DAT2 - Course Review

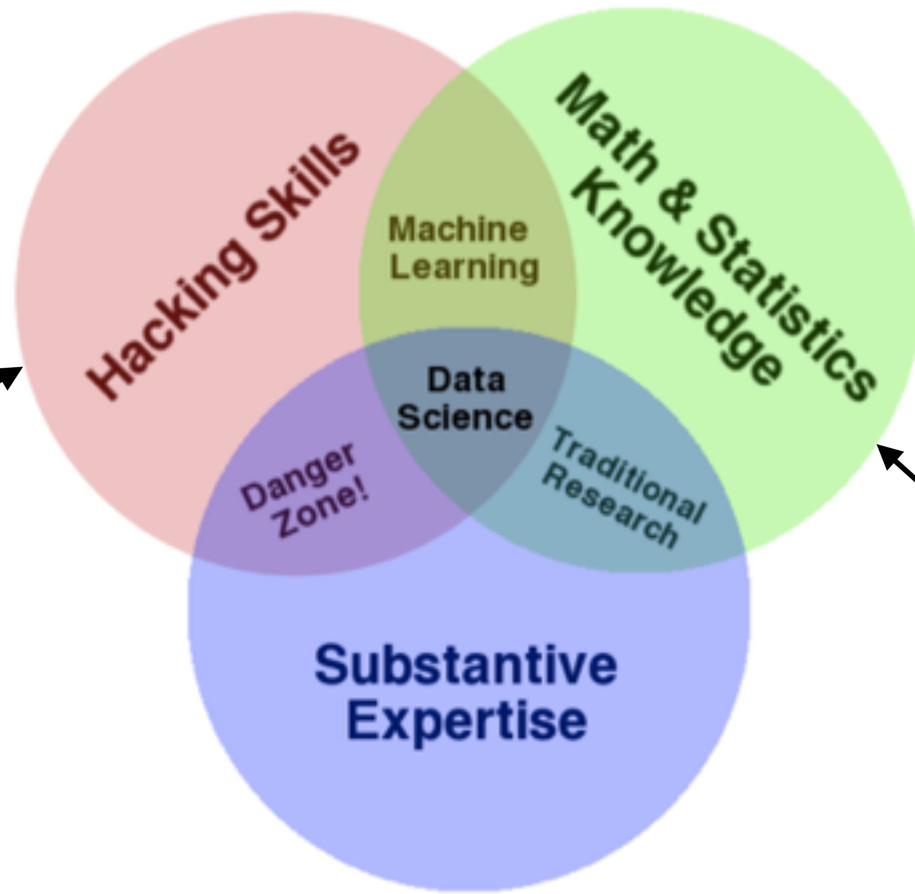
Agenda

- Basics
- Naïve Bayes
- Linear Regression
- knn
- logistic regression and GLMs
- cross validation
- k-means clustering, dbscan, spectral clustering
- map reduce and sql, Spark
- SVMs
- Decision trees and forests
- Dimensionality reduction: PCA and SVD
- Computer Science: github, command line, python, golang, recursion, scope, memory and pointers, documentation and debugging
- Learning theory, reinforcement learning, HMMs

Basics

WHAT IS A DATA SCIENTIST?

10

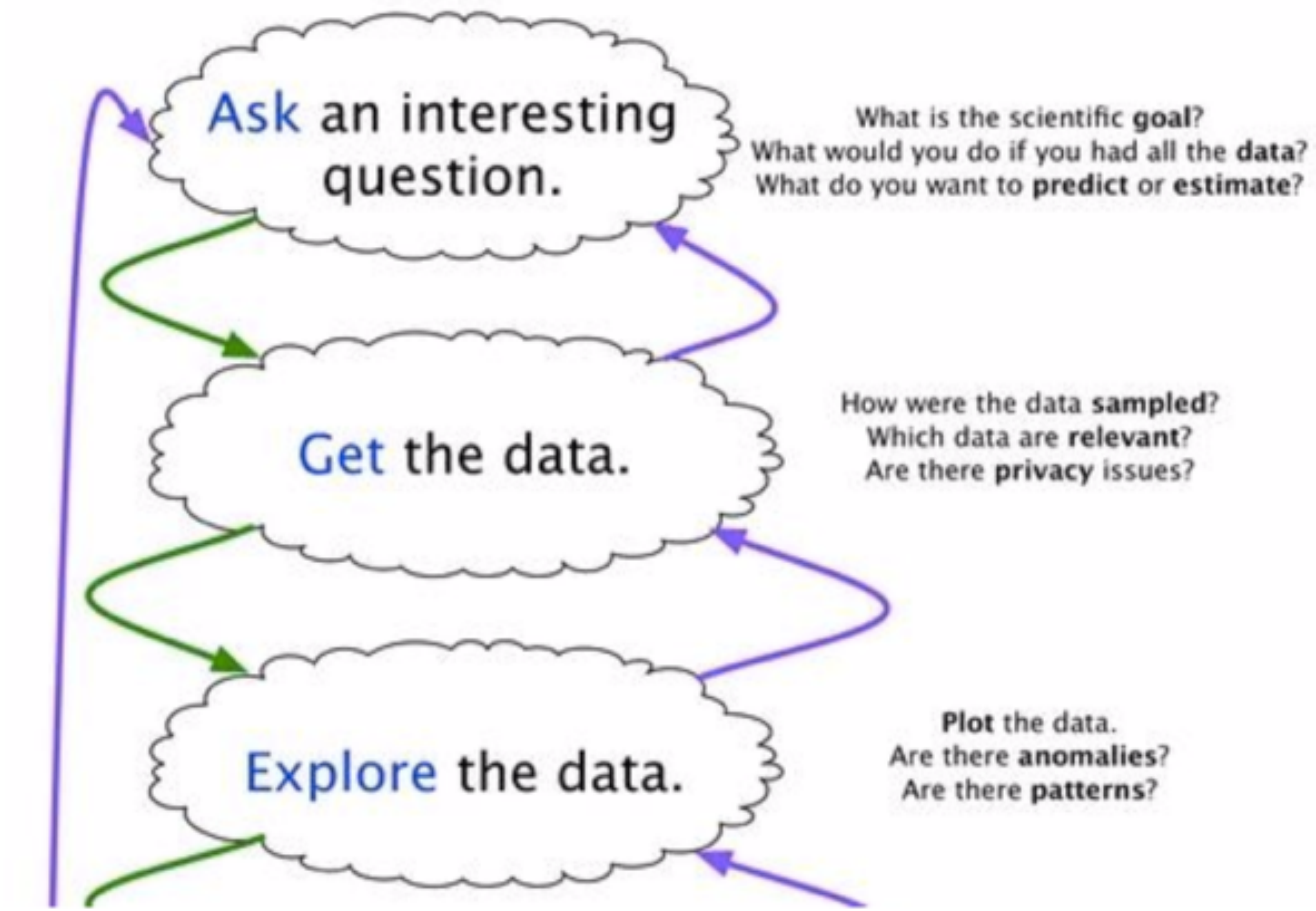


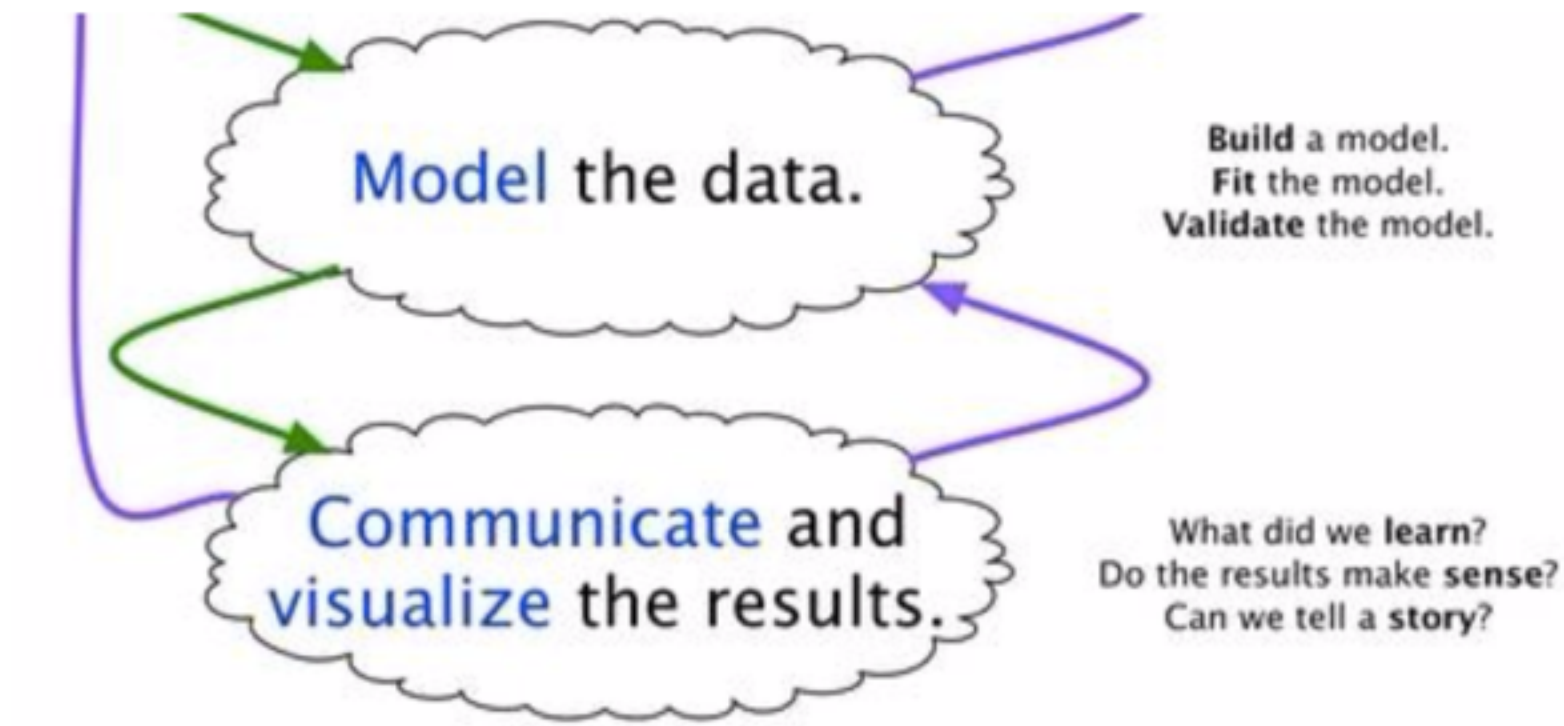
Wide variance in terms of skillsets: many job descriptions are more appropriate for a **team of data scientists!**

python
command line
golang
documentation
debugging

linear algebra
probability
Baye's Rule
GLMs
mathematical maturity

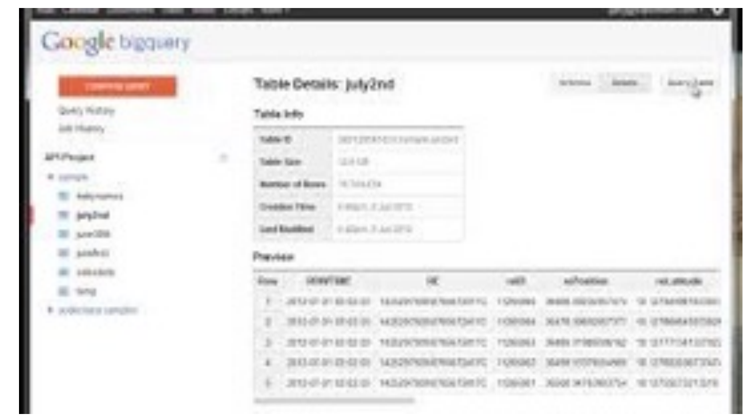
Source: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram


Ask an interesting question.
What is the scientific goal?
What would you do if you had all the data?
What do you want to predict or estimate?
Get the data.
How were the data sampled?
Which data are relevant?
Are there privacy issues?
Explore the data.
Plot the data.
Are there anomalies?
Are there patterns?

