Regression

WEEK 4

Outline

Bias - variance tradeoff

Regression with regularization

Advanced technique in regression

Bias-Variance Tradeoff

hidden mechanism $y = f(x) + \varepsilon \quad \varepsilon \sim N(0, \sigma)$

data $(x_0, y_0), (x_1, y_1) \cdots$

we want to get $\hat{f}(x)$ or for any x_0, get $\hat{y}_0$

Bias-Variance Tradeoff (details)

$$
\begin{aligned}
MSE &= E[(f(x)-\hat{y})^2] \\
&= E[(f(x)-E[\hat{y}]+E[\hat{y}]-\hat{y})^2] \\
&= E[(f(x)-E[\hat{y}]+E[\hat{y}]-\hat{y})(f(x)-E[\hat{y}]+E[\hat{y}]-\hat{y})] \\
&= E[f(x)^2-f(x)E[\hat{y}]+f(x)E[\hat{y}]-f(x)\hat{y} \\
&\quad -E[\hat{y}]f(x)+E[\hat{y}]^2-E[\hat{y}]^2+E[\hat{y}]\hat{y} \\
&\quad +E[\hat{y}]f(x)-E[\hat{y}]^2+E[\hat{y}]^2-E[\hat{y}]\hat{y} \\
&\quad -\hat{y}f(x)+\hat{y}E[\hat{y}]-\hat{y}E[\hat{y}]+\hat{y}^2]
\end{aligned}
$$

$$
\begin{aligned}
&= E[\hat{y}^2-2E[\hat{y}]\hat{y}+E[\hat{y}]^2] + E[E[\hat{y}]^2-2E[\hat{y}]f(x)+f(x)^2] \\
&\quad + E[f(x)E[\hat{y}]-f(x)\hat{y}-E[\hat{y}]^2+E[\hat{y}]\hat{y} \\
&\quad\quad +E[\hat{y}]f(x)-E[\hat{y}]^2-\hat{y}f(x)+\hat{y}E[\hat{y}]] \\
&= E[(\hat{y}-E[\hat{y}])^2] \cdots\cdots \text{varience} \\
&\quad + E[(E[\hat{y}]-f(x))^2] \cdots\cdots \text{bias} \\
&\quad + f(x)E[\hat{y}]-f(x)E[\hat{y}]-E[\hat{y}]^2+E[\hat{y}]E[\hat{y}] \\
&\quad + E[\hat{y}]f(x)-E[\hat{y}]^2-E[\hat{y}]f(x)+E[\hat{y}]E[\hat{y}] \\
&= \text{varience} + \text{bias}
\end{aligned}
$$

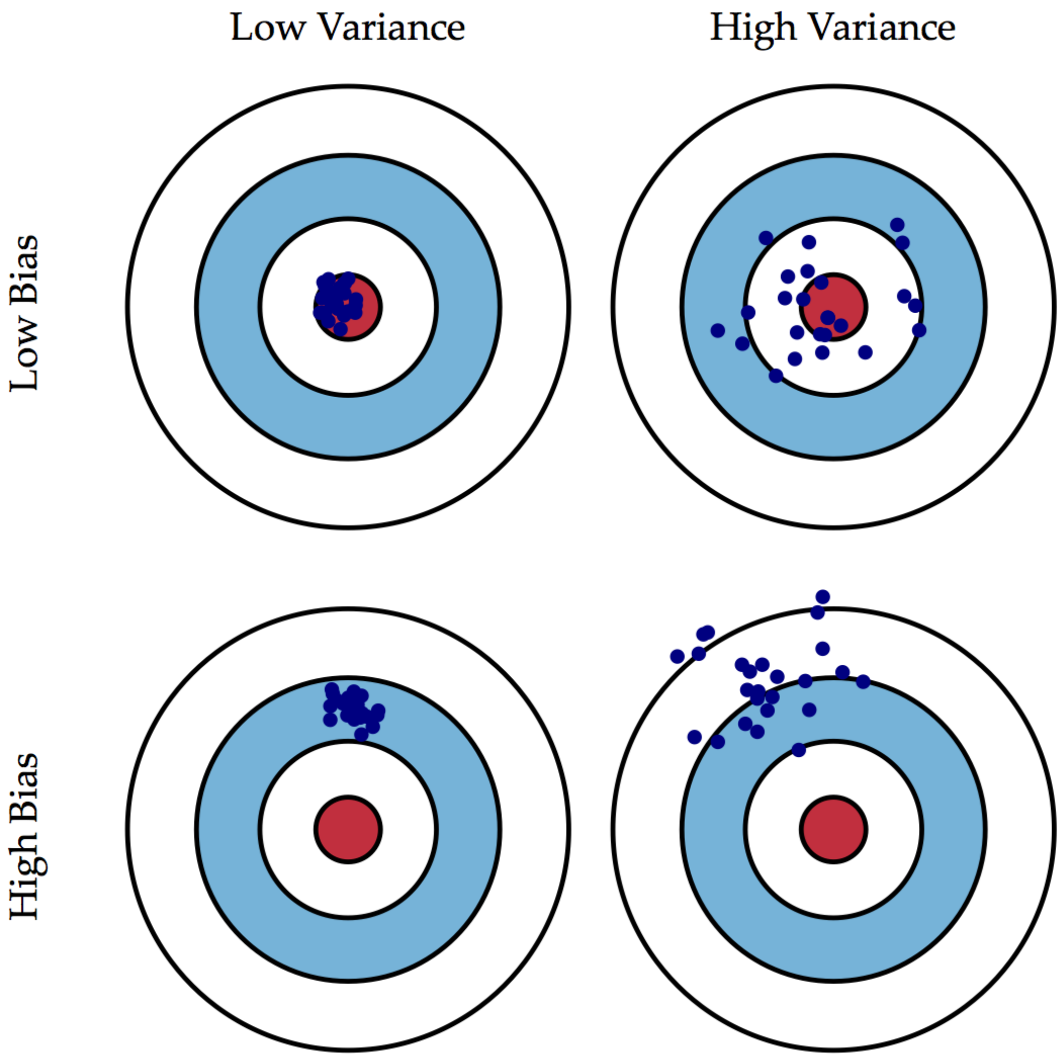
Bias-Variance Tradeoff (details)

