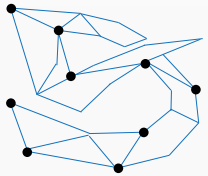
Data Application Lab

Regression

Outline

1. Regression
2. Bias-Variance Trade-off
3. Advanced Techniques
4. Q & A



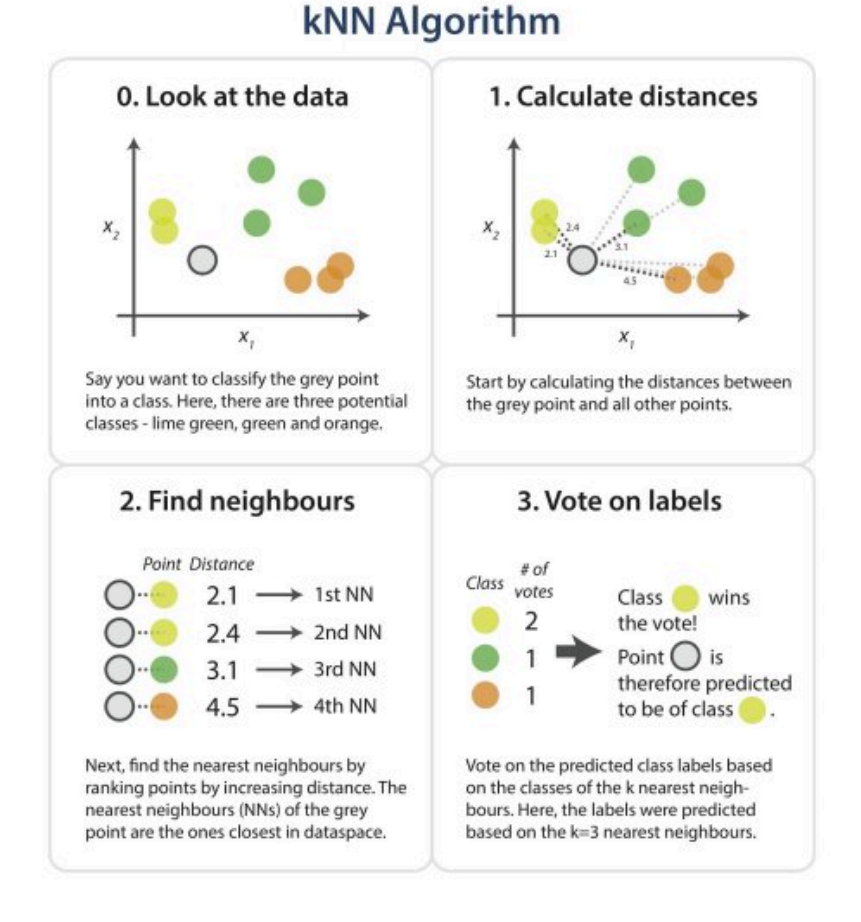
Regression

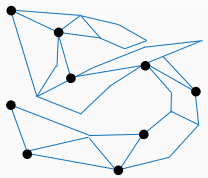
1

KNN

- Training stage
- Step 1. load all the training data (that's it!)

- Testing stage
- Step 1. given an Xnew
- Step 2. calculate distance between Xnew with all training data
- Step 3. find nearest k neighbors
- Step 4. vote on label or calculate average





KNN

Potential improvements?

Accuracy

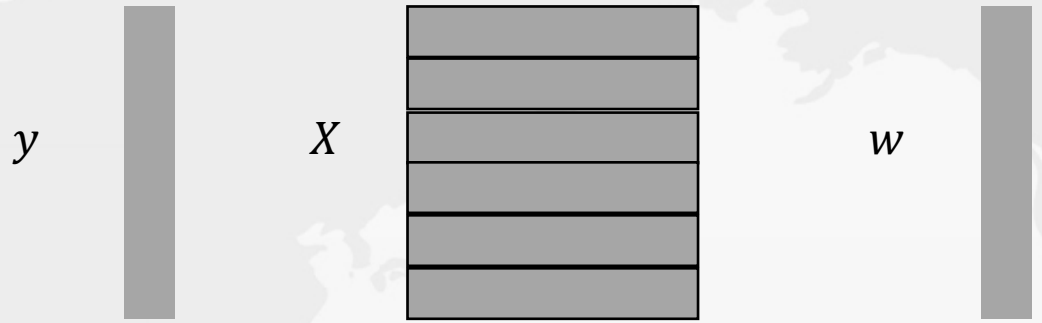
1. Could it be wrong under any situation?
    1. Dimension? => dimension reduction to smaller dimension
    2. Imbalanced data? => add weight to point based on class imbalance

Speed

1. How is the testing speed goes?
    1. Testing speed => decrease training data
    2. Calculate distance => k-d tree

Linear Regression

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_p x_p + w_0 = Xw$$



$$\begin{aligned} mse = (y - Xw)^2 &= (y - Xw)^T (y - Xw) \\ &= y^T y - (Xw)^T y - y^T (Xw) - (Xw)^T (Xw) \\ &= w^T (X^T X) w - 2\, X^T \mathrm{y} \mathrm{w} + y^T y \\ &= w^T A w + b w \end{aligned}$$

