UC Berkeley
Department of Electrical Engineering and Computer Science
Department of Statistics

EECS 281A / STAT 241A Statistical Learning Theory

Solution to Problem Set 6
Fall 2012

Issued: Tues. Oct. 30, 2012 **Due:** Tues. Nov. 13, 2012

Reading: Chapters 14, 15, 26; Sampling methods chapter (just after Chap. 20)

Problem 6.1
Factor analysis: Consider the factor analysis model

$$Y = \mu + \Lambda X + W$$

where $X \sim N(0, I)$ is a d-dimensional Gaussian; $\mu \in \mathbb{R}^n$ is a mean vector; $W \sim N(0, \sigma^2 I)$ is an n-dimensional Gaussian; and $\Lambda \in \mathbb{R}^{n \times d}$ is the factor matrix. From the course website, you can download the ASCII format files `Y.dat` and `Lambda.dat`, containing an observation vector $y \in \mathbb{R}^{121}$ and a 121×5 factor matrix Λ (i.e., $d = 5$ and $n = 121$).

(a) Assuming that $\mu = 0$ and $\sigma^2 = 0.25$, compute the conditional mean vector $\mathbb{E}[X|y]$ and covariance matrix $\mathrm{cov}[X|y]$. What does this estimate tell you about which factors were most heavily involved in generating y?

Solution: The random variables (X, Y) are jointly zero-mean Gaussian (since $\mu = 0$) with covariance matrix Σ with block partitions

$$\begin{aligned}
\Sigma_{XX} &= \mathrm{cov}(X, X) = I, \\
\Sigma_{XY} &= \mathrm{cov}(X, Y) = \mathrm{cov}(X, \Lambda X + W) = \Lambda^\top, \\
\Sigma_{YX} &= \Sigma_{XY}^\top = \Lambda, \\
\Sigma_{YY} &= \mathrm{cov}(\Lambda X + W, \Lambda X + W) = \Lambda\Lambda^\top + \sigma^2 I.
\end{aligned}$$

Therefore, we know that the conditional random variable $(X \mid Y)$ is also Gaussian with mean

$$\mathbb{E}[X \mid Y] = \Sigma_{XY}\Sigma_{YY}^{-1} Y = \Lambda^\top(\Lambda\Lambda^\top + \sigma^2 I)^{-1} Y$$

and covariance

$$\mathrm{cov}(X \mid Y) = \Sigma_{XX} - \Sigma_{XY}\Sigma_{XY}^{-1}\Sigma_{YX} = I - \Lambda^\top(\Lambda\Lambda^\top + \sigma^2 I)^{-1}\Lambda.$$

With the provided values of Λ, Y, and $\sigma^2 = 0.25$, we find that $\mathbb{E}[X \mid Y]$ is equal to

$$\begin{pmatrix} 0.1136 & 0.0703 & -0.1914 & 2.1110 & 2.1267 \end{pmatrix}^\top$$

and $\mathrm{cov}(X \mid Y)$ is equal to

$$\begin{pmatrix} 2.8642 \times 10^{-3} & -2.2109 \times 10^{-3} & -2.2109 \times 10^{-3} & -2.2109 \times 10^{-3} & -2.2109 \times 10^{-3} \\ -2.2109 \times 10^{-3} & 2.4385 \times 10^{-2} & -4.9023 \times 10^{-6} & -4.9023 \times 10^{-6} & -4.9023 \times 10^{-6} \\ -2.2109 \times 10^{-3} & -4.9023 \times 10^{-6} & 2.4385 \times 10^{-2} & -4.9023 \times 10^{-6} & -4.9023 \times 10^{-6} \\ -2.2109 \times 10^{-3} & -4.9023 \times 10^{-6} & -4.9023 \times 10^{-6} & 2.4385 \times 10^{-2} & -4.9023 \times 10^{-6} \\ -2.2109 \times 10^{-3} & -4.9023 \times 10^{-6} & -4.9023 \times 10^{-6} & -4.9023 \times 10^{-6} & 2.4385 \times 10^{-2} \end{pmatrix}.$$

Note that the conditional distribution $(X \mid Y)$ is our best guess of the process that generated the observed data Y. In particular, if $\mathbb{E}[X \mid Y]$ is large in certain entries, this means we are estimating that those factors were significant, and the conditional variance $\mathrm{cov}(X \mid Y)$ tells us how certain we are about those estimates.

(b) What is the relation between $\mathbb{E}[X|y]$ and the MAP estimate of X given $Y = y$?

