Learning theory
Lecture 8

David Sontag
New York University

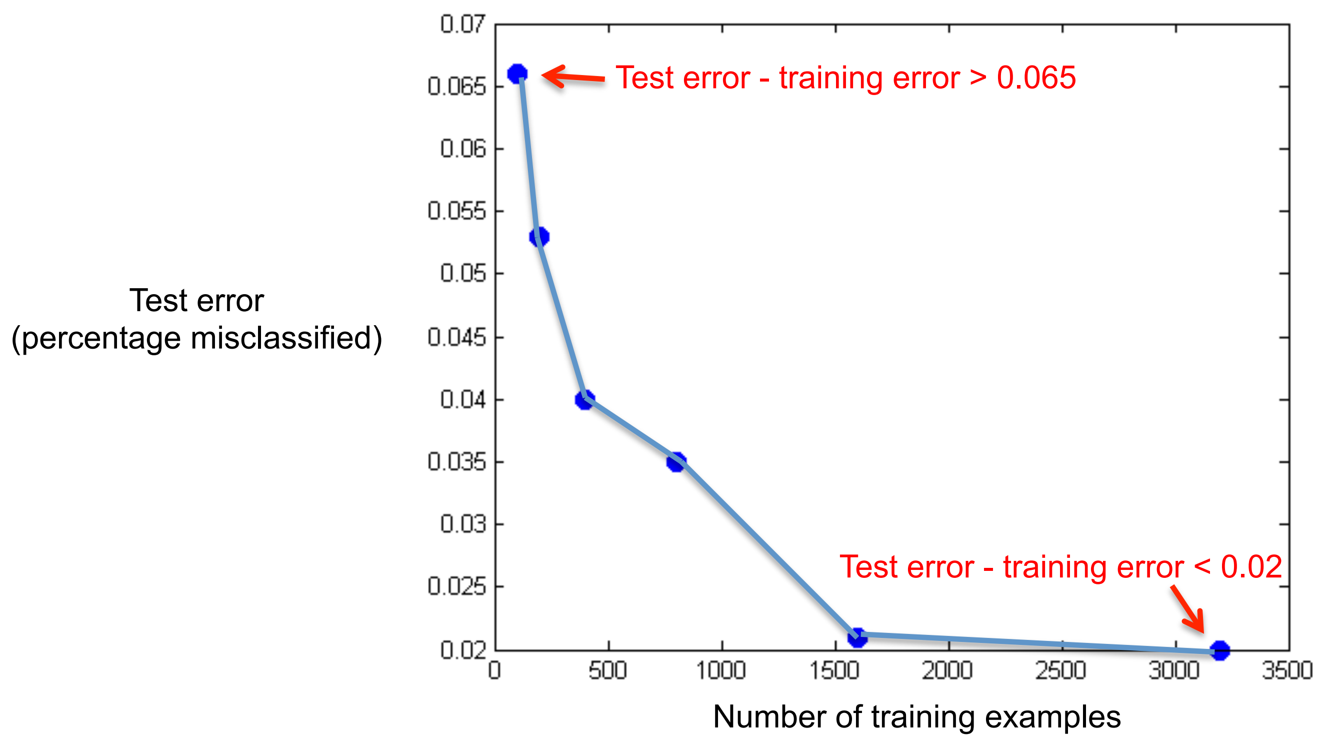
Slides adapted from Carlos Guestrin & Luke Zettlemoyer

What's next…

- We gave several machine learning algorithms:
  - Perceptron
  - Linear support vector machine (SVM)
  - 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

- In your homework, you trained a spam classifier using perceptron
  - The training error was always zero
  - With few data points, there was a big gap between training error and test error!



How much training data do you need?