$A = X^T X, \qquad b = -2X^T y$

Object: find w to minimize mse

Linear Regression

$$\min_{w} w^T Aw + bw$$

$$\frac{\partial mse}{\partial w} = 2Aw + b = 0$$

$$\Rightarrow \quad \mathrm{X}^T Xw = X^T y$$

$$\Rightarrow \quad \mathrm{w} = (X^T X)^{-1} X^T y$$

Linear Regression

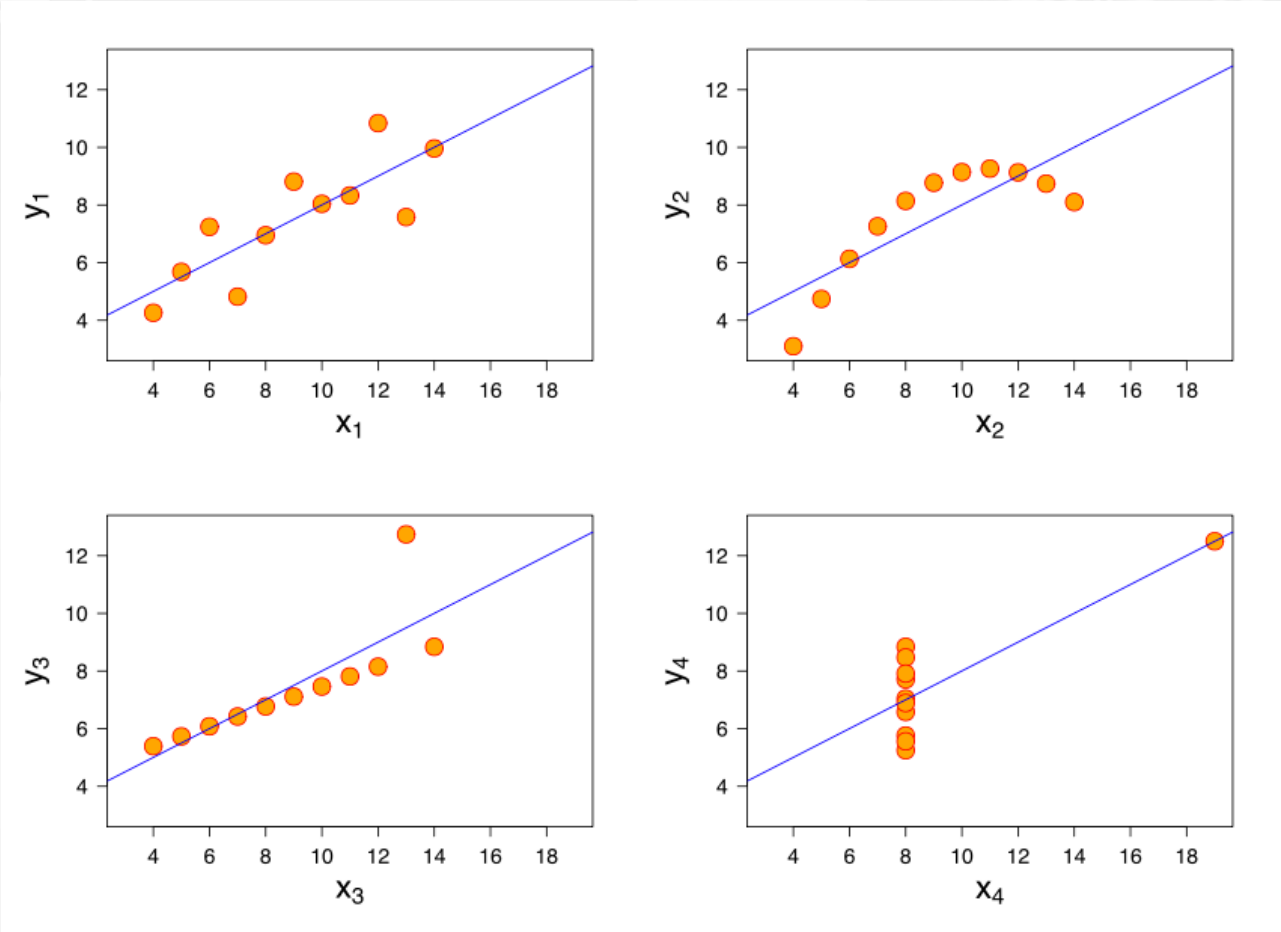
Gradient Descent: $w^{t+1} = w^t - \lambda \frac{\partial err}{\partial w}$

$$err = \sum_{j=1}^{n} (y_j - \sum_{i=1}^{p+1} X_{ji} w_i)^2$$

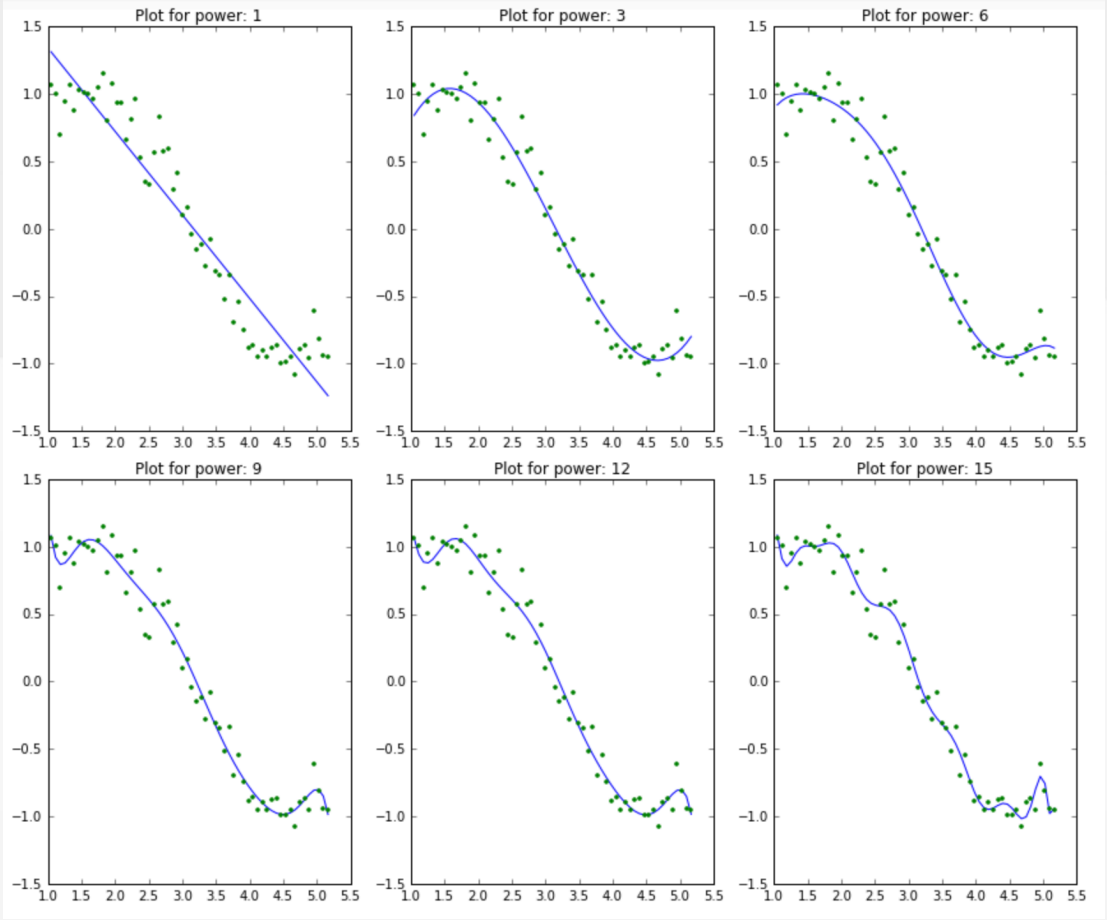
$$\frac{\partial err}{\partial w_i} = \sum_{j} -2(y_j - \sum_{i} X_{ji} w_i) X_{ji}$$