Solution: As noted in part (a), the conditional distribution $(X \mid Y)$ is Gaussian, so the MAP estimate of X given $Y = y$ is equal to the conditional expectation $\mathbb{E}[X \mid y]$.

Problem 6.2

Model selection for curve-fitting Suppose that we are interested in fitting curves to noisy data; in particular, consider the polynomial regression model linking the response variable $y \in \mathbb{R}$ to the covariate $x \in \mathbb{R}$ via

$$y = \sum_{k=1}^{D} \beta_k x^k + w, \tag{1}$$

where $w \sim N(0, 1)$ is Gaussian noise. One model selection problem is that of choosing the appropriate degree D of this polynomial fit, which we explore in this problem.

(a) The course website has two ASCII files `Ymodel.dat` and `Xmodel.dat`, containing samples $\{x_i, y_i\}_{i=1}^n$ with $n = 100$. For $d = 1, 2, \ldots, 10$, fit the model (1) to the data by minimizing the least squares loss

$L(\beta) = \frac{1}{2n}\sum_{i=1}^{n}\left\{y_i - \sum_{k=1}^{d}\beta_k (x_i)^k\right\}^2$. For which choice of d is this cost function smallest? On the same figure, plot the original data and the models fitted to the data for $d = 1, 2, 3, 4$.

Solution: Given $d = 1, \ldots, 10$, let

$$Y = \begin{pmatrix} y_1 \\ \cdots \\ y_n \end{pmatrix}, \quad X_{(d)} = \begin{pmatrix} x_1 & x_1^2 & \cdots & x_1^d \\ \cdots & & & \cdots \\ x_n & x_n^2 & \cdots & x_n^d \end{pmatrix}, \quad \beta_{(d)} = \begin{pmatrix} \beta_1 \\ \cdots \\ \beta_d \end{pmatrix}.$$

Then the cost function $L_d(\beta_{(d)})$ can be written as

$$L_d(\beta_{(d)}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{d}\beta_k x_i^k\right)^2 = \frac{1}{2n}\|Y - X_{(d)}\beta_{(d)}\|_2^2,$$

and thus by taking the derivative of L_d with respect to $\beta_{(d)}$, we see that the cost function is minimized by

$$\beta^*_{(d)} = (X_{(d)}^\top X_{(d)})^{-1} X_{(d)}^\top Y,$$

which can be computed with $\beta^*_{(d)} = (X_{(d)}^\top X_{(d)}) \backslash X_{(d)}^\top Y$ in MATLAB. The resulting cost functions are as follows:

d	1	2	3	4	5	
$L_d(\beta^*_{(d)})$	30.435	17.196	4.9210	4.9154	4.8520	$(\times 10^{-3})$
d	6	7	8	9	10	
$L_d(\beta^*_{(d)})$	4.7384	4.7378	4.7363	4.7238	4.6987	$(\times 10^{-3})$

We see that the cost function is smallest when $d = 10$, which is to be expected since using a higher degree polynomial gives us more power to fit the data better. However, it is also prone to overfitting, since if the degree is too high then we will just be fitting noise instead of the true structure of the data. Indeed, we see that the cost function remains essentially constant once we use a third-degree polynomial, and a visual inspection of the data clearly suggests that the data are generated from a cubic polynomial. Figure 1 shows the plot of the data, along with the fitted polynomials for $d = 1, 2, 3, 4$. The polynomial with $d = 4$ is very close to the polynomial with $d = 3$.

(b) Show that the AIC method, when applied to this problem, reduces to choosing the degree $\widehat{d}$ that minimizes $L(\widehat{\beta}[d]) + d/n$, where $\widehat{\beta}[d]$ is the

