Weining Bao weiningb@usc.edu 8287723653

Assignment 6: ML and MAP Estimation

Consider the signal $Y = X + W$ where X is a Bernoulli random variable sampled from $\{-1, 1\}$ and W is a Gaussian noise distributed as $\mathcal{N}(0, 1)$. X and W are independent. Answer the following questions.

1) Suppose that $p_X(1) = p_X(-1) = 0.5$. What is the value of c for the LMMSE estimate? Run a simulation of 10000 samples and show the empirical error $\hat{e}$ of the LMMSE estimate. Is it close to the theoretical value $\sigma_X^2 - \frac{\sigma_{XY}^2}{\sigma_Y^2}$?
2) Suppose that $p_X(1) = p_X(-1) = 0.5$. Run simulation for a 10000 samples of the ML estimate $\hat{x}_{ML}$. What is the error rate, that is, the number of errors ($\hat{x}_{ML} \neq x$) divided by the number of samples?
3) Suppose that $p_X(1) = 0.2$ and $p_X(-1) = 0.8$. Run simulation for a 10000 samples of the MAP estimate $\hat{x}_{MAP}$. What is the error rate? Compare the error rate with the ML case in 2). Why is one better than the other?

Matlab Simulation

with code at last

1).

X is Bernoulli RV sampled from {-1, 1} with p = 0.5, so E[X] = 0 and VAR[X] = 1. Y = X + W and W ~ N(0,1). So E[Y] = E[X] + E[W] = 0 and VAR[Y] = VAR[X] + VAR[W] = 2. The covariance of X and Y $\sigma_{XY} = E[(Y - E[Y])(X - E[X])] = E[(X - E[X])^2] + \sigma_{XW} = \sigma_X^2$ because X and W are independent.

Linear minimum mean square error estimate (LMMSE) minimizes the mean of cost function to give a good estimate of X given Y = y. Let $\hat{x}(y)$ denotes the estimate of X given Y = y, LMMSE minimizes $E[C(\hat{x}(Y), X)]$ where C(,) is the cost function defined as

$$C(\hat{x}(Y), X) \triangleq ((\hat{x}(Y) - E[\hat{x}(Y)]) - (X - E[X]))^2 .$$

while $\hat{x}(Y) = c(Y - E[Y]) + E[X]$ To minimize the mean of cost function, also known as mean square estimation error,

$c = \frac{\sigma_{XY}}{\sigma_Y^2}, e = \sigma_X^2 - \frac{\sigma_{XY}^2}{\sigma_Y^2}$. So here $c = \frac{\sigma_{XY}}{\sigma_Y^2} = \frac{\sigma_X^2}{\sigma_Y^2} = \frac{1}{2}$ and $\hat{x}(Y) = c(Y - E[Y]) + E[X] = \frac{Y}{2}$. The analytical error $e = \sigma_X^2 - \frac{\sigma_{XY}^2}{\sigma_Y^2} = \sigma_X^2 - \frac{\sigma_X^4}{\sigma_Y^2} = \frac{1}{2}$. $E[\hat{x}(Y)] = \frac{E[Y]}{2} = 0, VAR[\hat{x}(Y)] = \frac{VAR[Y]}{4} = \frac{1}{2}$.

10000 samples are generated with c = 0.5. First 5000 x are assigned as -1 while last 5000, 1. Here are the results:

Part 1:

Generating 10000 estimates of x:
Mean: -0.014039. Variance: 0.508063.
Empirical error: 0.493906.

As we can see, the results are all close to the analytical values with mean = 0, variance = 0.5 and error = 0.5. c = 0.5 with LMMSE is a good estimate for X given Y = y. However, LMMSE does not guarantee $\hat{x}(Y)$ in X's sample space χ.

2).
Maximum A Posteriori (MAP) Estimate is another estimation of X given Y = y which guarantees that $\hat{x}(Y)$ is in X's sample space χ: $\hat{x}_{MAP} = \arg\max_{x\in\chi} p_{X|Y}(x \mid y) = \arg\max_{x\in\chi} p_{Y|X}(y \mid x)\frac{p_X(x)}{p_Y(y)}$ $= \arg\max_{x\in\chi} p_{Y|X}(y \mid x)p_X(x)$ since $p_Y(y)$ is not relevant to the x to get maximum. If $p_X(x)$ are equal for all the $x \in \chi$, MAP estimate is simplified into Maximum likelihood (ML) estimate: $\hat{x}_{ML} = \arg\max_{x\in\chi} p_{Y|X}(y \mid x)$.