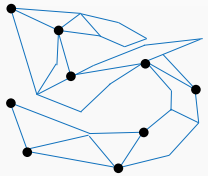
$$= -2 \sum_j err_j \cdot X_{ji}$$

Linear Regression



Linear Regression





Bias Variance Trade-off

1

Bias Variance

Hidden mechanism: $y = f(x) + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2)$

Data: $D = \{(x_1, y_1), \ldots, (x_i, y_i)\}$

Given data sample D -> find best fit $\tilde{f}(x, D)$

Question: if we sample another data set D_1, what's the expected error (variance)?

Bias Variance

$$E_D\left[\left(\tilde{f}(x,D)-y\right)^2\right] = E_D\left[\left(\tilde{f}(x,D)-\bar{f}(x)+\bar{f}(x)-y\right)^2\right] \qquad \bar{f}(x) = E_D[\tilde{f}(x,D)]$$

$$= E_D\left[\left(\tilde{f}(x,D)-\bar{f}(x)\right)^2\right] + E_D\left[\left(\bar{f}(x)-y\right)^2\right] + \cancel{2E_D\left[\left(\tilde{f}(x,D)-\bar{f}(x)\right)\left(\bar{f}(x)-y\right)\right]}$$

$$= E_D\left[\left(\tilde{f}(x,D)-\bar{f}(x)\right)^2\right] + E_D\left[\left(\bar{f}(x)-f(x)+f(x)-y\right)^2\right]$$

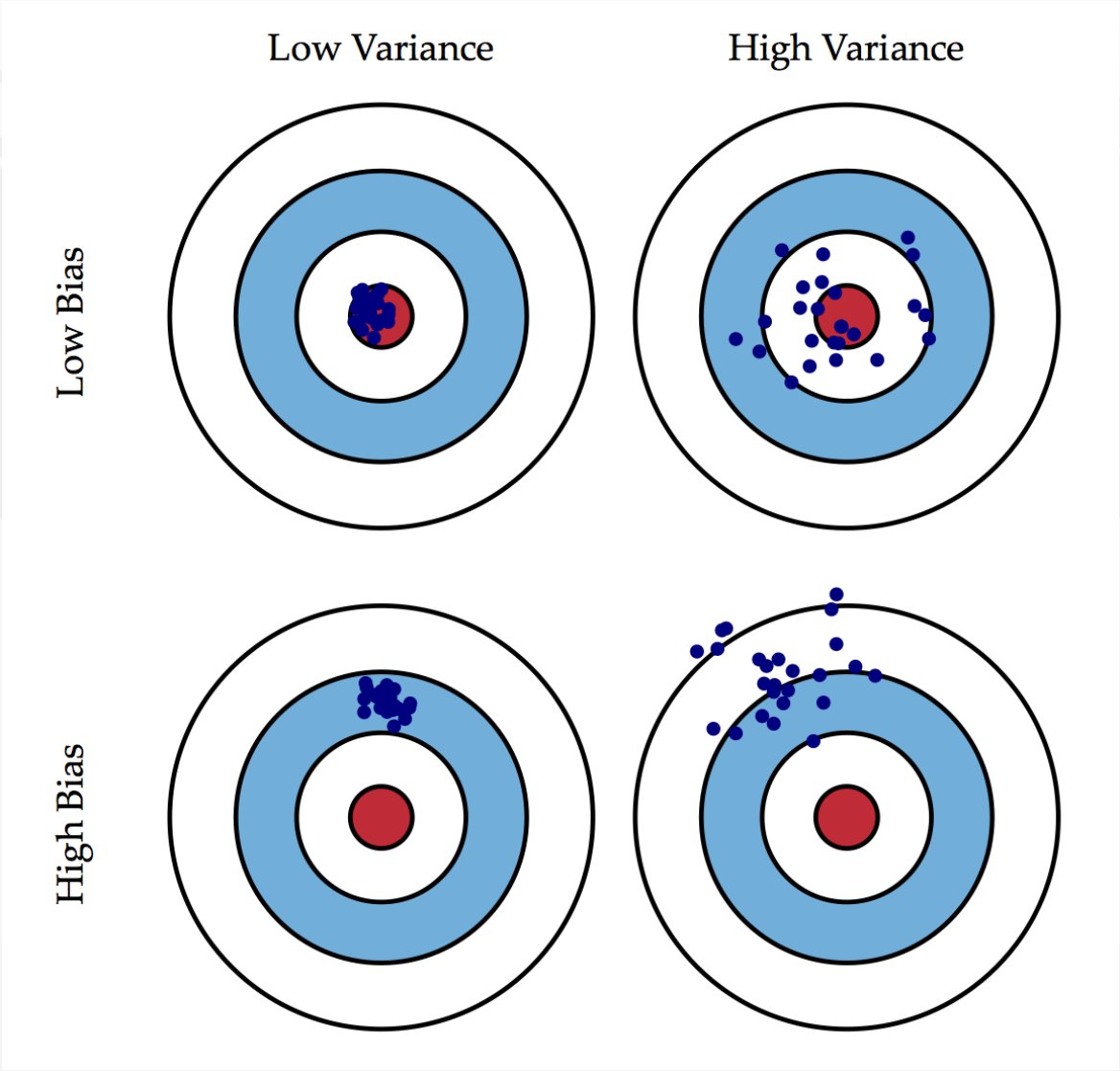
$$= E_D\left[\left(\tilde{f}(x,D)-\bar{f}(x)\right)^2\right] + E_D\left[\left(\bar{f}(x)-f(x)\right)^2\right] + E_D[(f(x)-y)^2] + \cancel{2E_D\left[\left(\bar{f}(x)-f(x)\right)\left(f(x)-y\right)\right]}$$

