CS260: Machine Learning Algorithms

Lecture 6: Nonlinear mapping, Model complexity

Cho-Jui Hsieh
UCLA

Jan 28, 2019

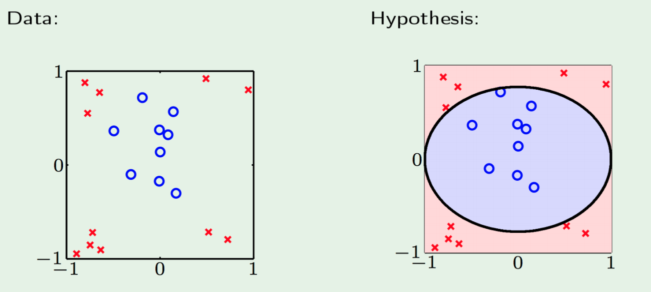
Nonlinear Transformation

Linear Hypotheses

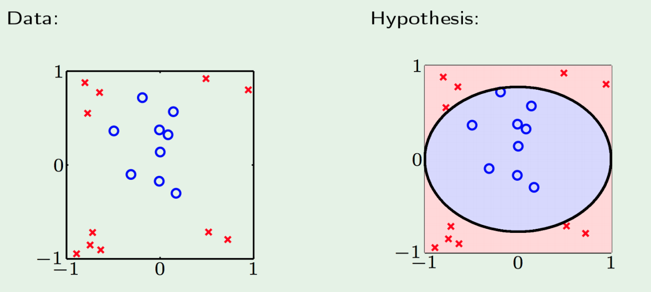
- Up to now: linear hypotheses

  Perceptron, Linear regression, Logistic regression, $\cdots$

- Many problems are not linearly separable

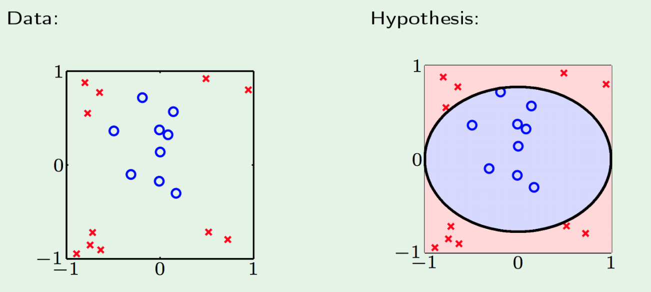


Circular Separable



- Data is not linear separable

Circular Separable



- Data is not linear separable
- but circular separable by a circle of radius $\sqrt{0.6}$ centered at origin:

$$h_{\text{SEP}}(\boldsymbol{x}) = \text{sign}(-x_1^2 - x_2^2 + 0.6)$$

Circular Separable and Linear Separable

$$h(\boldsymbol{x}) = \text{sign}(0.6 \cdot 1 + (-1) \cdot x_1^2 + (-1) \cdot x_2^2)$$

Circular Separable and Linear Separable

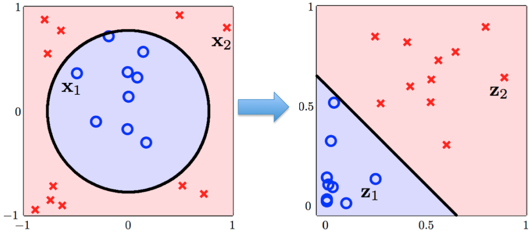
$$h(\boldsymbol{x}) = \text{sign}(\underbrace{0.6}_{\tilde{w}_0} \cdot \underbrace{1}_{z_0} + \underbrace{(-1)}_{\tilde{w}_1} \cdot \underbrace{x_1^2}_{z_1} + \underbrace{(-1)}_{\tilde{w}_2} \cdot \underbrace{x_2^2}_{z_2})$$

$$= \text{sign}(\tilde{\boldsymbol{w}}^T \boldsymbol{z})$$

Circular Separable and Linear Separable

$$h(\mathbf{x}) = \text{sign}(\underbrace{0.6}_{\tilde{w}_0} \cdot \underbrace{1}_{z_0} + \underbrace{(-1)}_{\tilde{w}_1} \cdot \underbrace{x_1^2}_{z_1} + \underbrace{(-1)}_{\tilde{w}_2} \cdot \underbrace{x_2^2}_{z_2})$$