$$f(x) \to f(x) + \varepsilon$$

$$E[(E[\hat{y}] - f(x) - \varepsilon)^2]$$

$$= E[(E[\hat{y}] - f(x))^2 - 2 \cdot \varepsilon \cdot (E[\hat{y}] - f(x)) + \varepsilon^2]$$

$$= E[(E[\hat{y}] - f(x))^2] + 0 + E[\varepsilon^2]$$


Low Variance
High Variance
Low Bias
High Bias

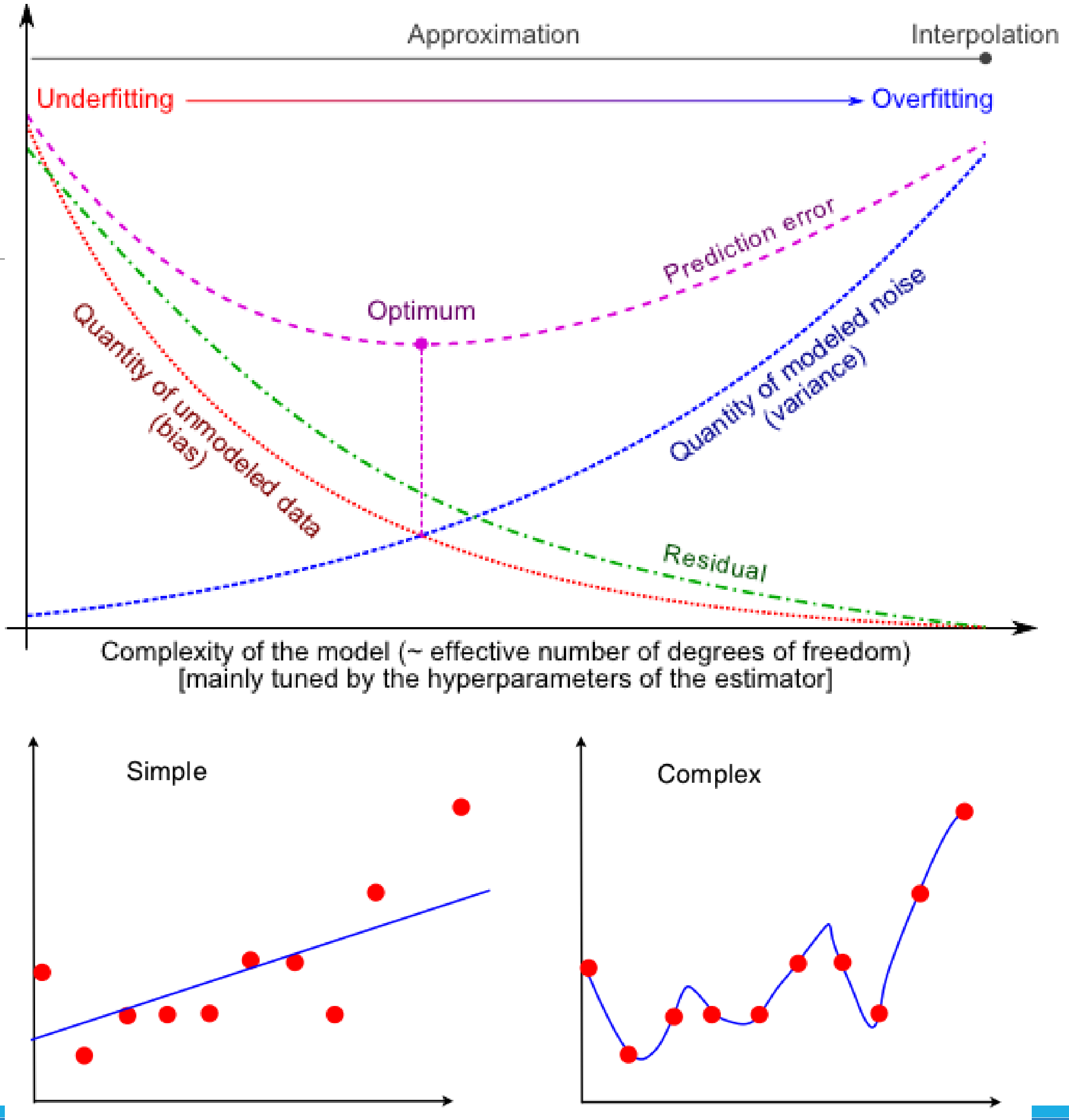
Under & Over fitting

Under fit = high bias

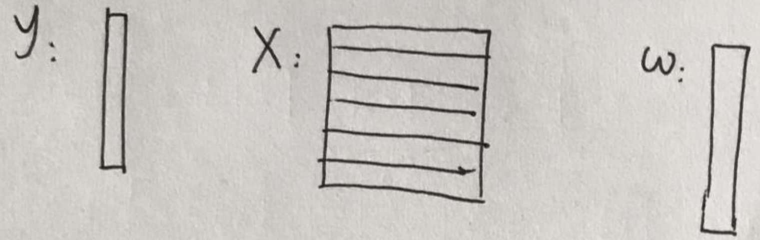
Over fit = high variance

Address over fitting:

1) Reduce number of features
2) Regularization



Linear regression (analytical solution)



$y = X\omega$

$$\text{error} = (y - X\omega)'(y - X\omega)$$

$$= y'y - (X\omega)'y - y'(X\omega) + (X\omega)'(X\omega)$$

[derivative w.r.t. ω should be zero, ignore $y'y$ (not related wrt ω)]

$$= \omega'(X'X)\omega - 2y'X\omega$$

Linear regression (analytical solution)

[$(Xw)'y = w'X'y =$ $\quad$ $y'Xw =$, same!]

$$= w'Aw - b'w \qquad [A = X'X \;,\; b = -2X'y]$$

$$\frac{\partial \, error}{\partial w} = (A+A')w + b = 0$$

$$\Downarrow$$

$$(X'X + (X'X)')w - 2X'y = 0 \qquad [X'X \text{ is symmetric } \; X'X = (X'X)' \,]$$

$$X'Xw = X'y$$

$$w = (X'X)^{\dagger} X'y$$

Linear regression (derivative only)

i : all features

j : all points.

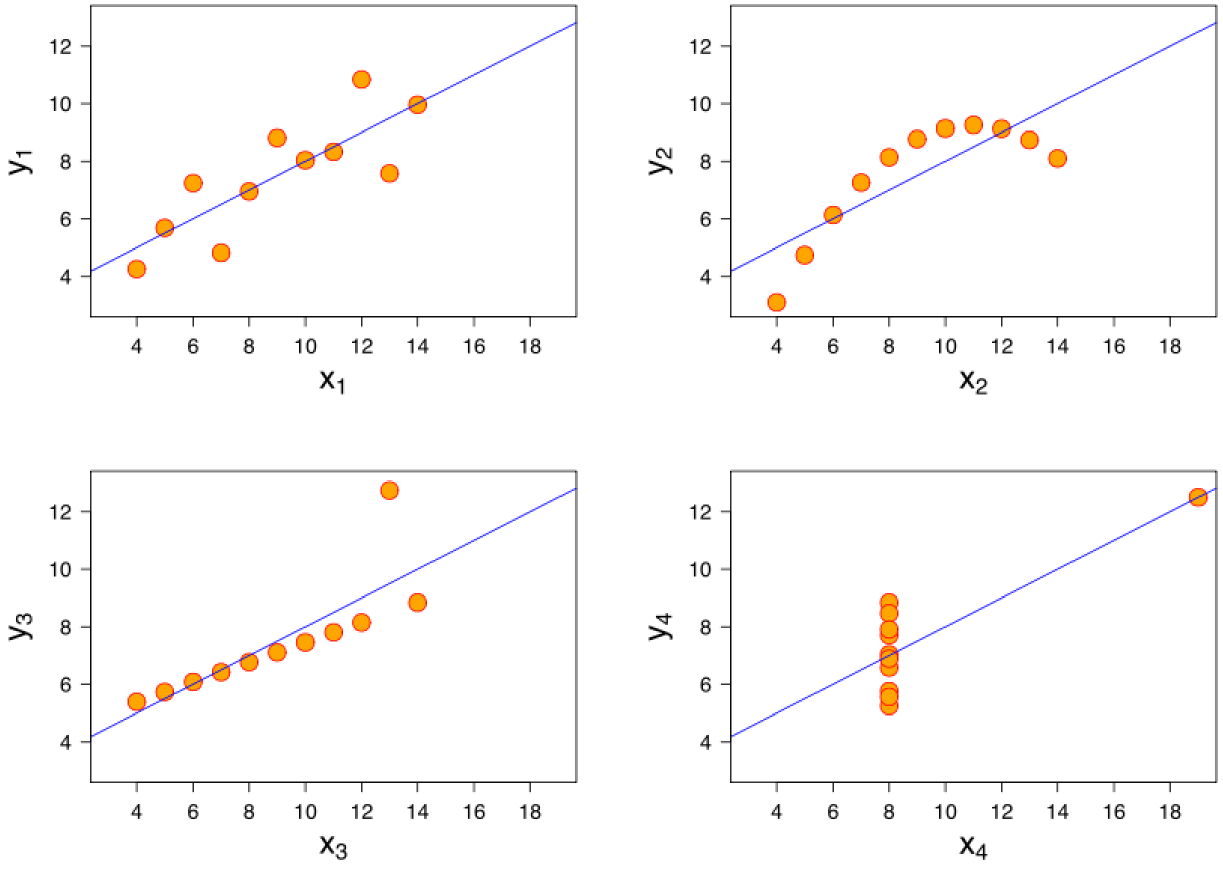
$$\text{error} = \sum_j \left(y_j - \sum_i x_{ij}\,\omega_i\right)^2$$