$$= \quad var(x) \quad + \quad bias^2(x) \quad + \quad \sigma^2$$

$$var(x) = E_D\left[\left(\tilde{f}(x,D)-\bar{f}(x)\right)^2\right]$$

$$bias^2(x) = E_D\left[\left(\bar{f}(x)-f(x)\right)^2\right]$$

Bias Variance

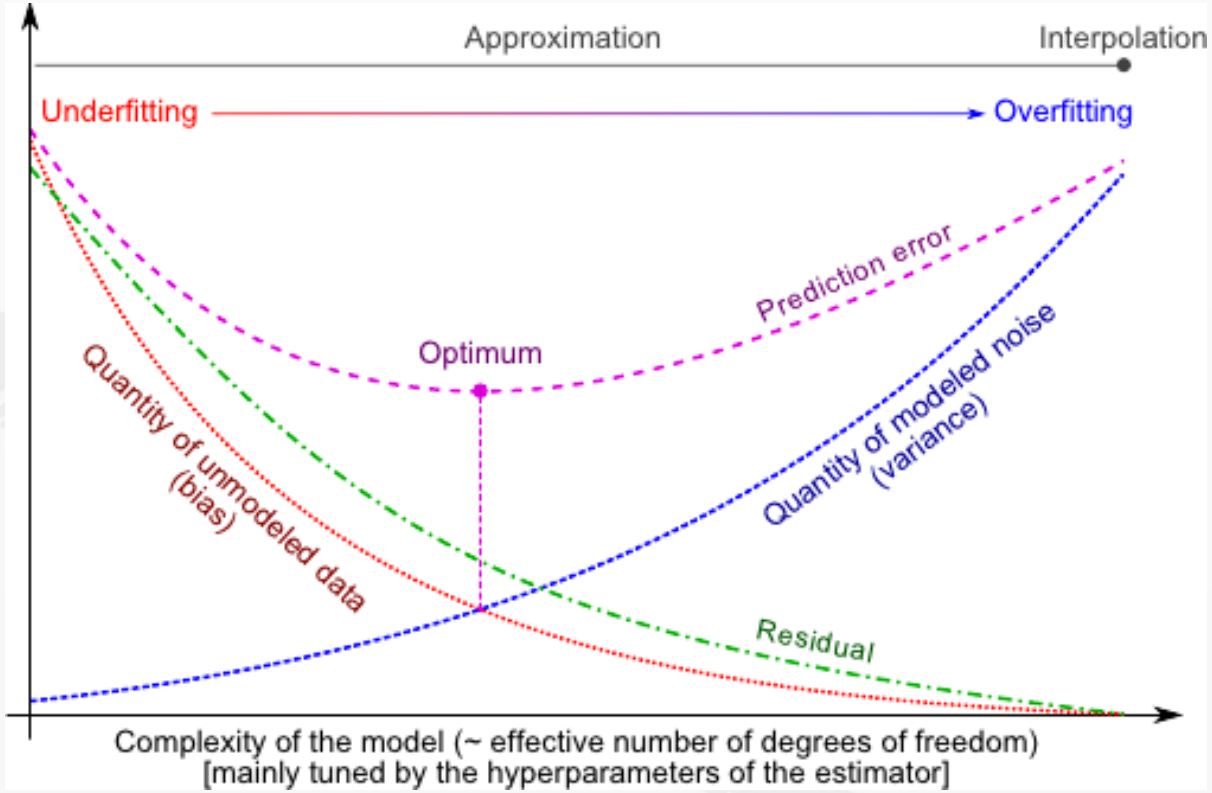


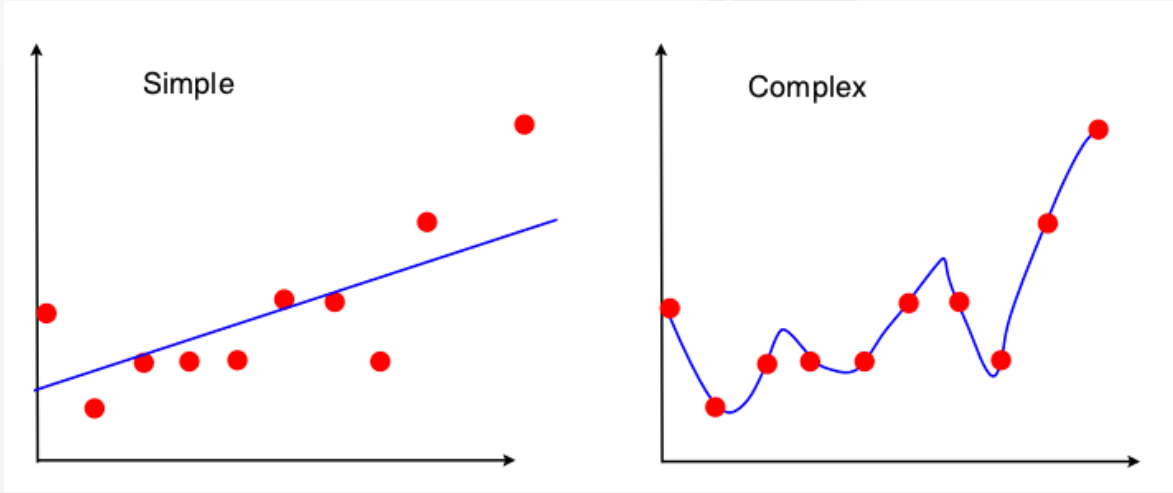
Under fit = high bias

Over fit = high variance

Address over fitting:

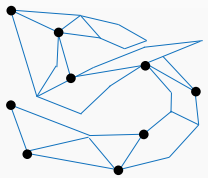
1) Reduce number of features
2) Regularization





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	-(



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

Regularization - Ridge

Introduce square of coefficient into the equation

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

Total cost =

