Recap

- Curse of dimension
  - Data tends to fall into the “rind” of space

    $\mathrm{d} \to \infty, \quad P(\{x \in \text{"}rind\text{"}\}) \to 1$
  - Variance gets larger (uniform cube):

    $E[x^t x] = \frac{d}{3}$
  - Data falls further apart from each other (uniform cube):

    $E[d(u,v)^2] = 2\frac{d}{3}$
  - Statistics becomes unreliable, difficult to build histograms, etc. ➔ use simple models

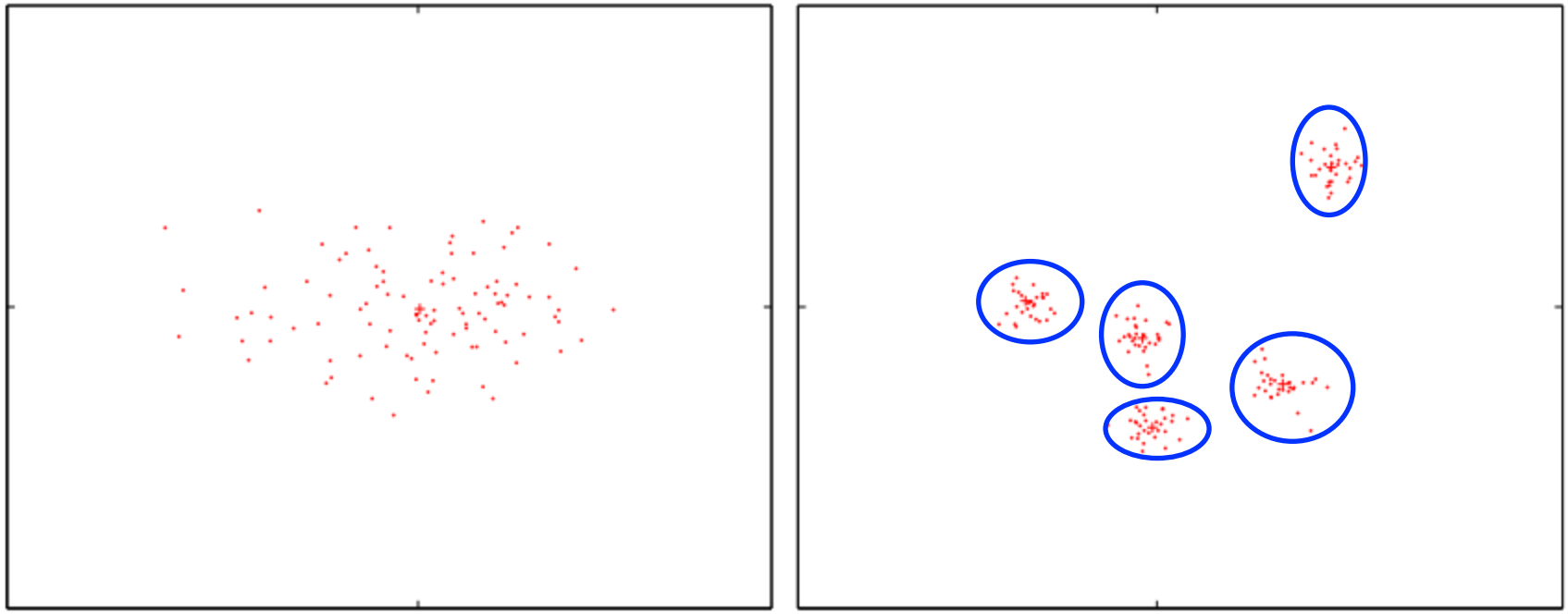
Recap

- Multivariate Gaussian

$$p(\mathbf{x}|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

  - By translation and rotation, it turns into multiplication of **normal distributions**
  - MLE of mean: $\hat{\mu} = \frac{\Sigma_i x_i}{N}$
  - MLE of covariance: $\hat{\Sigma} = \frac{\Sigma_i (x_i - \hat{\mu})(x_i - \hat{\mu})^T}{N}$

Be cautious..

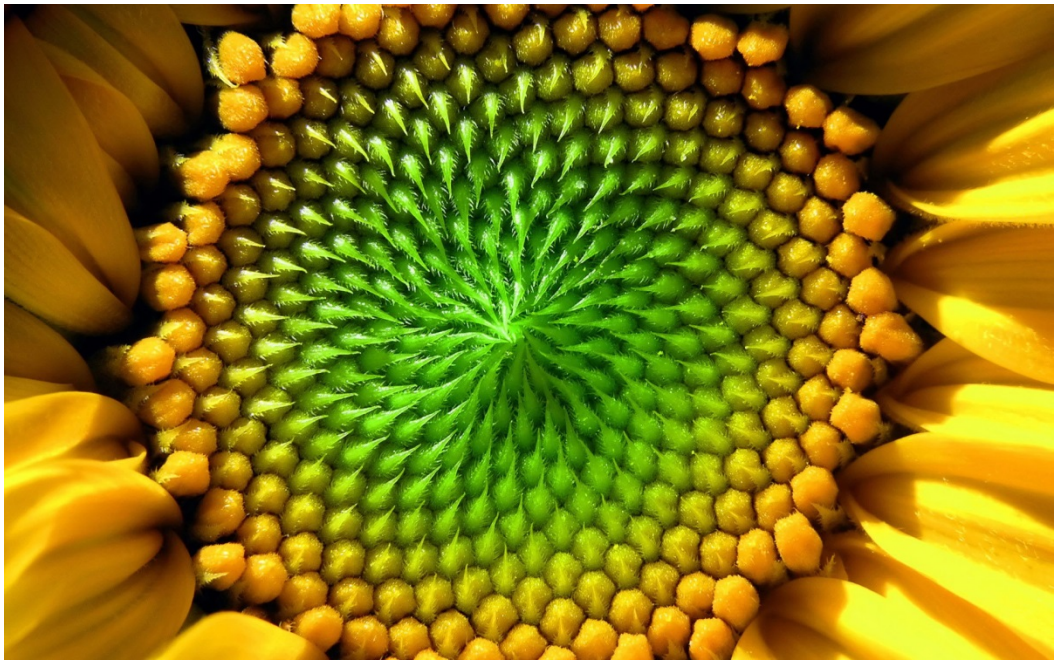


Data may not be in one blob, need to separate data into groups

Clustering

CS 498 Probability & Statistics

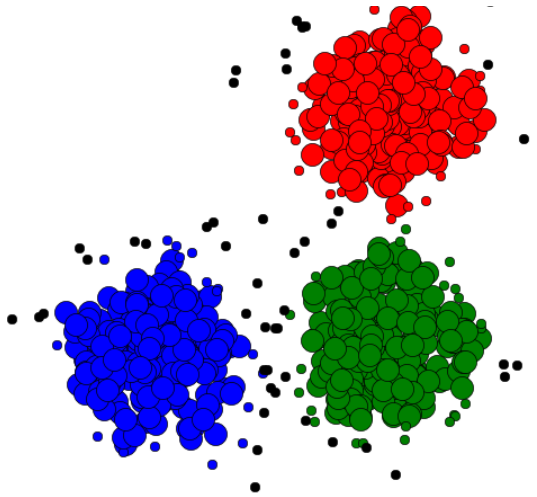
Clustering methods



Zicheng Liao

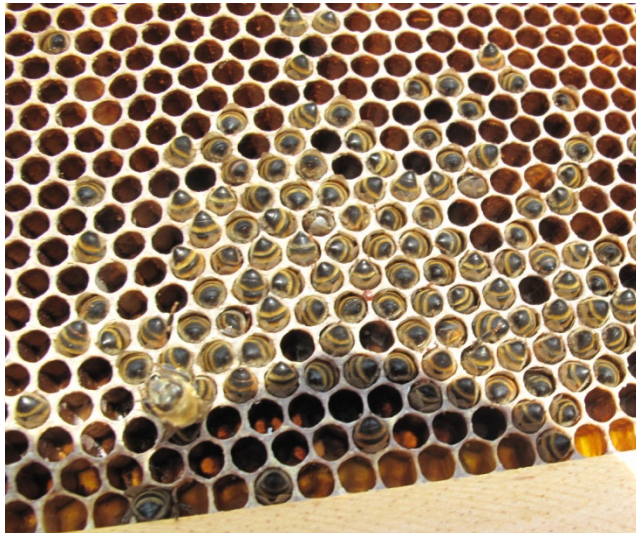
What is clustering?

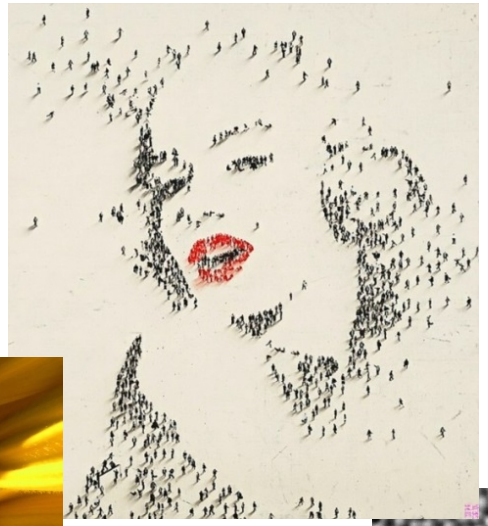
- "Grouping"
  - A fundamental part in signal processing
- "Unsupervised classification"
- Assign the same label to data points that are *close* to each other

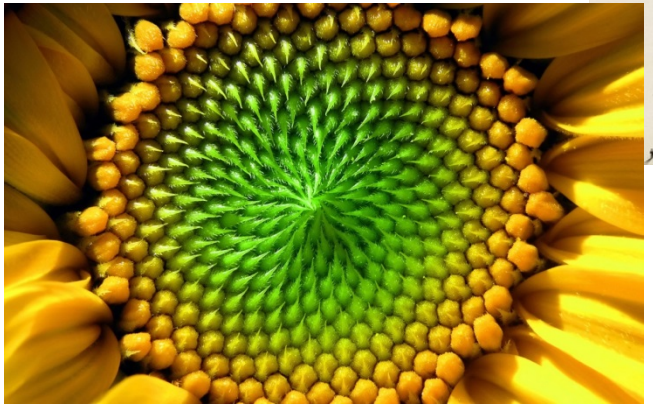


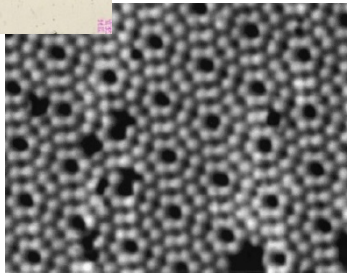
Why?