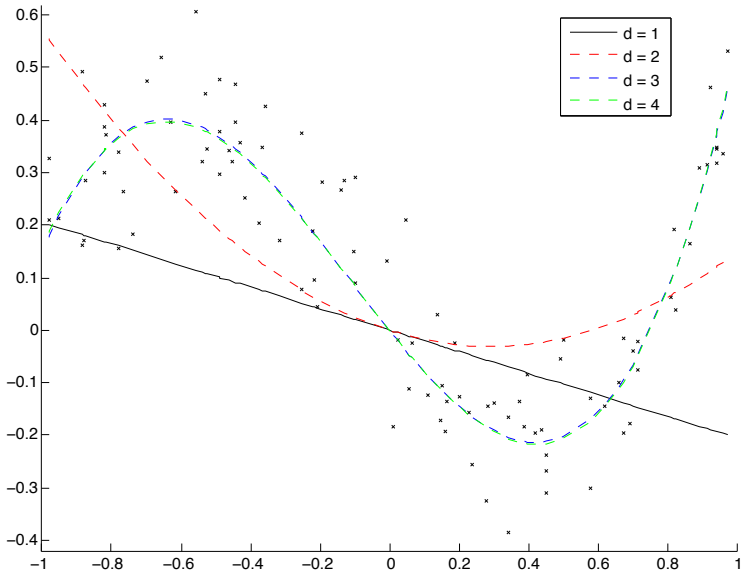

Figure 1: Fitted polynomials with degree at most 4.

fitted set of parameters of the polynomial with degree d. Implement this model selection criterion for this data set, where d ranges over $\{1, 2, \ldots, 10\}$. What $\widehat{d}$ is chosen by the procedure?

Solution: Under the Gaussian model assumption, the log likelihood of the data is precisely $-nL(\beta)$. The AIC attempts to find d that minimizes the penalized negative log likelihood:

$$\text{AIC} = nL(\widehat{\beta}[d]) + d,$$

or equivalently, we want to minimize $L_d^* = L(\widehat{\beta}[d]) + d/n$. The resulting regularized costs are shown in the table below. In this case we still choose the polynomial with degree $d = 3$.

d	1	2	3	4	5	
L_d^*	4.0435	3.7196	3.4921	4.4915	5.4852	$(\times 10^{-2})$
d	6	7	8	9	10	
L_d^*	6.4738	7.4738	8.4736	9.4724	10.4698	$(\times 10^{-2})$

(c) Given the model $\widehat{\beta} = \widehat{\beta}[\widehat{d}]$ chosen in part (b) and a new observed

covariate x, one can generate a predicted response $\hat{y}$ as

$$\hat{y} \;=\; \sum_{k=1}^{\widehat{d}} \widehat{\beta}_k x^k.$$

The course website also contains two ASCII files `Ynew.dat` and `Xnew.dat` with $m = 500$ new samples. Using the samples in `Xnew.dat`, generate predictions $\hat{y}_i, i = 1, \ldots, m$, and then compute the prediction error $\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$.

Solution: We compute the predicted third-degree polynomial response $\widetilde{y}$ using the computed $\widehat{\beta}[3]$ from part (b), giving us the plot in Figure 2, and the prediction error is

$$\sum_{i=1}^{m}(y_i - \widetilde{y}_i)^2 = 5.1013.$$

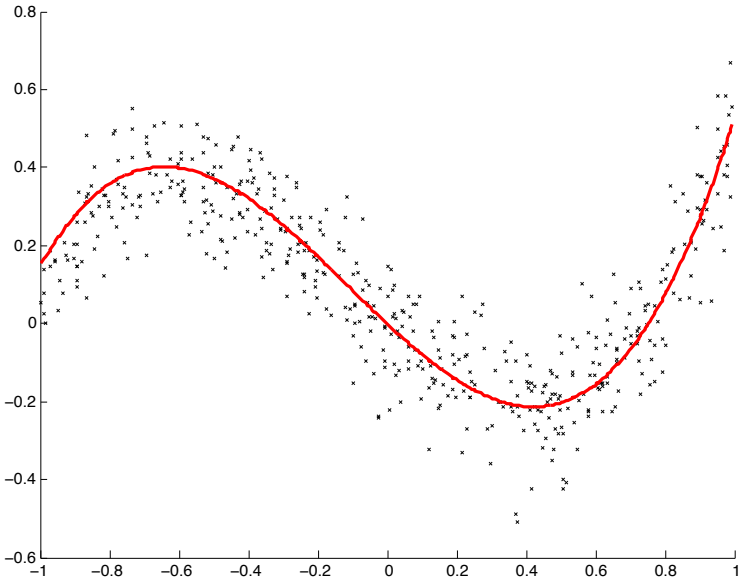

Figure 2: Plot of the new data and the fitted third-degree polynomial.

(d) Repeat part (c) for using the full model fit $\widehat{\beta}[10]$ with all $D = 10$ parameters. Is the prediction error of the full model higher/lower than your fitted model?

