(alpha(alp),W0);
        mean_n_bg=mean_bay(n_bay_bg,...
            mu0_BG',cov0_bay,...
            mean_bay_bg',cov_bay_bg)';
        cov_n_bg=cov_bay(n_bay_bg,cov0_bay,cov_bay_bg);

        mean_n_fg=mean_bay(n_bay_fg,...
            mu0_FG',cov0_bay,...

```

```

        mean_bay_fg',cov_bay_fg)';
cov_n_fg=cov_bay(n_bay_fg,cov0_bay,cov_bay_fg);

[im_bay_set(s,k,alp,,:), err_bay_set(s,k,alp)]=...
    take_im(cheetah_dct,...
    cheetah,pri_b,pri_f,truth,...
    mean_n_bg,cov_n_bg+cov_bay_bg,...
    mean_n_fg,cov_n_fg+cov_bay_fg);

%MAP
mean_map_bg=mean_n_bg;
cov_map_bg=take_cov(bg_set{k});
mean_map_fg=mean_n_fg;
cov_map_fg=take_cov(fg_set{k});
[im_map_set(s,k,alp,,:), err_map_set(s,k,alp)]=...
    take_im(cheetah_dct,...
    cheetah,pri_b,pri_f,truth,...
    mean_map_bg,cov_map_bg,...
    mean_map_fg,cov_map_fg);
    end
end
end
%end for calculation
for s=1:2
    for k=1:4
        figure();
        for a=1:size(alpha,2)
            ml_y(a)=err_ml(s,k,1);
            bay_y(a)=err_bay_set(s,k,a);
            map_y(a)=err_map_set(s,k,a);
        end
        semilogx(alpha,ml_y,alpha, ...
            bay_y,alpha,map_y,...
            'LineWidth',2)
        xlabel({'$log(\alpha)$'},'Interpreter','latex');
        ylabel({'$Error$'},'Interpreter','latex');
    end
end

```

```

        legend('ML','BAY','MAP','FontSize', 15,...
              'Location','best');
        title(['Strategy ' num2str(s) ' Data Set ' num2str(k)]);
    end
end
%end for plotting

function u=take_mean(sample)
    u=zeros(1,size(sample,2));
    total=0;
    for i=1:size(sample,2)
        for j=1:size(sample,1)
            total=total+sample(j,i);
        end
        u(1,i)=total/size(sample,1);
        total=0;
    end
end

function sig=take_cov(sample)
    sig=zeros(size(sample,2));
    for i=1:size(sample,2)
        for j=1:i
            temp=0;
            u_i=take_mean(sample(:,i));
            u_j=take_mean(sample(:,j));
            for k=1:size(sample,1)
                temp=temp+(sample(k,i)-u_i)*(sample(k,j)-u_j);
            end
            sig(i,j)=temp/size(sample,1);
            if(i~=j)
                sig(j,i)=sig(i,j);
            end
        end
    end
end
end
end

```

```

function post_u=mean_bay(n,mu0,cov0,u,cov)
    post_u=n*cov0*inv(cov+n*cov0)*u+...
        cov*inv(cov+n*cov0)*mu0;
end

```

```

function post_cov=cov_bay(n,cov0,cov)
    post_cov=inv(inv(cov0)+n*inv(cov));
end

```

```

function sig=take_cov0(a,w)
    sig= zeros(64,64);
    for i = 1:64
        sig(i,i) = a*w(i);
    end
end

```

```

function [image, err]=take_im(cheetah_dct,...
    cheetah,pri_b,pri_f,truth,...
    mean_bg,cov_bg,mean_fg,cov_fg)
    like_b=mvnpdf(cheetah_dct,mean_bg,cov_bg);
    like_f=mvnpdf(cheetah_dct,mean_fg,cov_fg);
    image=zeros(size(cheetah,1),size(cheetah,2));
    %disp(like_b(1:10))
    n=1;
    for i=1:size(cheetah,1)
        for j=1:size(cheetah,2)
            if(like_b(n)*pri_b>=like_f(n)*pri_f)
                image(i,j)=0;
            else
                image(i,j)=1;
            end
            n=n+1;
        end
    end
end
%figure();

```

```

    %imshow(image)
    %title("ML Set ")

    %calculate error
    err=error(image,truth);
end

function err=error(img,truth)
    %calculate error
    err=0;
    for i=1:size(truth,1)
        for j=1:size(truth,2)
            if (img(i,j)~= truth(i,j))
                err=err+1;
            end
        end
    end
    err=err/(size(truth,1)*size(truth,2));
end

```