Here, $\hat{x}_{ML} = \arg\max_{x\in\chi} p_{Y|X}(y \mid x) = \arg\max_{x\in\chi} \frac{p(x,y)}{p_X(x)} = \arg\max_{x\in\chi} \frac{p_X(x)p_W(y-x)}{p_X(x)} = \arg\max_{x\in\chi} p_W(y-x)$ because X and W are independent.

10000 samples of receiver signal are generated. Probabilities of all possible values at sender are calculated and compared to choose the x that maximizes this probability with given received signals. Here is the result:

Part 2:

Generating 10000 estimates of x:
Number of correct estimates: 8451. Error rate: 0.154900.

3).
$p_X(x)$ are not equal for all the $x \in \chi$ here so
$\hat{x}_{MAP} = \arg\max_{x\in\chi} p_{Y|X}(y \mid x)p_X(x) = \arg\max_{x\in\chi} p_W(y-x)p_X(x)$ where $p_X(1) = 0.2, p_X(-1) = 0.8$.
Simulation similar to the part 2 is done and here is the result:

Part 3:

Generating 10000 estimates of x:
Number of correct estimates: 8919. Error rate: 0.108100.

Compare the error rate of MAP with ML estimates, MAP has lower error rate than ML so MAP seems to be a better estimator than ML. According to their definitions, $\hat{x}_{MAP} = \arg\max_{x \in \chi} p_{Y|X}(y \mid x) p_X(x)$ and $\hat{x}_{ML} = \arg\max_{x \in \chi} p_{Y|X}(y \mid x)$, MAP estimator takes the prior probability information about X into account, that is why it is better than ML in general. However, only ML can be used in the case that no prior distribution information is available for estimate.

Code:

```
% part 1
fprintf('\nPart 1: \n\n');
num = 10000; % number of samples
c = 1/2; % c in LMMSE
x = zeros(num, 1);
fprintf('Generating %d estimates of x:\n', num);
for i = 1:(num/2) % generate Bernoulli RV from {-1, 1} with p = 0.5
   x(i) = -1;
   x(i + num/2) = 1;
end
y = zeros(num, 1); % receiver signal storage
for i = 1:num % receiver signals generation
   y(i) = (x(i) + normrnd(0, 1)) * c; % add noise
end
m = mean(y);
fprintf('Mean: %f. Variance: %f.\n', m, var(y));
error = 0;
for i = 1:num % sum error up
   error = error + (y(i) - m - x(i))^2;
end
error = error / num;
fprintf('Empirical error: %f.\n\n', error);

% part 2
fprintf('\nPart 2: \n\n');
num = 10000; % number of samples
x = zeros(num, 1);
fprintf('Generating %d estimates of x:\n', num);
for i = 1:(num/2) % generate Bernoulli RV from {-1, 1} with p = 0.5
   x(i) = -1;
   x(i + num/2) = 1;
end
y = zeros(num, 1); % receiver signal storage
for i = 1:num % receiver signals generation
   y(i) = x(i) + normrnd(0, 1); % add noise
end
esti = zeros(num, 1);
for i = 1:num % generate estimates
   p1 = normpdf(y(i)-1, 0, 1); % probability of x = 1
```

```
    pn1 = normpdf(y(i)+1, 0, 1); % probability of x = -1
    if p1 > pn1
        esti(i) = 1;
    else
        esti(i) = -1;
    end
end
correct = sum(x == esti);
fprintf('Number of correct estimates: %d. Error rate: %f.\n\n', correct, 1 - correct / num);

% part 3
fprintf('\nPart 3: \n\n');
num = 10000; % number of samples
p = 0.2; % probability of x = 1
x = -ones(num, 1);
fprintf('Generating %d estimates of x:\n', num);
for i = 1:(num*p) % generate Bernoulli RV from {-1, 1} with p = 0.2
    x(i) = 1;
end
y = zeros(num, 1); % receiver signal storage
for i = 1:num % receiver signals generation
    y(i) = x(i) + normrnd(0, 1); % add noise
end
esti = zeros(num, 1);
for i = 1:num % generate estimates
    p1 = normpdf(y(i)-1, 0, 1) * p; % probability of x = 1
    pn1 = normpdf(y(i)+1, 0, 1) * (1-p); % probability of x = -1
    if p1 > pn1
        esti(i) = 1;
    else
        esti(i) = -1;
    end
end
correct = sum(x == esti);
fprintf('Number of correct estimates: %d. Error rate: %f.\n\n', correct, 1 - correct / num);
```