Solution: We repeat part (c) using $\widehat{\beta}[10]$, giving us the plot in Figure 3, and the prediction error is now

$$\sum_{i=1}^{m}(y_i - \widetilde{y}_i)^2 = 5.4406.$$

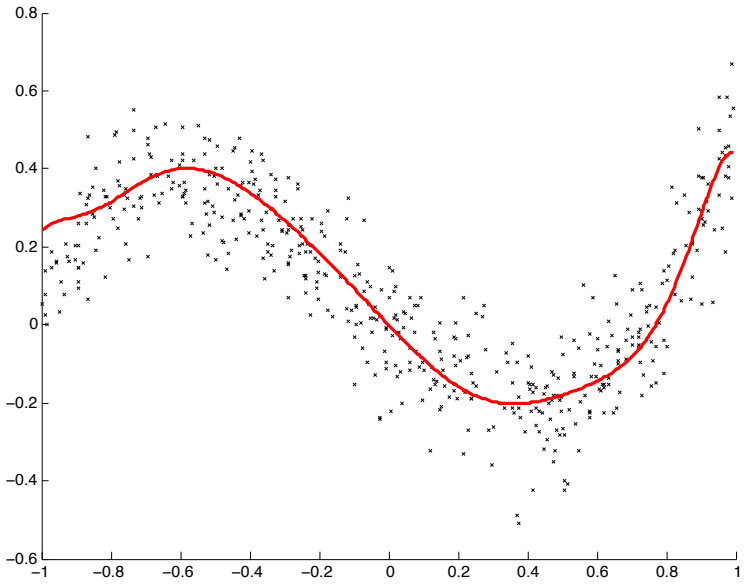

Figure 3: Plot of the new data and the fitted tenth-degree polynomial.

We see that now the prediction error is larger than the case $d = 3$ in part (c). This shows that the low training error of the $d = 10$ model actually overfits the data, and hence does not generalize well.

```
% Script for Problem 6.2

load Xmodel.dat
load Ymodel.dat

%% Part (a)
% Construct data matrices
deg = 10;   % maximum degree to fit
Y = Ymodel;
X = Xmodel(:, ones(1,deg));
```

```
for d = 2:deg
    X(:,d) = X(:,d).^d;
end

% Compute least square solution for each d
betas = cell(deg,1);
cost = zeros(deg,1);
for d = 1:deg
    betas{d} = X(:,1:d)\Y;
    cost(d) = sum((Y-X(:,1:d)*betas{d}).^2);
end

% Find minimum cost
[min_cost, i_cost] = min(cost);

% Plot data
figure;
scatter(Xmodel,Ymodel,25,'k','x');
hold all;
h1 = plot(Xmodel, X(:,1:1)*betas{1}, '-k');
h2 = plot(Xmodel, X(:,1:2)*betas{2}, '--r');
h3 = plot(Xmodel, X(:,1:3)*betas{3}, '-.b');
h4 = plot(Xmodel, X(:,1:4)*betas{4}, ':k');
legend([h1,h2,h3,h4], {'d = 1', 'd = 2', 'd = 3', 'd = 4'});
axis([-1 1 -0.42 0.62]);


%% Part (b)

% Compute regularized cost
reg_cost = cost + (1:deg)';

% Find minimum of regularized cost
[min_reg, i_reg] = min(reg_cost);


%% Part (c)

load Xnew.dat
load Ynew.dat

% Compute predicted response and error with d = 3
Ypred = Xnew*betas{3}(1) + Xnew.^2*betas{3}(2) ...
        + Xnew.^3*betas{3}(3);
Yerr = sum((Ynew-Ypred).^2);

% Plot new data and the fitted polynomial
figure;
scatter(Xnew,Ynew,25,'k','x');
```

```
hold all;
plot(Xnew,Ypred,'-r','LineWidth',2);


%% Part (d)

% Compute predicted response and error with d = 10
Ypred_d = 0;
for d = 1:10
    Ypred_d = Ypred_d + Xnew.^d*betas{10}(d);
end
Yerr_d = sum((Ynew-Ypred_d).^2);

% Plot new data and the fitted polynomial
figure;
scatter(Xnew,Ynew,25,'k','x');
hold all;
plot(Xnew,Ypred_d,'-r','LineWidth',2);
```

Problem 6.3

Accept/reject sampling. Suppose that we want to sample from a random variable X with density

$$p_X(x) = \begin{cases} cx(1-x) & \text{for } x \in [0,1] \\ 0 & \text{otherwise.} \end{cases},$$

where $c > 0$ is an appropriate constant.

(a) Suppose that you have a block-box routine to draw samples Y from a uniform distribution on $[0,1]$. Describe how to perform accept-reject sampling to generate samples $X \sim p_X$.