- Depends on what *hypothesis class* the learning algorithm considers

- For example, consider a memorization-based learning algorithm
  - Input: training data S = { ($\mathbf{x}_i$, y_i) }
  - Output: function f($\mathbf{x}$) which, if there exists ($\mathbf{x}_i$, y_i) in S such that $\mathbf{x}=\mathbf{x}_i$, 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 generalization performance is horrible)

- Linear classifiers are powerful precisely because of their simplicity
  - Generalization is easy to guarantee

Roadmap of next two lectures

1. Generalization of finite hypothesis spaces

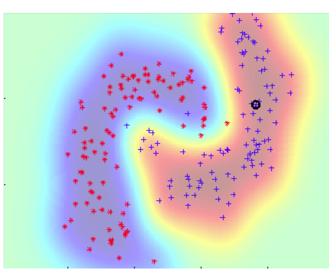
2. VC-dimension

   - Will show that linear classifiers need to see approximately **d** training points, where **d** is the dimension of the feature vectors

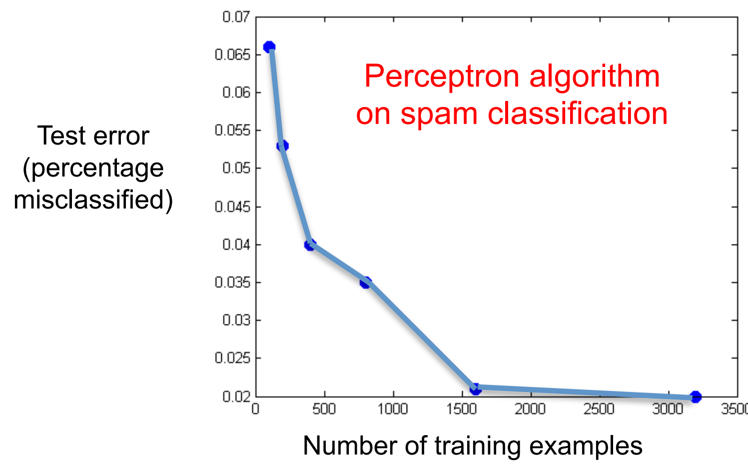
   - Explains the good performance we obtained using perceptron!!!! (we had 1899 features)

3. Margin based generalization

   - Applies to **infinite** dimensional feature vectors (e.g., Gaussian kernel)



[Figure from Cynthia Rudin]

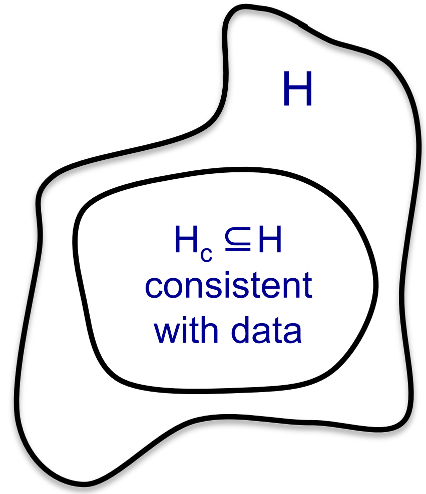


Choosing among several classifiers

- Suppose Facebook holds a competition for the best face recognition classifier (+1 if image contains a face, -1 if it doesn't)
- All recent worldwide graduates of machine learning and computer vision classes decide to compete
- Facebook gets back **|H|=** 20,000 face recognition algorithms
- They evaluate all 20,000 algorithms on **m** labeled images (not previously shown to the competitors) and chooses a winner
- The winner obtains 98% accuracy on these **m** images!!!
- Facebook already 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 public relations disaster!

[Fictional example]

A simple setting…



- Classification
  - m data points
  - **Finite** number of possible hypothesis (e.g., 20,000 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** labeled images (we'll handle the 98% case afterward)
- What is the probability that *h* has more than ε **true** error?
  - $error_{true}(h) \geq \varepsilon$

Introduction to probability: outcomes

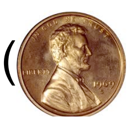
- An **outcome space** specifies the possible outcomes that we would like to reason about, e.g.