$$= \text{sign}(\tilde{\mathbf{w}}^T \mathbf{z})$$

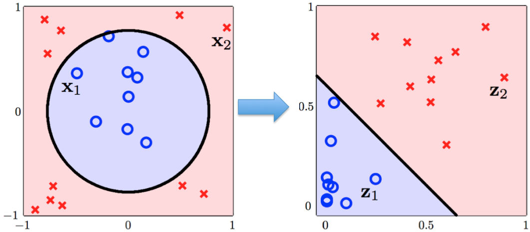


- $\{(\mathbf{x}_n, y_n)\}$ circular separable $\Rightarrow$ $\{(\mathbf{z}_n, y_n)\}$ linear separable

Circular Separable and Linear Separable

$$h(\mathbf{x}) = \text{sign}(\underbrace{0.6}_{\tilde{w}_0} \cdot \underbrace{1}_{z_0} + \underbrace{(-1)}_{\tilde{w}_1} \cdot \underbrace{x_1^2}_{z_1} + \underbrace{(-1)}_{\tilde{w}_2} \cdot \underbrace{x_2^2}_{z_2})$$

$$= \text{sign}(\tilde{\mathbf{w}}^T \mathbf{z})$$



- $\{(\mathbf{x}_n, y_n)\}$ circular separable $\Rightarrow$ $\{(\mathbf{z}_n, y_n)\}$ linear separable
- $\mathbf{x} \in \mathcal{X} \to \mathbf{z} \in \mathcal{Z}$ (using a nonlinear transformation Φ)

Nonlinear Transformation

- Define nonlinear transformation

$$\Phi(\boldsymbol{x}) = (1, x_1^2, x_2^2) = (z_0, z_1, z_2) = \boldsymbol{z}$$

Nonlinear Transformation

- Define nonlinear transformation

$$\Phi(\boldsymbol{x}) = (1, x_1^2, x_2^2) = (z_0, z_1, z_2) = \boldsymbol{z}$$

- Linear hypotheses in $\mathcal{Z}$ space:

$$\text{sign}(\tilde{h}(\boldsymbol{z})) = \text{sign}(\tilde{h}(\Phi(\boldsymbol{x}))) = \text{sign}(\boldsymbol{w}^T \Phi(\boldsymbol{x}))$$

Nonlinear Transformation

- Define nonlinear transformation

$$\Phi(\boldsymbol{x}) = (1, x_1^2, x_2^2) = (z_0, z_1, z_2) = \boldsymbol{z}$$

- Linear hypotheses in $\mathcal{Z}$ space:

$$\text{sign}(\tilde{h}(\boldsymbol{z})) = \text{sign}(\tilde{h}(\Phi(\boldsymbol{x}))) = \text{sign}(\boldsymbol{w}^T \Phi(\boldsymbol{x}))$$

- Lines in $\mathcal{Z}$ space $\Leftrightarrow$ some quadratic curves in $\mathcal{X}$-space

General Quadratic Hypothesis Set

- A "bigger" $\mathcal{Z}$-space:

$$\Phi_2(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

General Quadratic Hypothesis Set

- A "bigger" $\mathcal{Z}$-space:

$$\Phi_2(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

- Linear in $\mathcal{Z}$-space $\Leftrightarrow$ quadratic hypotheses in $\mathcal{X}$-space

General Quadratic Hypothesis Set

- A "bigger" $\mathcal{Z}$-space:

$$\Phi_2(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

- Linear in $\mathcal{Z}$-space $\Leftrightarrow$ quadratic hypotheses in $\mathcal{X}$-space
- The hypotheses space:

$$\mathcal{H}_{\Phi_2} = \{h(\boldsymbol{x}) : h(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^T \Phi_2(\boldsymbol{x}) \text{ for some } \tilde{\boldsymbol{w}}\}$$

(Quadratic hypotheses)