measure of fit + measure of magnitude of coefficients

RSS(**w**) $||\mathbf{w}||_2^2$

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

tuning parameter = balance of fit and magnitude

Regularization - Lasso

Introduce square of coefficient into the equation

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

Total cost =

measure of fit + λ measure of magnitude of coefficients

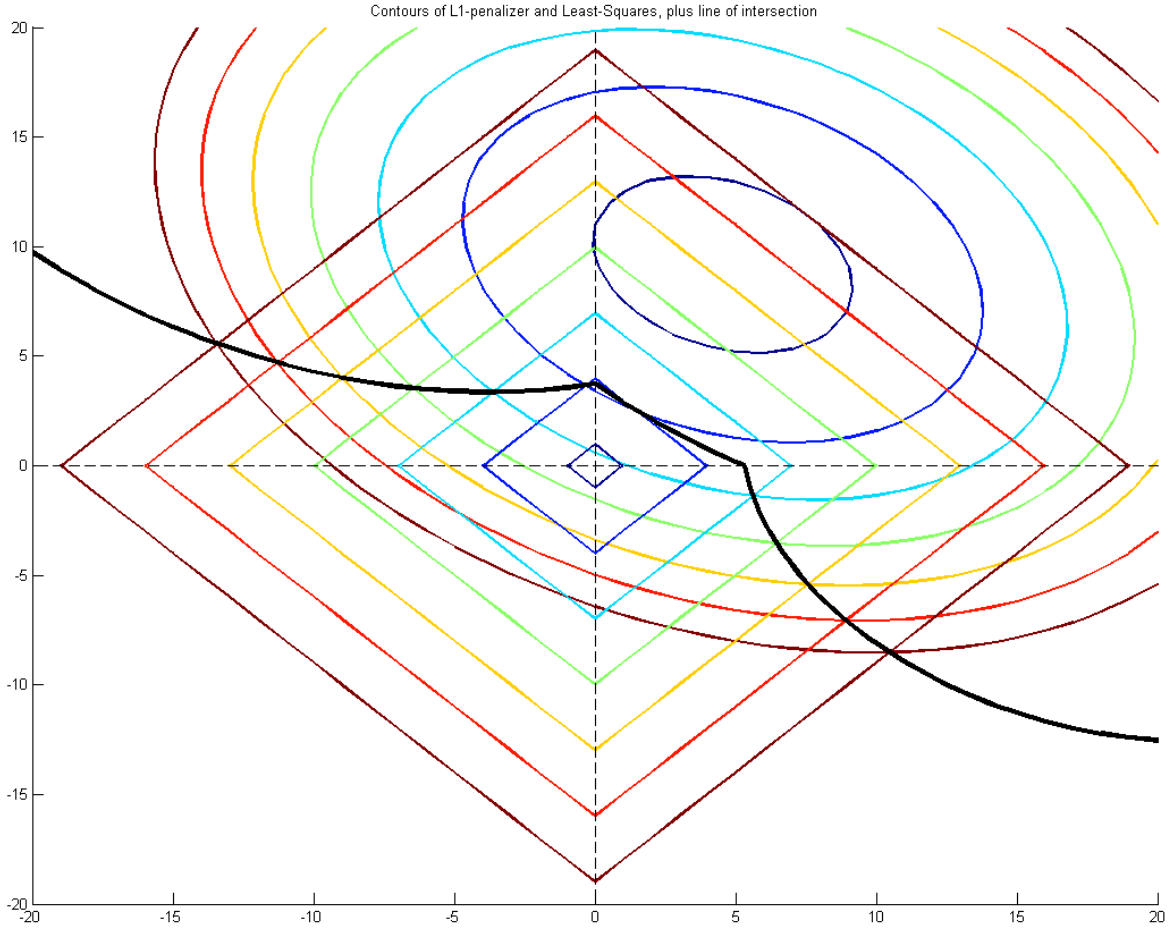
RSS(**w**)

$||\mathbf{w}||_1 = |w_0| + \ldots + |w_D|$

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

tuning parameter = balance of fit and **sparsity**

Regularization



- Think in the following geometry.
- Regularization term on Ridge is an eclipse; on lasso is a prismatic

