Introduction to Statistical Learning and Machine Learning

Chap 8
Computational Learning Theory&Mid-term Review (1)

Chap 8 Computational Learning Theory

Generalisation of finite hypothesis spaces;
VC-dimension
Margin based generalisation

What's next….

Optional subtitle

We gave several machine learning algorithms:

- Perceptron
- Linear Support vector Machine
- SVM with kernels, e.g. polynomial or Gaussian

How do we guarantee that the learned classifier will perform well on test data?

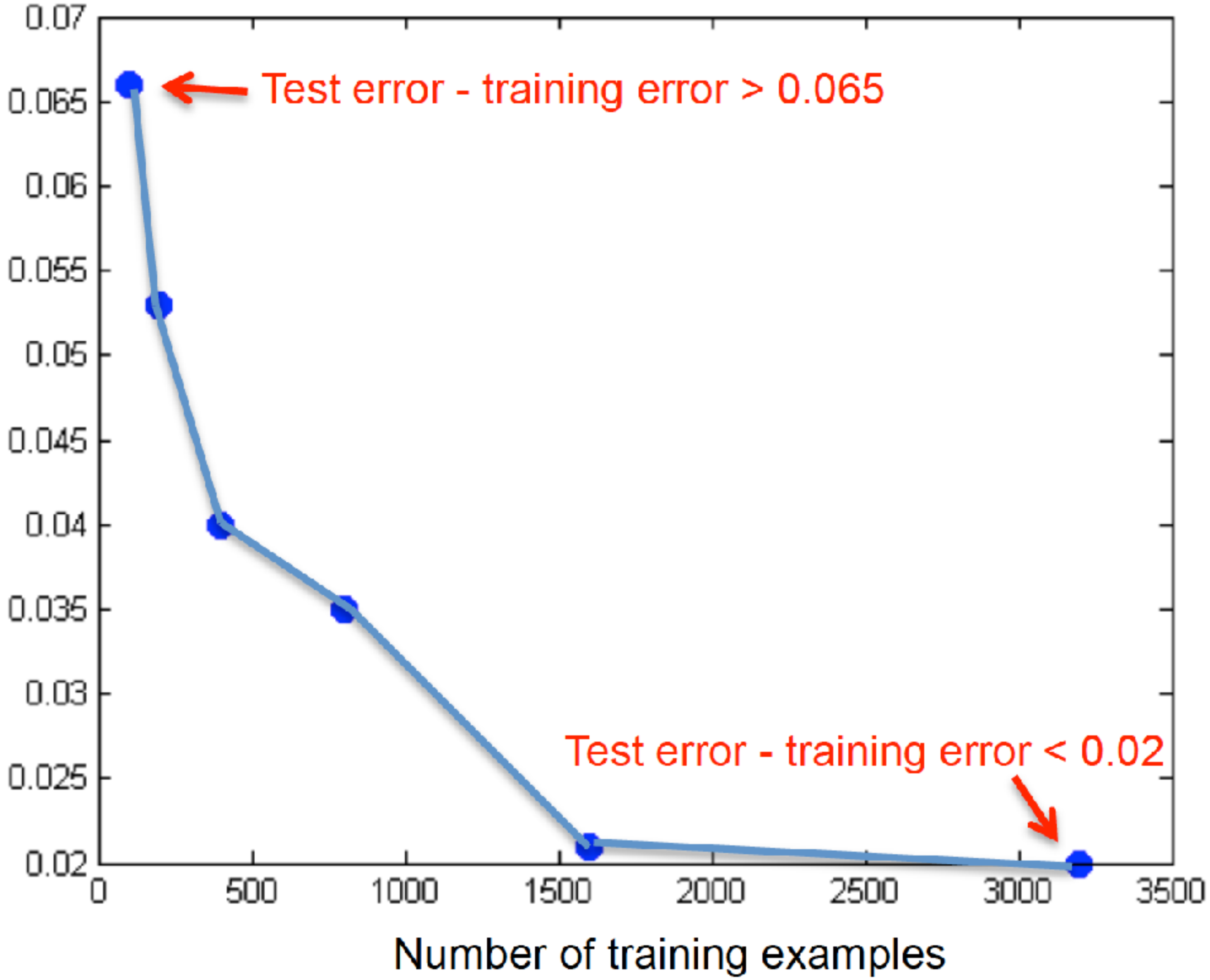
How much training data do we need?

Example: Perceptron applied to spam classification

Optional subtitle

With few data points,
there was a big gap between
training error and test error!

This is the difficulty of one-shot learning.



How much training data do you need?

In general, not the one-shot learning case

- Depends on what Hypothesis class the learning algorithm considers
- For example, consider an Instance-based Learning algorithm
  - Input: training data $S=\{(x_i,y_i)\}$
  - Output: function $f(x)$ which, if there exists (x_i,y_i) in S such that $x=x_i$, predicts y_i, and otherwise predicts the majority label,
  - this learning algorithm will always obtain zero training error
  - But, it will take a huge amount of training data to obtain small test error (i.e. its generalisation performance is horrible).

- Linear classifiers are powerful precisely because of its simplicity
  - Generalisation is easy to guarantee

Choosing among several classifiers

A fictional example

Suppose Alibaba holds a competition for the best face recognition classifier (+1 if image contains a face, -1, otherwise)

Lots of teams compete …

Alibaba get back 20,000 recognition algorithm

They evaluate all 20,000 algorithm on *m* labelled images which is not previously shown to the competitors) and chooses a winner.

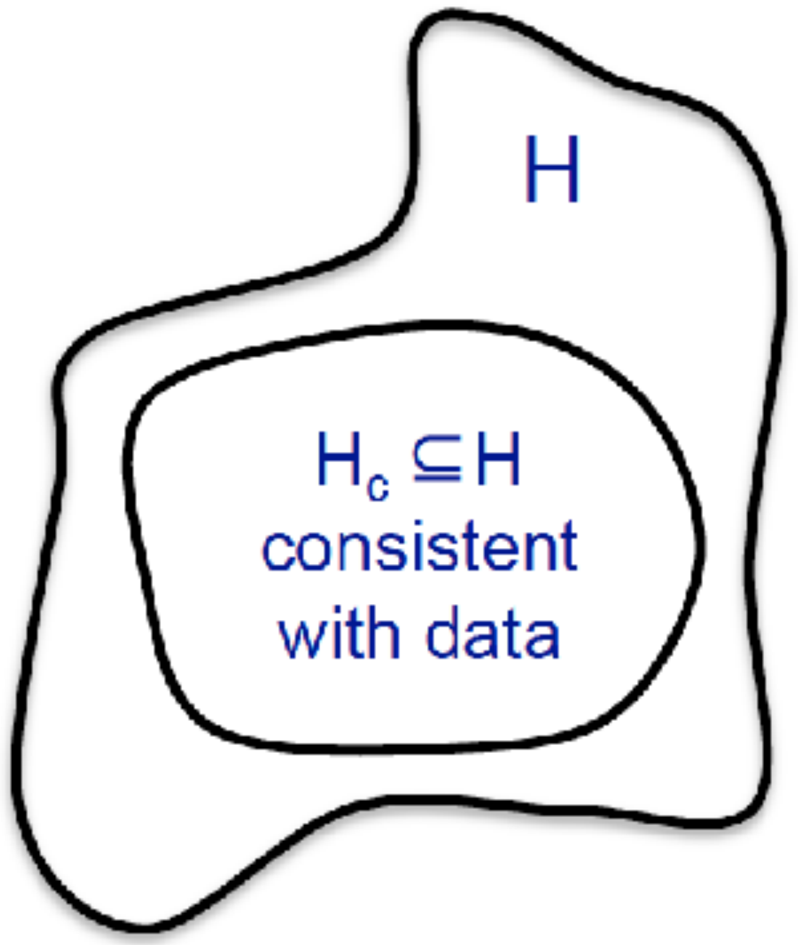
The winner obtains 98% accuracy on *m* labelled images!