Solution: We need to find a tractable distribution $g(x)$ with the property that, for some k, $kg(x) \geq p_X(x)$ for $x \in \text{supp}\{X\}$. In this case, the support of $p_X(x)$ is the interval $[0,1]$, so we can choose $g(x)$ to be the uniform distribution over $[0,1]$. Since $\int_0^1 x(1-x) = \frac{1}{6}$, we have that the constant c in $p_X(x)$ is equal to 6, giving $p_X(x)$ a maximum value of $\frac{3}{2}$. Therefore, we can choose $k = \frac{3}{2}$ and perform rejection sampling as follows:

1. Sample $Y \sim \text{Uniform}[0,1]$.
2. Sample $U \sim \text{Uniform}[0,1]$.
   - If $U > \frac{1}{k} p_X(Y)$, reject Y and return to step 1.
   - Otherwise, accept and return Y as a sample from p_X.

(b) Suppose that you standardize your sampler so that, conditioned on the event $Y = 1/2$, it accepts with probability $1/2$. Implement this version of the sampler, and plot the histogram of $n = 10000$ randomly drawn samples. Hand in this histogram, and your code. (Be sure to document your code, explaining what you are doing.)

Solution: In order to satisfy the given condition, we choose $k = 3$ instead of $k = \frac{3}{2}$ as above. With this threshold, we have the desired acceptance probability conditioned on the event $Y = 1/2$:

$$\mathbb{P}(\text{accept } Y \mid Y = 1/2) = \mathbb{P}\Big(U \leq \frac{1}{3}p_X(Y) \;\Big|\; Y = \frac{1}{2}\Big) = \frac{1}{3}p_X(1/2) = \frac{1}{2}.$$

We implement this version of the sampler in MATLAB (code given below), and draw $n = 10000$ samples. The histogram (on 100 bins) of the samples is shown in Figure 4, along with the curve of the density p_X (on rescaled axis so the scales match). In this case, the average number of uniform samples needed to draw one sample is

$$\overline{T} = 3.0039.$$

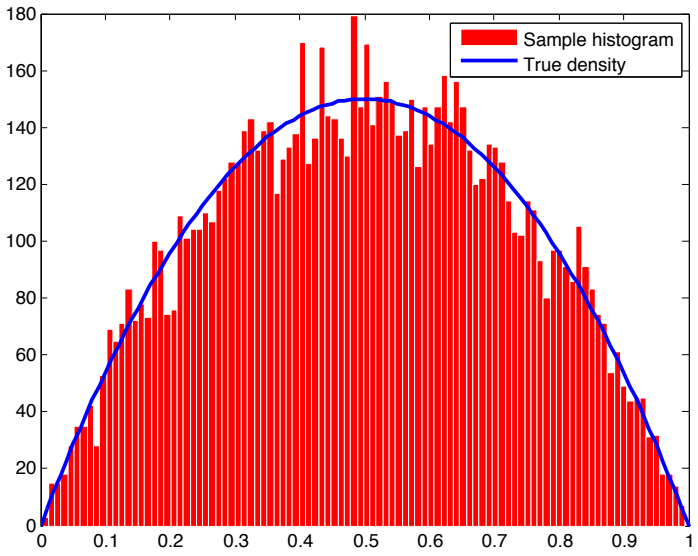

Figure 4: Histogram of $n = 10000$ randomly drawn samples and the (rescaled) density.

(c) Let T denote the random number of uniform samples that must be drawn in order for the algorithm to output one sample. Compute $\mathbb{E}[T]$, and compare this theoretical mean to the empirical mean over your $n = 10000$ samples. (Note: Your code will need to record the number of uniform samples that were generated for each of the $n = 10000$ samples in (b)).

Solution: If we accept the first sample that we generate, then $T = 1$; otherwise, we have used one trial and we have to repeat the procedure from the beginning. Thus, we have the equation

$$\begin{aligned} \mathbb{E}[T] &= \mathbb{P}(\text{accept}) \cdot 1 + \mathbb{P}(\text{reject})\big(1 + \mathbb{E}[T]\big) \\ \implies \quad \mathbb{E}[T] &= \frac{\mathbb{P}(\text{accept}) + \mathbb{P}(\text{reject})}{1 - \mathbb{P}(\text{reject})} = \frac{1}{\mathbb{P}(\text{accept})}. \end{aligned}$$