We live in a universe full of clusters





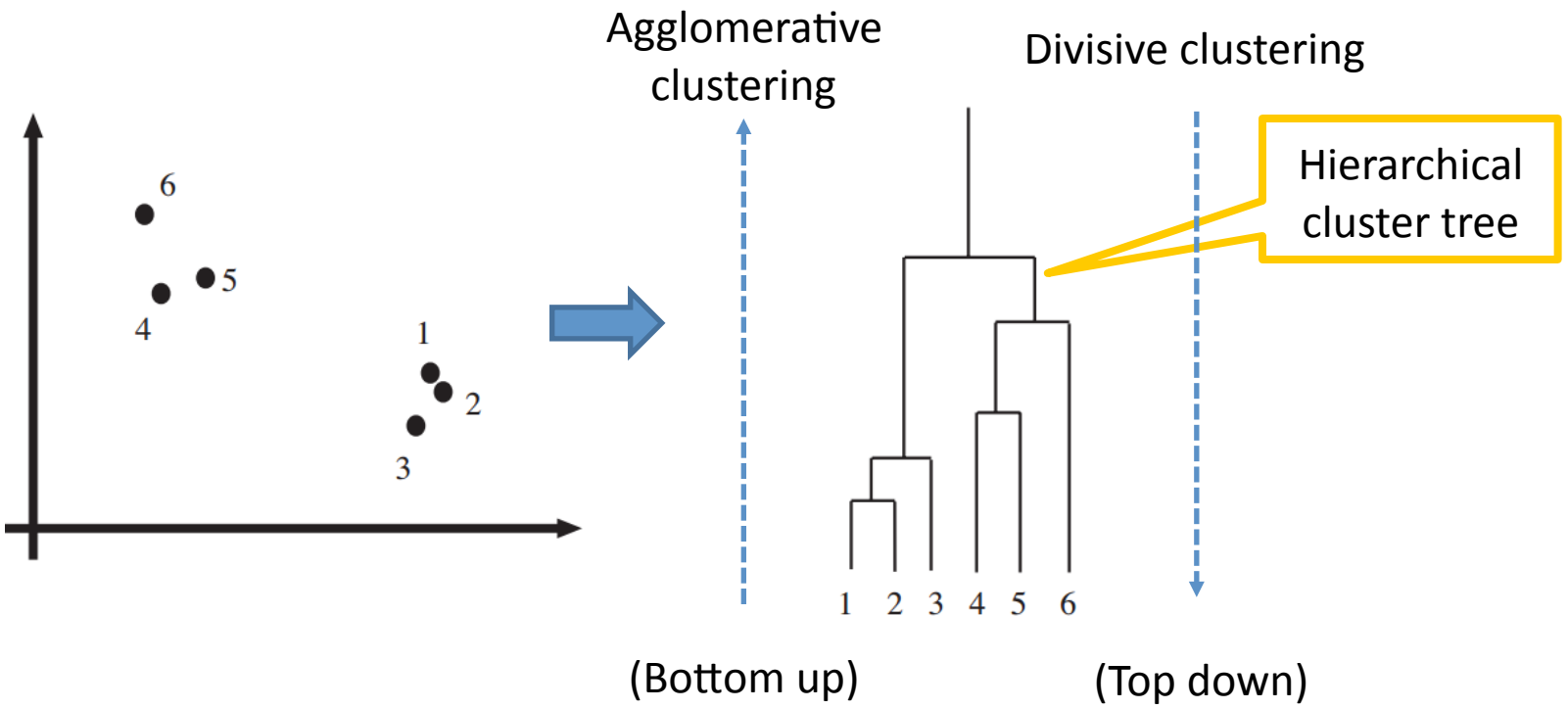




Two (types of) clustering methods

- Agglomerative/Divisive clustering
- K-means

Agglomerative/Divisive clustering



Algorithm

```
Make each point a separate cluster
Until the clustering is satisfactory
      Merge the two clusters with the
         smallest inter-cluster distance
end
```

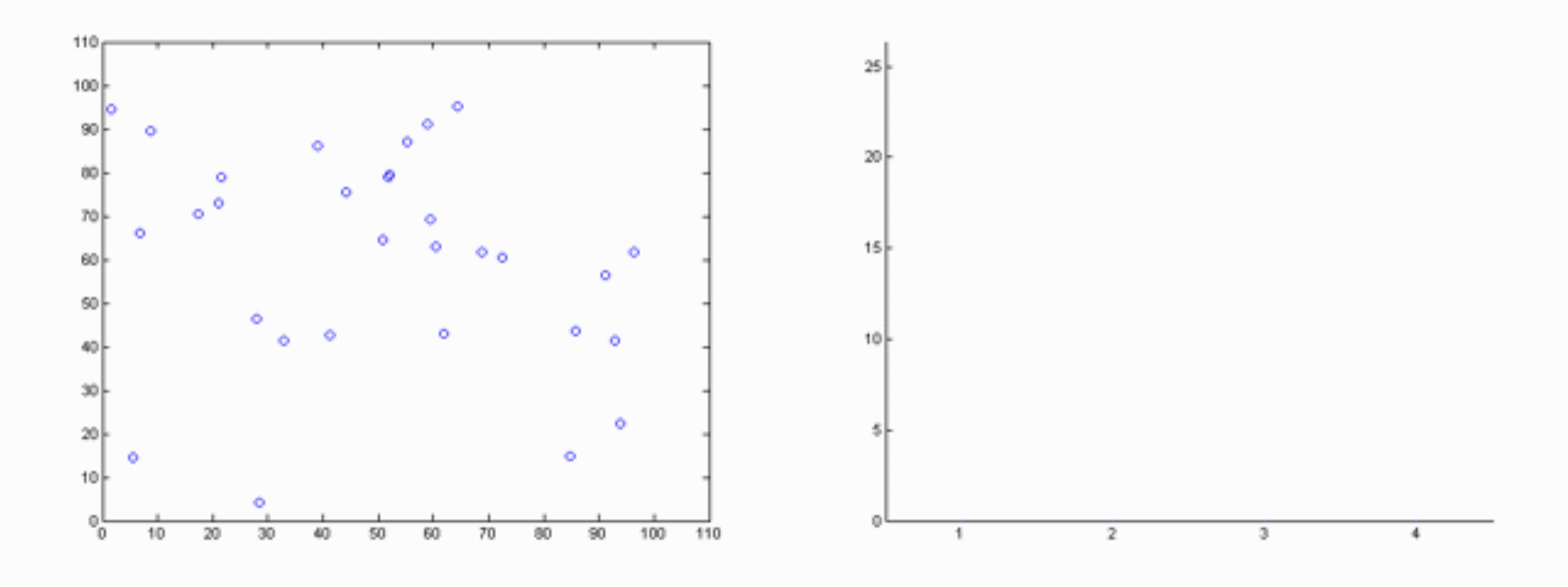
Algorithm 12.1: *Agglomerative Clustering or Clustering by Merging.*

```
Construct a single cluster containing all points
Until the clustering is satisfactory
      Split the cluster that yields the two
         components with the largest inter-cluster distance
end
```

Algorithm 12.2: *Divisive Clustering, or Clustering by Splitting.*

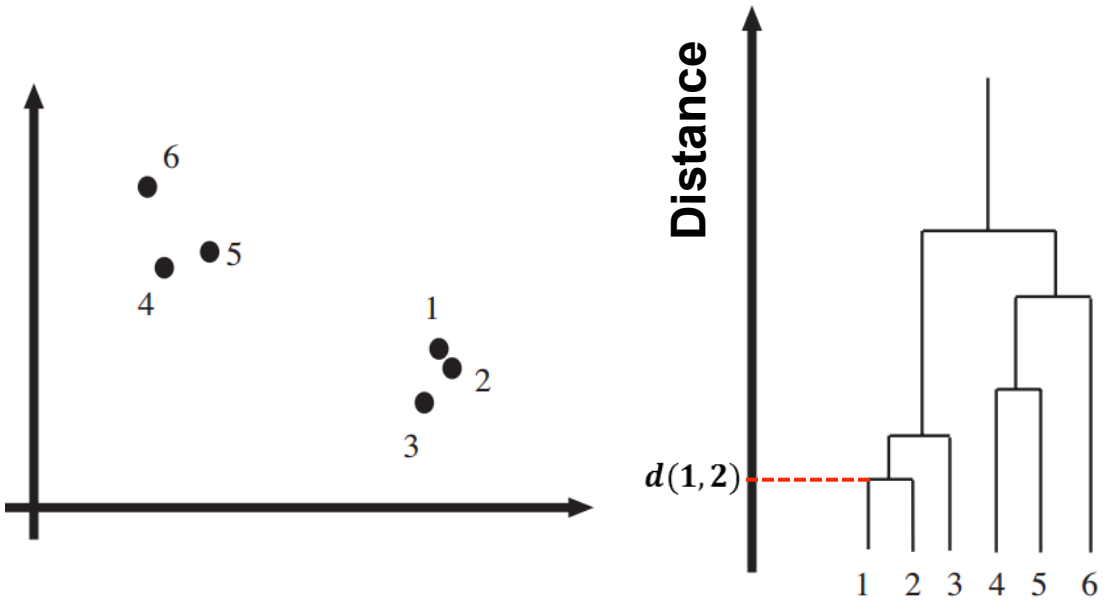
Agglomerative clustering: an example

- "merge clusters bottom up to form a hierarchical cluster tree"