$$\text{derivative} = \frac{\partial\,\text{error}}{\partial \omega_i} = \sum_j 2 \cdot \left(y_j - \sum_i x_{ij}\,\omega_i\right) \cdot \left(-x_{ij}\right)$$

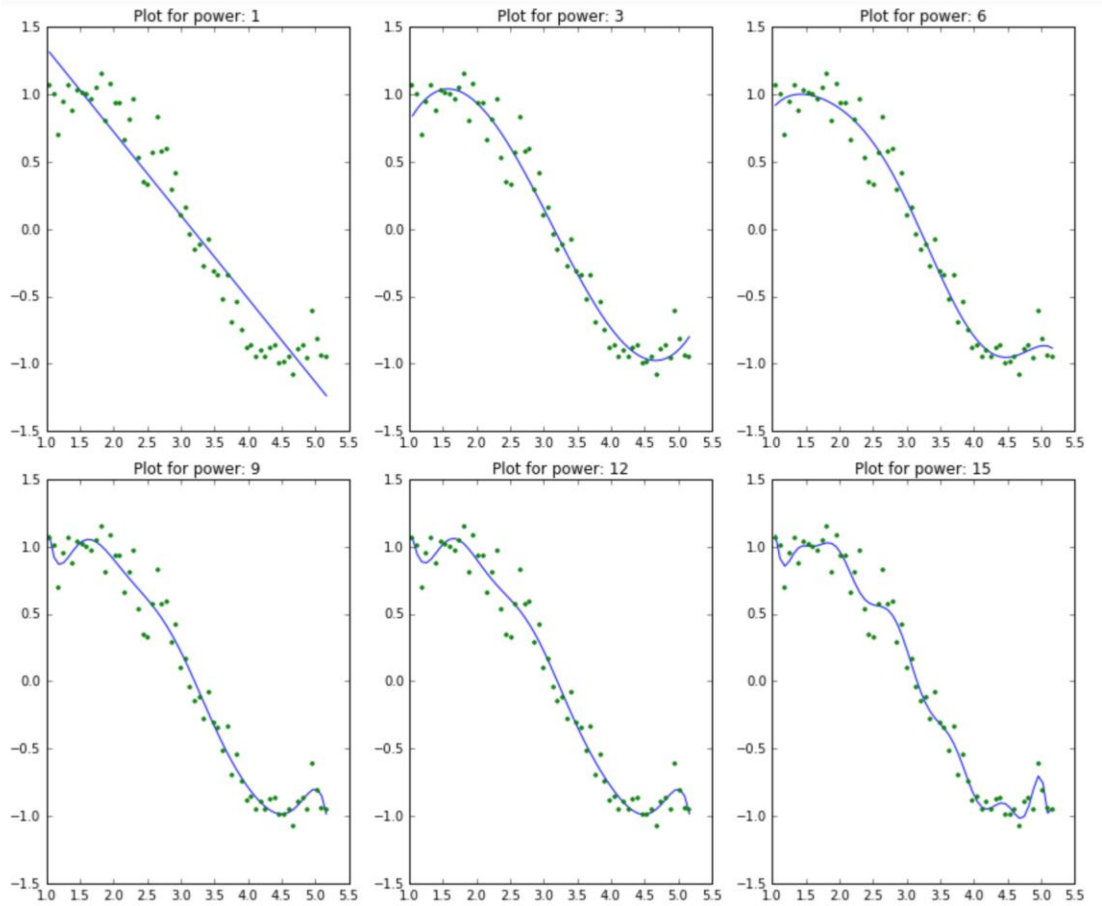
$$= -2 \sum_j (\Delta y_j) \cdot x_{ij}$$

$\Delta y_j = y_j - \sum_i x_{ij}\,\omega_i \Rightarrow$ difference between actual & prediction.

Linear regression (possible problem)



Linear regression (overfitting)



Linear regression (why need regularization)