Regularization – Elastic Net

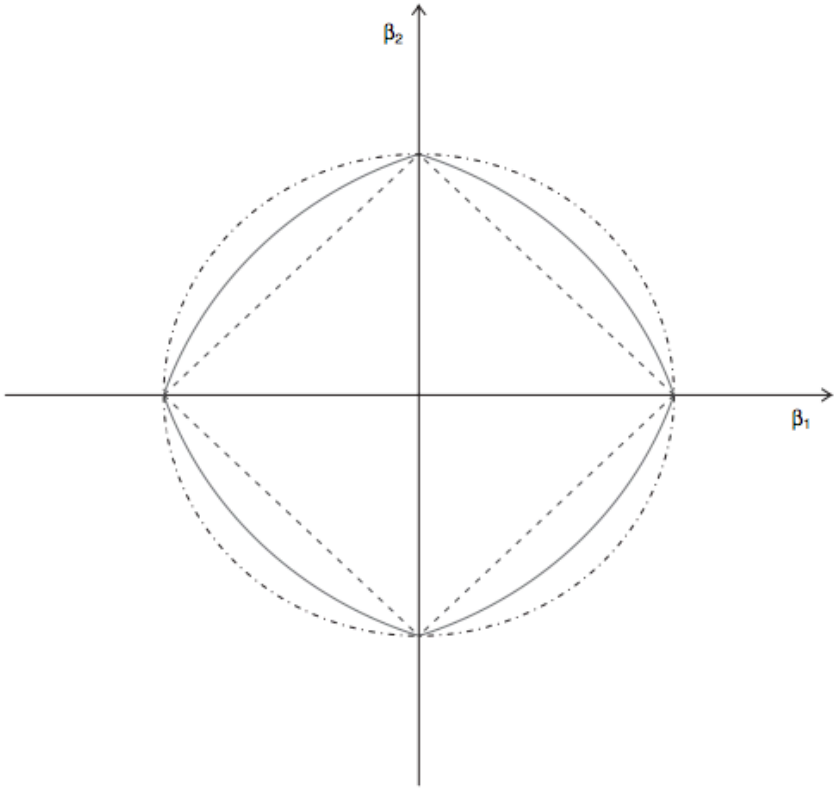
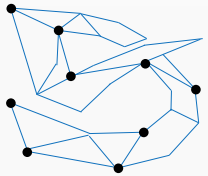


Fig. 1. Two-dimensional contour plots (level 1) (· - - - - -, shape of the ridge penalty; -------, contour of the lasso penalty; ———, contour of the elastic net penalty with $\alpha = 0.5$): we see that singularities at the vertices and the edges are strictly convex; the strength of convexity varies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

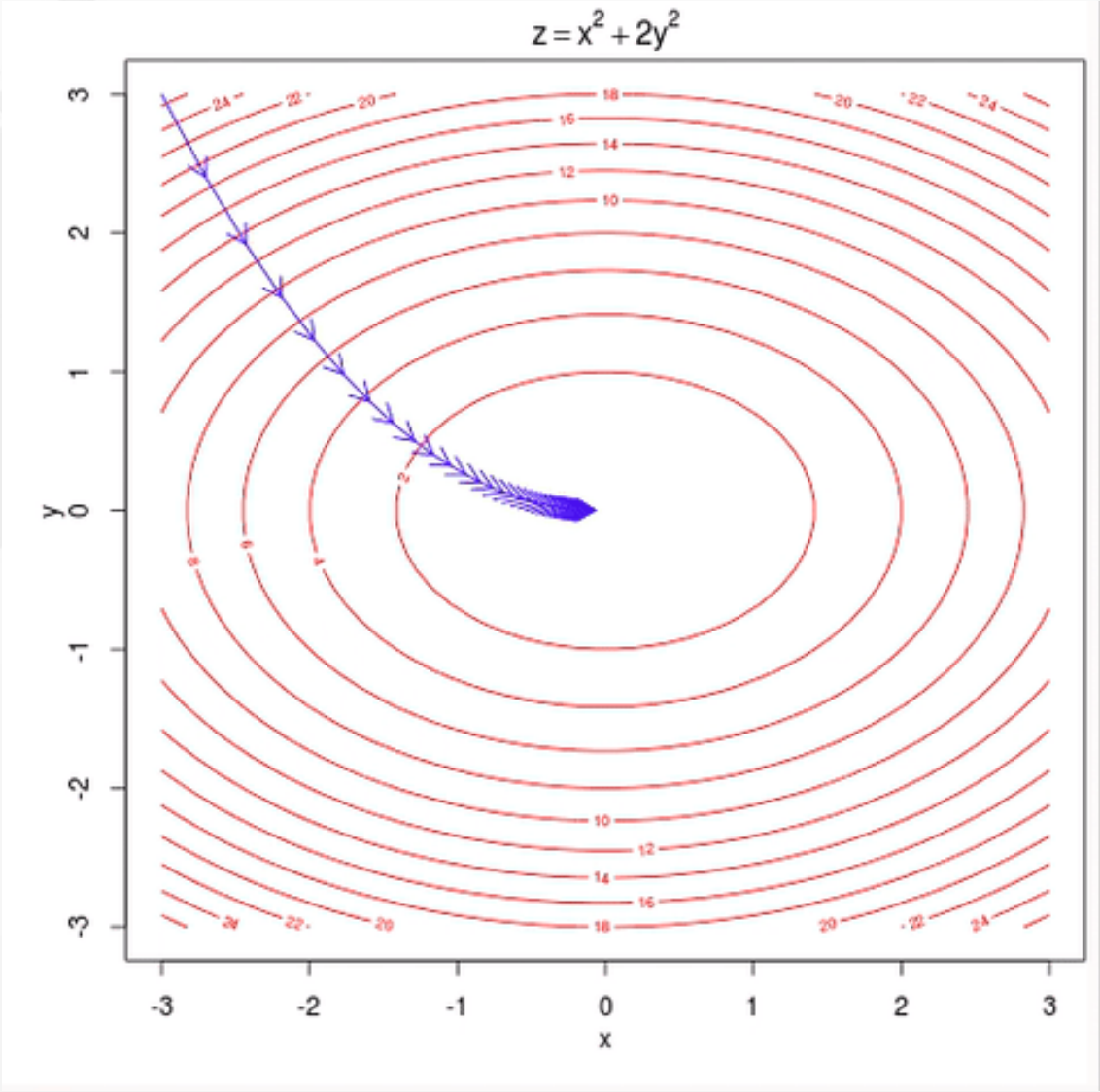
- How it is different from Lasso and Ridge