Animation from Georg Berber
www.mit.edu/~**georg**/papers/lecture6.ppt

Dendrogram




```
>> X = rand(6, 2);     %create 6 points on a plane

>> Z = linkage(X);     %Z encodes a tree of hierarchical clusters

>> dendrogram(Z);      %visualize Z as a dendrograph
```

Distance measure

- Popular choices: *Euclidean, hamming, correlation, cosine,…*
- A metric
  - $d(x, y) \geq 0$
  - $d(x, y) = 0 \; iff \; x = y$
  - $d(x, y) = d(y, x)$
  - $d(x, y) \leq d(x, z) + d(z, y) \quad (\text{triangle inequality})$

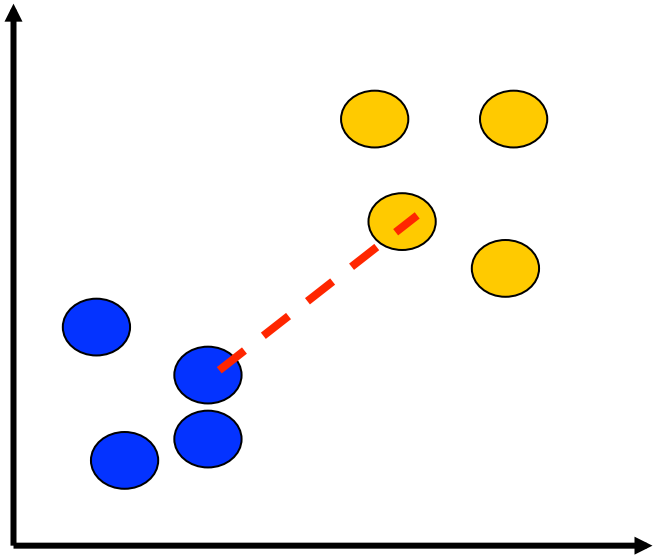
- Critical to clustering performance
- No single answer, depends on the data and the goal
- *Data whitening* when we know little about the data

Inter-cluster distance

- Treat each data point as a single cluster
- Only need to define inter-cluster distance
  - Distance between one set of points and another set of points
- 3 popular inter-cluster distances
  - Single-link
  - Complete-link
  - Averaged-link

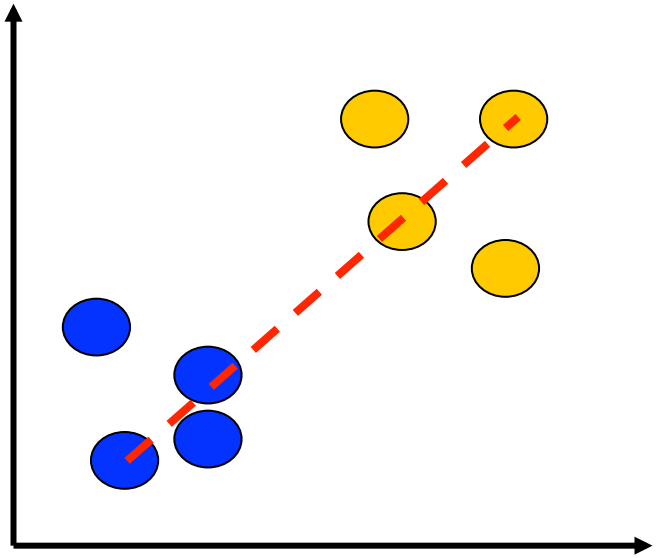
Single-link

- Minimum of all pairwise distances between points from two clusters
- Tend to produce long, loose clusters



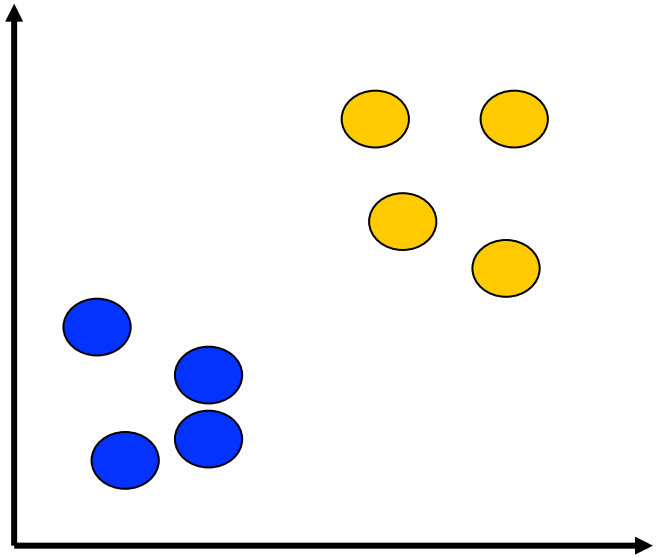
Complete-link

- Maximum of all pairwise distances between points from two clusters
- Tend to produce tight clusters



Averaged-link

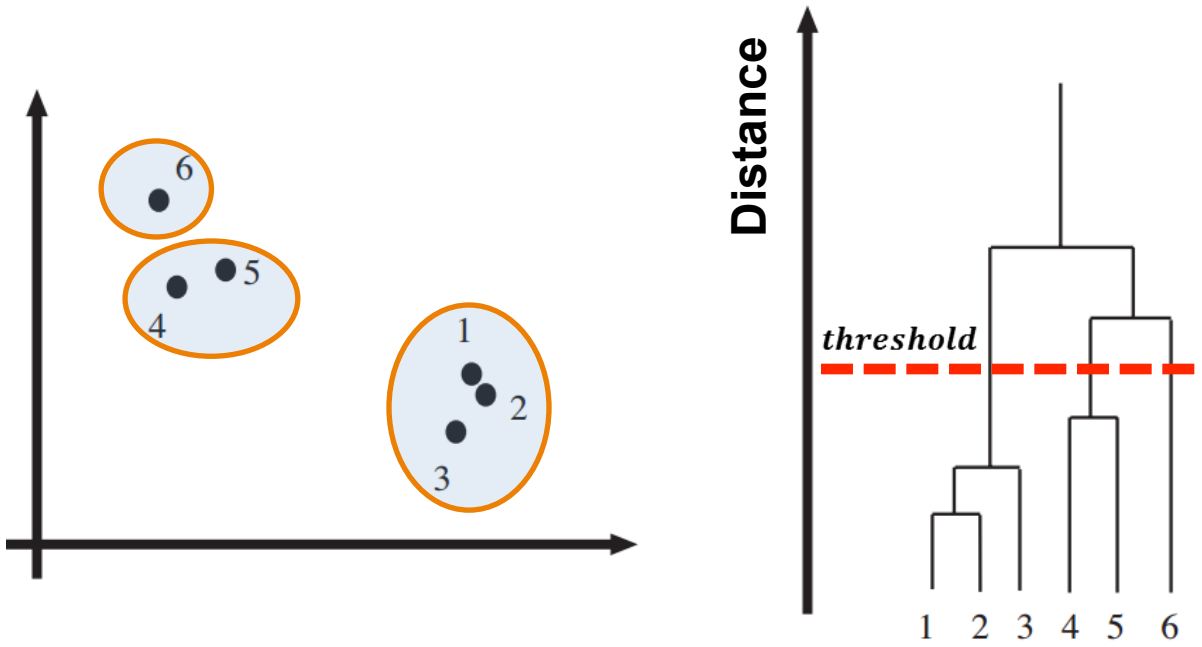
- Average of all pairwise distances between points from two clusters



$$D(C_1, C_2) = \frac{1}{N} \sum_{p_i \in C_1, p_2 \in C_2} d(p_i, p_j)$$

How many clusters are there?