General Quadratic Hypothesis Set

- A "bigger" $\mathcal{Z}$-space:

$$\Phi_2(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

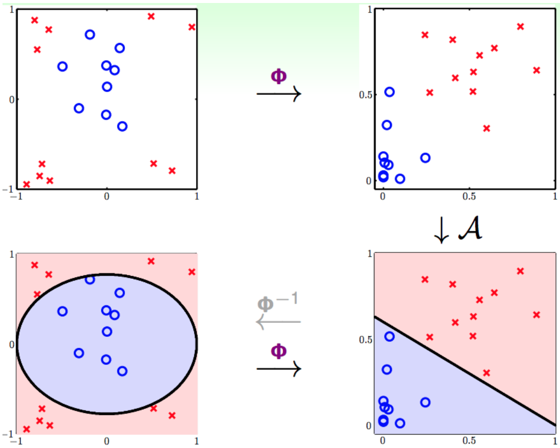
- Linear in $\mathcal{Z}$-space $\Leftrightarrow$ quadratic hypotheses in $\mathcal{X}$-space
- The hypotheses space:

$$\mathcal{H}_{\Phi_2} = \{h(\boldsymbol{x}) : h(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^T \Phi_2(\boldsymbol{x}) \text{ for some } \tilde{\boldsymbol{w}}\}$$

  (Quadratic hypotheses)
- Also include linear model as a degenerate case

Learning a good quadratic function

- Transform original data $\{\mathbf{x}_n, y_n\}$ to $\{\mathbf{z}_n = \Phi(\mathbf{x}_n), y_n\}$
- Solve a linear problem on $\{\mathbf{z}_n, y_n\}$ using your favorite algorithm $\mathcal{A}$ to get a good model $\tilde{\mathbf{w}}$
- Return the model $h(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^T \Phi(\mathbf{x}))$



Polynomial mappings

- Can now freely do quadratic classification, quadratic regression, $\cdots$

Polynomial mappings

- Can now freely do quadratic classification, quadratic regression, $\cdots$
- Can easily extend to any degree of polynomial mappings

  E.g., $\Phi(\boldsymbol{x}) = (x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x1x_2^2, x_1x_3^2, x_1x_2^2, x_2^2x_3, x_1^2x_3, x_2^2x_3, x_1^3, x_2^3, x_3^3)$

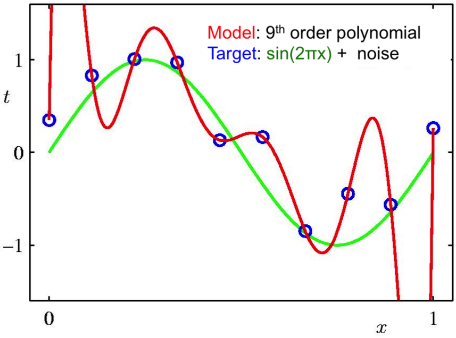
The price we pay: Computational complexity

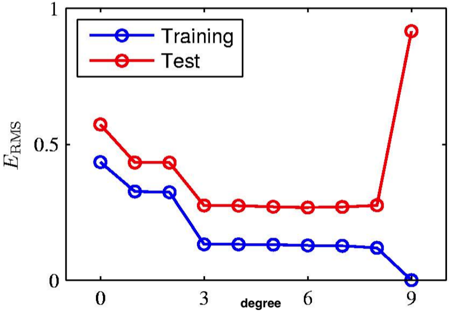
- Q-th order polynomial transform:

$$\begin{aligned}\Phi(\boldsymbol{x}) = (1, x_1, x_2, \cdots, x_d,\\ x_1^2, x_1 x_2, \cdots, x_d^2,\\ \cdots\\ x_1^Q, x_1^{Q-1}x_2, \cdots, x_d^Q)\end{aligned}$$

- $O(d^Q)$ dimensional vector $\Rightarrow$ High computational cost

The price we pay: overfitting





- Overfitting: the model has low training error but high prediction error.

Theory of Generalization

Material is from "Learning from data"

Training versus testing

Machine learning pipeline:

- Training phase:
  - Obtain the best model h by minimizing training error
- Test (inference) phase:
  - For any incoming test data $\boldsymbol{x}$:
    Make prediction by $h(\boldsymbol{x})$
  - Measure the performance of h: test error

Training versus testing

Does low training error imply low test error?

- They can be totally different if

$$\text{train distribution} \neq \text{test distribution}$$

Training versus testing

Does low training error imply low test error?

