Risk Minimization, Learning theory

Xuezhi Wang

Computer Science Department
Carnegie Mellon University

10701-recitation, Mar 5

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Loss

- Data: $X_1, ..., X_n$
- Estimate: $\hat{\theta} = \hat{\theta}(X_1, ..., X_n)$
- How good is this estimator $\hat{\theta}$?

Loss

Suppose the true parameter is θ, we need to quantify how far is $\hat{\theta}$ from θ.

- Squared error loss: $L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2$
- Absolute error loss/L-1 loss: $L(\theta, \hat{\theta}) = |\theta - \hat{\theta}|$
- L-p loss: $L(\theta, \hat{\theta}) = |\theta - \hat{\theta}|^p$
- Zero-one loss: $L(\theta, \hat{\theta}) = I(\theta = \hat{\theta})$
- Large deviation loss: $L(\theta, \hat{\theta}) = I(|\theta - \hat{\theta}| > c)$
- KL loss: $L(\theta, \hat{\theta}) = \int \log(\frac{p(x;\theta)}{p(x;\hat{\theta})}) p(x;\theta) dx$

Loss

If $\theta = (\theta_1, ..., \theta_k)$ is a vector, then some common loss functions are:

- Squared error loss: $L(\theta, \hat{\theta}) = ||\theta - \hat{\theta}||^2 = \sum_{j=1}^{k} (\hat{\theta}_j - \theta_j)^2$
- L-p loss: $L(\theta, \hat{\theta}) = ||\theta - \hat{\theta}||_p = (\sum_{j=1}^{k} |\hat{\theta}_j - \theta_j|^p)^{1/p}$

Loss: examples

- For classification, we usually want to predict $Y \in \{0, 1\}$ based on some classifier $h(x)$
  - Zero-one loss: $L(Y, h(X)) = I(Y \neq h(X))$
- For regression, we usually want to predict $Y \in \mathbb{R}$ based on some regressor $h(x)$
  - Squared-error loss: $L(Y, h(X)) = (Y - h(X))^2$

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Risk

The risk of an estimator $\hat{\theta}$ is:

$$R(\theta, \hat{\theta}) = E_\theta(L(\theta, \hat{\theta})) = \int L(\theta, \hat{\theta}) p(x; \theta) dx$$

When the loss function is squared error, the risk is the **MSE**:

$$R(\theta, \hat{\theta}) = E_\theta(\theta - \hat{\theta})^2 = Var_\theta(\hat{\theta}) + bias^2$$

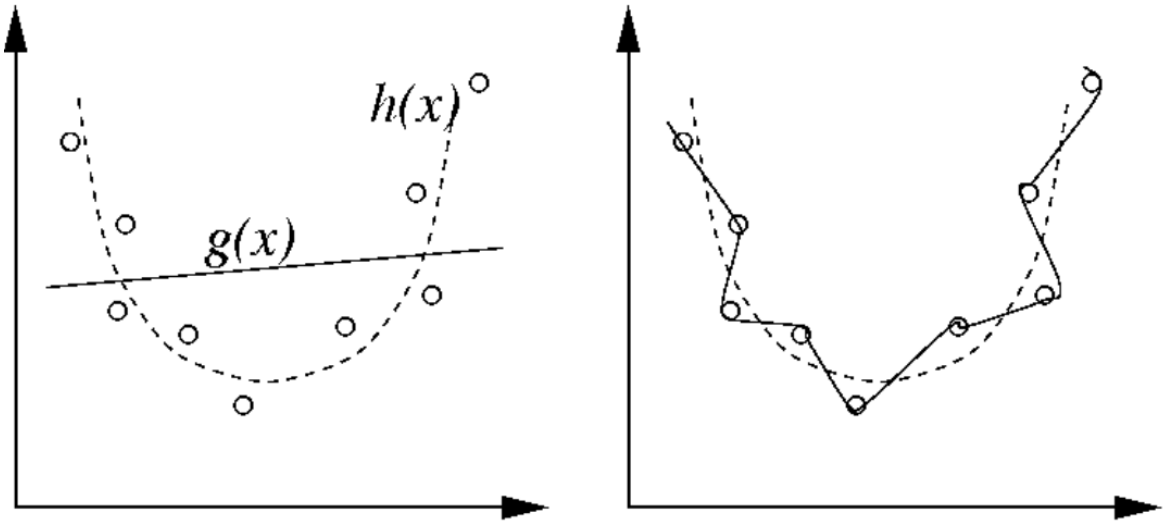
MSE

- $Bias = E_\theta(\hat{\theta}) - \theta$
- Variance=$Var_\theta(\hat{\theta}) = E_\theta(\hat{\theta} - E_\theta(\hat{\theta}))^2$