- Intrinsically hard to know
- The dendrogram gives insights to it
- Choose a threshold to split the dendrogram into clusters

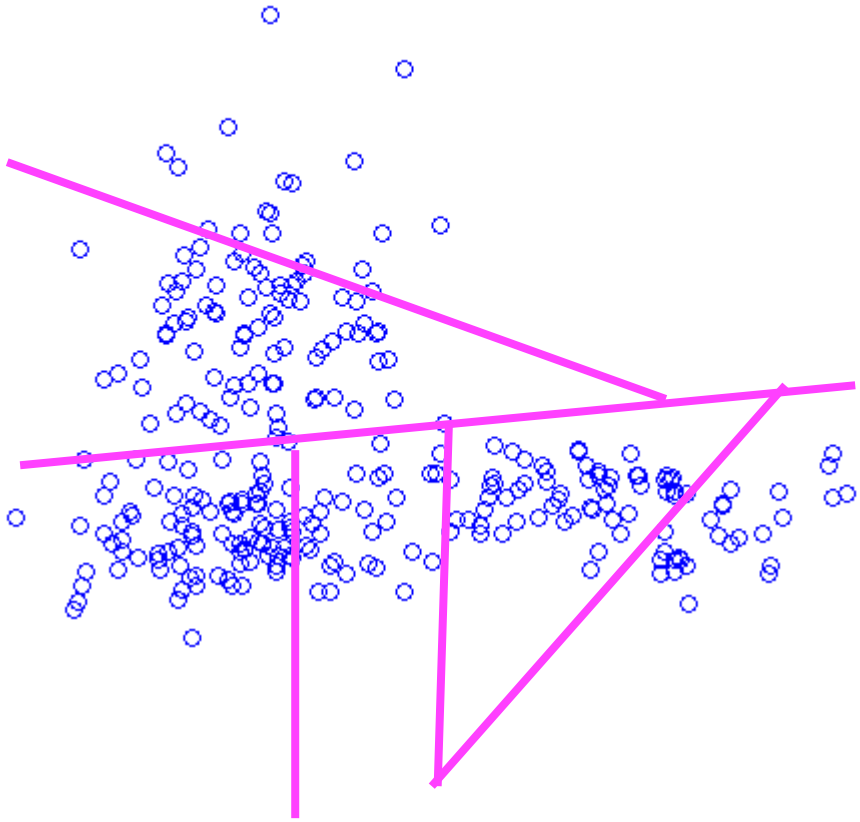


An example

do_agglomerative.m

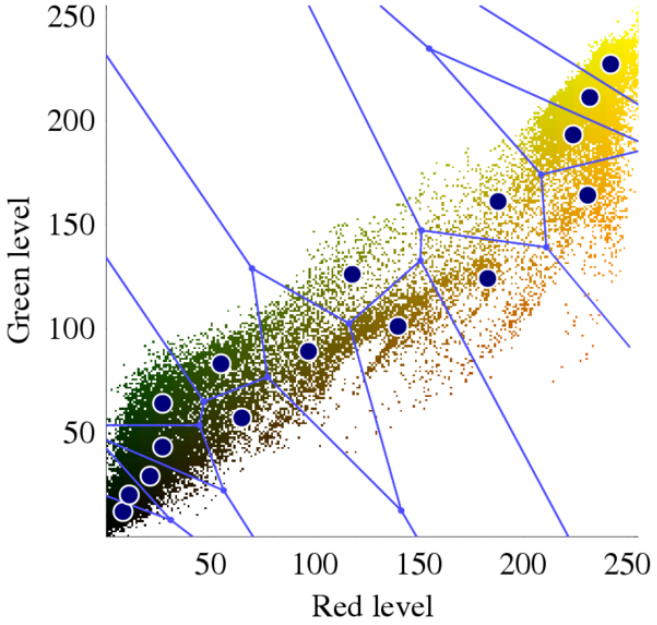
Divisive clustering

- “recursively split a cluster into smaller clusters”
- It’s hard to choose where to split: combinatorial problem
- Can be easier when data has a special structure (pixel grid)



K-means

- Partition data into clusters such that:
  - Clusters are tight (distance to cluster center is small)
  - Every data point is closer to its own cluster center than to all other cluster centers (Voronoi diagram)



[figures excerpted from Wikipedia]

Formulation

- Find K clusters that minimize:

Cluster center

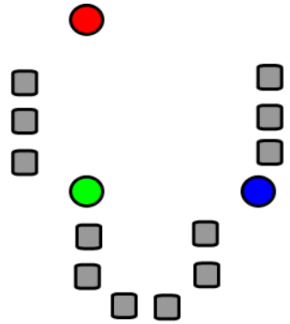
$$\Phi(\boldsymbol{C}, \boldsymbol{x}) = \sum_{i \in ||\boldsymbol{C}||} \left\{ \sum_{\boldsymbol{x}_j \in \boldsymbol{C}_i} \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right)\right)^T \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i\right) \right\}$$

- Two parameters: $\{label, cluster\ center\}$
- NP-hard for global optimal solution
- Iterative procedure (local minimum)

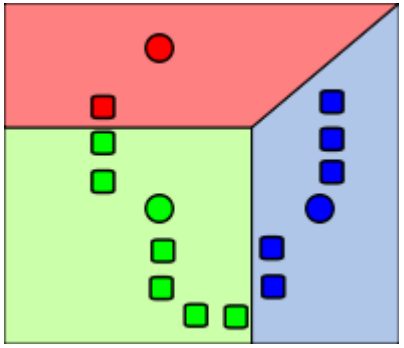
K-means algorithm

1. *Choose cluster number K*
2. *Initialize cluster center* $\mu_1, \ldots \mu_k$
   a. Randomly select K data points as cluster centers
   b. Randomly assign data to clusters, compute the cluster center
3. Iterate:
   a. Assign each point to the closest cluster center
   b. Update cluster centers (take the mean of data in each cluster)
4. Stop when the assignment doesn't change

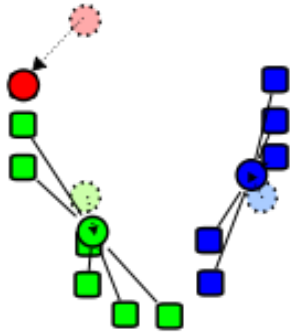
Illustration



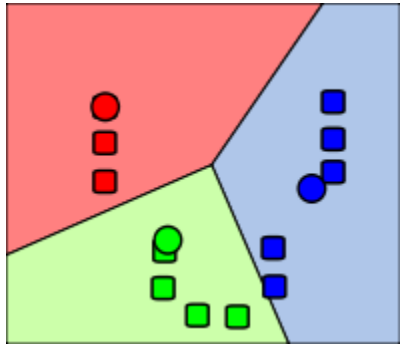
Randomly initialize 3 cluster centers (circles)



Assign each point to the closest cluster center



Update cluster center



Re-iterate step2

[figures excerpted from Wikipedia]

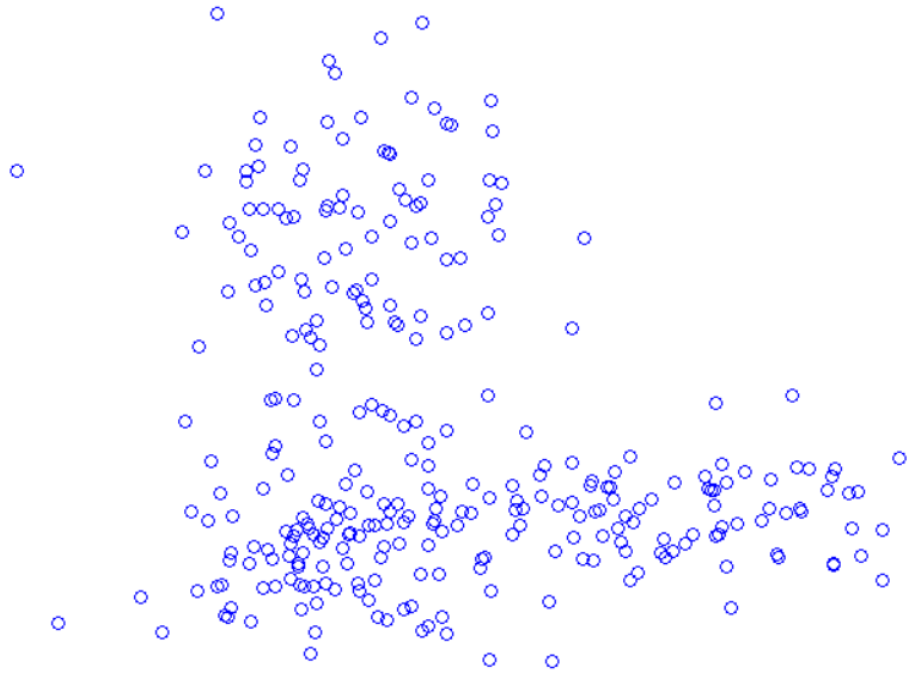
Example

do_Kmeans.m

(show step-by-step updates and effects of cluster number)

Discussion

- How to choose cluster number K?
  - No exact answer, guess from data (with visualization)
  - Define a **cluster quality** measure $Q(K)$ then optimize K