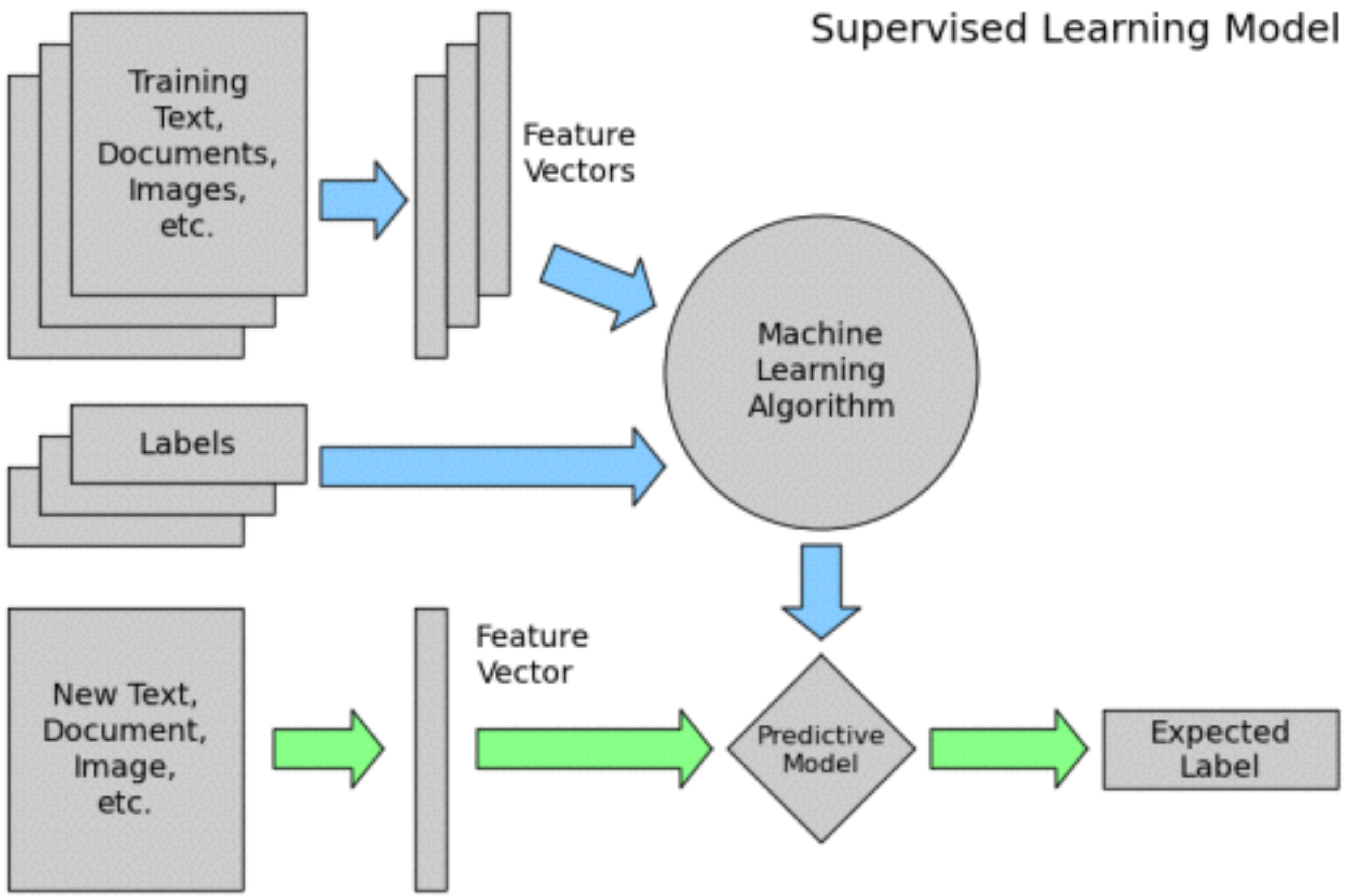


Source: https://www.quora.com/What-is-the-work-flow-or-process-of-a-data-scientist-analyst-and-what-tools-do-you-use-for-this/answer/Ryan-Fox-Squire

Exploring the data - from Excel to BigQuery



- summarising: min, max, mean, variance
- cleaning: outliers, junk data
- initial visualisation: pie, histogram, line
- analytical transformations: machine learning


Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Simultaneous System - Overdetermined (n > d)

“More equations (constraints) than variables”

Usually the case!! No solution?

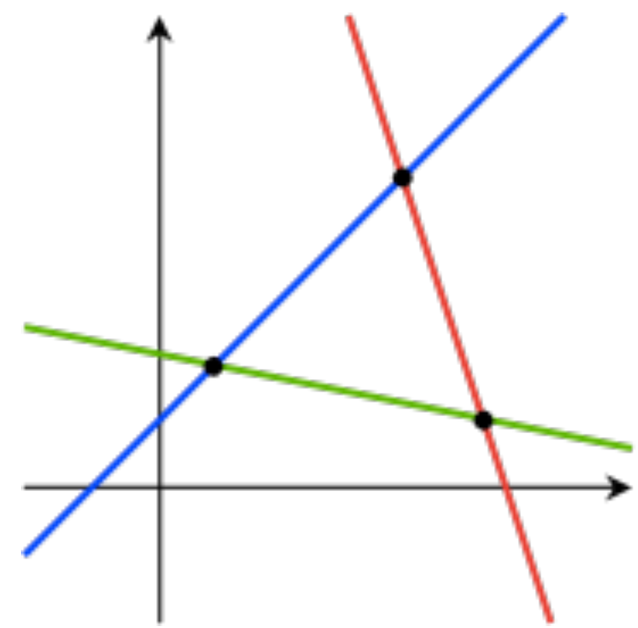
Fisher's *Iris* Data

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*

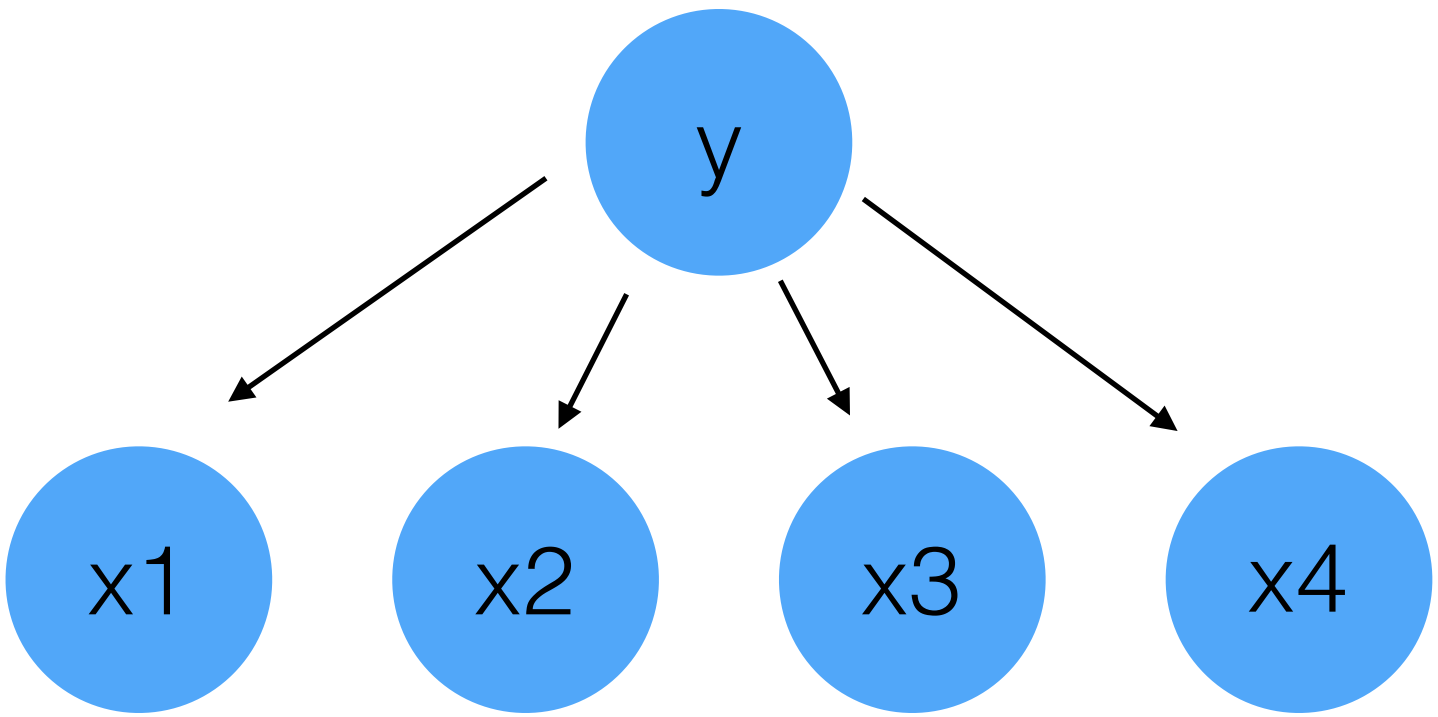
150 observations *(n = 150)*

4 features (p = 4)

response



Naïve Bayes



core idea? equation?

use cases?

tradeoffs?

p(x_i, x_j | y) = p(x_i | y)

Linear Regression

Squared Error Loss

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient Descent

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}
$$

Update Rule: $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every } j\text{)}.$$

}

The Normal Equations: the analytical approach

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T (X\theta - \vec{y}) \\
&= \frac{1}{2}\nabla_\theta \left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \operatorname{tr}\left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right) \\
&= \frac{1}{2}\nabla_\theta \left(\operatorname{tr}\theta^T X^T X\theta - 2\operatorname{tr}\vec{y}^T X\theta\right) \\
&= \frac{1}{2}\left(X^T X\theta + X^T X\theta - 2X^T \vec{y}\right) \\
&= X^T X\theta - X^T \vec{y}
\end{aligned}
$$

Probabilistic Interpretation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right).$$