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	c
model_pow_1	3.3	2	-0.62	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_2	3.3	1.9	-0.58	-0.006	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_3	1.1	-1.1	3	-1.3	0.14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_4	1.1	-0.27	1.7	-0.53	-0.036	0.014	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_5	1	3	-5.1	4.7	-1.9	0.33	-0.021	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_6	0.99	-2.8	9.5	-9.7	5.2	-1.6	0.23	-0.014	NaN	NaN	NaN	NaN	NaN	N
model_pow_7	0.93	19	-56	69	-45	17	-3.5	0.4	-0.019	NaN	NaN	NaN	NaN	N
model_pow_8	0.92	43	-1.4e+02	1.8e+02	-1.3e+02	58	-15	2.4	-0.21	0.0077	NaN	NaN	NaN	N
model_pow_9	0.87	1.7e+02	-6.1e+02	9.6e+02	-8.5e+02	4.6e+02	-1.6e+02	37	-5.2	0.42	-0.015	NaN	NaN	N
model_pow_10	0.87	1.4e+02	-4.9e+02	7.3e+02	-6e+02	2.9e+02	-87	15	-0.81	-0.14	0.026	-0.0013	NaN	N
model_pow_11	0.87	-75	5.1e+02	-1.3e+03	1.9e+03	-1.6e+03	9.1e+02	-3.5e+02	91	-16	1.8	-0.12	0.0034	N
model_pow_12	0.87	-3.4e+02	1.9e+03	-4.4e+03	6e+03	-5.2e+03	3.1e+03	-1.3e+03	3.8e+02	-80	12	-1.1	0.062	-
model_pow_13	0.86	3.2e+03	-1.8e+04	4.5e+04	-6.7e+04	6.6e+04	-4.6e+04	2.3e+04	-8.5e+03	2.3e+03	-4.5e+02	62	-5.7	0
model_pow_14	0.79	2.4e+04	-1.4e+05	3.8e+05	-6.1e+05	6.6e+05	-5e+05	2.8e+05	-1.2e+05	3.7e+04	-8.5e+03	1.5e+03	-1.8e+02	1
model_pow_15	0.7	-3.6e+04	2.4e+05	-7.5e+05	1.4e+06	-1.7e+06	1.5e+06	-1e+06	5e+05	-1.9e+05	5.4e+04	-1.2e+04	1.9e+03	-

Regression with regularization

Extend the cost function from regular RSS to RSS + extra terms

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

$$\min_{w} ||Xw - y||_2{}^2$$

Total cost =

measure of fit + measure of magnitude of coefficients

Ridge regression

Introduce square of coefficient into the equation

$$\min_{w} ||Xw - y||_2{}^2 + \alpha ||w||_2{}^2$$

Total cost =

$$\underbrace{\text{measure of fit}}_{\text{RSS}(\mathbf{w})} + \underbrace{\text{measure of magnitude of coefficients}}_{||\mathbf{w}||_2^2}$$