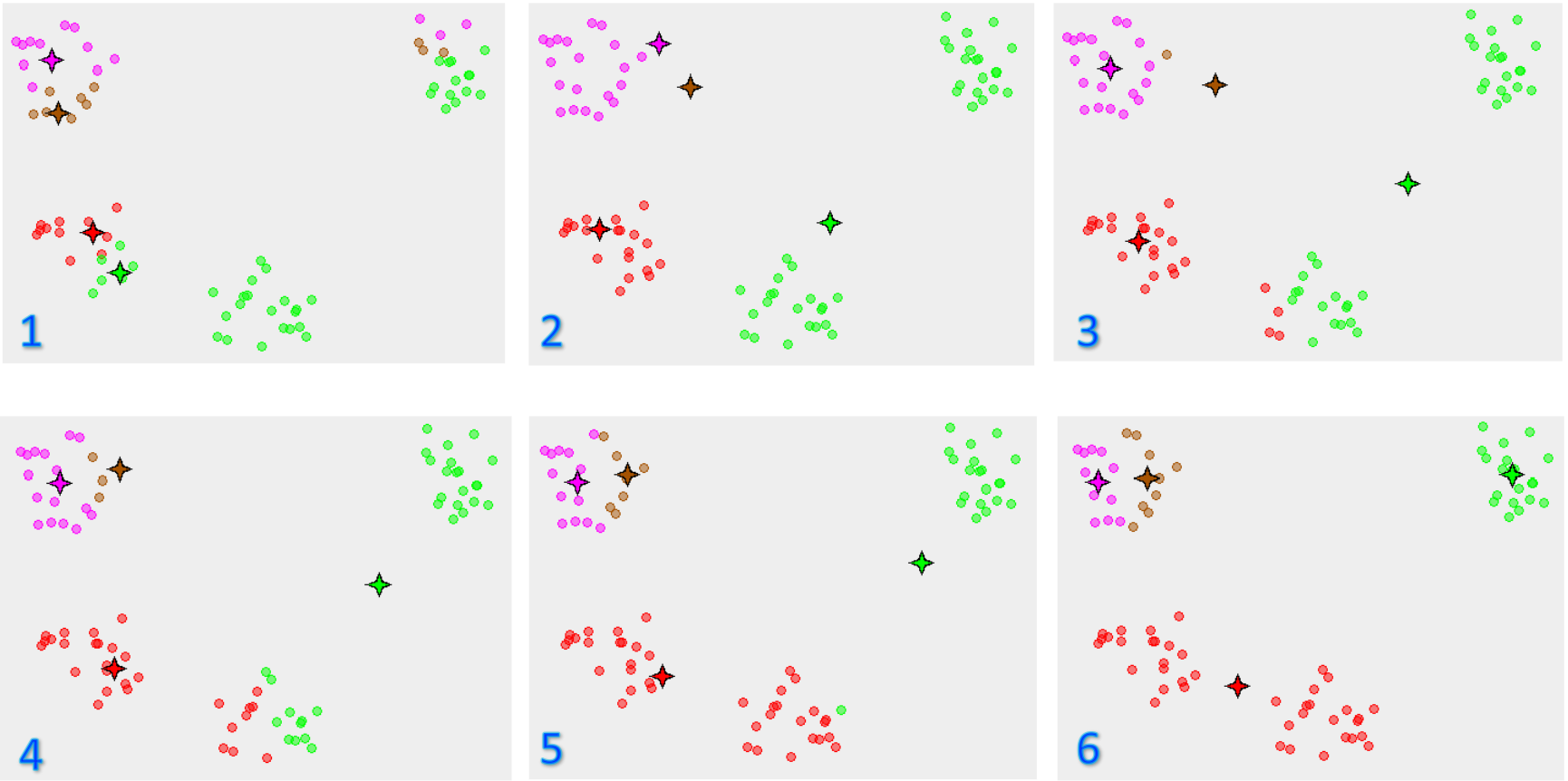
K = 2?

K = 3?

K = 5?

Discussion

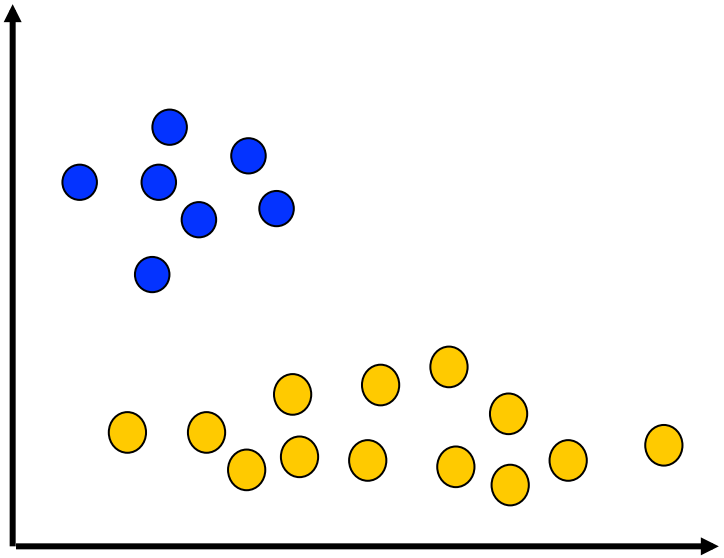
- Converge to local minimum => counterintuitive clustering



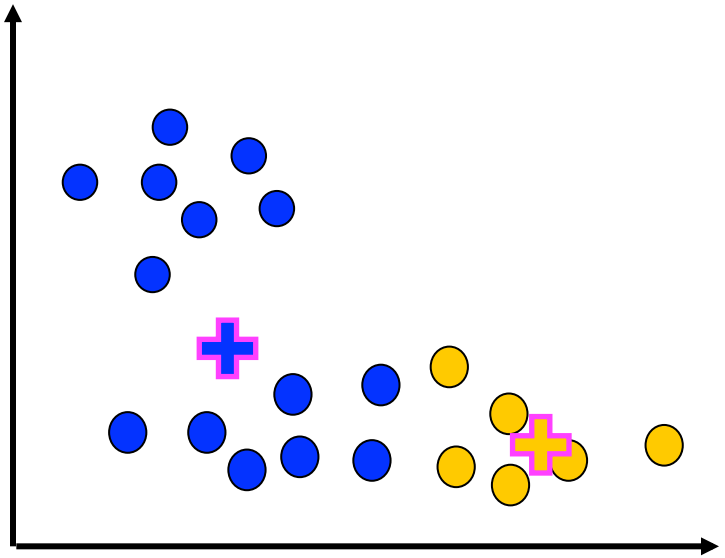
[figures excerpted from Wikipedia]

Discussion

- Favors ***spherical*** clusters;
- Poor results for long/loose/stretched clusters



Input data(color indicates true labels)



K-means results

Discussion

- Cost is guaranteed to decrease in every step
  - Assign a point to the closest cluster center minimizes the cost for current *cluster center configuration*
  - Choose the mean of each cluster as new cluster center minimizes the squared distance for current *clustering configuration*

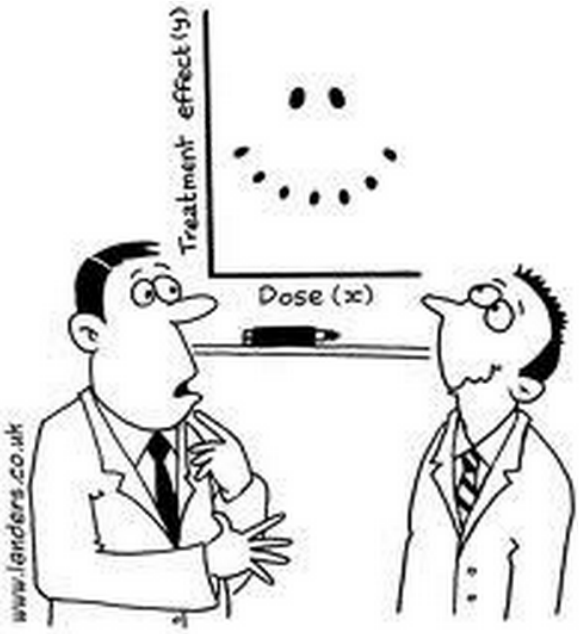
- Finish in polynomial time

Summary

- Clustering as grouping “similar” data together
- A world full of clusters/patterns
- Two algorithms
  - Agglomerative/divisive clustering: hierarchical clustering tree
  - K-means: vector quantization