We can minimise **cost**
or
maximise likelihood

$$\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2.
\end{aligned}$$

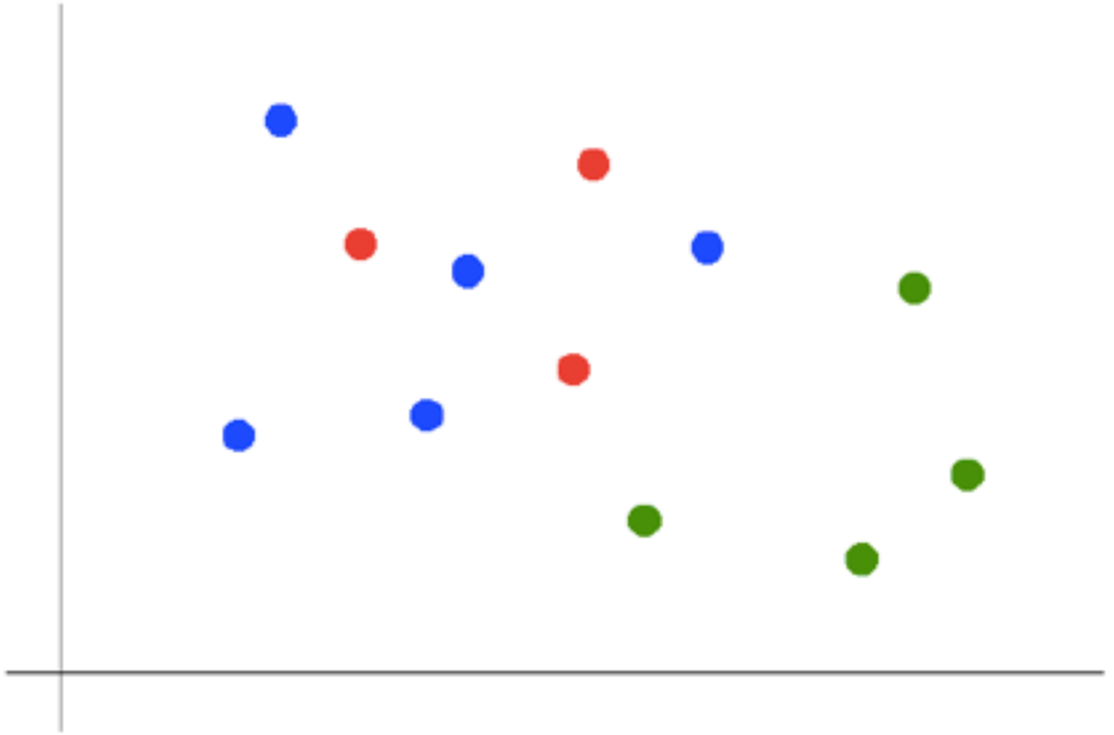
Hence, maximizing $\ell(\theta)$ gives the same answer as minimizing

$$\frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2,$$

kNN

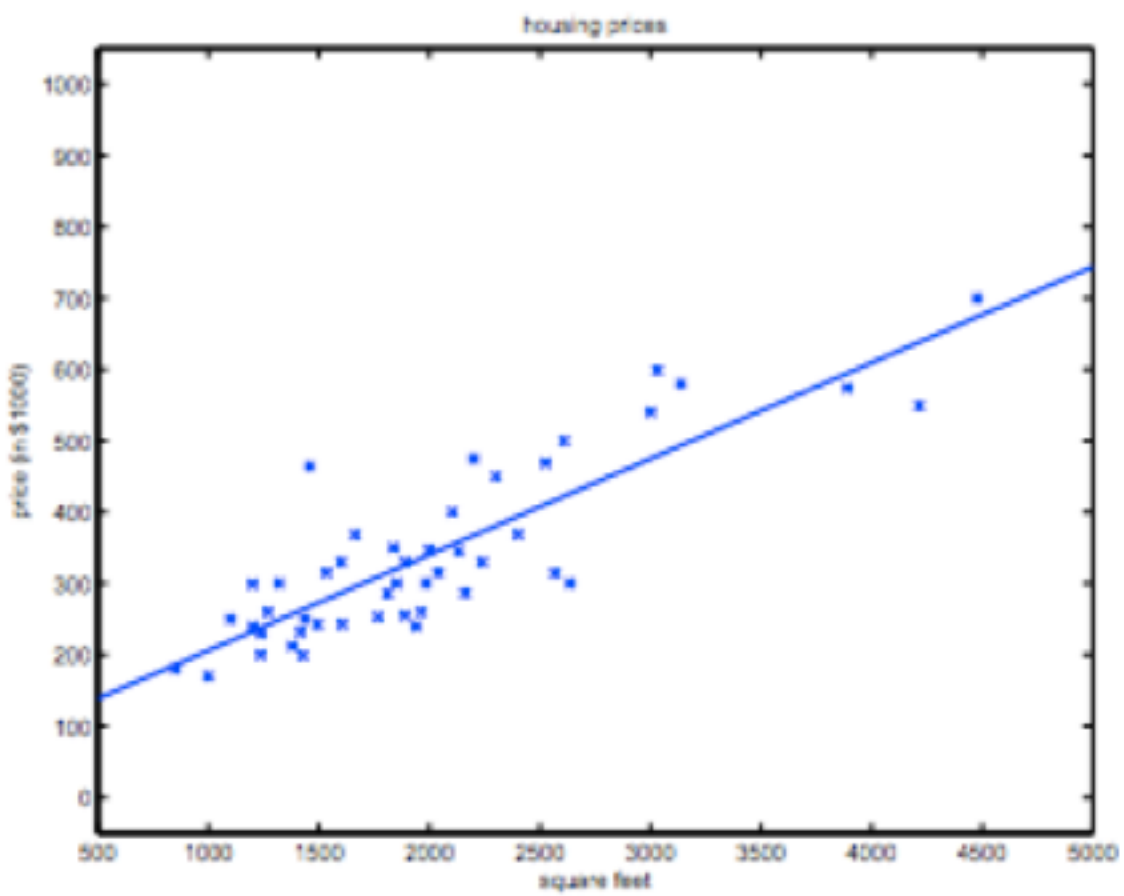
Suppose we want to predict the color of the gray dot.

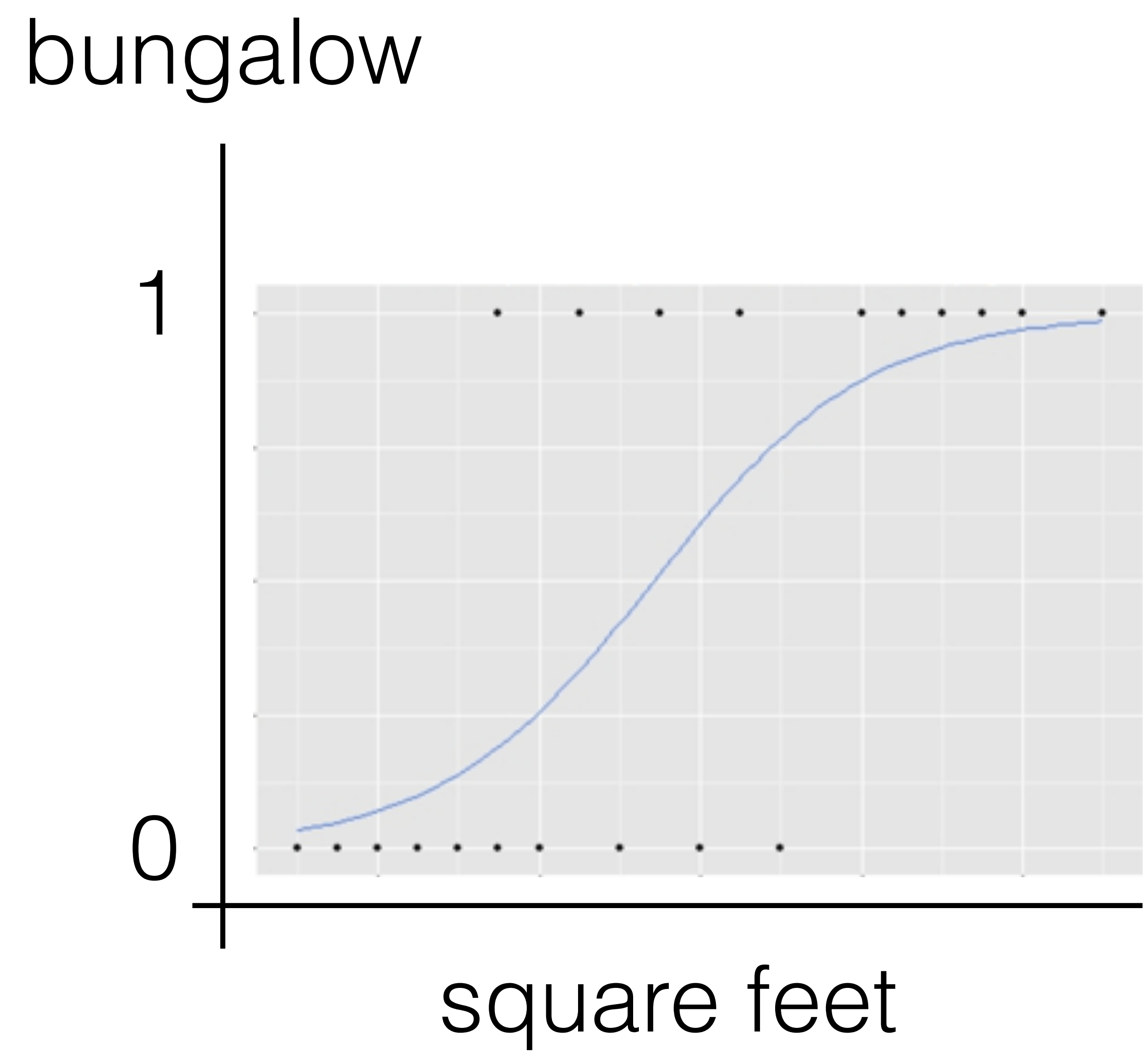
1) Pick a value for k.
2) Find colors of k nearest neighbors.
3) Assign the most common color to the gray dot.



Logistic Regression & GLMs

Linear to Logistic





Problems with just using linear regression to classify?

Classification vs Clustering?

Examples of Classification?

GLMs

We’ve seen

y | x ~ N(mu, sigma) $\longrightarrow$ linear regression

y | x ~ Bernoulli(phi) $\longrightarrow$ logistic classification

Can we find common ground?

The Exponential Family

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta))$$

The Exponential Family: Bernoulli

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta))$$

$$\begin{aligned}
p(y;\phi) &= \phi^y (1-\phi)^{1-y} \\
&= \exp(y \log\phi + (1-y)\log(1-\phi)) \\
&= \exp\left(\left(\log\left(\frac{\phi}{1-\phi}\right)\right) y + \log(1-\phi)\right)
\end{aligned}$$

What are…

η = log(ϕ/(1 − ϕ)).

$$\begin{aligned}
T(y) &= y \\
a(\eta) &= -\log(1-\phi) \\
&= \log(1+e^\eta) \\
b(y) &= 1
\end{aligned}$$

The Exponential Family: Normal

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta))$$

$$\begin{aligned}
p(y;\mu) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(y-\mu)^2\right) \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}y^2\right)\cdot\exp\left(\mu y - \frac{1}{2}\mu^2\right)
\end{aligned}$$

What are…

$$\begin{aligned}
\eta &= \mu \\
T(y) &= y \\
a(\eta) &= \mu^2/2 \\
&= \eta^2/2 \\
b(y) &= (1/\sqrt{2\pi})\exp(-y^2/2)
\end{aligned}$$

Okay, okay…so who cares?