The bias-variance decomposition of MSE:

$$\begin{aligned} E_\theta(\theta - \hat{\theta})^2 &= E_\theta(\theta - E_\theta(\hat{\theta}) + E_\theta(\hat{\theta}) - \hat{\theta})^2 \\ &= E_\theta(\hat{\theta} - E_\theta(\hat{\theta}))^2 + (E_\theta(\hat{\theta}) - \theta)^2 \\ &\quad + 2(E_\theta(\hat{\theta}) - \theta)E_\theta(\hat{\theta} - E_\theta(\hat{\theta})) \\ &= Var_\theta(\hat{\theta}) + bias^2 \end{aligned}$$

Bias-Variance Decomposition



- An estimator is **unbiased** if the bias is 0. Then MSE=Var.
- Usually there is a tradeoff between bias and variance.
- Low bias can imply high variance and vice versa.
- Underfitting: high bias, low variance
- Overfitting: low bias, high variance

MSE example

Suppose $X_1, ..., X_n \sim N(\mu, \sigma^2)$.
Estimate μ, σ^2 using $\bar{X} = \frac{1}{n}\sum X_i$, $S^2 = \frac{1}{n-1}\sum (X_i - \bar{X})^2$.

- $\bar{X}$ is unbiased, since $E(\bar{X}) = \mu$.
- Hence MSE=$Var(\bar{X}) = E(\bar{X} - \mu)^2 = \frac{\sigma^2}{n}$
- S^2 is also unbiased, since $E(S^2) = \sigma^2$.
- Hence MSE=$Var(S^2) = E(S^2 - \sigma^2)^2 = \frac{2\sigma^4}{n-1}$

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Examples

- Empirical cdf
  - $F_n(t) = \frac{1}{n} \sum I(X_i \leq t)$
  - $P(|F_n(t) - F(t)| > \epsilon) \leq 2e^{-2n\epsilon^2}$
  - $P(\sup_t |F_n(t) - F(t)| > \epsilon) \leq ?$
- Classification
  - $R(h) = P(Y \neq h(X)), R_n(h) = \frac{1}{n} \sum I(Y_i \neq h(X_i))$
  - $P(|R_n(h) - R(h)| > \epsilon) \leq 2e^{-2n\epsilon^2}$
  - $P(\sup_h |R_n(h) - R(h)| > \epsilon) \leq ?$

Uniform Bounds

Why do we care about this?

- $R(h) = P(Y \neq h(X)), R_n(h) = \frac{1}{n} \sum I(Y_i \neq h(X_i))$
- $P(\sup_h |R_n(h) - R(h)| > \epsilon) \leq ?$

If it holds, we can say something nice about the training procedure in Machine Learning.
In supervised learning we usually minimize the training error:

$$R_n(h) = \frac{1}{n} \sum I(Y_i \neq h(X_i))$$

Suppose we get $\hat{h}$ that minimizes $R_n(h)$.
How can we expect it performs well on the test data, i.e., how small is $R(h) = P(Y \neq h(X))$?

Uniform Bounds

Let h_* be the function that minimize the true error $R(h)$.
If the following holds:

$$P(\sup_h |R_n(h) - R(h)| > \epsilon) \leq \text{something small}$$

Then with high probability,

$$R(\hat{h}) \leq R_n(\hat{h}) + \epsilon \leq R_n(h_*) + \epsilon \leq R(h_*) + 2\epsilon$$

So we know if we minimize the training error, the smallest true error will only be 2ϵ away from the test error using our trained model.

Finite classes

- Union bound $P(A_1 \cup ... \cup A_N) \leq \sum_{i=1}^{N} P(A_i)$
- Uniform Bounds
  Suppose $\max_{1 \leq j \leq N} \sup_x |f_j(x)| \leq B$

$$P(\sup_f |P_n(f) - P(f)| > \epsilon) = P(A_1 \cup ... \cup A_N) \leq \sum_{i=1}^{N} P(A_i)$$
$$\leq \sum_{i=1}^{N} 2e^{-n\epsilon^2/(2B^2)}$$
$$= 2Ne^{-n\epsilon^2/(2B^2)}$$

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Infinite classes: Shattering

Let A be a class of sets, $F = \{x_1, ..., x_n\}$. Let G be a subset of F. Say that A **picks out** G if $A \cap F = G$.
Let $s(A, F)$ be the number of subsets picked out by A.
Examples: $A = \{(a, b) : a \leq b\}$.

- $F = \{1, 2, 3\}$. Then A can pick out:

$$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}$$

  $s(A, F) = 7$.
- $F = \{1, 2\}$ Then A can pick out:

$$\emptyset, \{1\}, \{2\}, \{1, 2\}$$

  $s(A, F) = 4$.

Infinite classes: Shattering

n is the number of points in F.