CS 498 Probability & Statistics

Regression

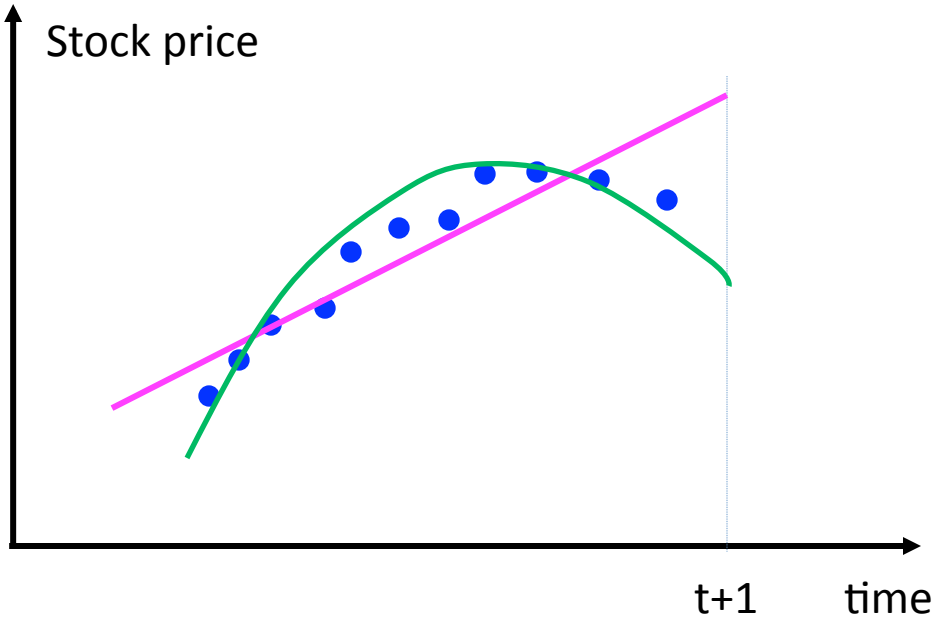


"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

Zicheng Liao

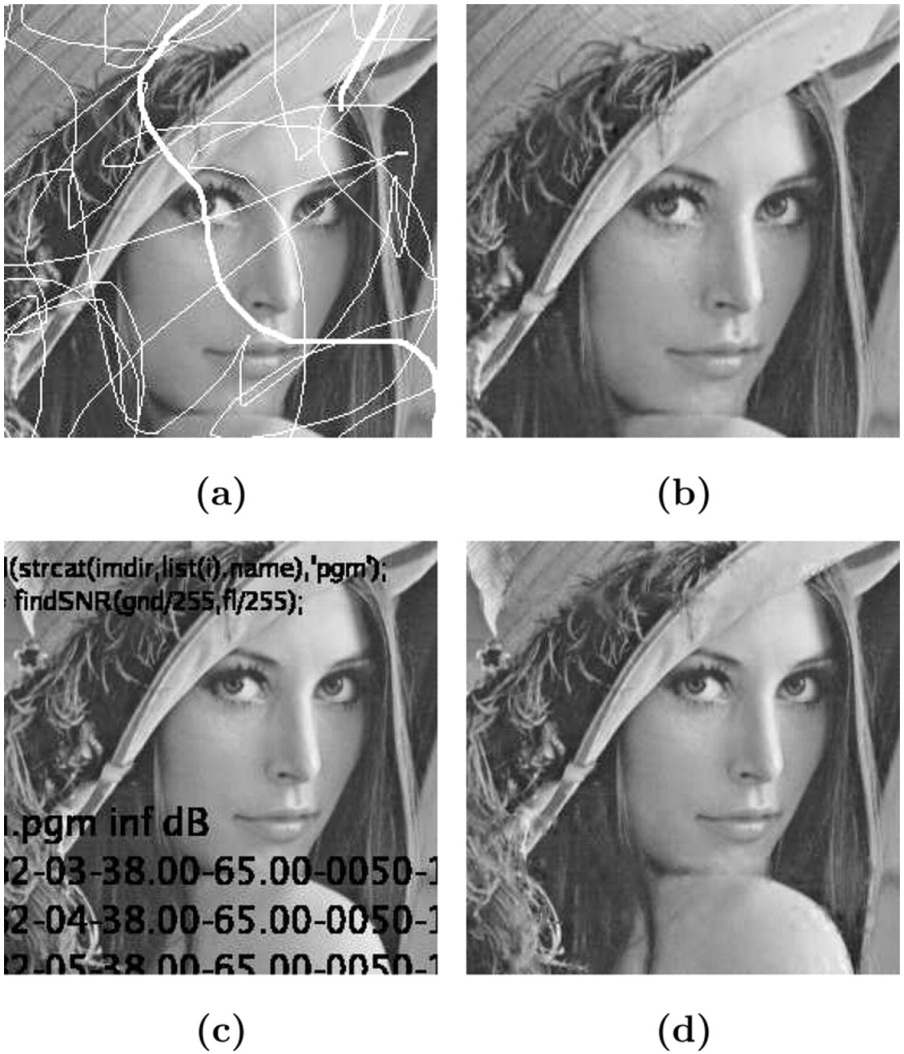
Example-I

- Predict stock price



Example-II

- Fill in missing pixels in an image: inpainting



Example-III

- Discover relationship in data

Number	A	B	C
1.	25.8	16.3	28.8
2.	20.5	11.6	22.0
3.	14.3	11.8	29.7
4.	23.2	32.5	28.9
5.	20.6	32.0	32.8
6.	31.1	18.0	32.5
7.	20.9	24.1	25.4
8.	20.9	26.5	31.7
9.	30.4	25.8	28.5

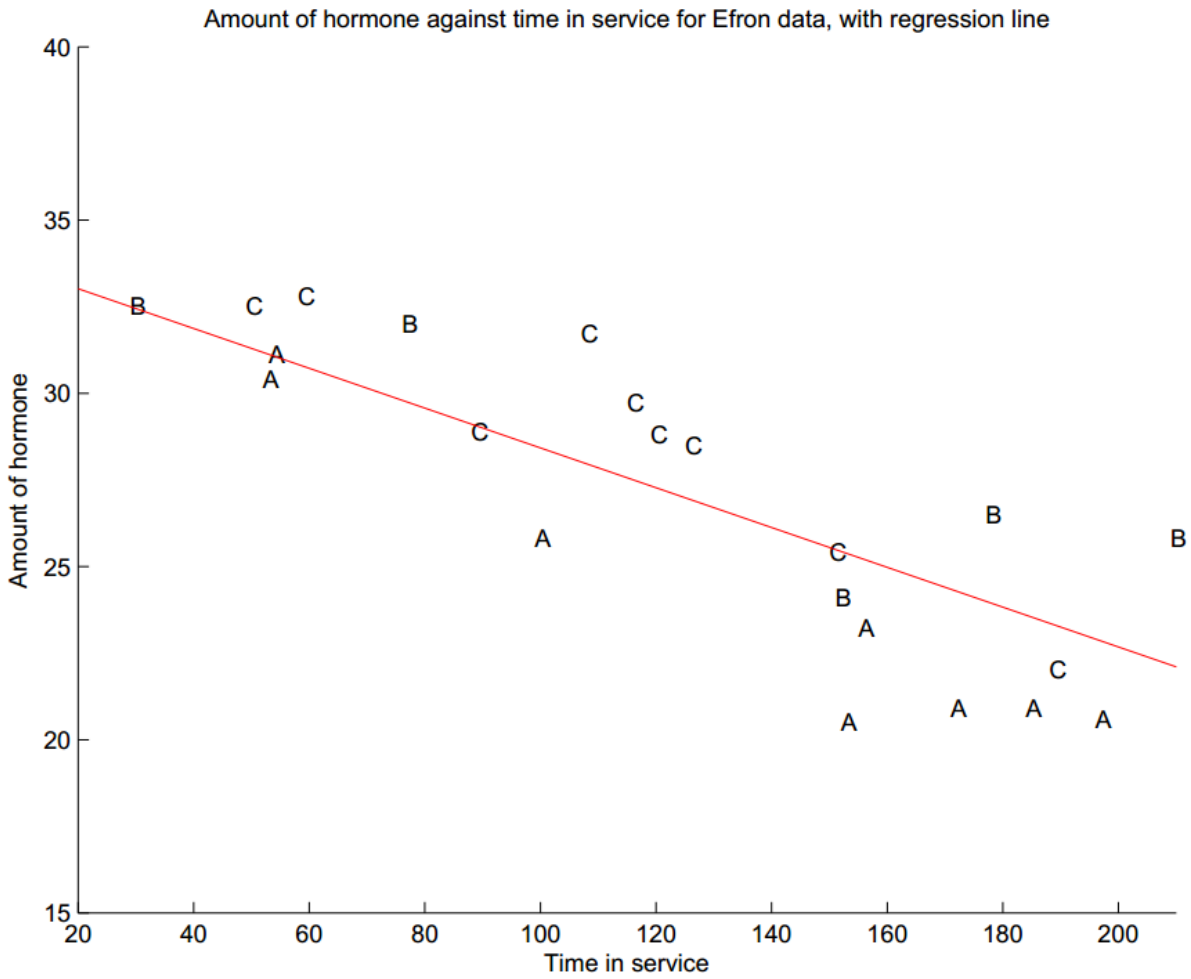
Amount of hormones by devices from 3 production lots

Number	A	B	C
1.	99	376	119
2.	152	385	188
3.	293	402	115
4.	155	29	88
5.	196	76	58
6.	53	296	49
7.	184	151	150
8.	171	177	107
9.	52	209	125

Time in service for devices from 3 production lots

Example-III

- Discovery relationship in data



Linear regression

- Input:

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_M, y_M)\}$$

y: $house\ price$
$\boldsymbol{x}$: $\{size, age\ of\ house, \#bedroom, \#bathroom, yard\}$

- Linear model with Gaussian noise

$$y = \boldsymbol{x}^T \beta + \xi \qquad \boldsymbol{x^T} = (x_1, x_2, \ldots x_N, 1)$$

  - x: explanatory variable
  - y: dependent variable
  - β: parameter of linear model
  - ξ: zero mean *Gaussian random variable*

Parameter estimation

- MLE of linear model with Gaussian noise

$$maximize: \quad P(\{(\boldsymbol{x}_i, y_i)\}^M | \beta)$$

Likelihood function

$$= \prod_{i=1}^{M} g(y_i - \boldsymbol{x}_i{}^T \beta; 0, \sigma)$$

$$= \frac{1}{const} \exp\{-\frac{\sum_{i=1}^{M}(y_i - \boldsymbol{x}_i{}^T \beta)^2}{2\sigma}\}$$

➔ $minimize:$ $\sum_{i=1}^{M}(y_i - x_i{}^T \beta)^2$

[Least squares, Carl F. Gauss, 1809]

Parameter estimation

- Closed form solution

Cost function

$$\Phi(\beta) = \sum_{i=1}^{M} (y_i - \boldsymbol{x_i}^T \beta)^2 = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta) \qquad \boldsymbol{X} = \begin{pmatrix} \boldsymbol{x_1}^T \\ \boldsymbol{x_2}^T \\ \cdots \\ \boldsymbol{x_M}^T \end{pmatrix} \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_M \end{pmatrix}$$

$$\frac{\partial \Phi(\beta)}{\partial \beta} = \boldsymbol{X}^T \boldsymbol{X}\beta - \boldsymbol{X}^T \boldsymbol{y}$$

➔ $\boldsymbol{X}^T \boldsymbol{X}\beta - \boldsymbol{X}^T \boldsymbol{y} = \boldsymbol{0}$ **Normal equation**

➔ $\beta = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$

(expensive to compute the matrix inverse for high dimension)

Gradient descent

- http://openclassroom.stanford.edu/MainFolder/VideoPage.php?course=MachineLearning&video=02.5-LinearRegressionI-GradientDescentForLinearRegression&speed=100 (Andrew Ng)

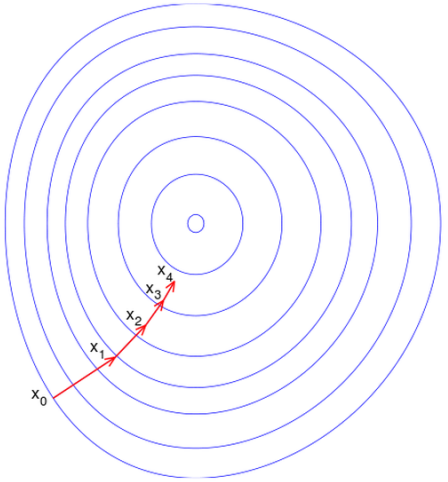
$$\frac{\partial \Phi(\beta)}{\partial \beta} = \boldsymbol{X}^T \boldsymbol{X}\beta - \boldsymbol{X}^T \boldsymbol{y}$$

Init: $\beta^{(0)} = (0,0, \ldots 0)$

Repeat:

$$\beta^{(t+1)} = \beta^{(t)} - \alpha \frac{\partial \Phi(\beta)}{\partial \beta}$$

Until converge.