Constructing GLMs

1. Assume $y \mid x; \theta \sim \text{ExponentialFamily}(\eta)$

2. Given x, we want to predict T(y), usually = y.
We choose h(x) = E[y|x]

3. Further assume $\eta = \theta^T x$

So we have a machinery we can crank

Constructing GLMs

Linear Regression

$$\begin{aligned} h_\theta(x) &= E[y|x;\theta] \\ &= \mu \\ &= \eta \\ &= \theta^T x. \end{aligned}$$

Logistic Classification

$$\begin{aligned} h_\theta(x) &= E[y|x;\theta] \\ &= \phi \\ &= 1/(1+e^{-\eta}) \\ &= 1/(1+e^{-\theta^T x}) \end{aligned}$$

Coincidentally, this is how we get softmax regression…

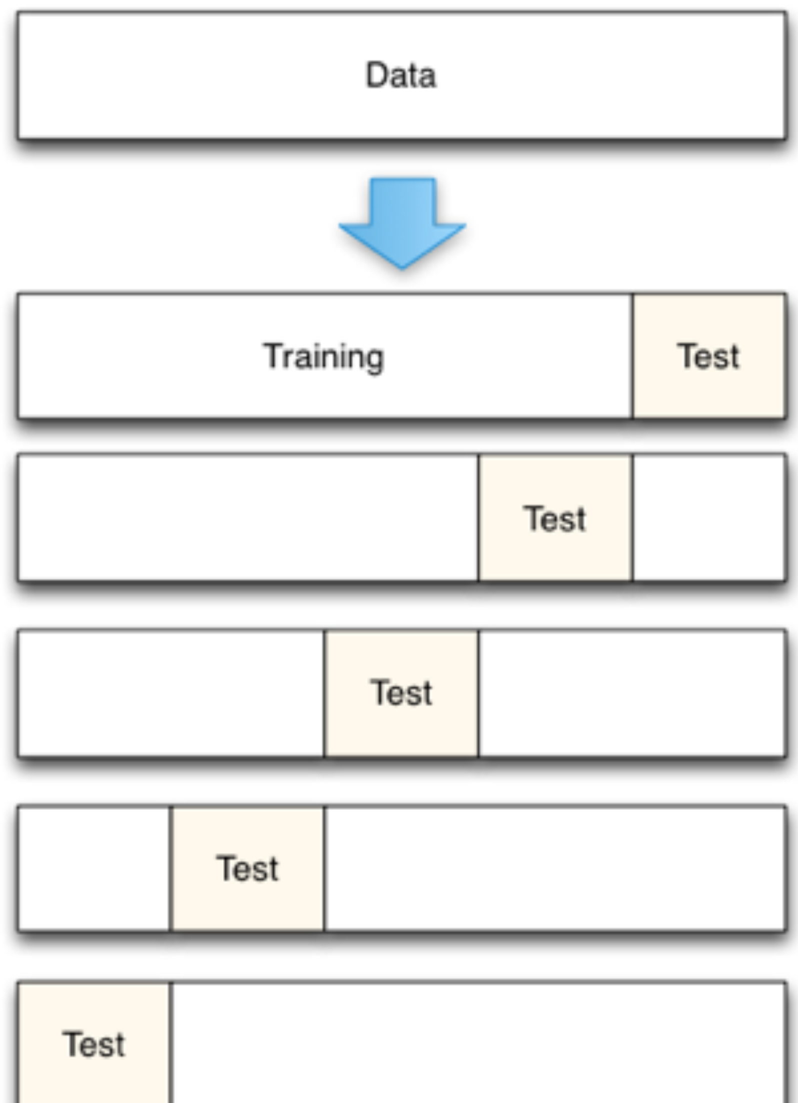
Relationship to Naïve Bayes

Assuming $y \mid x \sim$ some distribution

Assuming $x \mid y \sim$ some distribution

e.g.
text classification
Gaussian Discriminant Analysis (GDA)

Model Evaluation



import data, clean dataframe, visualise

instantiate Model(), fit_(transform), predict

cross-validation on parameters (external), features, and models

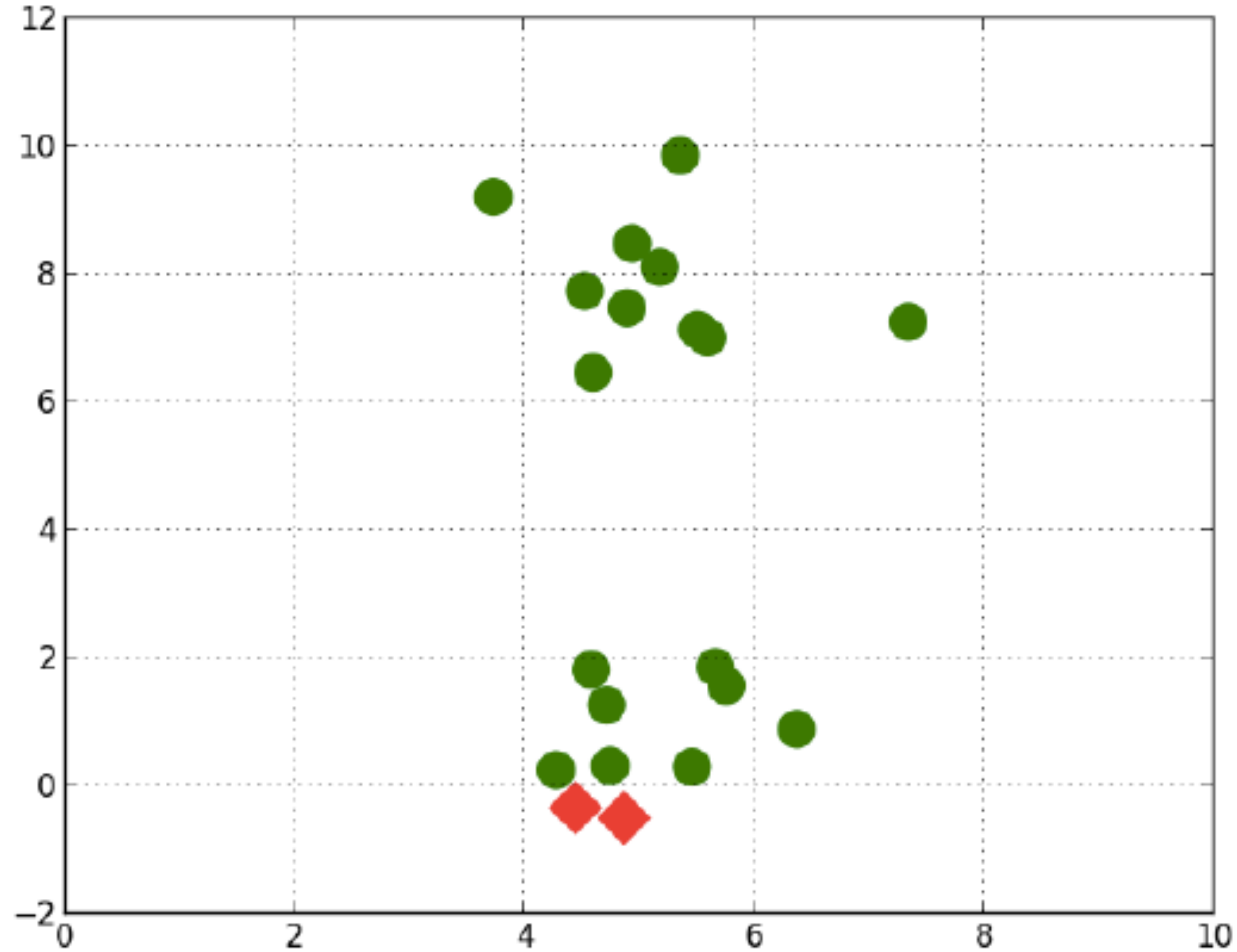
Clustering

- **Clustering**, or **cluster analysis**, is the task of grouping observations such that members of the same group, or **cluster**, are more similar to each other by some metric than they are to the members of the other clusters