Alibaba has a face recognition algorithm that is known to be 95% accurate,

- Should they deploy the winner's algorithm instead?
- Can't risk doing worse … would be a disaster for Alibaba.

A simple setting…

Optional subtitle



- Classification
  - *m* data points
  - Finite number of possible hypothesis (e.g. 20000 face recognition classifiers)
- A learner finds a hypothesis *h* that is consistent with training data
  - Gets zero error in training: $error_{train}(h)=0$
  - i.e. assume for now that the winner gets 100% accuracy on the *m* labelled images (we'll handle 98% case afterward)
- What is the probability that *h* has more than ε true error?
  - $error_{true}(h) \geq \varepsilon$

A simple setting— Finite number of possible hypothesis

- Empirical Risk Minimisation(ERM)
  - training set $\mathcal{S}$ from an unknown distribution $\mathcal{D}$; labeled by target function f; Output: h_S: $\mathcal{X} \rightarrow \mathcal{Y}$;
  - Empirical error/empirical risk/training error: errors of classifier incurs over the training sample.
  - ERM may go wrong —- Overfitting.
- Empirical Risk Minimisation with Inductive Bias
  - A common solution is to apply the ERM learning rule over a restricted search space.
  - the learner should choose in advance (before seeing the data) a set of predictors. This set is called a hypothesis class and is denoted by $\mathcal{H}$. Each h in $\mathcal{H}$ function mapping from $\mathcal{X}$ to $\mathcal{Y}$. For a given class $\mathcal{H}$, and a training sample $\mathcal{S}$, the ERM_H learner uses the ERM rule to choose a predictor h **with the lowest possible error over** $\mathcal{S}$.
  - **Such restrictions are often called an inductive bias.**

(通常的解决方案是在一个受限的搜索空间使用ERM学些规则）。

Some Concepts

Optional subtitle

- Empirical Risk Minimisation (ERM) 经验风险最小化
  - 对于Learner 而言，训练样本是真实世界的一个缩影，因此利用训练集来寻找一个对于数据的可行解是合理的。
- Overfitting： 一个预测器在训练集上的效果非常优秀，但是在真实世界中的表示非常糟糕。
  - 正如日常生活中，一个人如果能对自己的每个行为都做出完美的解释，那么这个人是容易令人产生怀疑的。

Chap8

Recap — probability

Introduction to probability: outcomes

- An **outcome space** specifies the possible outcomes that we would like to reason about, e.g.

$\Omega = \{$  ,  $\}$ Coin toss

$\Omega = \{$  $\}$ Die toss

- We specify a **probability** p(**x**) for each outcome **x** such that

$$p(x) \geq 0, \qquad \sum_{x \in \Omega} p(x) = 1$$

E.g., p() = .6

p() = .4

Introduction to probability: events

Optional subtitle

- An **event** is a subset of the outcome space, e.g.

E = { , ,  } Even die tosses

O = { , ,  } Odd die tosses

- The **probability** of an event is given by the sum of the probabilities of the outcomes it contains,

$$p(E) = \sum_{x \in E} p(x)$$

E.g., p(E) = p() + p() + p()

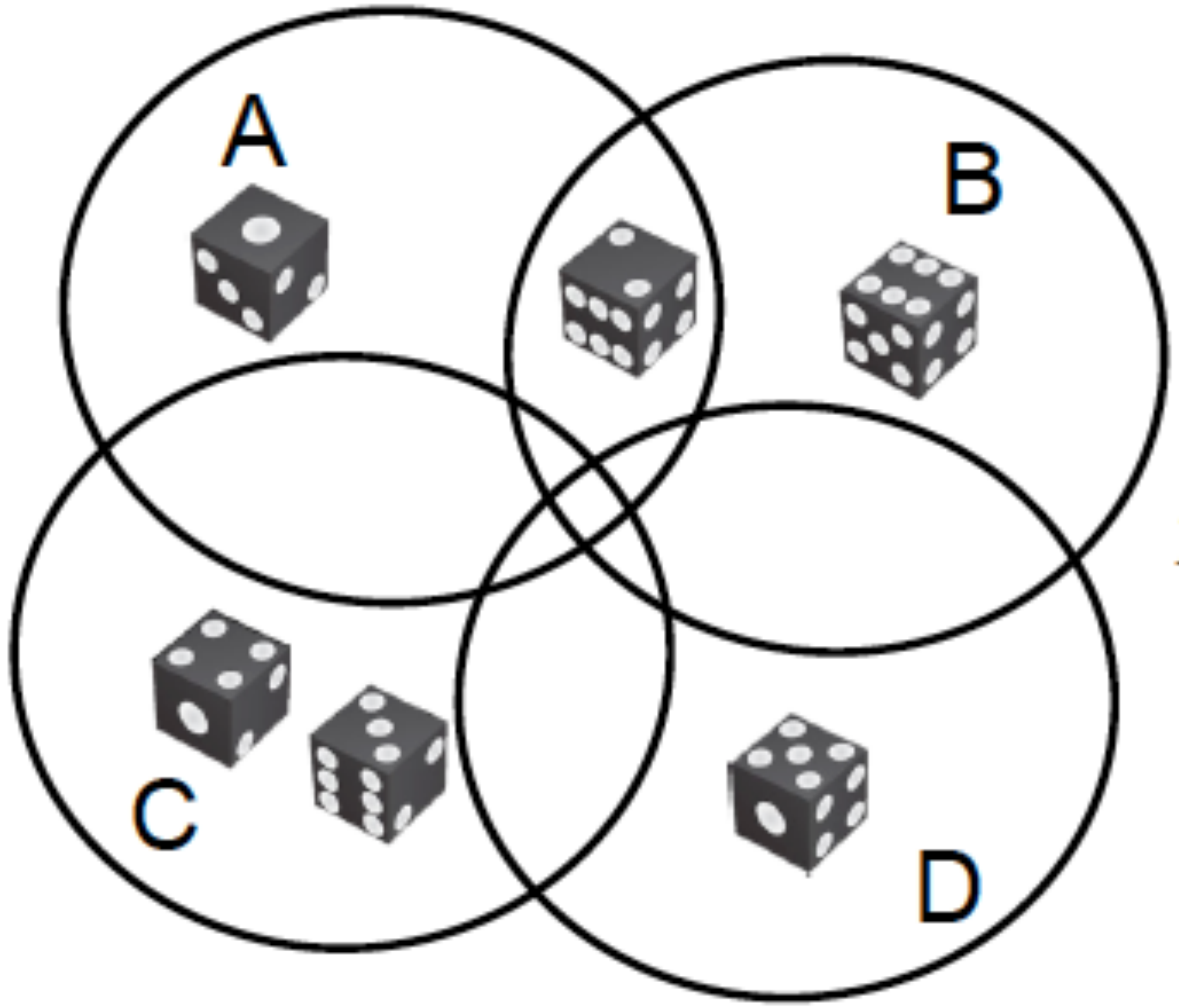
= 1/2, if fair die

Introduction to probability: union bounds

Optional subtitle

- P(A or B or C or D or …)

  $\leq$ P(A) + P(B) + P(C) + P(D) + …



$$p(A \cup B) = p(A) + p(B) - p(A \cap B)$$
$$\leq p(A) + p(B)$$

Q: When is this a tight bound?