- They can be totally different if

$$\text{train distribution} \neq \text{test distribution}$$

- Even under the same distribution, they can be very different:

  Because h is picked to minimize training error, not test error

Formal definition

- Assume training and test data are both sampled from D
- The ideal function (for generating labels) is f: $f(x) \rightarrow y$
- Training error: Sample $x_1, \cdots, x_N$ from D and

$$E_{\text{tr}}(h) = \frac{1}{N} \sum_{n=1}^{N} e(h(x_n), f(x_n))$$

h is determined by $x_1, \cdots, x_N$

- Test error: Sample $x_1, \cdots, x_M$ from D and

$$E_{\text{te}}(h) = \frac{1}{M} \sum_{m=1}^{M} e(h(x_m), f(x_m))$$

h is independent to $x_1, \cdots, x_M$

Formal definition

- Assume training and test data are both sampled from D
- The ideal function (for generating labels) is f: $f(x) \rightarrow y$
- Training error: Sample $x_1, \cdots, x_N$ from D and

$$E_{\text{tr}}(h) = \frac{1}{N} \sum_{n=1}^{N} e(h(x_n), f(x_n))$$

  h is determined by $x_1, \cdots, x_N$

- Test error: Sample $x_1, \cdots, x_M$ from D and

$$E_{\text{te}}(h) = \frac{1}{M} \sum_{m=1}^{M} e(h(x_m), f(x_m))$$

  h is independent to $x_1, \cdots, x_M$

- Generalization error = Test error = Expected performance on D:

$$E(h) = E_{x \sim D}[e(h(x), f(x))] = E_{\text{te}}(h)$$

The 2 questions of learning

- $E(h) \approx 0$ is achieved through:

$$E(h) \approx E_{\text{tr}}(h) \quad \text{and} \quad E_{\text{tr}}(h) \approx 0$$

The 2 questions of learning

- $E(h) \approx 0$ is achieved through:

$$E(h) \approx E_{\text{tr}}(h) \quad \text{and} \quad E_{\text{tr}}(h) \approx 0$$

- Learning is split into 2 questions:
  - Can we make sure that $E(h) \approx E_{\text{tr}}(h)$?
    today's focus
  - Can we make $E_{\text{tr}}(h)$ small?
    Optimization (done)

How to bound $\|E(h) - E_{\text{tr}}(h)\|$?

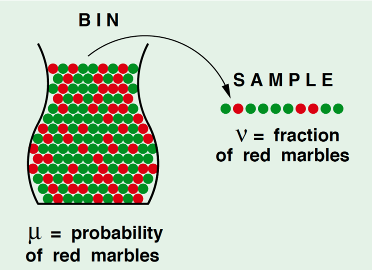
Inferring Something Unknown

- Consider a bin with red and green marbles

$$P[\text{picking a red marble}] = \mu$$

$$P[\text{picking a green marble}] = 1 - \mu$$

- The value of μ is unknown to us
- How to infer μ?
    - Pick N marbles independently
    - ν: the fraction of red marbles



Inferring with probability

- Do you **know** μ?

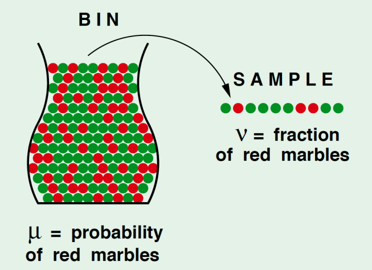
  No

  Sample can be mostly green while bin is mostly red

- Can you say something about μ?

  Yes

  ν is "probably" close to μ



Hoeffding's Inequality

- In big sample (large N), ν (sample mean) is probably close to μ:

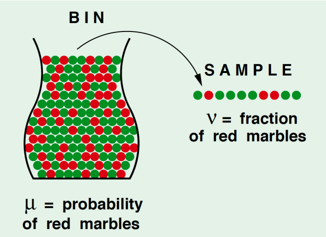
$$P[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

  This is called **Hoeffding's inequality**

- The statement "$\mu = \nu$" is probably approximately correct (PAC)

Hoffding's Inequality

$$P[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$



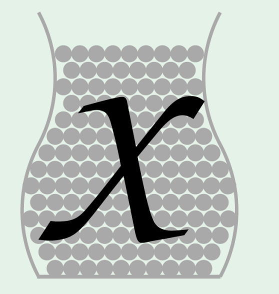
- Valid for all N and $\epsilon > 0$
- Does not depend on μ (no need to know μ)
- Larger sample size N or looser gap ϵ

  $\Rightarrow$ higher probability for $\mu \approx \nu$

Connection to Learning

How to connect this to learning?