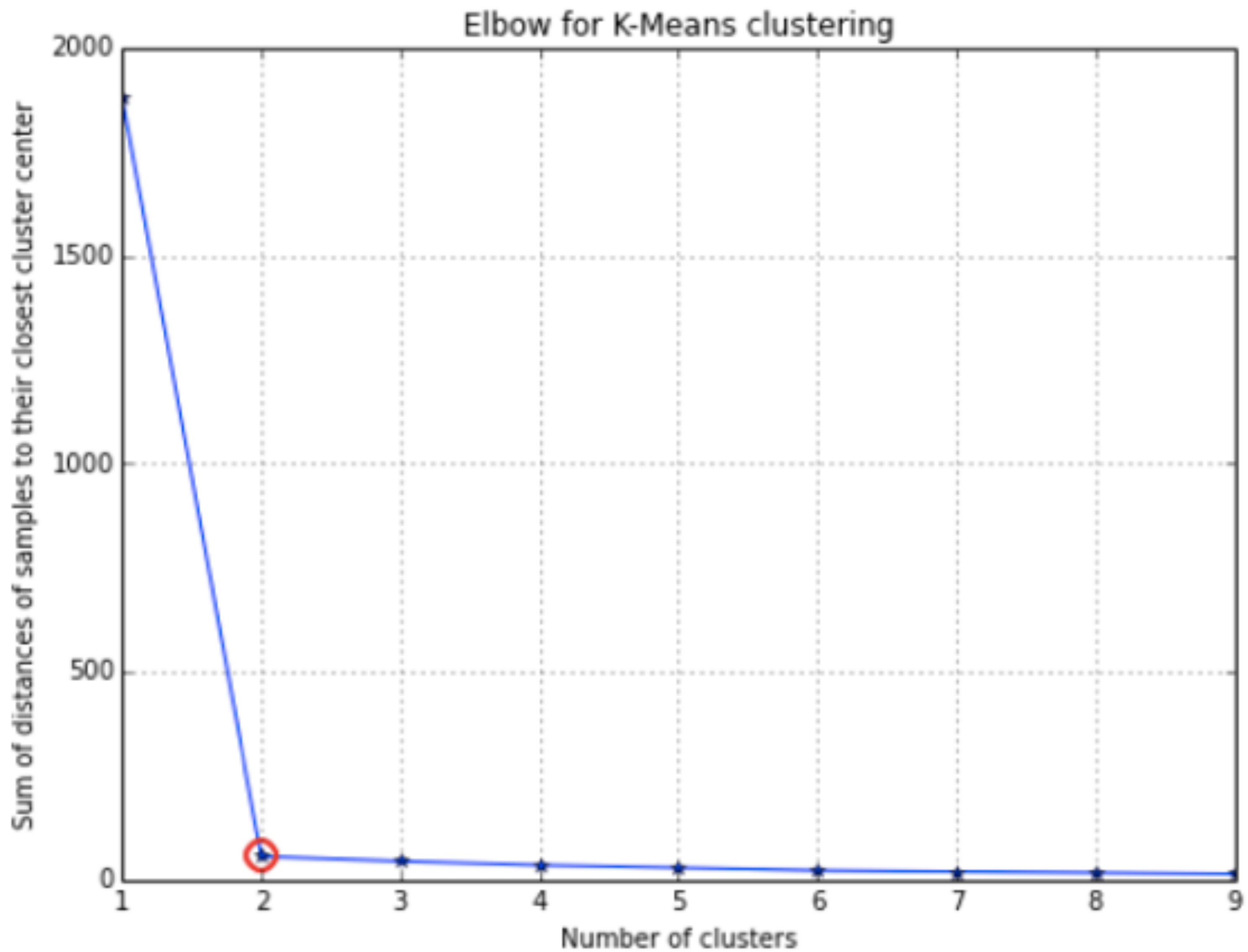
THE BASIC K-MEANS ALGORITHM

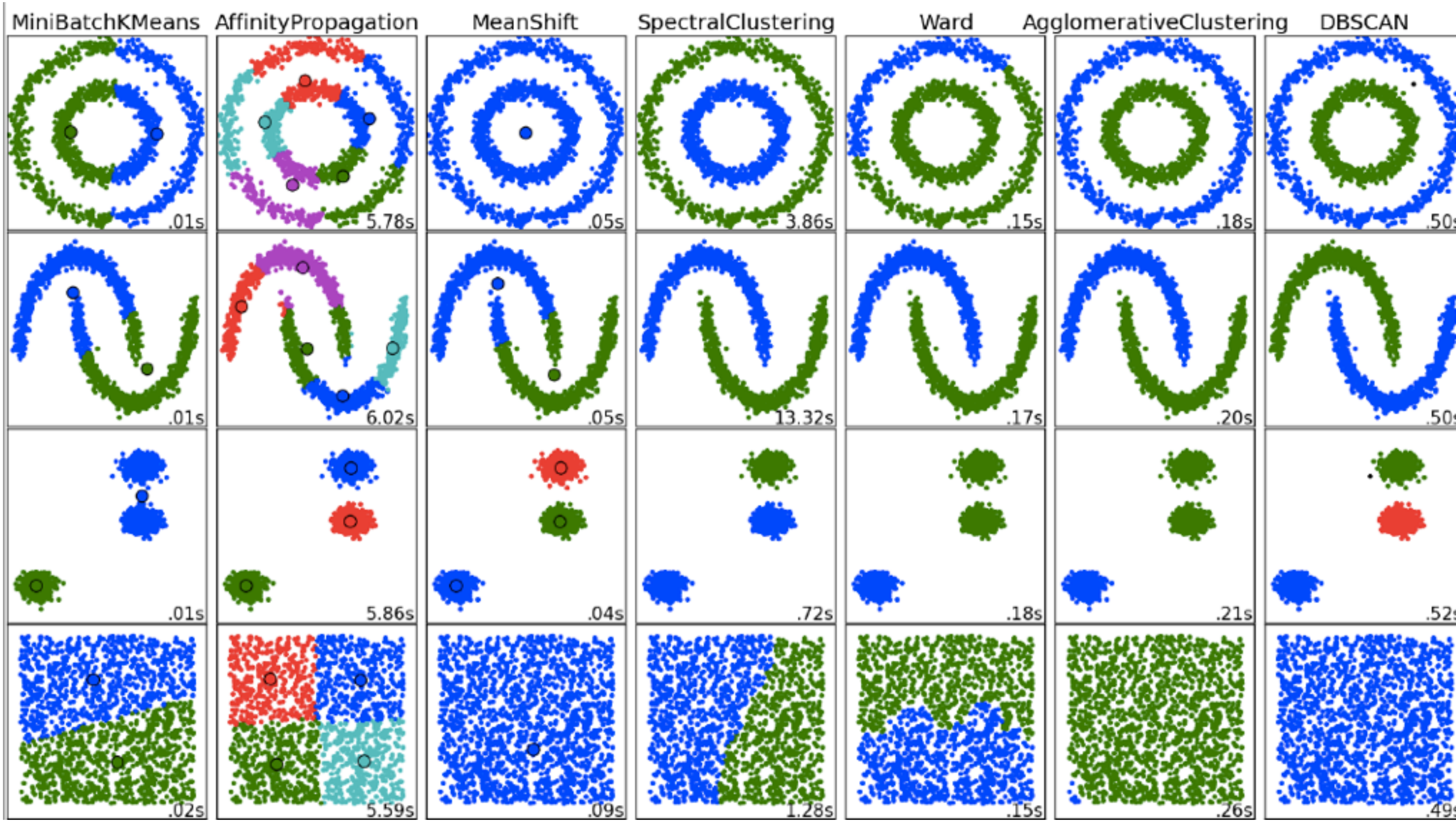
1. choose ***k*** initial centroids (note that k is an input)
2. for each data point:
   - find distance to each centroid (k)
   - assign point to nearest centroid
3. recalculate centroid positions
4. repeat steps 2-3 until stopping criteria met

DISADVANTAGES OF K-MEANS



SELECTING K WITH THE ELBOW METHOD




MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
.01s
5.78s
.05s
3.86s
.15s
.18s
.50s
.01s
6.02s
.05s
13.32s
.17s
.20s
.50s
.01s
5.86s
.04s
.72s
.18s
.21s
.52s
.02s
5.59s
.09s
1.28s
.15s
.26s
.49s

SQL and MapReduce

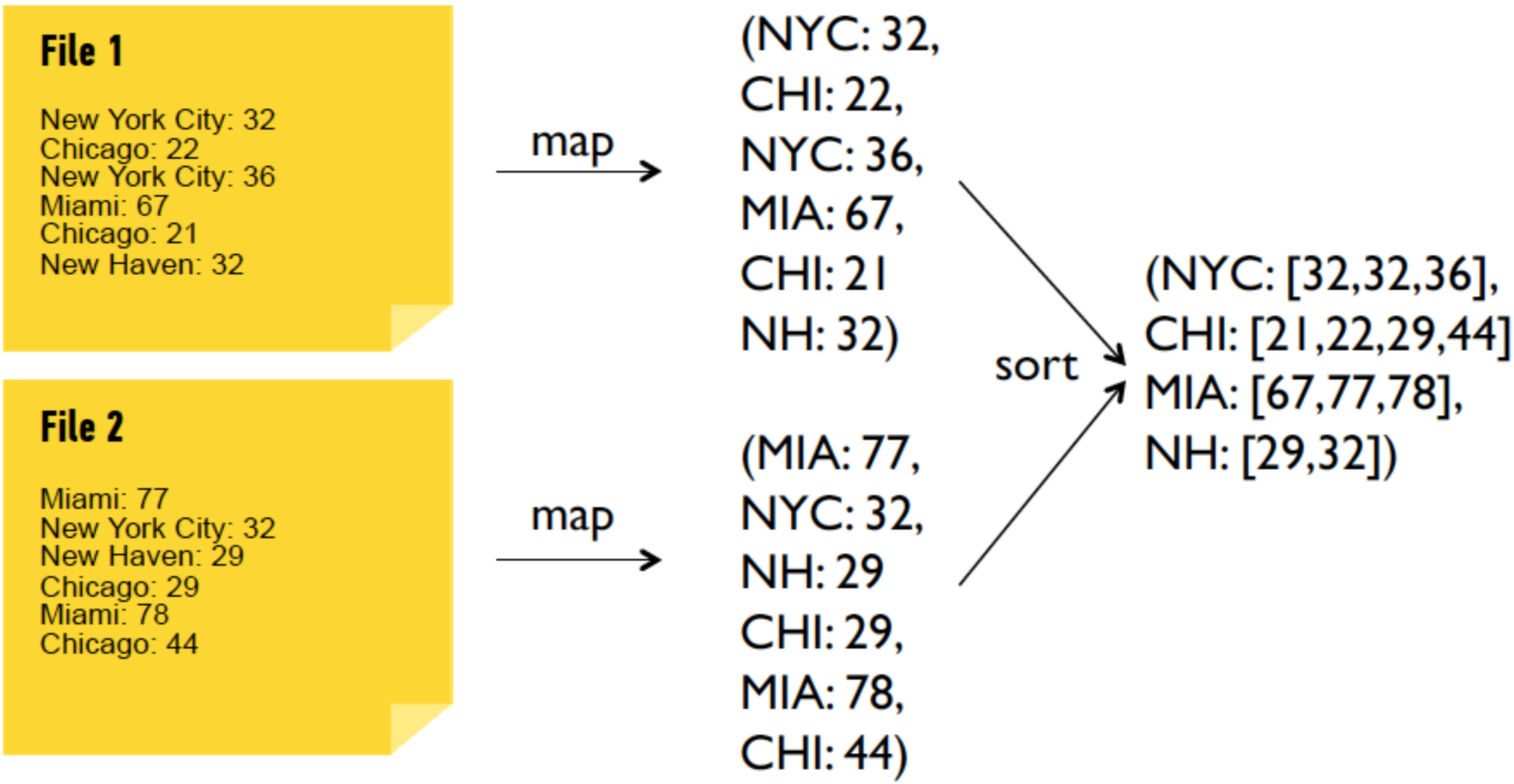
```
Product(PName, Price, Category, Manufacturer)
Company(CName, StockPrice, Country)
```

Several equivalent ways to write a basic join in SQL:

```
SELECT PName, Price
FROM   Product, Company
WHERE  Manufacturer = CName
        AND Country='Japan'
       AND Price <= 200
```

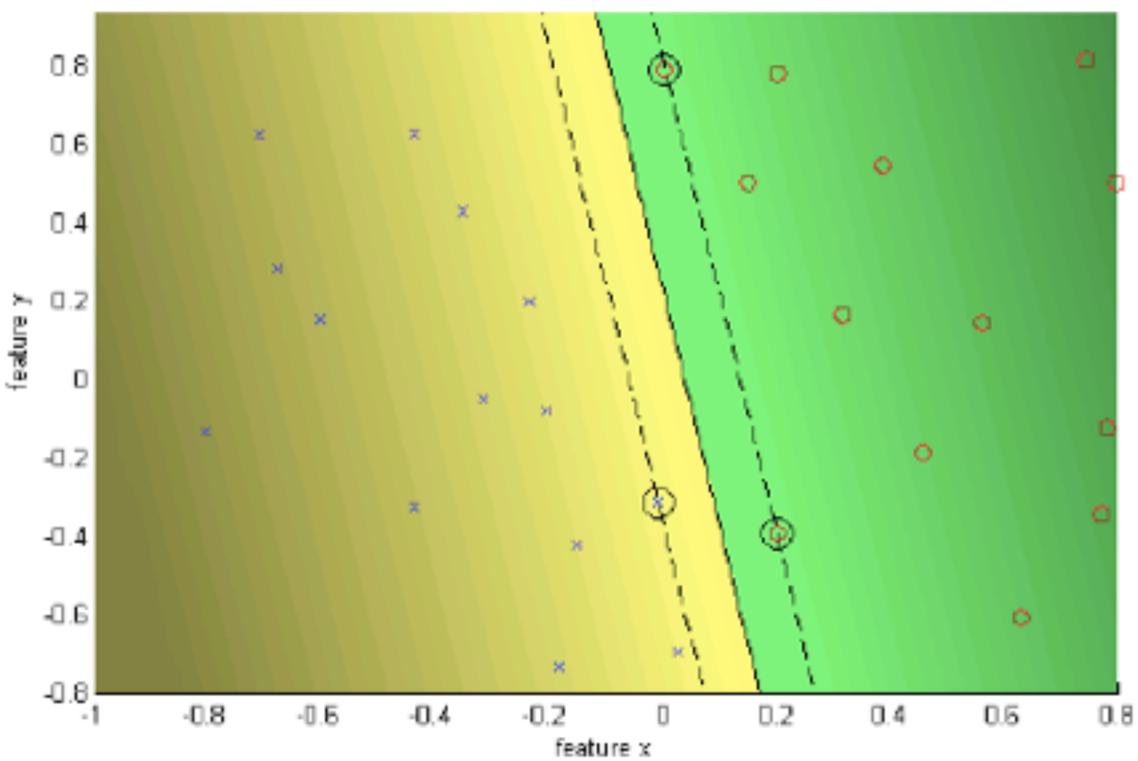
```
SELECT PName, Price
FROM   Product
JOIN   Company ON Manufacturer = Cname
                 AND Country='Japan'
WHERE  Price <= 200
```