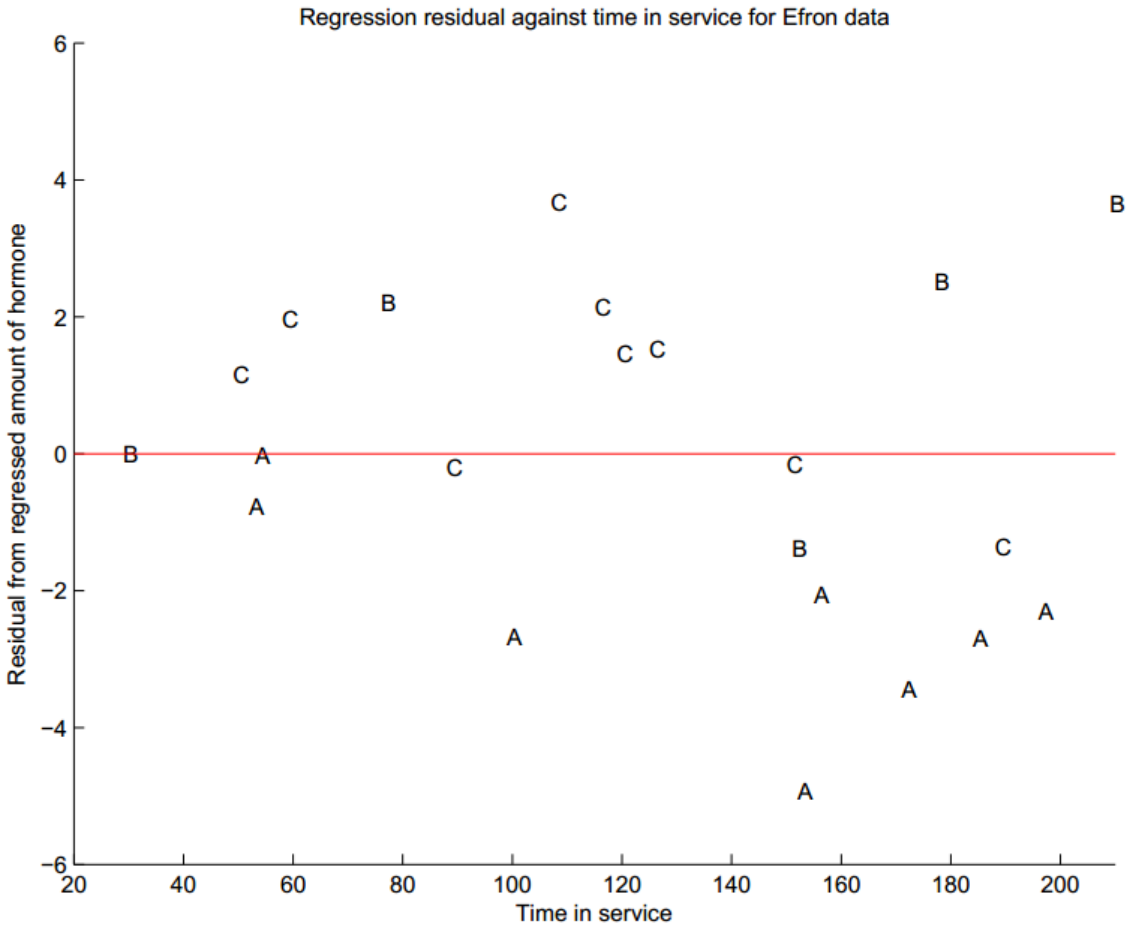
(Guarantees to reach global minimum in finite steps)

Example

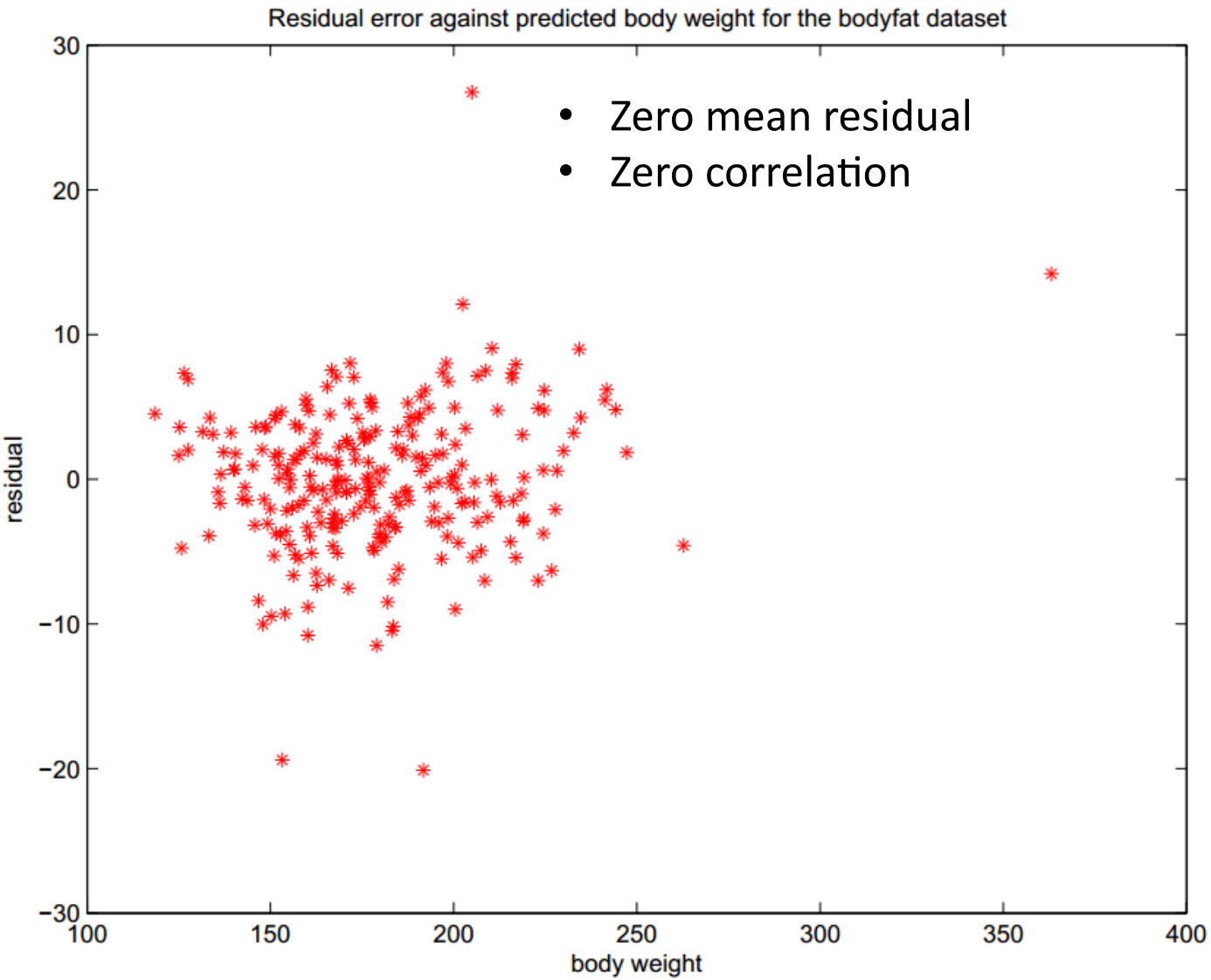
do_regression.m

Interpreting a regression

$$y = -0.0574t + 34.2$$



Interpreting a regression



Interpreting the residual

Useful Facts: 13.1 *Regression*

We write $\mathbf{y} = \mathcal{X}\beta + \mathbf{e}$, where $\mathbf{e}$ is the residual. Assume $\mathcal{X}$ has a column of ones, and β is chosen to minimize $\mathbf{e}^T\mathbf{e}$. Then we have

1. $\mathbf{e}^T\mathcal{X} = \mathbf{0}$, i.e. that $\mathbf{e}$ is orthogonal to any column of $\mathcal{X}$. This is because, if $\mathbf{e}$ is not orthogonal to some column of $\mathbf{e}$, we can increase or decrease the β term corresponding to that column to make the error smaller. Another way to see this is to notice that *beta* is chosen to minimize $\mathbf{e}^T\mathbf{e}$, which is $(\mathbf{y}-\mathcal{X}\beta)^T(\mathbf{y}-\mathcal{X}\beta)$. Now because this is a minimum, the gradient with respect to β is zero, so $(\mathbf{y} - \mathcal{X}\beta)^T(-\mathcal{X}) = -\mathbf{e}^T\mathcal{X} = 0$.
2. $\mathbf{e}^T\mathbf{1} = 0$ (recall that $\mathcal{X}$ has a column of all ones, and apply the previous result).
3. $\mathbf{1}^T(\mathbf{y} - \mathcal{X}\beta) = 0$ (same as previous result).
4. $\mathbf{e}^T\mathcal{X}\beta = 0$ (first result means that this is true).

