

Maschinelles Lernen 1

Wintersemester 2008/2009

Blatt 10

Abgabe bis Mittwoch, 14. Dezember 2008, 14:00 Uhr

Ausarbeitung im Sekretariat FR6052, oder bei Mikio Braun, FR6058 (notfalls unter der Tür durchschieben),
praktischen Teil unter <https://ml01.service.tu-berlin.de/~mikio/pass.pl?conf=blatt10.conf>.

Aufgaben (Least-Squares Regression)

Ergänze den fehlenden Code im Programmskelett (jeweils 5 Punkte).

1. `center`: Zentriert eine Matrix so, dass die Spaltensummen anschließend null sind.
2. `lsr`: Berechnet den least-squares-fit (ohne Zentrieren).
3. `lsr_with_centering`: Berechnet den least-squares-fit (mit Zentrieren). Die Zentrierung muß so umgerechnet werden, dass man anschließend eine Funktion $g(x) = w'x + b$ erhält (d.h. zuerst wird zentriert, dann w ausgerechnet, und anschließend daraus das b).
4. `lsr_poly`: Berechnet den least-squares-fit für eine Basis der Monome

$$1, x, x^2, \dots, x^k,$$

mit Regularisierung der Quadratnorm des Parametervektors (ohne Zentrierung).

5. `poly_design_matrix`: Stellt die Designmatrix für die obige Polynombasis auf.
6. `plot_poly_fit`: Plottet den resultierenden Fit für das Polynom.

```
function sheet10

% generate some data
N = 50;
X = 2 * pi * rand(N, 1);
Y = fct1(X) + randn(N, 1);

% plot the data
hold off;
plot(X, Y, '.');
grid;

% plot the linear fit without centering
hold on;
W = lsr(X, Y)
plot_fit(X, Y, W, 0, 'r-');

% ... and with centering
[W, B] = lsr_with_centering(X, Y);
plot_fit(X, Y, W, B, 'b-');

% plot a polynomial fit for 4 different regularization constants
for lambda = [1, 10, 100, 1000]
    W = lsr_poly(4, lambda, X, Y);
    plot_poly_fit(4, W, X, Y, 'k-');
end

hold off;
```

```

legend('data', 'without centering', 'with centering', 'poly')

function Y = fct1(X)
Y = 0.2 * (X - pi - 1) .* (X - pi - 2.5) .* (X - pi + 1) .* X - 3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fill in your solutions below

% 1. Center the data
function X = center(X)
% ...

% 2. Compute the least-squares fit from X and Y
function W = lsr(X, Y)
% ...

% 3. Compute the centered version. Compute W and B accordingly.
function [W, B] = lsr_with_centering(X, Y)
XM = mean(X);
YM = mean(Y);
X = center(X);
Y = center(Y);

% ...

function plot_fit(X, Y, W, B, style)
X = sort(X);
Yh = W'*X + B;
plot(X, Yh, style);

% 4. Compute the least-squares fit using the function
% poly_design_matrix.
function W = lsr_poly(K, lambda, X, Y)
% ...

% 5. Compute the design matrix for polynomials from degree 0 to K
function PSI = poly_design_matrix(K, X)
% ...

% 6. Compute the prediction using for polynomials from degree 0 to K
function W = plot_poly_fit(K, W, X, Y, style)
X = sort(X);
% ...
plot(X, Yh, style);

```