$$\text{RSS}(\mathbf{w}) + \lambda ||\mathbf{w}||_2^2$$

tuning parameter = balance of fit and magnitude

Lasso regression

Introduce square of coefficient into the equation

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$$

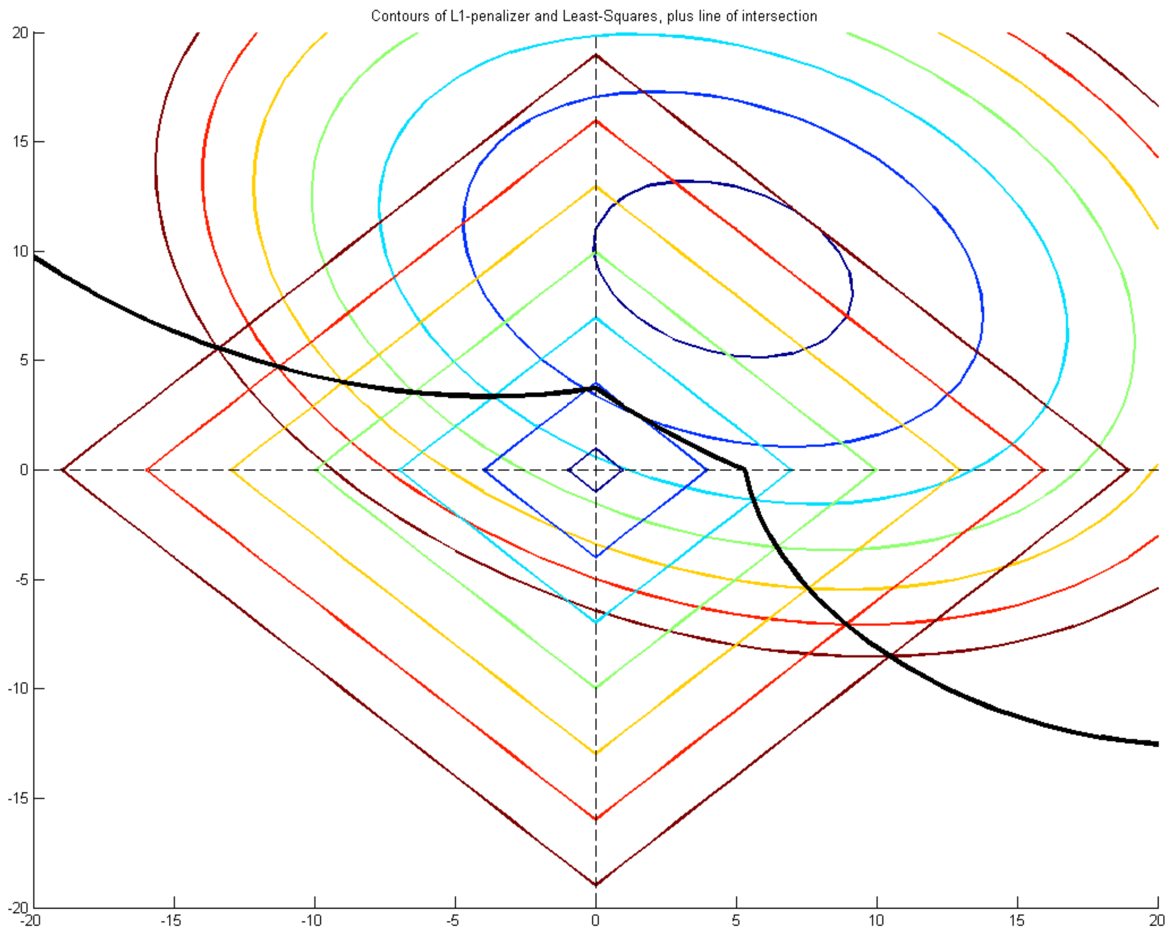
Total cost =

$$\underbrace{\text{measure of fit}}_{RSS(\mathbf{w})} + \lambda \underbrace{\text{measure of magnitude of coefficients}}_{||\mathbf{w}||_1 = |w_0| + \ldots + |w_D|}$$