Interpreting the residual

- e has zero mean

- e is orthogonal to every column of $\boldsymbol{X}$
  - e is also **de-correlated** from every column of $\boldsymbol{X}$

$$\mathrm{cov}(\mathrm{e}, \mathbf{X}^{(i)}) = \frac{1}{M}(e-0)^T\left(\boldsymbol{X}^{(i)} - mean\left(\boldsymbol{X}^{(i)}\right)\right)$$
$$= \frac{1}{M}e^T\boldsymbol{X}^{(i)} - \ mean(e) * mean\left(\boldsymbol{X}^{(i)}\right)$$
$$= 0 \ - 0$$

- e is orthogonal to the regression vector $\boldsymbol{X}\beta$
  - e is also **de-correlated** from the regression vector $\boldsymbol{X}\beta$

    (follow the same line of derivation)

How good is a fit?

- Information ~ variance

- Total variance is decoupled into regression variance and error variance

$$var[\boldsymbol{y}] = var[\boldsymbol{X}\beta] + var[e]$$

(Because e and $X\beta$ have zero covariance)

- How good is a fit: How much variance is explained by regression: $\boldsymbol{X}\beta$

How good is a fit?

- R-squared measure
  - The percentage of variance explained by regression

$$R^2 = \frac{var[\boldsymbol{X}\beta]}{var[\boldsymbol{y}]}$$

  - Used in hypothesis test for model selection

Regularized linear regression

- Cost

$$\sum_i (y_i - \mathbf{x}_i^T \beta)^2 = (\mathbf{y} - \mathcal{X}\beta)^T (\mathbf{y} - \mathcal{X}\beta)$$

$$\sum_i (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \beta^T \beta = (\mathbf{y} - \mathcal{X}\beta)^T (\mathbf{y} - \mathcal{X}\beta) + \lambda \beta^T \beta$$

Penalize large values in β

- Closed-form solution

$$\beta = (\boldsymbol{X}^T \boldsymbol{X} + \lambda I)^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

- Gradient descent

Init: $\beta = (0, 0, \ldots 0)$

Repeat:

$$\beta^{t+1} = \beta^t \left(1 - \frac{\alpha}{M} \lambda\right) - \alpha \frac{\partial \Phi(\beta)}{\partial \beta}$$

Until converge.

Why regularization?

- Handle small eigenvalues
  - Avoid dividing by small values by adding the regularizer

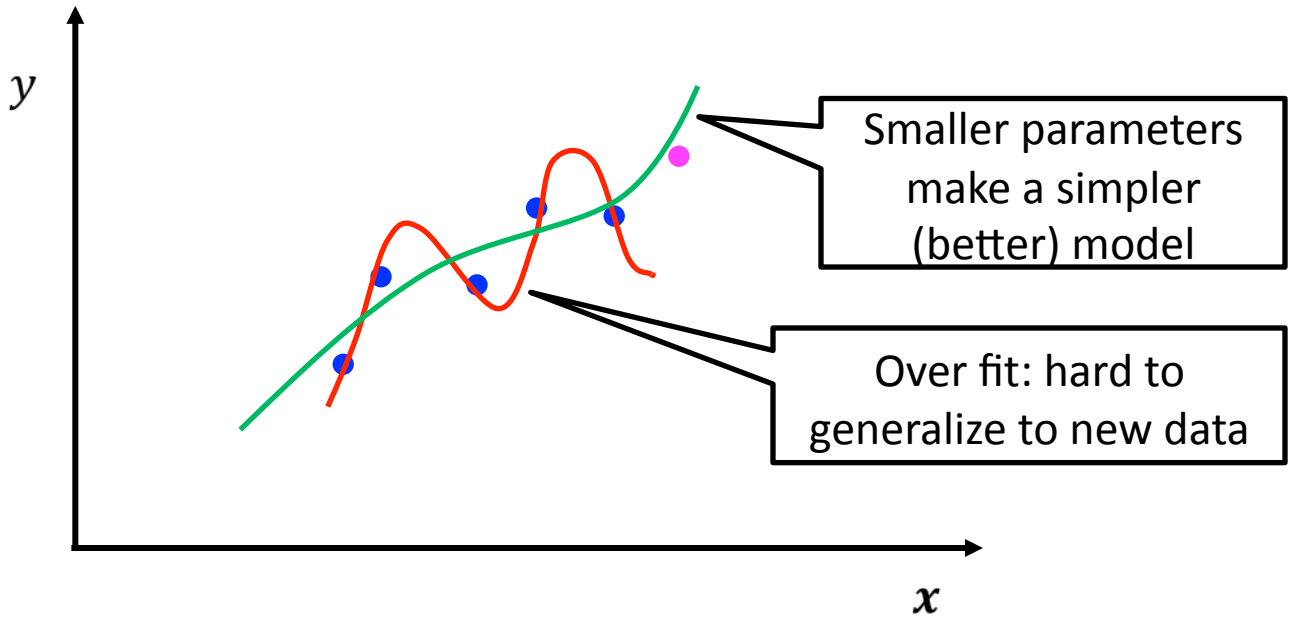
$$\beta = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\Downarrow$$

$$\beta = (\boldsymbol{X}^T\boldsymbol{X} + \lambda I)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Why regularization?

- Avoid over-fitting:
  - Over fitting
  - Small parameters ➔ simpler model ➔ less prone to over-fitting



L1 regularization (Lasso)

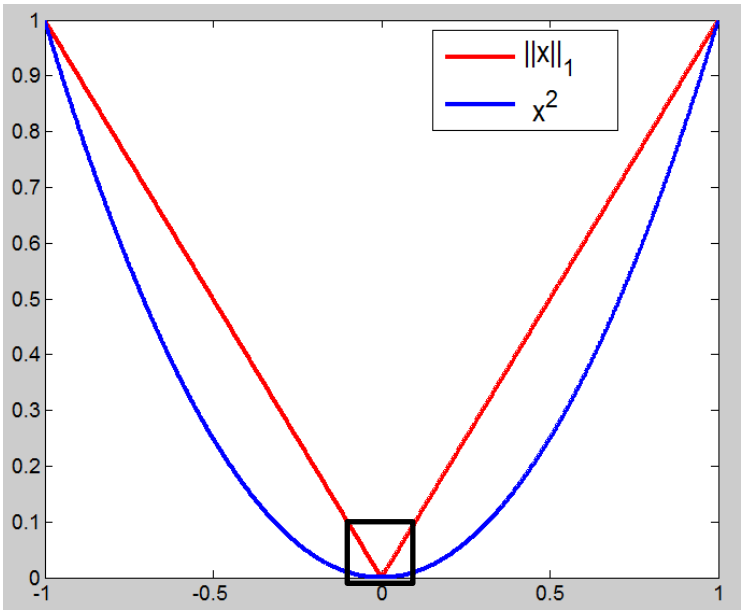
- $$\sum_i (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \beta^T \beta = (\mathbf{y} - \mathcal{X}\beta)^T (\mathbf{y} - \mathcal{X}\beta) + \lambda \beta^T \beta$$

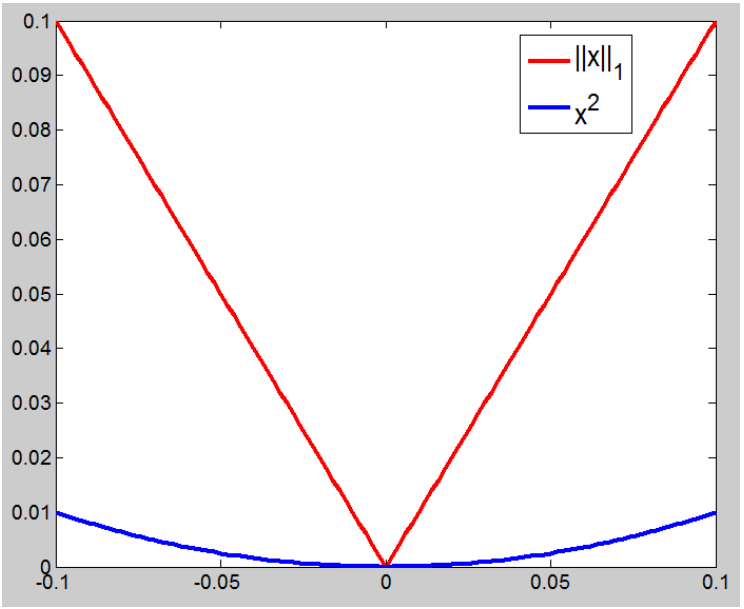
  ⇩

  $$\sum_i (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \|\beta\|_1 = (\mathbf{y} - \mathcal{X}\beta)^T (\mathbf{y} - \mathcal{X}\beta) + \lambda \|\beta\|_1$$

  - Some features may be irrelevant but still have a small non-zero coefficient in β
  - L1 regularization pushes small values of β to zero
  - "Sparse representation"

How does it work?

- 



  - When β is small, the L1 penalty is much larger than squared penalty.
  - Causes trouble in optimization (gradient non-continuity)

Summary

- Linear regression
  - Linear model + Gaussian noise
  - Parameter estimation by MLE ➔ Least squares
  - Solving least square by the normal equation
  - Or by gradient descent for high dimension
- How to interpret a regression model
  - R^2 measure
- Regularized linear regression
  - Squared norm
  - L1 norm: Lasso