A: For disjoint events

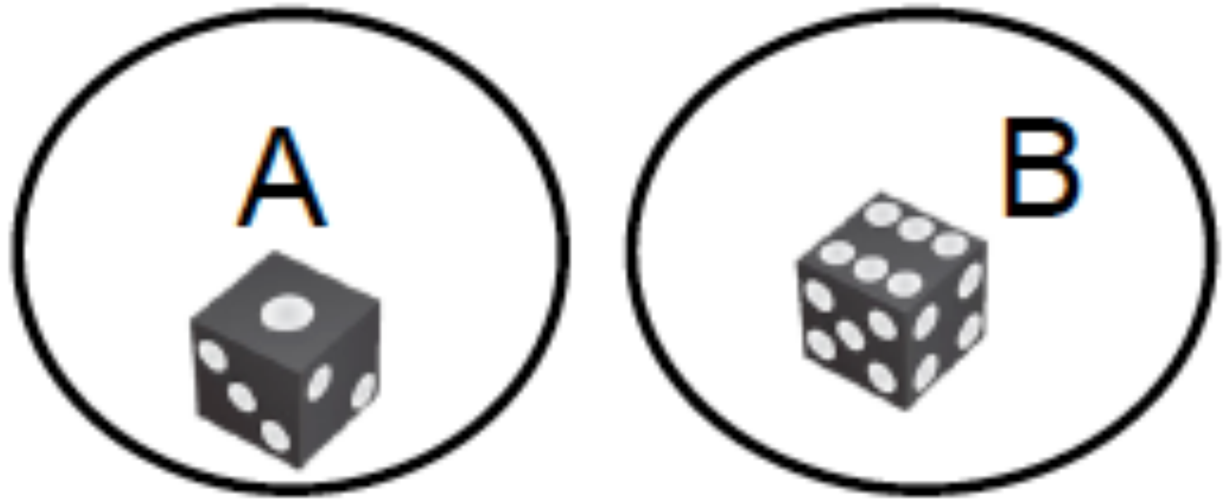
(i.e., non-overlapping circles)

Introduction to probability: independence

Optional subtitle

- Two events A and B are **independent** if

$$p(A \cap B) = p(A)p(B)$$



Are these events independent?

No! $p(A \cap B) = 0$

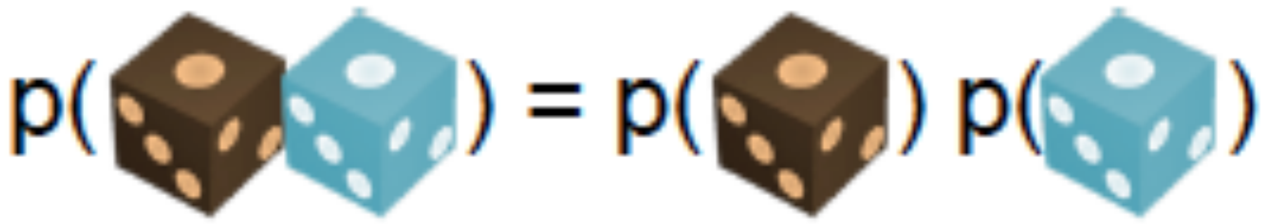
$$p(A)p(B) = \left(\frac{1}{6}\right)^2$$

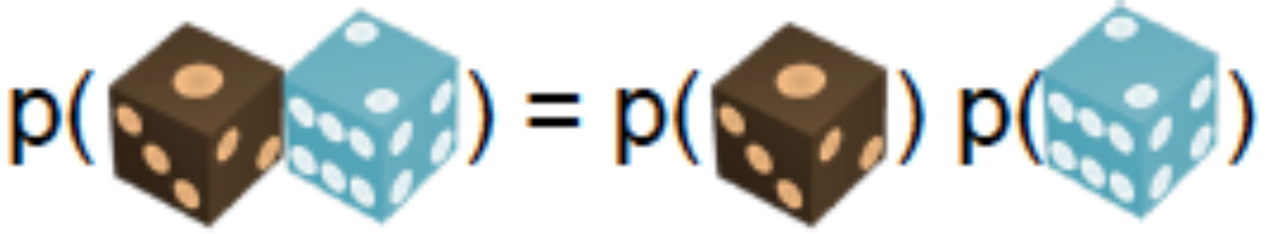
- Suppose our outcome space had two different die:

2 die tosses

$6^2 = 36$ outcomes

and each die is (defined to be) independent, i.e.