MAPREDUCE: EXAMPLE

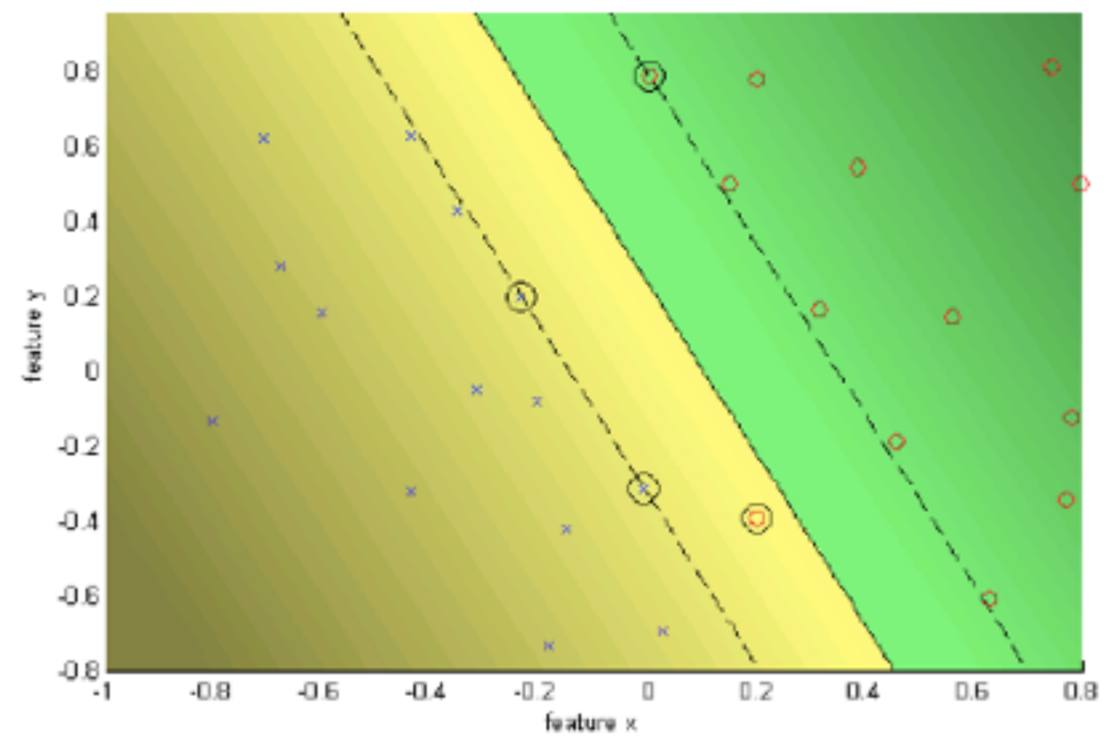


SVMs

C = Infinity   hard margin



C = 10   soft margin



The optimization problem becomes

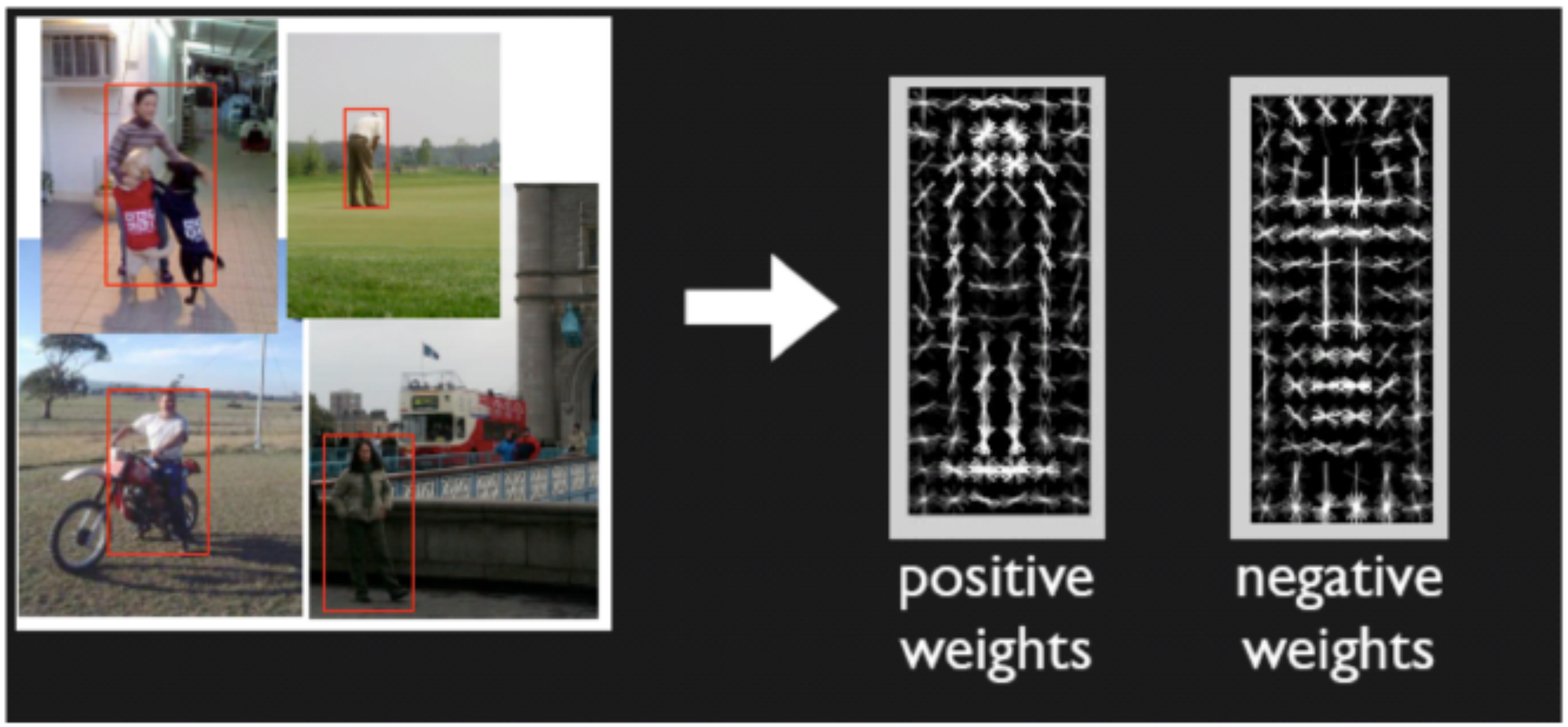
$$\min_{\mathbf{w}\in\mathbb{R}^d, \xi_i\in\mathbb{R}^+} ||\mathbf{w}||^2 + C\sum_i^N \xi_i$$

subject to

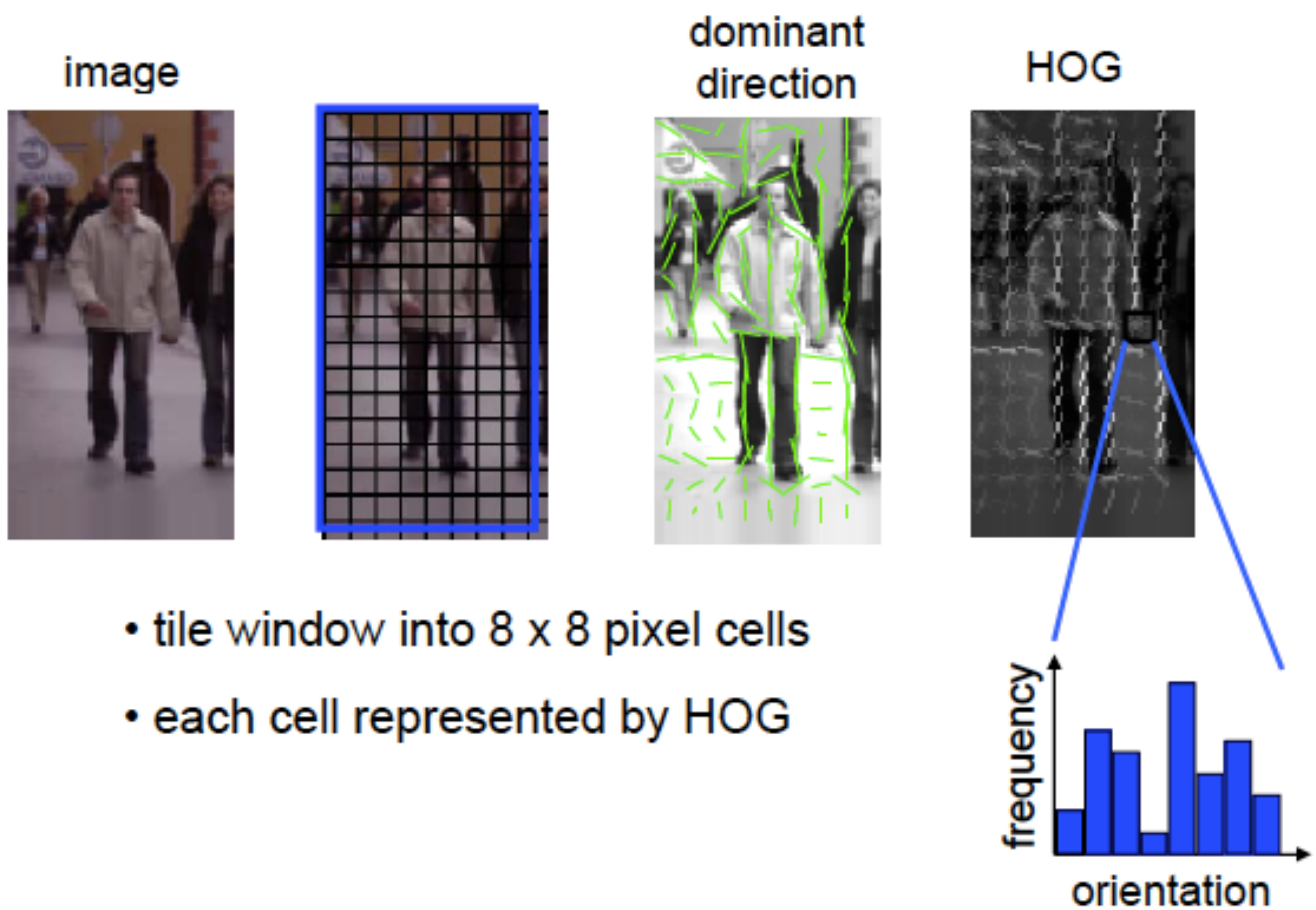
$$y_i\left(\mathbf{w}^\top\mathbf{x}_i + b\right) \geq 1 - \xi_i \text{ for } i = 1\dots N$$