- Each marble (uncolored) is a data point $\boldsymbol{x} \in \mathcal{X}$

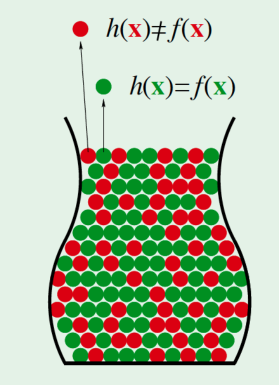


Connection to Learning

How to connect this to learning?

- Each marble (uncolored) is a data point $\boldsymbol{x} \in \mathcal{X}$
- red marble: $h(\boldsymbol{x}) \neq f(\boldsymbol{x})$ (h is correct)

  green marble: $h(\boldsymbol{x}) = f(\boldsymbol{x})$ (h is wrong)



Connection to Learning

- Given a function h:
- If we randomly draw $x_1, \cdots, x_N$ (independent to h):
  - $E(h) = E_{\boldsymbol{x} \sim D}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})]$ (generalization error, unknown)
    $\Leftrightarrow \mu$
  - $\frac{1}{N}\sum_{n=1}^{N}[h(\boldsymbol{x}_n) \neq y_n]$ (error on sampled data, known)
    $\Leftrightarrow \nu$

Connection to Learning

- Given a function h:
- If we randomly draw $x_1, \cdots, x_N$ (independent to h):
  - $E(h) = E_{\boldsymbol{x} \sim D}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})]$ (generalization error, unknown) $\Leftrightarrow \mu$
  - $\frac{1}{N}\sum_{n=1}^{N}[h(\boldsymbol{x}_n) \neq y_n]$ (error on sampled data, known) $\Leftrightarrow \nu$
- Based on Hoeffding's inequality:

$$P[|\mu - \nu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

- "$\mu = \nu$" is probably approximately correct (PAC)

Connection to Learning

- Given a function h:
- If we randomly draw $x_1, \cdots, x_N$ (independent to h):
  - $E(h) = E_{\boldsymbol{x} \sim D}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})]$ (generalization error, unknown)
    $\Leftrightarrow \mu$
  - $\frac{1}{N}\sum_{n=1}^{N}[h(\boldsymbol{x}_n) \neq y_n]$ (error on sampled data, known)
    $\Leftrightarrow \nu$
- Based on Hoeffding's inequality:

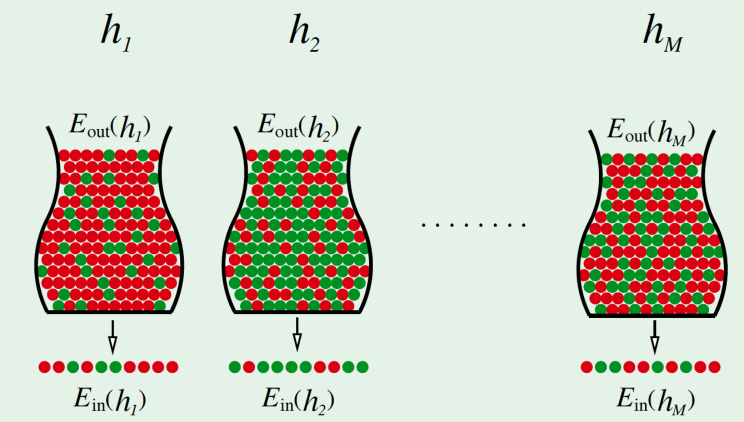
$$P[|\mu - \nu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

- "$\mu = \nu$" is probably approximately correct (PAC)
- However, this can only "verify" the error of a hypothesis:

  h and $x_1, \cdots, x_N$ must be independent

Apply to multiple bins (hypothesis)

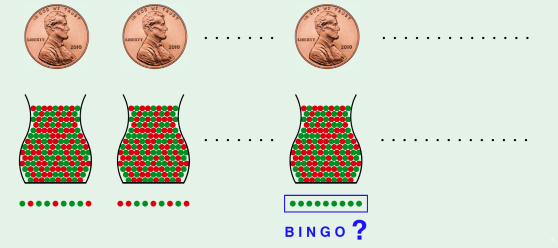
Can we apply to multiple hypothesis?