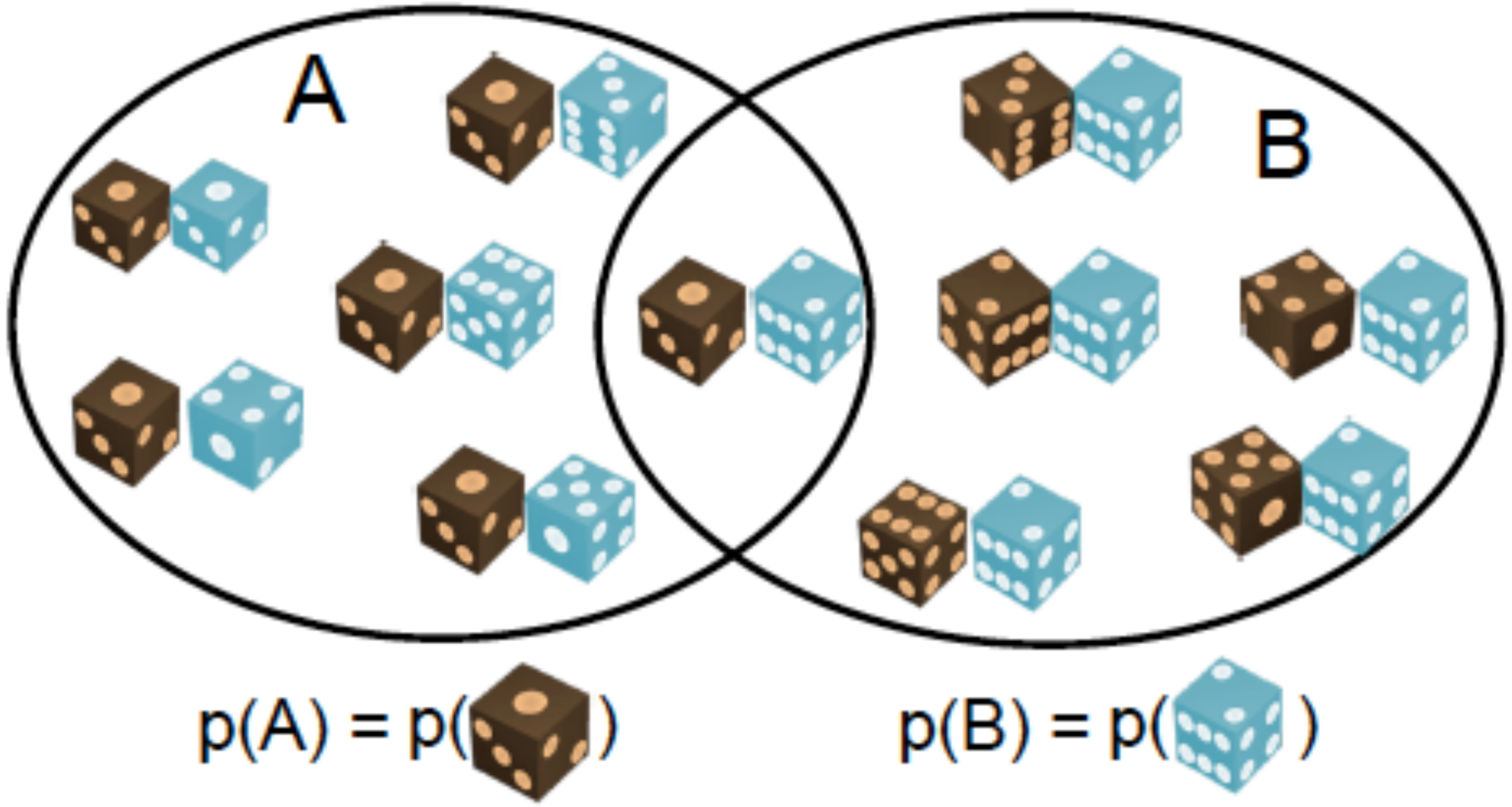


Introduction to probability: independence

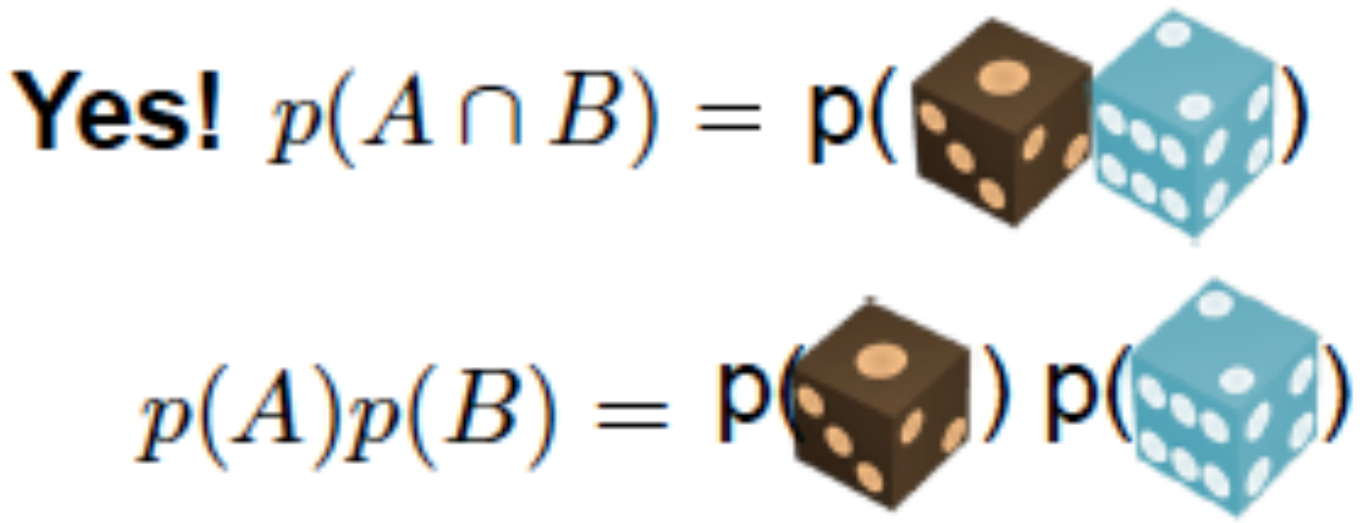
Optional subtitle

- Two events A and B are **independent** if

$$p(A \cap B) = p(A)p(B)$$

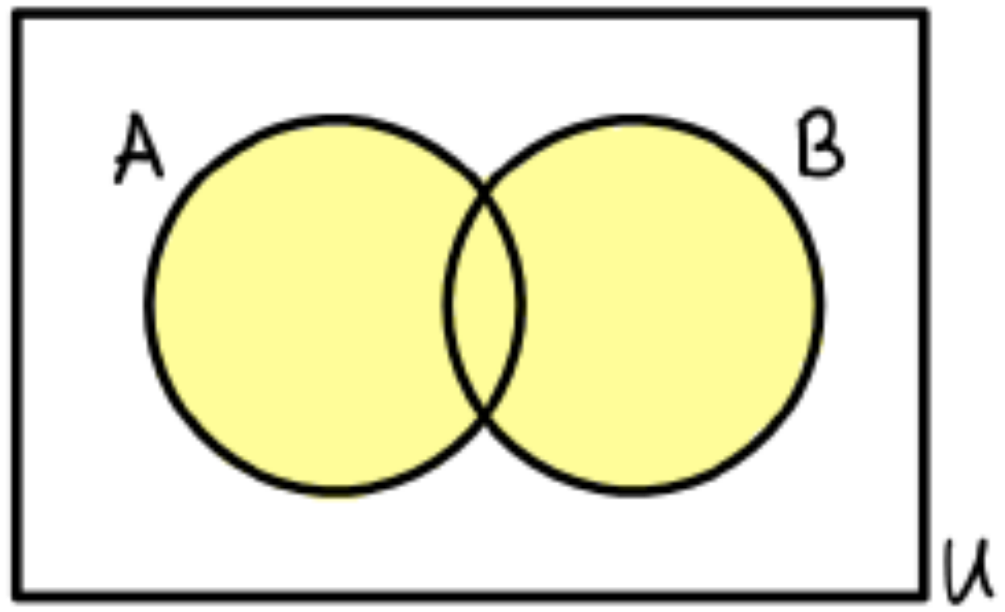


Are these events independent?