Ω = {  ,  } Coin toss

Ω = { , , , , ,  } 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

- 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,

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

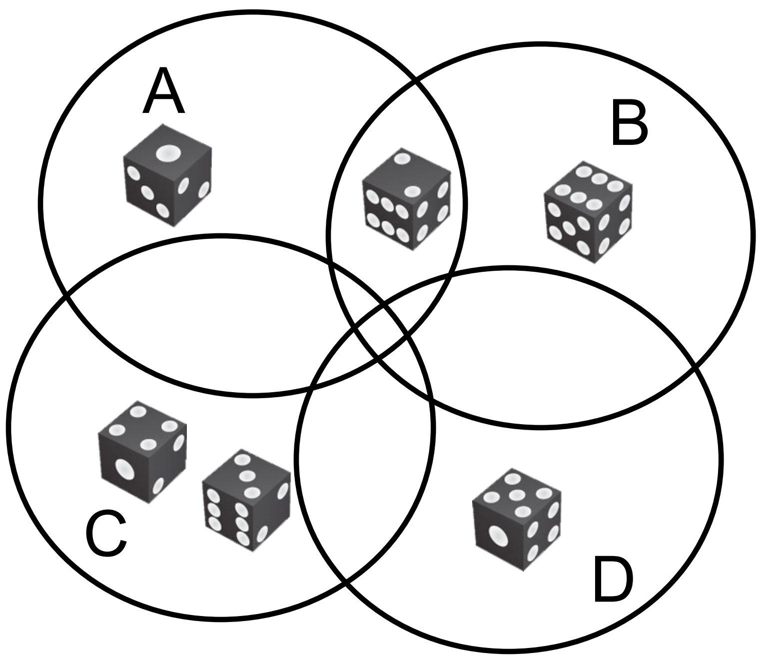
E.g., p(E) = p() + p() + p()

= 1/2, if fair die

Introduction to probability: union bound

- P(A or B or C or D or …)

  ≤ P(A) + P(B) + P(C) + P(D) + …



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

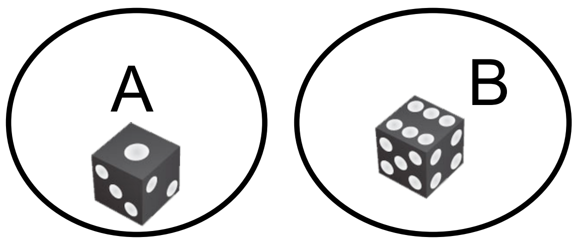
Q: When is this a tight bound?

A: For disjoint events
(i.e., non-overlapping circles)

Introduction to probability: independence

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

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

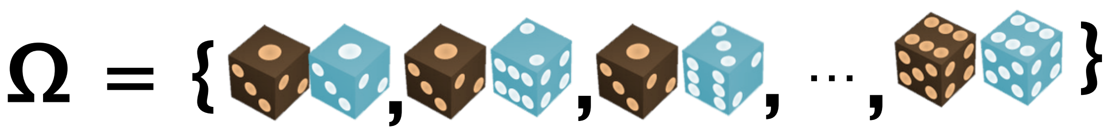


Are these events independent?

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

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

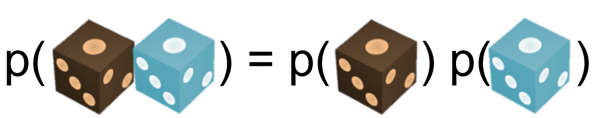
- Suppose our outcome space had two different die:

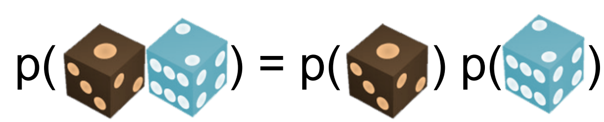


2 die tosses

6^2 = 36 outcomes

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

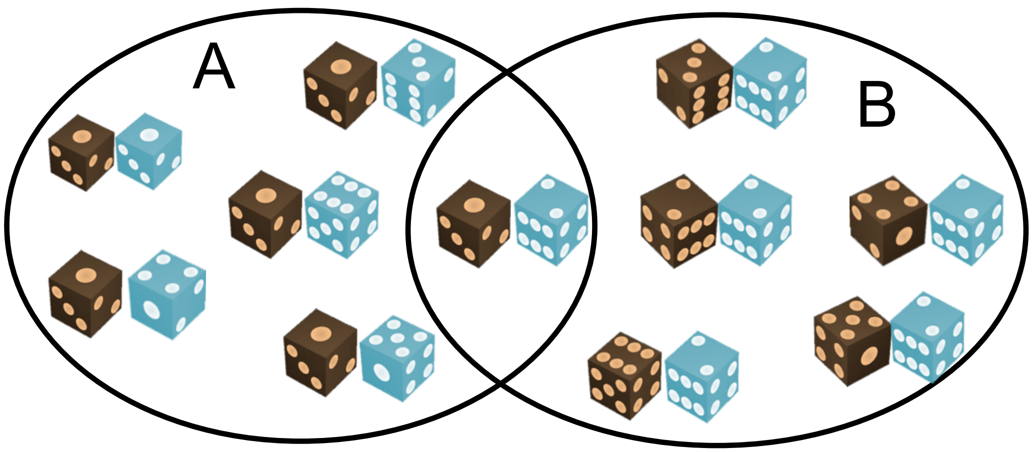


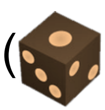


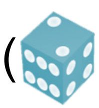
Introduction to probability: independence

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

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

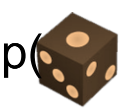
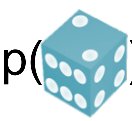


p(A) = p()

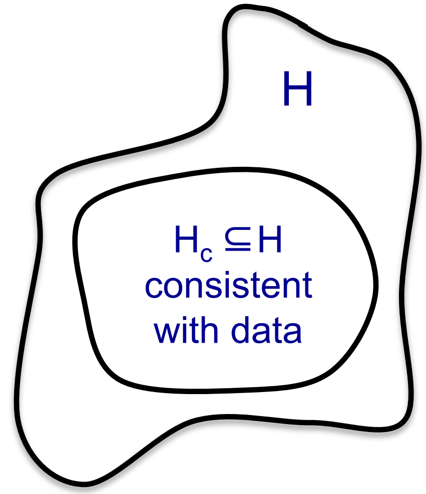
p(B) = p()

Are these events independent?

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

$p(A)p(B) =$ p() p()

A simple setting…



- Classification
  - m data points
  - **Finite** number of possible hypothesis (e.g., 20,000 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** labeled images (we'll handle the 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* | $error_{true}(h) = \varepsilon$) = ε E.g., probability of a biased coin coming up tails

  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:

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

E.g., probability of m biased coins coming up heads

Are we done?

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

- Says "if h gets m data points correct, then with very high probability (i.e. $1-e^{-\varepsilon m}$) 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?

Pr(h gets m *iid* data points right | $error_{true}(h) \geq \varepsilon$) $\leq e^{-\varepsilon m}$

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!

$$P(error_{true}(h_1) \geq \varepsilon \text{ OR } error_{true}(h_2) \geq \varepsilon \text{ OR } \ldots \text{ OR } error_{true}(h_{|H_c|}) \geq \varepsilon)$$

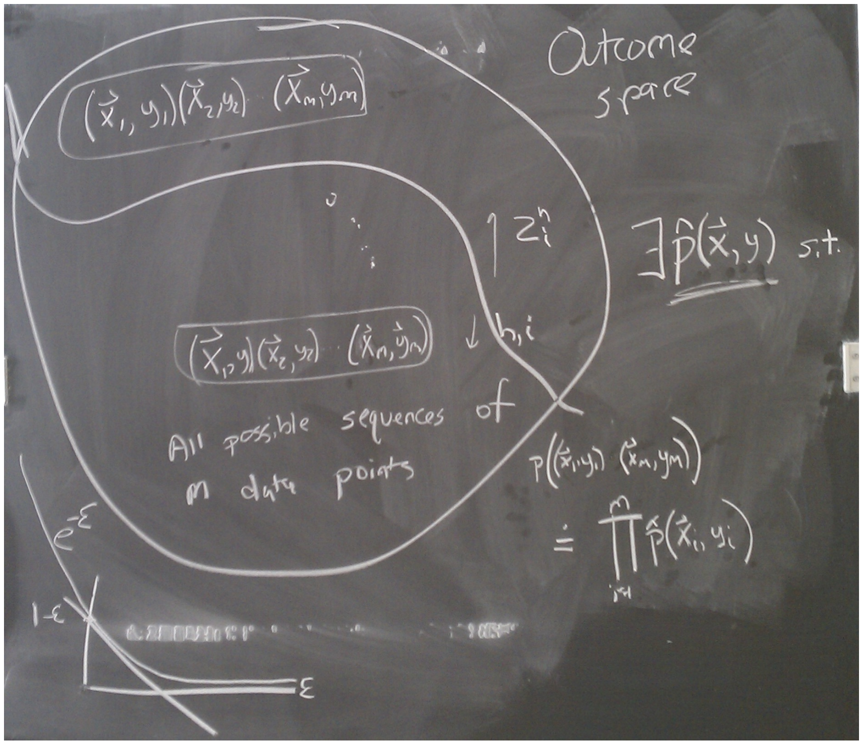
$\leq \sum_k P(error_{true}(h_k) \geq \varepsilon)$ ← Union bound