Color in each bin depends on different hypothesis

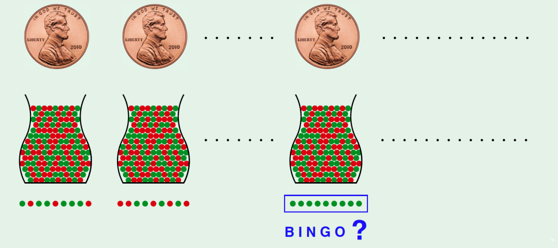
Bingo when getting all green balls?

Coin Game



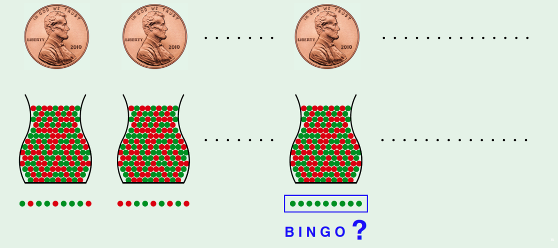
- If you have 150 **fair** coins, flip each coin 5 times, and one of them gets 5 heads. Is this coin (g) special?

Coin Game



- If you have 150 **fair** coins, flip each coin 5 times, and one of them gets 5 heads. Is this coin (g) special?
- No. The probability of existing one of the coin results in 5 heads is $1-(\frac{31}{32})^{150} > 99\%$

Coin Game



- If you have 150 **fair** coins, flip each coin 5 times, and one of them gets 5 heads. Is this coin (g) special?
- No. The probability of existing one of the coin results in 5 heads is $1 - (\frac{31}{32})^{150} > 99\%$
- Because: there can exist some h such that E and E_{tr} are far away if M is large.

A Simple Solution

- For each particular h,

$$P\left[|E_{tr}(h) - E(h)| > \epsilon\right] \leq 2e^{-2\epsilon^2 N}$$

- We want a "union bound":

$$P\left[|E_{tr}(h_1) - E(h_1)| > \epsilon \text{ or } \cdots \text{ or } |E_{tr}(h_M) - E(h_M)| > \epsilon\right]$$
$$\leq \sum_{m=1}^{M} P\left[|E_{tr}(h_m) - E(h_m)|\right] \leq 2Me^{-2\epsilon^2 N}$$

When is learning successful?

When our Learning Algorithm $\mathcal{A}$ picks the hypothesis g:

$$P[|E_{tr}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

- If M is small and N is large enough:

  If $\mathcal{A}$ finds $E_{tr}(g) \approx 0$

  $\Rightarrow E(g) \approx 0$ (Learning is successful!)

Feasibility of Learning

$$P[|E_{tr}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

Two questions:

(1) Can we make sure $E(g) \approx E_{tr}(g)$?

(2) Can we make sure $E(g) \approx 0$?

Feasibility of Learning

$$P[|E_{tr}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

Two questions:

(1) Can we make sure $E(g) \approx E_{tr}(g)$?

(2) Can we make sure $E(g) \approx 0$?

M: complexity of model

- Small M: (1) holds, but (2) may not hold (too few choices) (under-fitting)

Feasibility of Learning

$$P[|E_{tr}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

Two questions:

(1) Can we make sure $E(g) \approx E_{tr}(g)$?

(2) Can we make sure $E(g) \approx 0$?

M: complexity of model

- Small M: (1) holds, but (2) may not hold (too few choices) (under-fitting)
- Large M: (1) doesn't hold, but (2) may hold (over-fitting)

What the theory will achieve

- Currently we only know

$$P[|E_{\text{tr}}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

What the theory will achieve

- Currently we only know

$$P[|E_{\text{tr}}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

- What if $M = \infty$?

  (e.g., linear hyperplanes)

What the theory will achieve

- Currently we only know

$$P[|E_{\text{tr}}(g) - E(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

- What if $M = \infty$?

  (e.g., linear hyperplanes)

- Todo:

  We will establish a finite quantity to replace M

$$P[|E_{\text{tr}}(g) - E(g)| > \epsilon] \overset{?}{\leq} 2m_{\mathcal{H}}(N)e^{-2\epsilon^2 N}$$

- Study $m_{\mathcal{H}}(N)$ to understand the trade-off for model complexity

Conclusions

- Polynomial feature expansion: Our first nonlinear model
- Bounding the generalization (test) error

Questions?