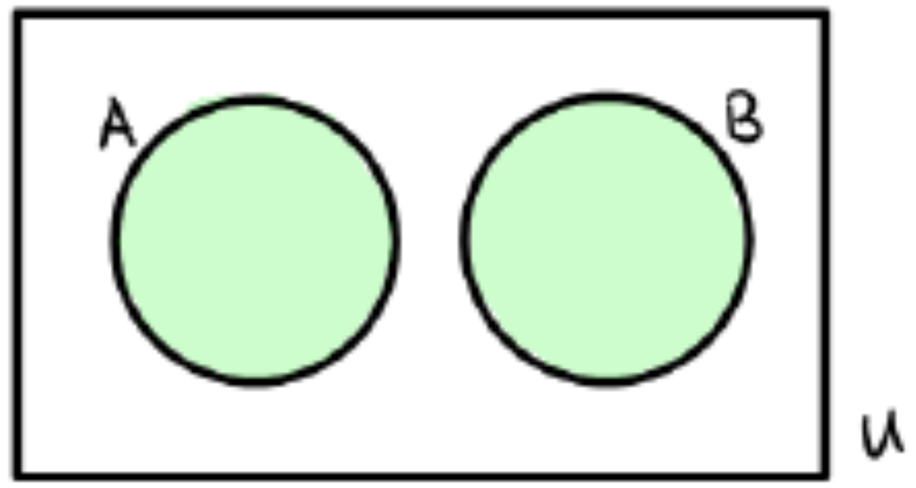
Introduction to probability

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$



Mutually Exclusive

$$P(A \cap B) = 0$$
$$P(A \cup B) = P(A) + P(B)$$

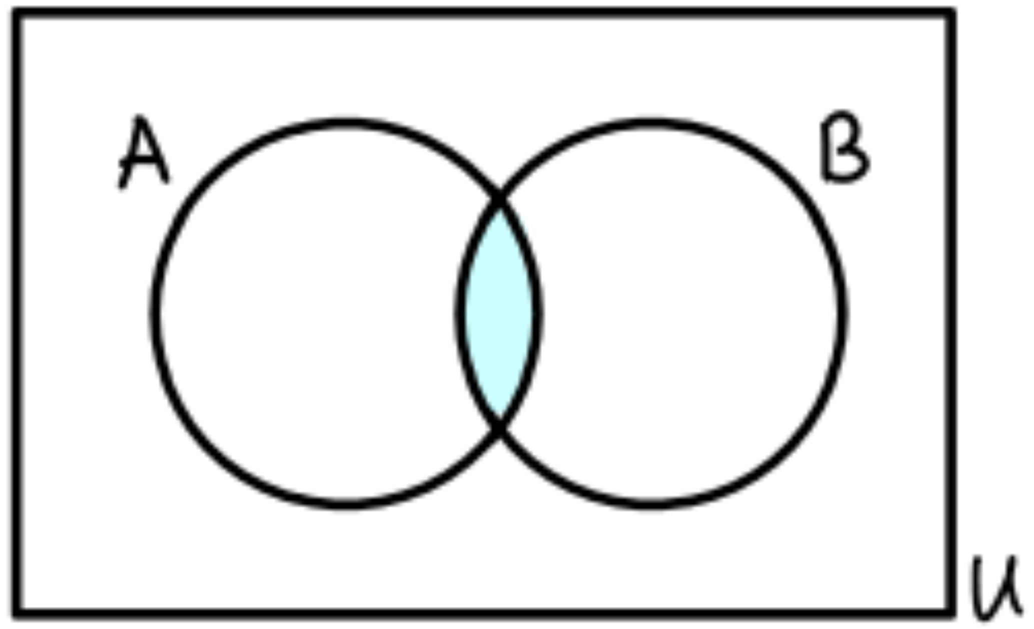


U = outcome space

A,B events

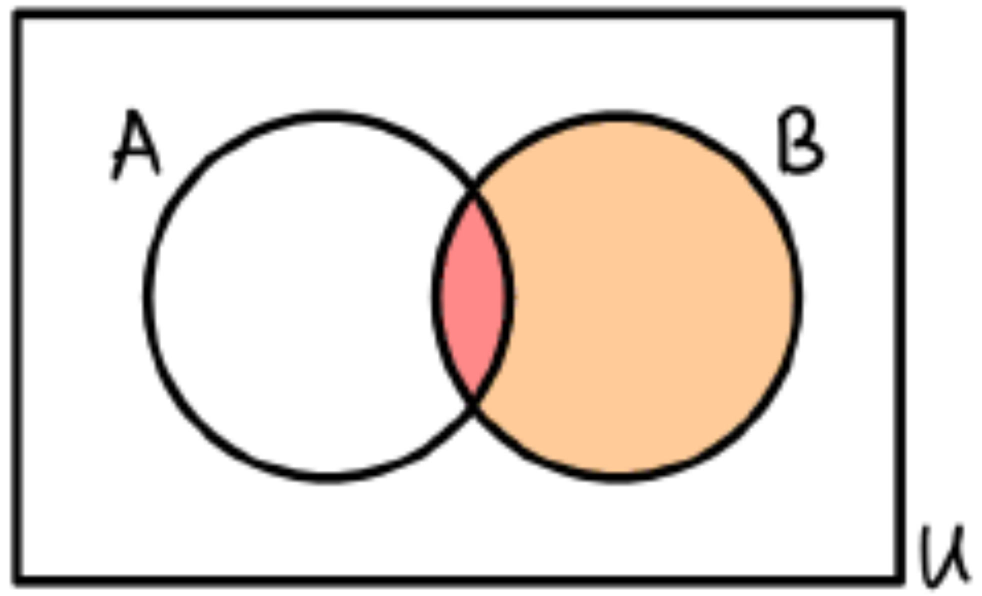
Independence

$$P(A \cap B) = P(A)P(B)$$



Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$



[Figures from http://ibscrewed4maths.blogspot.com/]

Introduction to probability

Optional subtitle

Notation: Val(X) = set D of all values assumed by variable X

p(X) specifies a distribution: $p(X = x) \geq 0 \ \ \forall x \in \mathrm{Val}(X)$

$$\sum_{x \in \mathrm{Val}(X)} p(X = x) = 1$$

X=x is simply an event, so can apply union bound, conditioning, etc.

Two random variables **X** and **Y** are **independent** if:

$$p(X = x, Y = y) = p(X = x)p(Y = y) \quad \forall x \in \mathrm{Val}(X), y \in \mathrm{Val}(Y)$$

The **expectation** of **X** is defined as: $E[X] = \sum_{x \in \mathrm{Val}(X)} p(X = x)x$