$$RSS(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

tuning parameter = balance of fit and sparsity

Why zero for Lasso?



Think in the following geometry.

Regularization term on Ridge is an eclipse; on lasso is a prismatic

Elastic net

How it is different from Lasso and Ridge

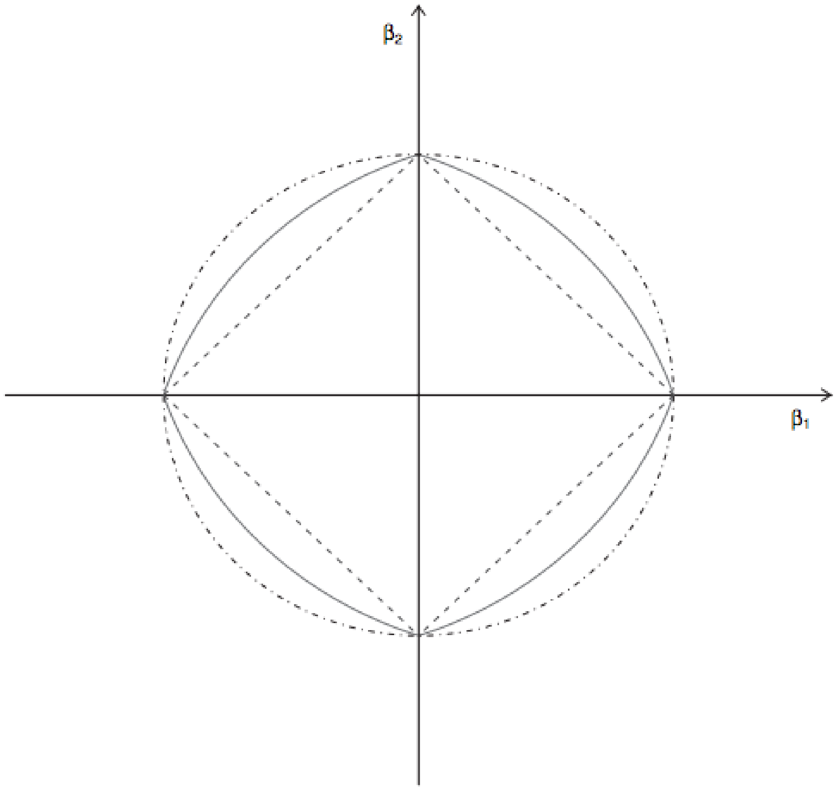
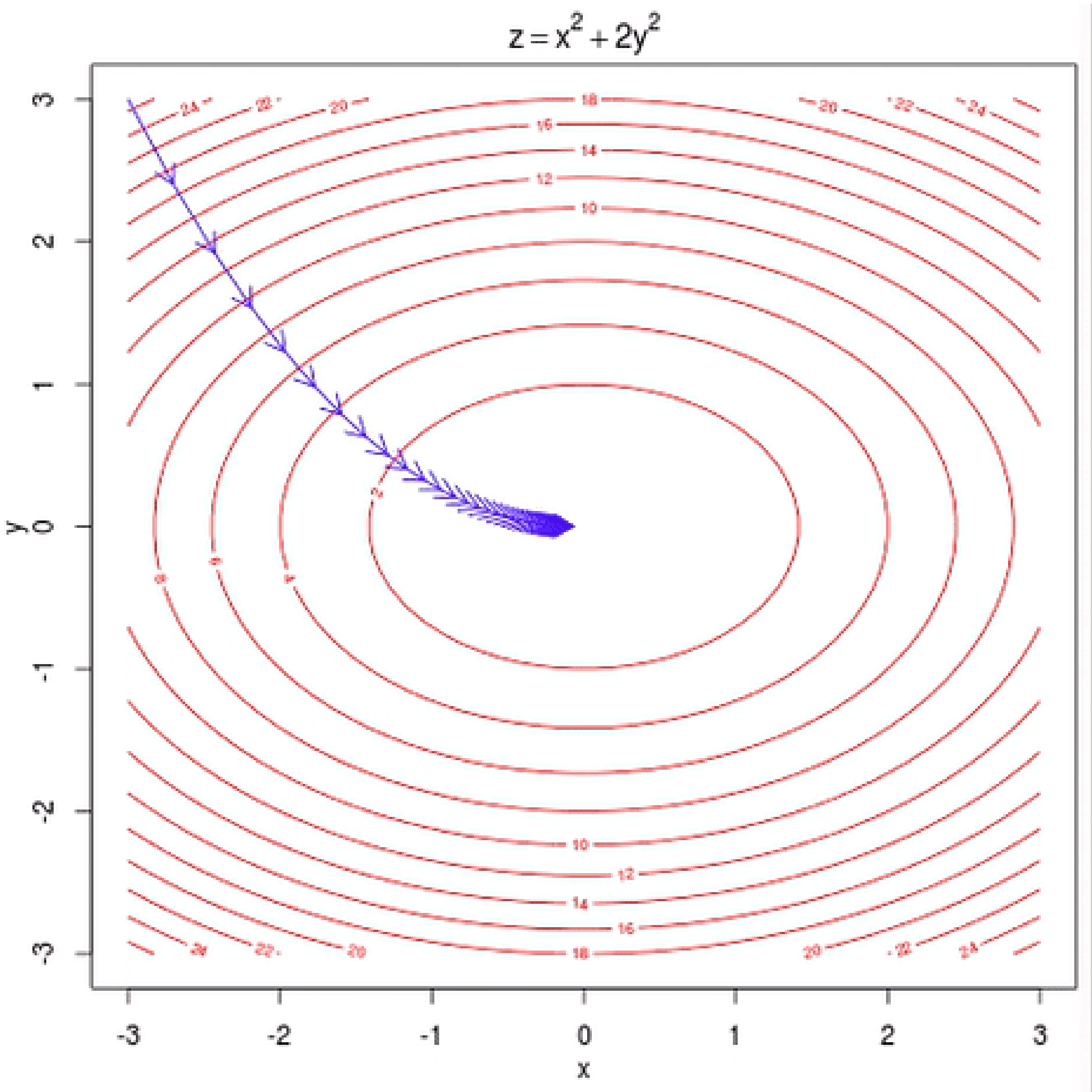