Learned model

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



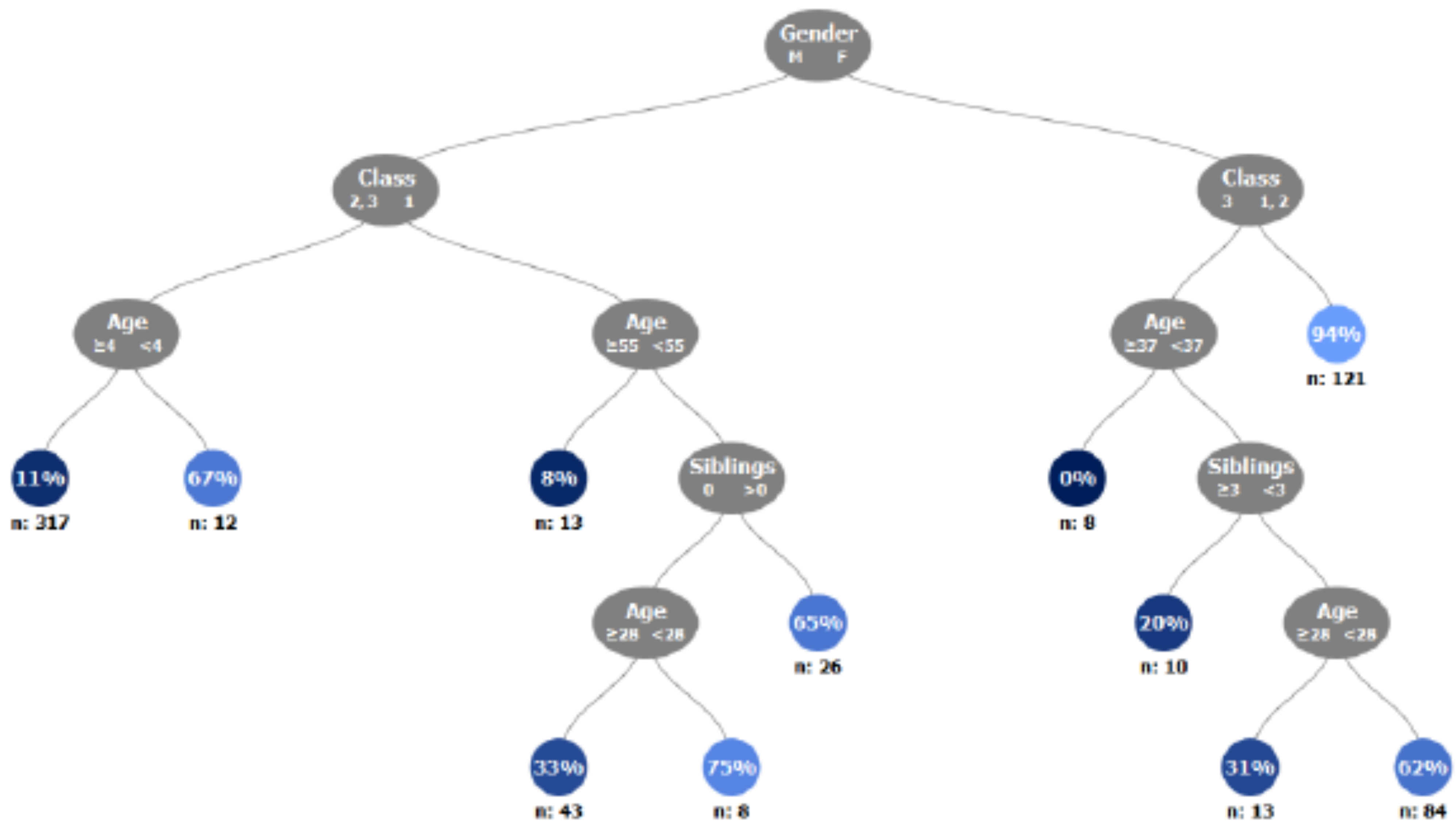
Feature: histogram of oriented gradients (HOG)



- tile window into 8 x 8 pixel cells
- each cell represented by HOG

Feature vector dimension = 16 x 8 (for tiling) x 8 (orientations) = 1024

Decision Trees



Before Split	All
Survived	10
Died	15

$$1 - \sum \left(\frac{class_i}{total} \right)^2$$

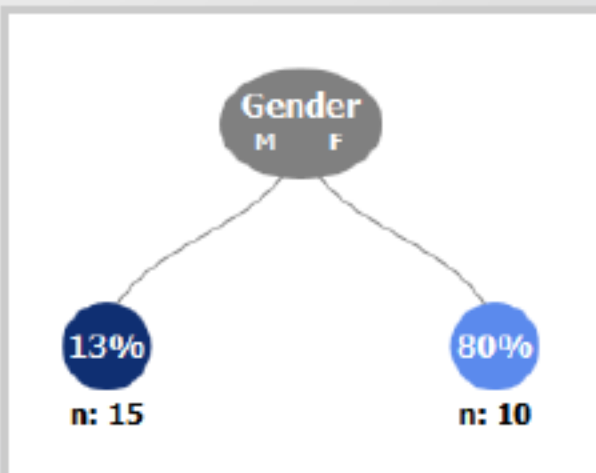
Before Split	All
Survived	10
Died	15

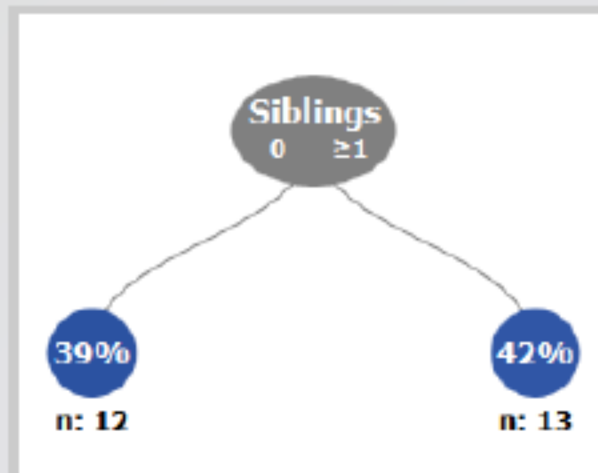
$$1 - \left(\frac{survived}{total}\right)^2 - \left(\frac{died}{total}\right)^2$$

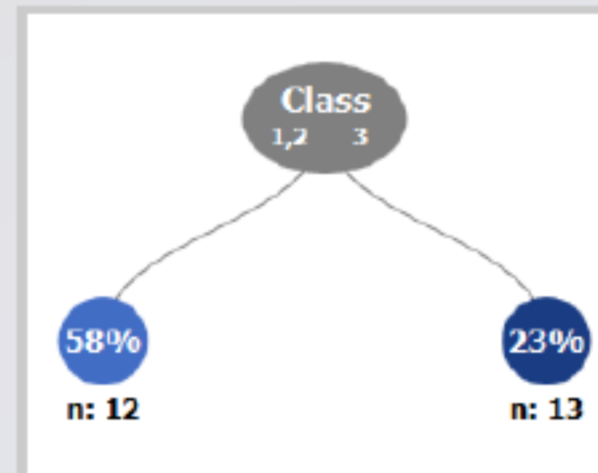
$$1 - \left(\frac{10}{25}\right)^2 - \left(\frac{15}{25}\right)^2 = 0.48$$

Choosing the Split

How does the gini coefficient compare for the Siblings and class variables?





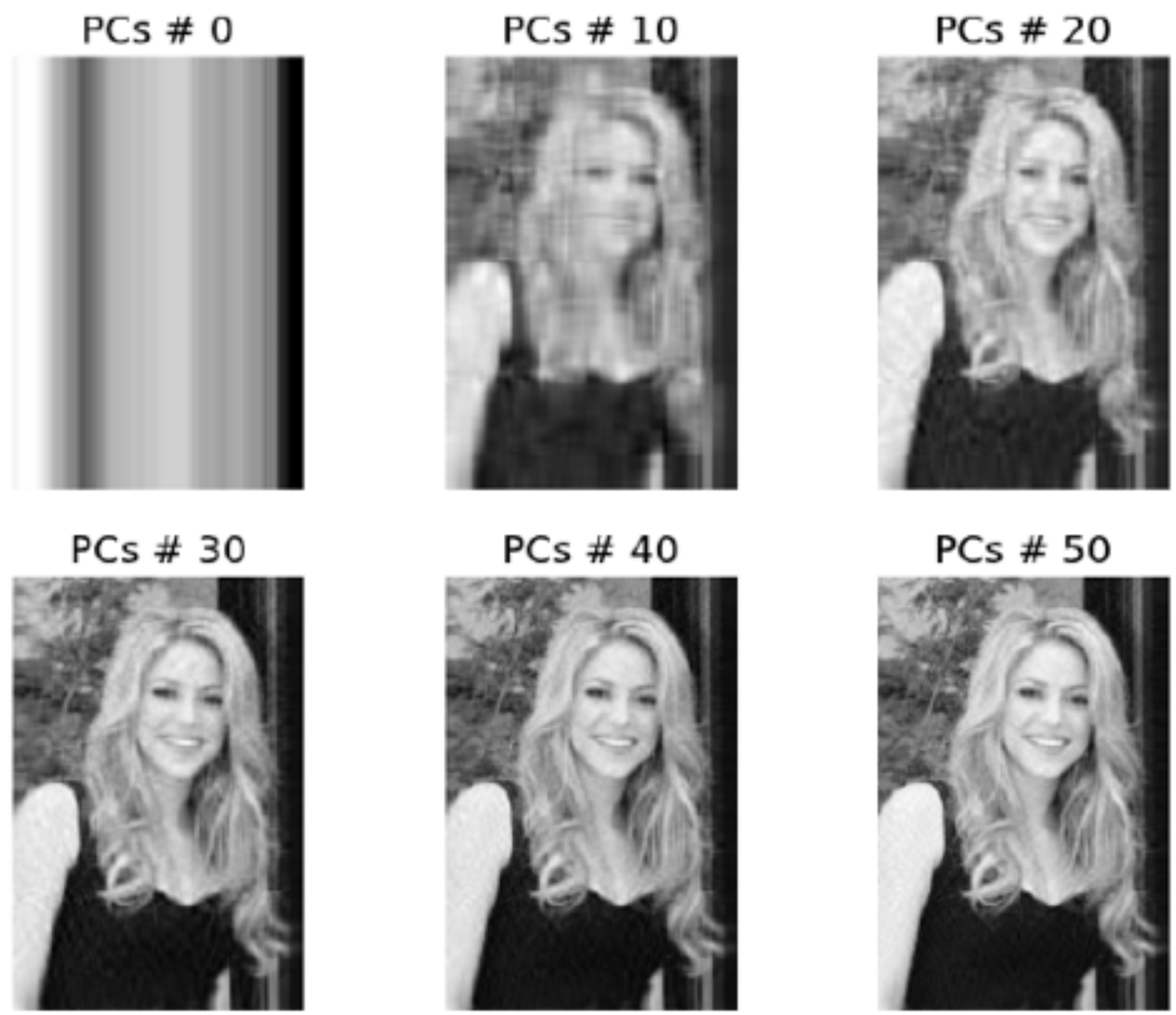


Gender	M	F
Survived	2	8
Died	13	2
$Gini_C$	0.27	

Siblings	0	≥1
Survived	5	5
Died	7	8
$Gini_C$	0.48	

Class	1,2	3
Survived	7	3
Died	5	10
$Gini_C$	0.42	

Dimensionality Reduction



source: http://glowingpython.blogspot.it/2011/07/pca-and-image-compression-with-numpy.html

ASIDE: EIGENVALUE DECOMPOSITION

The *eigenvalue decomposition* of a square matrix C is given by:

$$C = Q \Lambda Q^{-1}$$

The columns of Q are the eigenvectors of C, and the values in Λ are the associated eigenvalues of C.