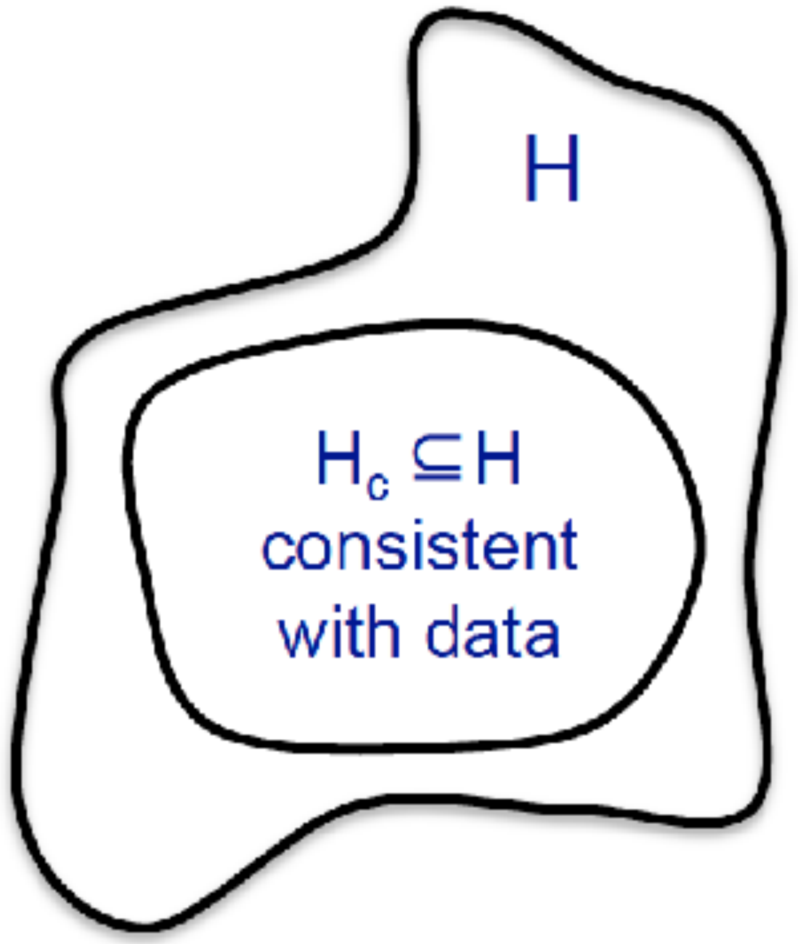
For example, $E[Z_i^h] = \sum_{z \in \{0,1\}} p(Z_i^h = z)z \ = p(Z_i^h = 1)$

Chap 8

PAC bound

A simple setting…

Optional subtitle



- Classification
  - *m* data points
  - Finite number of possible hypothesis (e.g. 20000 face recognition classifiers)
- A learner finds a hypothesis h that is consistent with training data
  - Gets zero error in training: $error_{train}(h)=0$
  - i.e. assume for now that the winner gets 100% accuracy on the *m* labelled images (we'll handle 98% case afterward)
- What is the probability that *h* has more than ε true error?
  - $error_{true}(h) \geq \varepsilon$

How likely is a *bad* hypothesis to get *m* data points right?

- Hypothesis *h* that is **consistent** with training data
  - got *m* i.i.d. points right
  - h "bad" if it gets all this data right, but has high true error
  - What is the probability of this happening?

- Probability that *h* with $error_{true}(h) \geq \varepsilon$ classifies a randomly drawn data point correctly:

  1. Pr(h gets data point *wrong* | $error_{true}(h) = \varepsilon) = \varepsilon$
  2. Pr(h gets data point *wrong* | $error_{true}(h) \geq \varepsilon) \geq \varepsilon$
  3. Pr(h gets data point *right* | $error_{true}(h) \geq \varepsilon$) = 1 - Pr(h gets data point *wrong* | $error_{true}(h) \geq \varepsilon$)

     $\leq 1 - \varepsilon$

- Probability that *h* with $error_{true}(h) \geq \varepsilon$ gets *m* iid data points correct:

$$\Pr(\text{h gets m } \textit{iid} \text{ data points right} \mid error_{true}(h) \geq \varepsilon) \leq (1-\varepsilon)^m \leq e^{-\varepsilon m}$$

Are we done?

Optional subtitle

$$\Pr(\text{h gets m } iid \text{ data points right} \mid \text{error}_{true}(h) \geq \varepsilon) \leq e^{-\varepsilon m}$$

- Says "if h gets m data points correct, then with very high probability (i.e. $1\text{-}e^{-\varepsilon m}$) it is close to perfect (i.e., will have error $\leq \varepsilon$)"
- This only considers **one** hypothesis!
- Suppose 1 billion people entered the competition, and each person submits a *random* function
- For **m** small enough, one of the functions will classify all points correctly – but all have very large true error

How likely is learner to pick a bad hypothesis?

Optional subtitle

$\boxed{\Pr(\text{h gets m } \textit{iid} \text{ data points right} \mid \text{error}_{true}(h) \geq \varepsilon) \leq e^{-\varepsilon m}}$

Suppose there are $|H_c|$ hypotheses consistent with the training data

- How likely is learner to pick a bad one, i.e. with *true* error $\geq \varepsilon$?
- We need to a bound that holds for all of them!

$$P(\text{error}_{true}(h_1) \geq \varepsilon \text{ OR } \text{error}_{true}(h_2) \geq \varepsilon \text{ OR } \dots \text{ OR } \text{error}_{true}(h_{|Hc|}) \geq \varepsilon)$$

$\leq \sum_k P(\text{error}_{true}(h_k) \geq \varepsilon)$ ← Union bound

$\leq \sum_k (1-\varepsilon)^m$ ← bound on individual h_js

$\leq |H|(1-\varepsilon)^m$ ← $|H_c| \leq |H|$

$\leq |H|\, e^{-m\varepsilon}$ ← $(1-\varepsilon) \leq e^{-\varepsilon}$ for $0 \leq \varepsilon \leq 1$

Generalisation error of finite hypothesis spaces [Haussler '88]

Optional subtitle

We just proved the following result:

Theorem: Hypothesis space *H* finite, dataset *D* with *m* i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis *h* that is consistent on the training data:

$$P(\text{error}_{true}(h) > \epsilon) \leq |H|e^{-m\epsilon}$$

Using a PAC bound

Typically, 2 use cases:

- 1: Pick ε and δ, compute *m*
- 2: Pick m and δ, compute ε

Argument: Since for all *h* we know that

$$P(\text{error}_{true}(h) > \epsilon) \leq |H|e^{-m\epsilon}$$

... with probability 1-δ the following holds... (either case 1 or case 2)

$$p(\text{error}_{true}(h) \geq \epsilon) \leq |H|e^{-m\epsilon} \leq \delta$$

Says: we are willing to tolerate a δ probability of having ≥ ε error

$$\ln\left(|H|e^{-m\epsilon}\right) \leq \ln\delta$$

$$\ln|H| - m\epsilon \leq \ln\delta$$

Case 1

$$m \geq \frac{\ln|H| + \ln\frac{1}{\delta}}{\epsilon}$$

Case 2

$$\epsilon \geq \frac{\ln|H| + \ln\frac{1}{\delta}}{m}$$

Log dependence on |H|,

ε has stronger influence than δ

ε shrinks at rate O(1/m)

Limitations of Haussler '88 bound

Optional subtitle

- There may be no consistent hypothesis h (where $error_{train}(h)=0$)
- Size of hypothesis space
  - What if |H| is really big?
  - What if it is continuous?

- First Goal: Can we get a bound for a learner with $error_{train}(h)$ in training set?

Question: what's the expected error of a hypothesis?

- The probability of a hypothesis incorrectly classifying: $\sum_{(\vec{x},y)} \hat{p}(\vec{x},y)1[h(\vec{x}) \neq y]$

- We showed that the Z_i^h random variables are **independent** and **identically distributed** (i.i.d.) with $\Pr(Z_i^h = 0) = \sum_{(\vec{x},y)} \hat{p}(\vec{x},y)1[h(\vec{x}) \neq y]$

- Estimating the true error probability is like estimating the parameter of a coin!