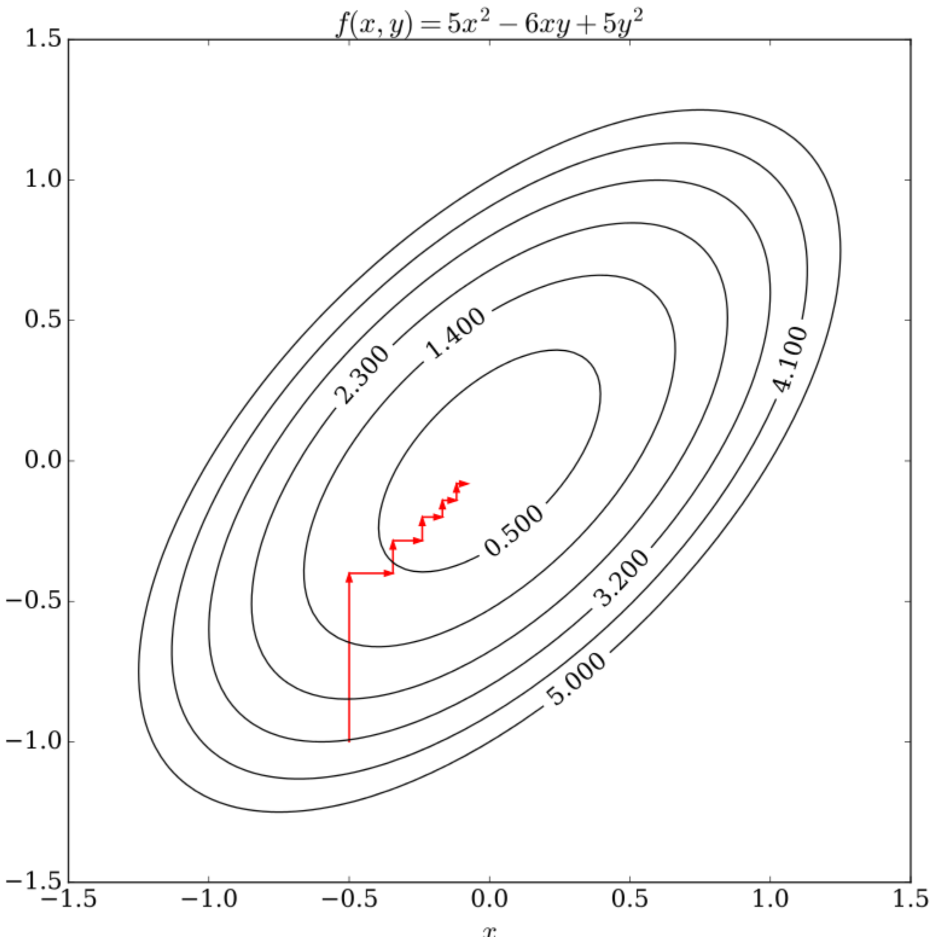
Fig. 1. Two-dimensional contour plots (level 1) (· - - - - -, shape of the ridge penalty; - - - - - - -, contour of the lasso penalty; ———, contour of the elastic net penalty with $\alpha = 0.5$): we see that singularities at the vertices and the edges are strictly convex; the strength of convexity varies with α