- Obviously, $s(A, F) \leq 2^n$.
- F is **shattered** if $s(A, F) = 2^n$.
- **Shatter coefficient**: $s_n(A) = \sup_{F \in \mathcal{F}_n} s(A, F)$.
- Still $s_n(A) \leq 2^n$.

Infinite classes: Shattering

Let $\mathcal{A}$ be a class of sets. Then

$$P(\sup_{A \in \mathcal{A}} |P_n(A) - P(A)| > \epsilon) \leq 8 s_n(\mathcal{A}) e^{-n\epsilon^2/32}$$

How large is $s_n(\mathcal{A})$?

VC dimension

The VC dimension is:

$$d = d(\mathcal{A}) = \text{largest n such that } s_n(\mathcal{A}) = 2^n$$

d is the size of the largest set that can be shattered. Hence

- For $n \leq d$, $s_n(\mathcal{A}) = 2^n$
- For $n > d$, $s_n(\mathcal{A}) \leq 2^n$

But for $n > d$, how does $s_n(\mathcal{A})$ behave?

Sauer's Theorem

Suppose $\mathcal{A}$ has finite VC dimension d. Then for all $n > d$,

$$s(\mathcal{A}, n) \leq (n+1)^d$$

So now we can conclude:

$$P(\sup_{A \in \mathcal{A}} |P_n(A) - P(A)| > \epsilon) \leq 8(n+1)^d e^{-n\epsilon^2/32}$$

VC dimension: examples

- Intevals $[a, b]$ on the real line: $d = 2$
- Halfspace in $\mathbb{R}^2$: $d = 3$
- Discs in $\mathbb{R}^2$: $d = 3$
- Convex polygons in $\mathbb{R}^2$: $d = \infty$
- $\sin(\pi a x)$ for $a \in \mathbb{R}$: $d = \infty$

Exercise: What is the VC dimension for $\{a\} \cup [b, c] \cup \{d\}$?

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Conditional Gaussian

if $\mathbf{x}_1, \mathbf{x}_2$ is jointly Gaussian, i.e.,

$$p(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} | \mu, \Sigma) = \mathcal{N}(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} | \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix})$$

The conditional distribution is also Gaussian:

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1|\mathbf{m}_{1|2}, V_{1|2})$$

$$m_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2), \mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

Gaussian Process Regression

The joint Gaussian is:

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^\top & K_{**} \end{bmatrix})$$

Then

$$y^*|y \sim \mathcal{N}(\mu_2 + K_*^\top K^{-1}(y - u_1), K_{**} - K_*^\top K^{-1} K_*)$$

where

$$K(x, x') = \sigma_f^2[\exp\{\frac{-(x - x')^2}{2\sigma^2}\}] + \sigma_n^2 \delta(x, x')$$

Outline

1. Risk Minimization
   - Loss
   - Risk
2. Learning theory
   - From Empirical to truth
   - VC dimension
3. Gaussian Process
   - Gaussian Process Regression
   - Gaussian Process Classification

Gaussian Process Classification

We recieve $x_1, ..., x_n$, but $y_i \in \{-1, 1\}$.
Can we assume

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^\top & K_{**} \end{bmatrix})$$

If not, how can we connect this with previous results in regression?

Gaussian Process Classification

Solution: We assume doing regression for $\{x_1, t_1\}, ..., \{x_n, t_n\}$, where $t_i \in \mathbb{R}$. Now we add a model from t_i to y_i:

$$p(y_i|t_i) = \frac{1}{1 + e^{-t_i y_i}}$$

We can still assume that: $t \sim \mathcal{N}(\mu, K)$

Gaussian Process Classification

Solution: We observe $x_i, y_i, i = 1, ..., n$, and we want to maximize

$$\begin{aligned} p(t|y,x) &\propto p(t|x)p(y|t) \\ &\propto p(t|x)\prod_i p(y_i|t_i) \\ &\propto exp\{-\frac{1}{2}t^\top K^{-1}t\}\prod_i \frac{1}{1+e^{-t_i y_i}} \end{aligned}$$

Equivalently we maximize:

$$\log p(t|y,x) = -\frac{1}{2}t^\top K^{-1}t - \sum_i \log(1+e^{-t_i y_i})$$

Gaussian Process Classification

Solution:

$$\min_t \frac{1}{2} t^\top K^{-1} t + \sum_i \log(1 + e^{-t_i y_i})$$

We get t, which is continuous. For a new point x^*, we can first do regression:

$$\begin{bmatrix} t \\ t^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^\top & K_{**} \end{bmatrix}\right)$$

$$t^* | t \sim \mathcal{N}(\mu_2 + K_*^\top K^{-1}(y - u_1), K_{**} - K_*^\top K^{-1} K_*)$$

After we get t^*, we can predict y^* using

$$p(y^* | t^*) = \frac{1}{1 + e^{-y^* t^*}}$$