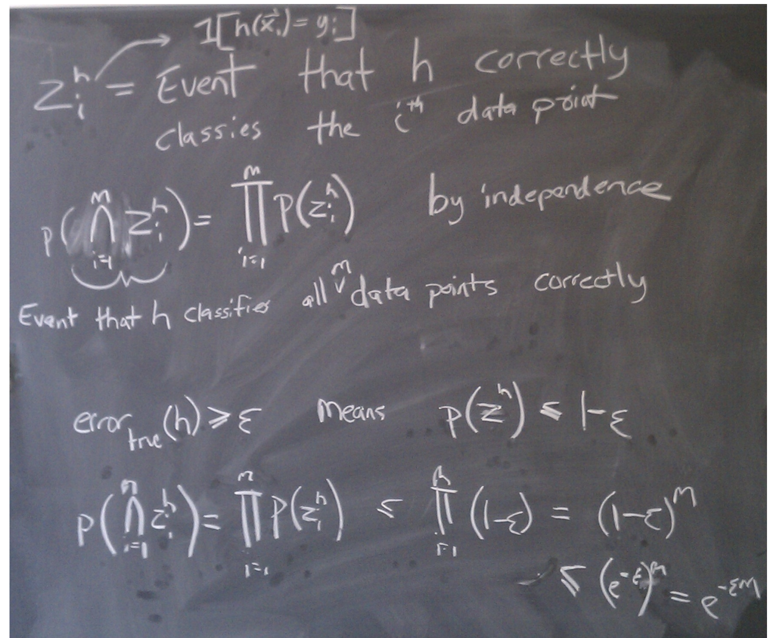
$\leq \sum_k (1-\varepsilon)^m$ ← bound on individual h_js

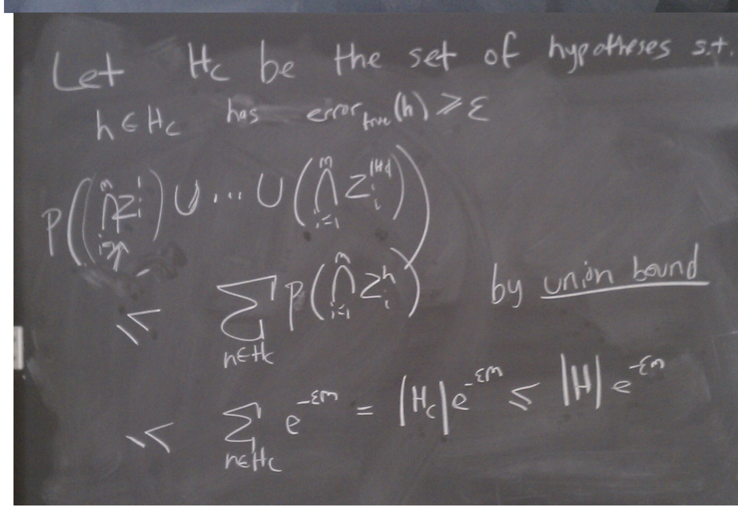
$\leq |H|(1-\varepsilon)^m$ ← $|H_c| \leq |H|$

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

Analysis done on blackboard







Generalization error of finite hypothesis spaces [Haussler '88]

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|, OK if exponential size (but not doubly)

ε has stronger influence than δ

ε shrinks at rate O(1/m)