Gradient Descendent

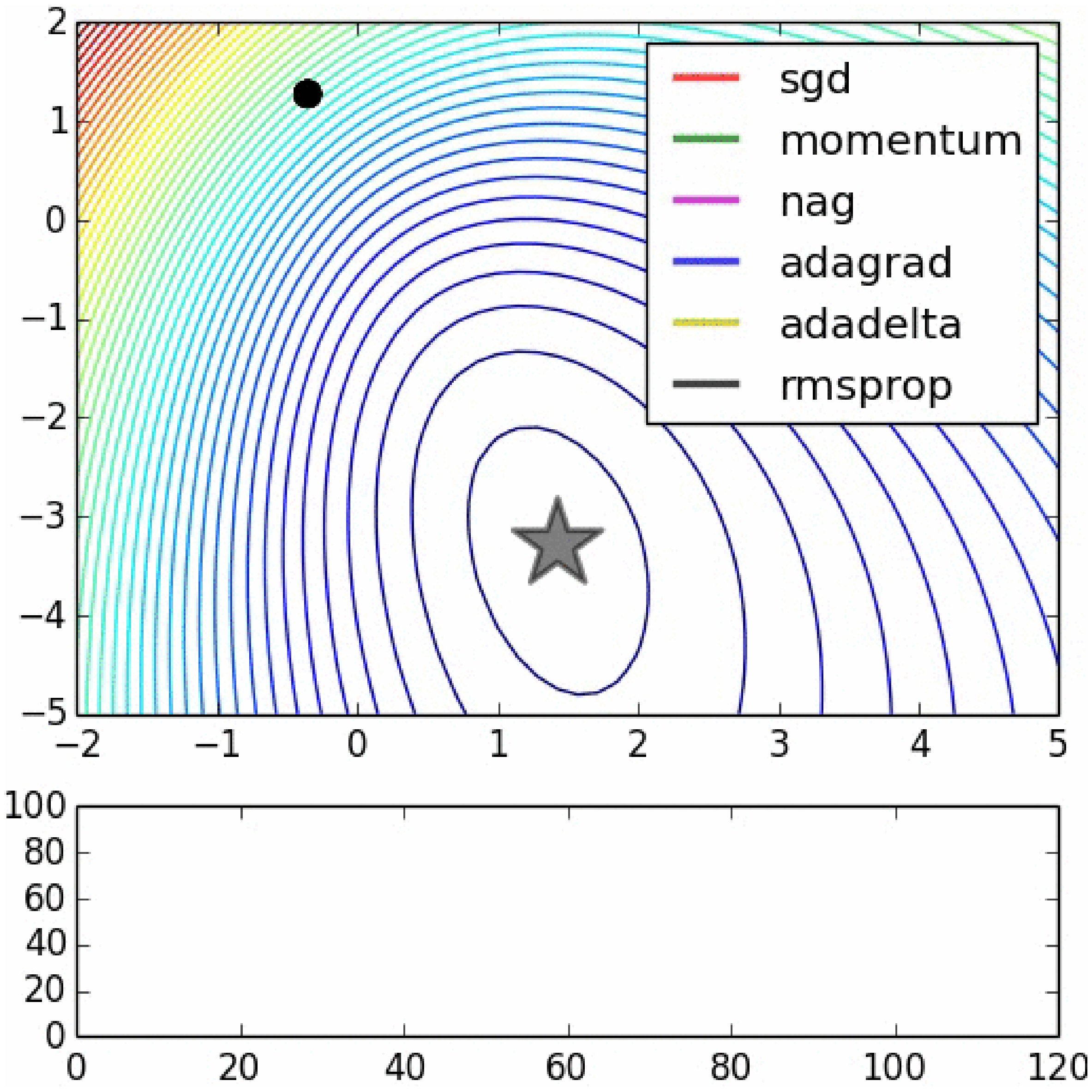


Coordinated Descendent

- Choose an initial parameter vector $\mathbf{x}$.
- Until convergence is reached, or for some fixed number of iterations:
  - Choose an index i from 1 to n.
  - Choose a step size α.
  - Update x_i to $x_i - \alpha \frac{\partial F}{\partial x_i}(\mathbf{x})$.



Stochastic Gradient Descendent



Random sample consensus (RANSAC)

```
iterations = 0
bestfit = nul
besterr = something really large
while iterations < k {
    maybeinliers = n randomly selected values from data
    maybemodel = model parameters fitted to maybeinliers
    alsoinliers = empty set
    for every point in data not in maybeinliers {
        if point fits maybemodel with an error smaller than t
             add point to alsoinliers
    }
    if the number of elements in alsoinliers is > d {
        % this implies that we may have found a good model
        % now test how good it is
        bettermodel = model parameters fitted to all points in maybeinliers and alsoinliers
        thiserr = a measure of how well model fits these points
        if thiserr < besterr {
            bestfit = bettermodel
            besterr = thiserr
        }
    }
    increment iterations
}
return bestfit
```