Note that

$$\begin{aligned} \mathbb{P}(\text{accept}) &= \mathbb{P}\Big(U \le \frac{1}{3} p_X(Y)\Big) = \int_0^1 \mathbb{P}\Big(U \le \frac{1}{3} p_X(Y) \;\Big|\; Y = y\Big)\, dy \\ &= \int_0^1 \frac{1}{3} p_X(y)\, dy = \frac{1}{3}, \end{aligned}$$

so we conclude that

$$\mathbb{E}[T] = 3.$$

Thus, the empirical average $\overline{T} = 3.0039$ from part (b) is very close to the true mean $\mathbb{E}[T] = 3$.

```
function [samples num_tries] = sample(n)
% Rejection sampling algorithm for Problem 6.3(b)

samples = zeros(n,1);   % generated samples
num_tries = zeros(n,1); % number of trials for each sample

for i = 1:n
    % Generate the i-th sample
    accept = 0; % whether we have found an accepted sample
    tries = 0;  % number of tries so far

    while (accept==0)
        y = rand;
        u = rand;
        if (u ≤ 2*y*(1-y))
            accept = 1; % accept y
```

```
        end
        tries = tries + 1;
    end

    % Record results
    samples(i) = y;
    num_tries(i) = tries;
end
```

```
% Script for Problem 1(b)

% Generate samples
n = 10000;   % number of samples
[samples num_tries] = sample(n);
fprintf('Average number of trials = %.4f\n', ...
    mean(num_tries));

% Plot histogram and density
bins = 100;  % number of bins used
hist(samples, bins);
set(findobj(gca,'Type','patch'), 'FaceColor',' r', ...
    'EdgeColor', 'w');
hold on;
x = 0:1/bins:1;
plot(x, (n/bins)*6*x.*(1-x),'b','LineWidth',2);
legend({'Sample histogram', 'True density'});
```

Problem 6.4

Cautionary tale about importance sampling: Suppose that we wish to estimate the normalizing constant $Z(p)$ of a Gaussian density $p(\cdot) \sim \mathcal{N}(0, \sigma_p^2)$. Given i.i.d. samples $y_1, \ldots, y_n$ from a standard normal $q(\cdot) \sim \mathcal{N}(0, 1)$, consider the importance sampling estimate

$$\widehat{Z} = \frac{1}{n} \sum_{i=1}^{n} \frac{p^*(y_i)}{q(y_i)} \qquad \text{where} \quad p^*(y) = \exp(-\frac{1}{2\sigma_p^2} y^2).$$

(a) Show that $\widehat{Z}$ is an unbiased estimator of Z_p.

Solution: Let $f(y) = p^*(y)/q(y)$, and let Y be a random variable sampled from q. Since $y_1, \ldots, y_n \sim q$ i.i.d., we have

$$\mathbb{E}[\widehat{Z}] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[f(y_i)] = \mathbb{E}[f(Y)] = \int_{-\infty}^{\infty} \frac{p^*(y)}{q(y)} q(y) dy = \int_{-\infty}^{\infty} p^*(y) dy = Z_p.$$

(b) Letting $f(y) = p^*(y)/q(y)$, show that $\text{var}(\widehat{Z}) = \frac{\text{var}(f(Y))}{n}$ whenever $\text{var}(f(Y))$ is finite. For what values of σ_p^2 is this variance actually finite?

Solution: Since $y_1, \ldots, y_n \sim q$ i.i.d., we have

$$\text{var}(\widehat{Z}) = \frac{1}{n^2} \sum_{i=1}^{n} \text{var}(f(y_i)) = \frac{1}{n} \text{var}(f(Y))$$

whenever $\text{var}(f(Y)) < \infty$. Now, since

$$\text{var}(f(Y)) = \mathbb{E}[f(Y)^2] - \mathbb{E}[f(Y)]^2 = \mathbb{E}[f(Y)^2] - Z_p^2,$$

we see that $\text{var}(f(Y))$ is finite if and only if $\mathbb{E}[f(Y)^2]$ is. Note that

$$\mathbb{E}[f(Y)^2] = \int_{-\infty}^{\infty} \frac{(p^*(y))^2}{q^2(y)} \, q(y) \, dy = \sqrt{2\pi} \int_{-\infty}^{\infty} \exp\left(\frac{y^2}{2} - \frac{y^2}{\sigma_p^2}\right) \, dy.$$

Therefore, $\text{var}(f(Y)) < \infty$ if and only if $\sigma_p < 2$.