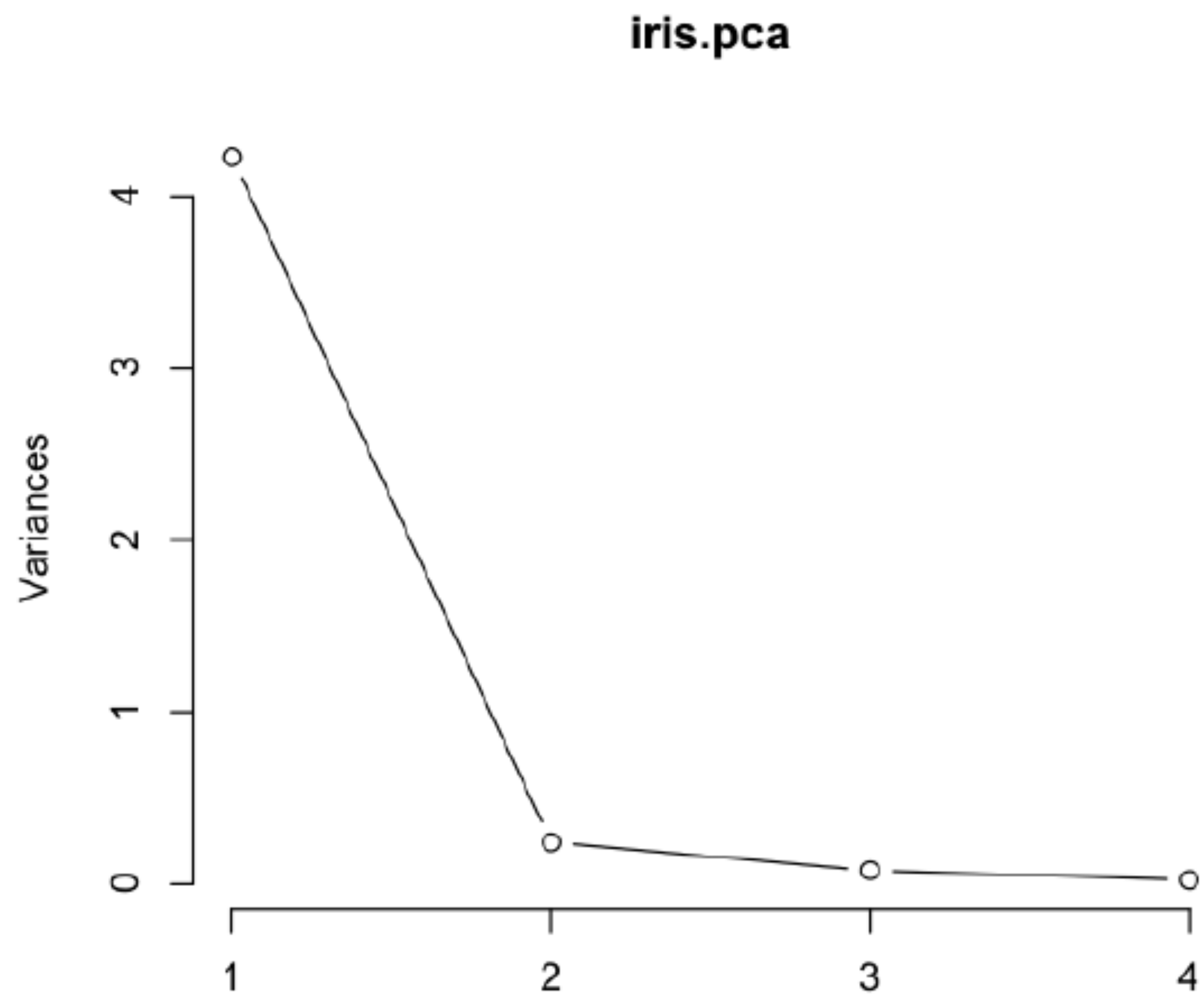
For an eigenvector v of C and its eigenvalue λ, w
the important relation:

$$Cv = \lambda v$$

NOTE

This relationship *defines* what it means to be an eigenvector of C.

PRINCIPAL COMPONENT ANALYSIS



Computer Science

- Python, command line, web HTML/CSS/JS, Golang
- Decomposition, scope, recursion, memory hierarchy and pointer
- Documentation and debugging, soft skills

Learning Theory

Reinforcement Learning, HMMs

Theorem. Let $|\mathcal{H}| = k$, and let any m, δ be fixed. Then with probability at least $1 - \delta$, we have that

$$\varepsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \varepsilon(h) \right) + 2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}.$$

lower bias: increasing H class

variance tradeoff of increasing H class

Example

For instance, we can ask the following question: Given γ and some $\delta > 0$, how large must m be before we can guarantee that with probability at least $1 - \delta$, training error will be within γ of generalization error? By setting $\delta = 2k \exp(-2\gamma^2 m)$ and solving for m, [you should convince yourself this is the right thing to do!], we find that if

$$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta},$$

then with probability at least $1 - \delta$, we have that $|\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma$ for all $h \in \mathcal{H}$. (Equivalently, this shows that the probability that $|\varepsilon(h) - \hat{\varepsilon}(h)| > \gamma$ for some $h \in \mathcal{H}$ is at most δ.) This bound tells us how many training examples we need in order make a guarantee. The training set size m that a certain method or algorithm requires in order to achieve a certain level of performance is also called the algorithm's **sample complexity**.

Reinforcement Learning

A Markov decision process is a tuple $(S, A, \{P_{sa}\}, \gamma, R)$, where:

- S is a set of **states**. (For example, in autonomous helicopter flight, S might be the set of all possible positions and orientations of the helicopter.)

- A is a set of **actions**. (For example, the set of all possible directions in which you can push the helicopter's control sticks.)

- P_{sa} are the state transition probabilities. For each state $s \in S$ and action $a \in A$, P_{sa} is a distribution over the state space. We'll say more about this later, but briefly, P_{sa} gives the distribution over what states we will transition to if we take action a in state s.

- $\gamma \in [0, 1)$ is called the **discount factor**.

- $R : S \times A \mapsto \mathbb{R}$ is the **reward function**. (Rewards are sometimes also written as a function of a state S only, in which case we would have $R : S \mapsto \mathbb{R}$).

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots$$

$$s_1 \sim P_{s_0 a_0}.$$

Upon visiting the sequence of states $s_0, s_1, \ldots$ with actions $a_0, a_1, \ldots$, our total payoff is given by

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots .$$

Or, when we are writing rewards as a function of the states only, this becomes

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots .$$

Policy to choose actions

$$\pi : S \mapsto A$$

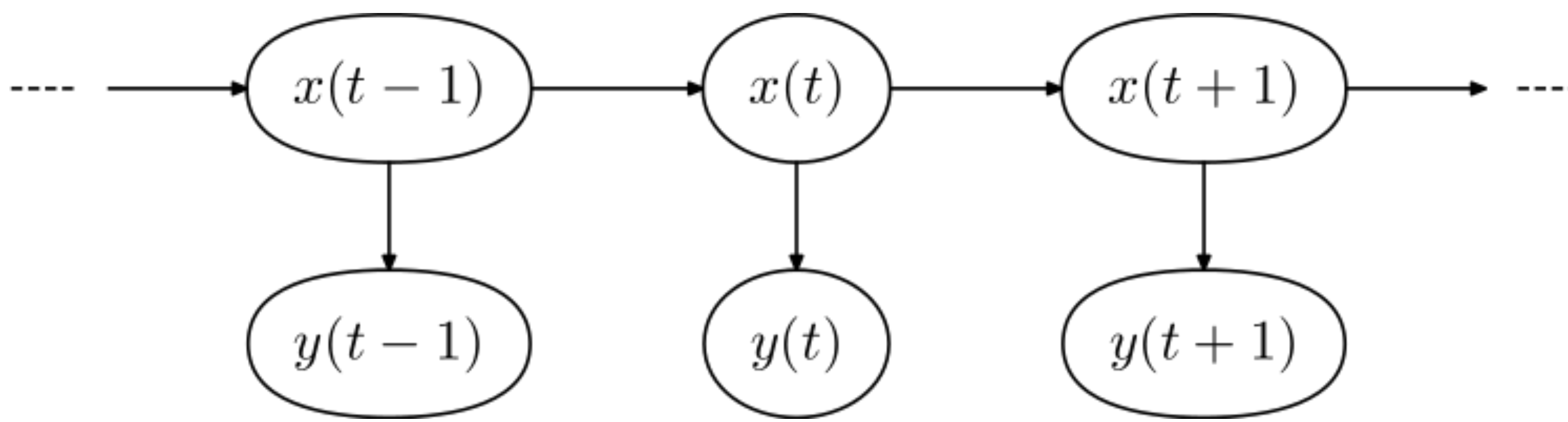
$$a = \pi(s)$$

Value function based on policy

$$V^{\pi}(s) = \mathrm{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \,\middle|\, s_0 = s, \pi\right].$$

Hidden Markov Models

We don't observe the sequence directly



e.g. words vs audio waves

use observe sequence **z**

$$
\begin{aligned}
P(\vec{x}; A, B) &= \sum_{\vec{z}} P(\vec{x}, \vec{z}; A, B) \\
&= \sum_{\vec{z}} P(\vec{x}|\vec{z}; A, B) P(\vec{z}; A, B)
\end{aligned}
$$

use HMM assumptions

$$
\begin{aligned}
P(\vec{x}; A, B) &= \sum_{\vec{z}} P(\vec{x}|\vec{z}; A, B) P(\vec{z}; A, B) \\
&= \sum_{\vec{z}} \left(\prod_{t=1}^{T} P(x_t|z_t; B)\right) \left(\prod_{t=1}^{T} P(z_t|z_{t-1}; A)\right) \\
&= \sum_{\vec{z}} \left(\prod_{t=1}^{T} B_{z_t\, x_t}\right) \left(\prod_{t=1}^{T} A_{z_{t-1}\, z_t}\right)
\end{aligned}
$$