Advanced Technique

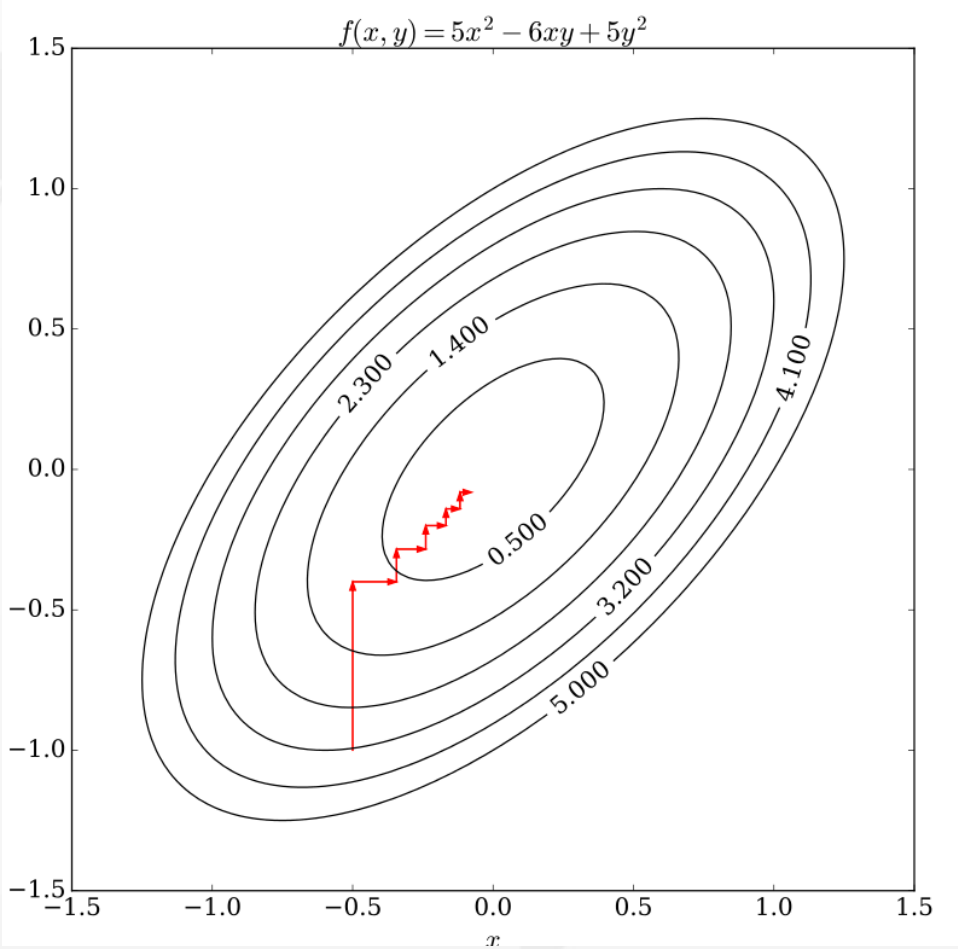
3

Gradient Descendent

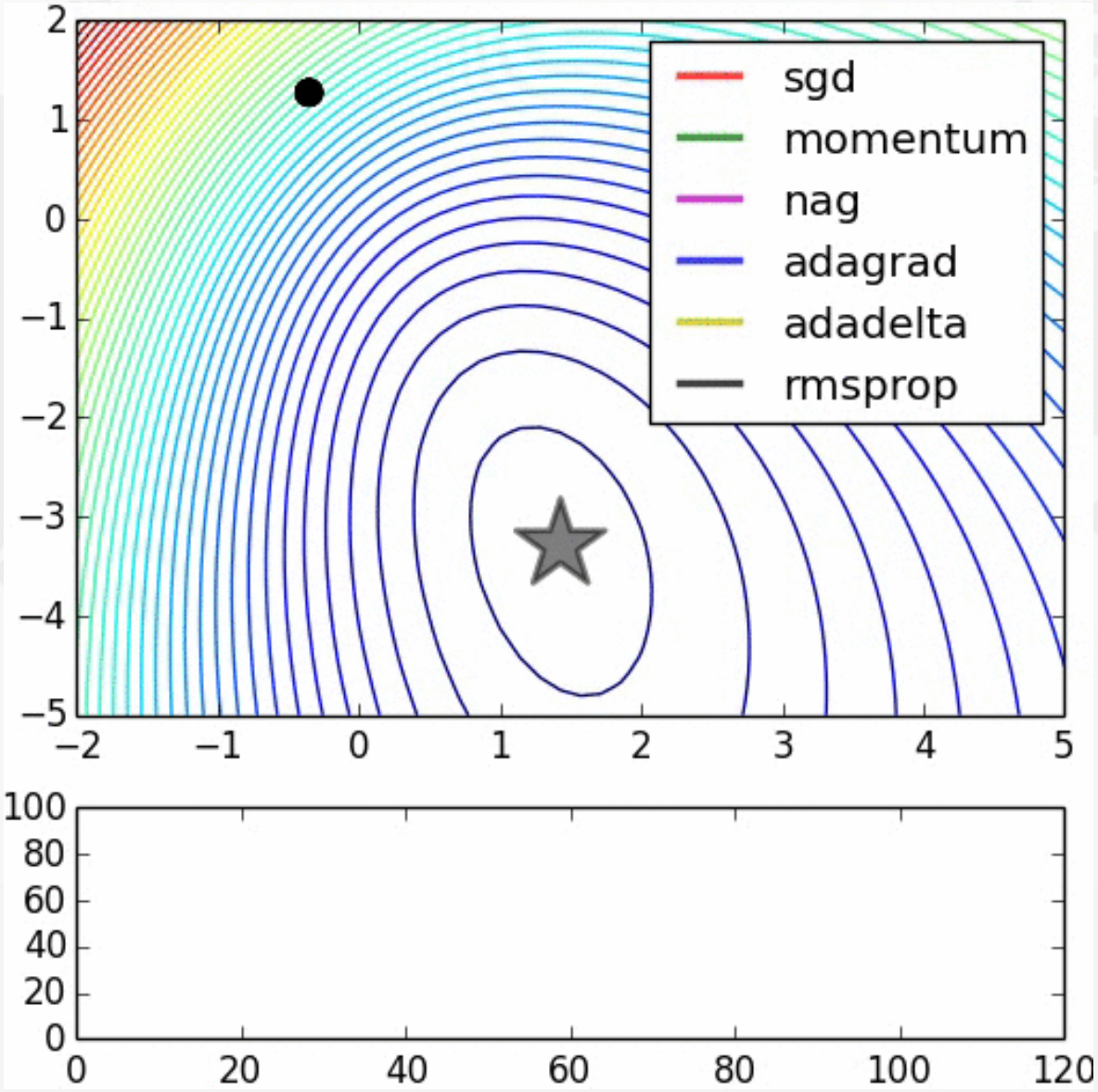


Coordinated Descendent

- 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 $x_i - \alpha \frac{\partial F}{\partial x_i}(\mathbf{x})$.

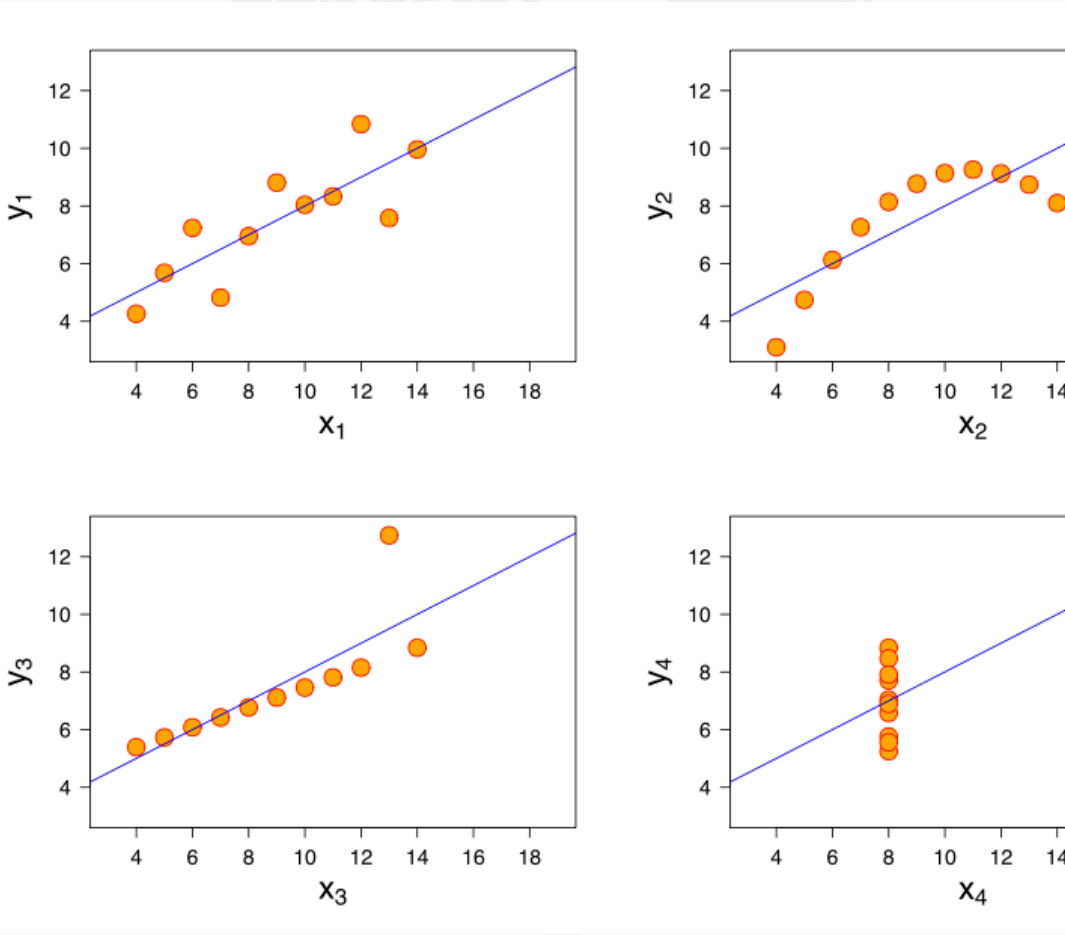


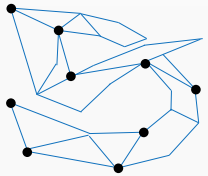
Stochastic Gradient Descendent



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
```





Q & A

4