- **Chernoff bound**: for *m* i.i.d. coin flips, $X_1,\ldots,X_m$, where $X_i \in \{0,1\}$. For $0<\varepsilon<1$:

$$p(X_i = 1) = \theta$$

$$\boxed{P\left(\theta - \frac{1}{m}\sum_i x_i > \epsilon\right) \leq e^{-2m\epsilon^2}}$$

True error probability

Observed fraction of points incorrectly classified

$$E[\frac{1}{m}\sum_{i=1}^{m} X_i] = \frac{1}{m}\sum_{i=1}^{m} E[X_i] = \theta$$

(by linearity of expectation)

Generalisation bound for |*H*| hypothesis

Optional subtitle

Theorem: Hypothesis space *H* finite, dataset *D* with *m* i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis *h*:

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H|e^{-2m\epsilon^2}$$

Why? Same reasoning as before. Use the Union bound over individual Chernoff bounds

Optional subtitle

PAC bound and Bias-Variance tradeoff

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H|e^{-2m\epsilon^2}$$

or, after moving some terms around, with probability at least 1-δ:

$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$$

- **Important: PAC bound holds for all *h*, but doesn't guarantee that algorithm finds best *h*!!!**

PAC bound and Bias-Variance tradeoff

Optional subtitle

for all h, with probability at least 1-δ:

$$\text{error}_{true}(h) \leq \underbrace{\text{error}_{train}(h)}_{\text{“bias”}} + \underbrace{\sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2m}}}_{\text{“variance”}}$$

- For large |H|
  - low bias (assuming we can find a good h)
  - high variance (because bound is looser)
- For small |H|
  - high bias (is there a good h?)
  - low variance (tighter bound)

PAC bound: How much data?

Optional subtitle

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H| e^{-2m\epsilon^2}$$

$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$$

- Given δ,ε how big should m be?

$$\boxed{m \geq \frac{1}{2\epsilon^2}\left(\ln|H| + \ln\frac{1}{\delta}\right)}$$

Returning to our example…

A fictional example

Suppose Alibaba holds a competition for the best face recognition classifier (+1 if image contains a face, -1, otherwise)

Lots of teams compete …

Alibaba get back 20,000 recognition algorithm

They evaluate all 20,000 algorithm on m labelled images which is not previously shown to the competitors) and chooses a winner.

The winner obtains 98% accuracy on m labelled images!

Alibaba has a face recognition algorithm that is known to be 95% accurate,

- Should they deploy the winner's algorithm instead?
- Can't risk doing worse … would be a disaster for Alibaba.

Returning to our example…

Optional subtitle

$$\text{error}_{true}(\text{Alibaba}\quad) = .05$$

|H|=20,000 competitors

$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$$

=.02 error on the m images

m = 100 images

Suppose δ=0.01 and m=100: $.02 + \sqrt{\frac{\ln(20,000)+\ln(100)}{200}} \approx .29$

Suppose δ=0.01 and m=10,000: $.02 + \sqrt{\frac{\ln(20,000)+\ln(100)}{20,000}} \approx .047$

So, with only ~100 test images, confidence interval too large! Do not deploy!

But, if the competitor's error is still .02 on m>10,000 images, then we can say that it is truly better with probability at least 99/100

Appendix

VC dimension

What about continuous hypothesis spaces?

Optional subtitle

$$\text{error}_{true}(h) \le \text{error}_{train}(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$$

- Continuous hypothesis space:
  - $|H| = \infty$
  - Infinite variance???

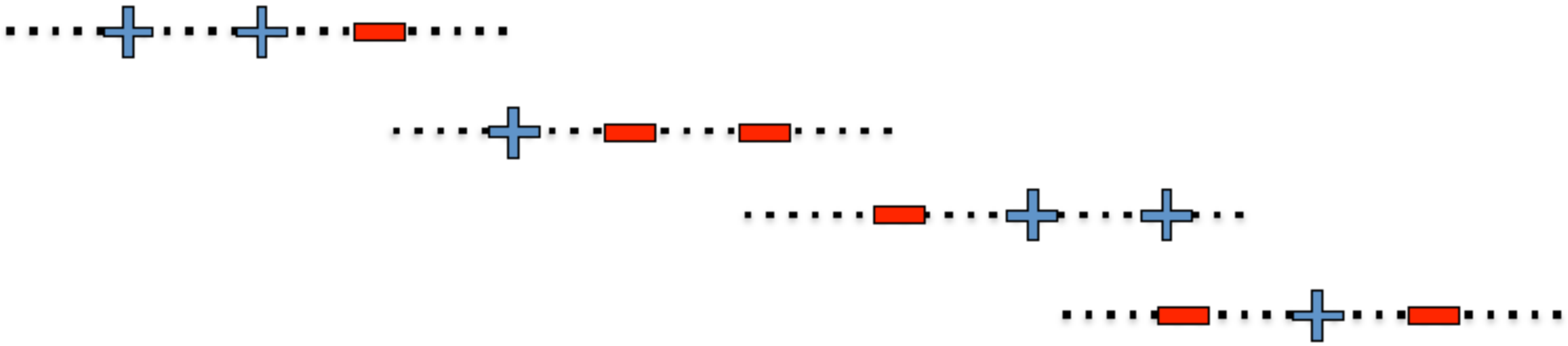
- **Only care about the maximum number of points that can be classified exactly!**

How many points can a linear boundary classify exactly? (1-D)

Optional subtitle

2 Points: Yes!!

3 Points: No…



etc (8 total)

Shattering and Vapnik-Chervonenkis Dimension

Optional subtitle

A **set of points** is *shattered* by a hypothesis space H iff:

- For all ways of *splitting* the examples into positive and negative subsets
- There exists some *consistent* hypothesis h

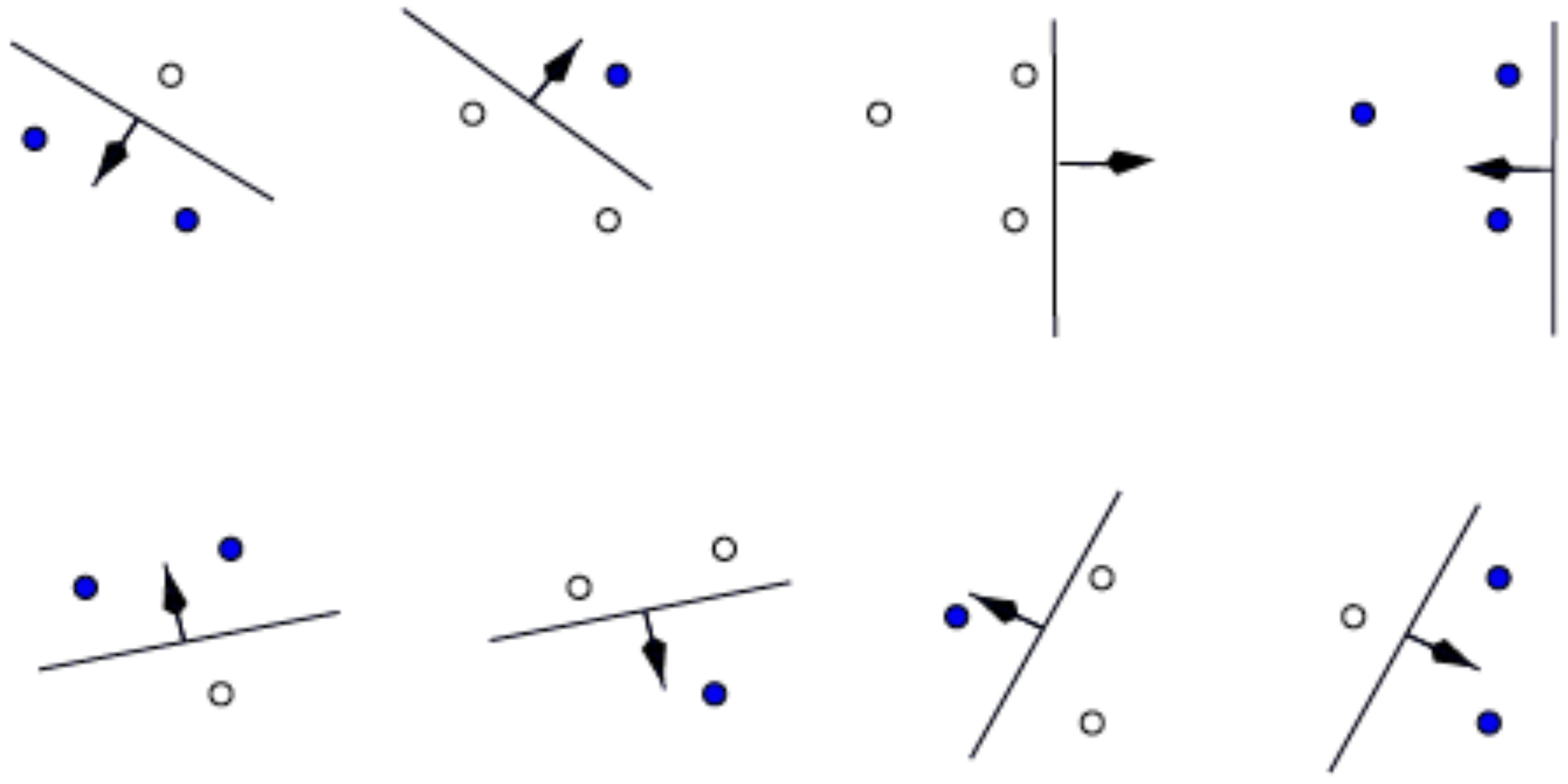
The *VC Dimension* of H over input space X

- The size of the *largest* finite subset of X shattered by H

How many points can a linear boundary classify exactly? (3-D)

Optional subtitle

3 Points: Yes!!



4 Points: No…

etc.

[Figure from Chris Burges]

How many points can a linear boundary classify exactly? (d-D)

Optional subtitle

- A linear classifier $w_0+\sum_{j=1..d} w_j x_j$ can represent all assignments of possible labels to d+1 points
  - But not d+2!!
  - Thus, VC-dimension of d-dimensional linear classifiers is d+1
  - Bias term w_0 required
  - Rule of Thumb: number of parameters in model often matches max number of points

- Question: Can we get a bound for error in as a function of the number of points that can be completely labeled?

PAC bound using VC dimension

Optional subtitle

- VC dimension: number of training points that can be classified exactly (shattered) by hypothesis space H!!!
  - Measures relevant size of hypothesis space

$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

- Same bias / variance tradeoff as always
  - Now, just a function of VC(H)

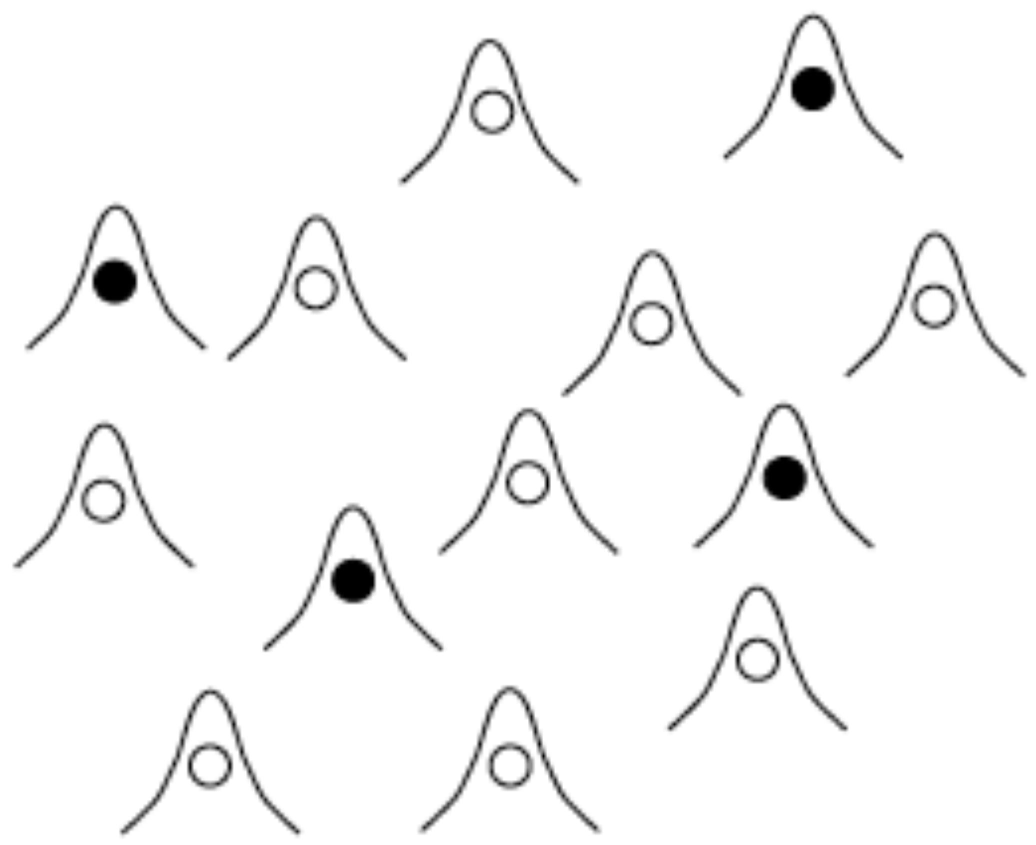
- Note: all of this theory is for **binary** classification
  - Can be generalized to multi-class and also regression

Example of VC dimension

Optional subtitle

$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

- Linear classifiers:
  - VC(H) = d+1, for *d* features plus constant term *b*
- SVM with Gaussian Kernel
  - VC(H) = ∞



[Figure from Chris Burges]

What you need to know

Optional subtitle

- Finite hypothesis space
  - Derive results
  - Counting number of hypothesis
  - Mistakes on Training data
- Complexity of the classifier depends on number of points that can be classified exactly
  - Finite case – number of hypotheses considered
  - Infinite case – VC dimension
- Bias-Variance tradeoff in learning theory