Lecture 06 - Support Vector Machines

NIPS

j: 60↑

(1) Naive Bayes
- Laplace Smoothing
- Event Models

(2) Comments on apply ML

(3) SVM intros

(1) Recap

$$x = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 1 \\ 0 \\ \dots \end{bmatrix} \begin{matrix} a \\ aadv \\ \dots \\ buy \\ \vdots \end{matrix}$$

$x_j = 1\{\text{whether } j \text{ appears in email}\}$

ith → training examples

jth → index in features

→ Generative Model (Gaussian or Bernoulli)

$$p(x|y) = \prod_{j=1}^{n} p(x_j|y) \quad p(y)$$

→ Maximum likelihood

→ parameters $P(y=1) = \phi_y$

$P(x_j = 1 | y=0) = \phi_{j|y=0}$

$P(x_j = 1 | y=1) = \phi_{j|y=1}$

At prediction time:

$$P(y=1|x) = \frac{P(x|y=1) \cdot P(y=1)}{P(x|y=1) \cdot P(y=1) + P(x|y=0) \cdot P(y=0)}$$

→ Laplace smoothing

#1 +1

#0 +1

More Generally:

$x \in \{1 \dots K\}$

$$\Rightarrow \text{Estimate } P(x=j) = \frac{\sum_{j=1}^{m} 1\{x^{(i)} = j\} + 1}{M + K}$$

$$\Rightarrow \phi_{j|y=0} = \frac{\sum_{j=1}^{m} 1\{x_j^{(i)} = 1, y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} + 2}$$

$x_i \in \{1 \dots K\}$

Size	<400	400 - 800	800 - 1200	>1200
x_i	1	2	3	4

$$P(x|y) = \prod_{j=1}^{n} P(x_j|y) \rightarrow \text{multinomial}$$

- New representation $x = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$ a ← 1, ad... ← 2, buy ← 800, drugs ← 1600, now ← 6200

$x \in \begin{bmatrix} 1600 \\ 800 \\ 1600 \\ 6200 \end{bmatrix} \in \mathbb{R}^w$

$x_j \in \{1 \dots 10000\}$

R_i = length of email i

Multivariate Bernoulli event model $\{0,1 \dots 1\}$

↳ Multinomial event model {new representation}

$P(x,y) = p(x|y) \cdot P(y)$

assume $= \prod_{j=1}^{R} p(x_i|y) \cdot P(y)$

- Parameters: $\phi_y = P(y=1)$

$\phi_{k|y=0} = P(x_j = k | y=0)$

↳ Chance of word j being k if $y=0$

- doesn't depend on j^{th}

$\phi_{k|y=1} = P(x_j = k | y=1)$

- MLE:

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\} \sum_{j=0}^{n_i} 1\{x_j^{(i)}=k\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)}=0\} n_i + 10000}$$

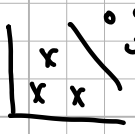
$\phi_{k|y=0}$ ↓ $1 \dots 10000$

word embedding ↳ $C \leq 230$

→ Support Vector Machine

non-linear boundaries

- optimal margin classifier (separable case)

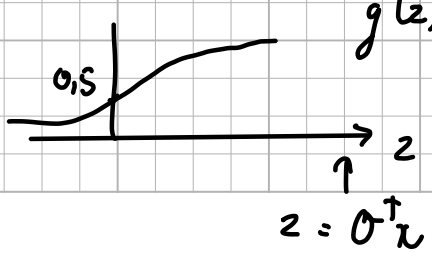


- Kernels $x \to \phi(x) \to \mathbb{R}^{\infty}$

- Inseperable case

Functional margin → how confidently

$h_\theta(x) = g(\theta^T x)$



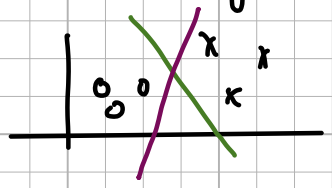
Predict "1" if $\theta^T x \geq 0$ ($h_\theta(x) = g(\theta^T x) \geq 0.5$)

"0" otherwise

- if $y^{(i)} = 1$, hope that $\theta^T x^{(i)} \gg 0$
- if $y^{(i)} = 0$, hope that $\theta^T x^{(i)} \ll 0$

Geometric margin → optimal margin classifier

to max geometric margin



Notation:

Labels $y \in \{-1, +1\}$

→ have h output $\{-1, +1\}$

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \dots \end{bmatrix} \begin{matrix} \} b \\ \\ \} w \end{matrix}$$

$$h_\theta(x) = g(\theta^T x)$$

$\uparrow \mathbb{R}^{n+1}, x_0 = 1$

$$h_{w,b}(x) = g(w^T x + b)$$

$\uparrow \mathbb{R}^n \quad \uparrow \mathbb{R}$

Drop $x_0 = 1$ Convention

- functional margin of hyperplane defined by (w, b) wrt $(x^{(i)}, y^{(i)})$

$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$ → very large $\gg 0$

if $y^{(i)} = 1$ want $w^T x^{(i)} + b \gg 0$

if $y^{(i)} = -1$ want $w^T x^{(i)} + b \ll 0$

→ if $\hat{\gamma}^{(i)} \gg 0$ that means $h(x^{(i)}) = y^{(i)}$

and $\hat{\gamma} = \min_{i=1\dots n} \hat{\gamma}^{(i)}$

$$\Rightarrow \gamma^{(i)} = \frac{(w^T x^{(i)} + b) \cdot y^{(i)}}{\|w\|}$$

geometric margin $= \frac{\hat{\gamma}^{(i)}}{\|w\|}$ ← functional margin

$\|w\| = 1$, $(w, b) \rightarrow \left(\frac{w}{\|w\|}, \frac{b}{\|w\|}\right)$ Normalize

→ optimal margin classifier

Choose w, b to max γ

$$\max_{\gamma, w, b} \gamma \quad \text{s.t} \quad \frac{y^{(i)}(w^T x^{(i)} + b)}{\|w\|} \geq \gamma \quad i = 1 \dots m$$

→ $$\min_{w,b} \|w\|^2$$
$$\text{s.t} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1$$

Lecture 07 SVM

- Optimization problem
- Representer theorem
- Kernels
- Examples of Kernels

(1) Optimal margin classifier

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1$$

$$\gamma^{(i)} = \frac{y^{(i)}(w^T x^{(i)} + b)}{\|w\|}$$ (Geometric margin)

$$\gamma = \min_{i=1\dots m} \gamma^{(i)}$$

$$y = g(w^T x + b)$$

$x^{(i)} \in \mathbb{R}^{100}$ $\quad y^{(i)} \in \{-1, +1\}$

Suppose $w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$

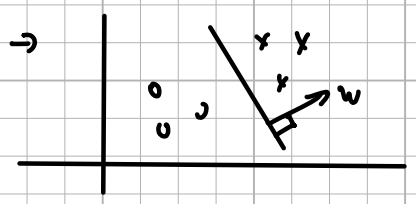
$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix} \rightarrow \begin{bmatrix} b \\ w_1 \\ \dots \\ w_n \end{bmatrix}$$

(Intuition #1)

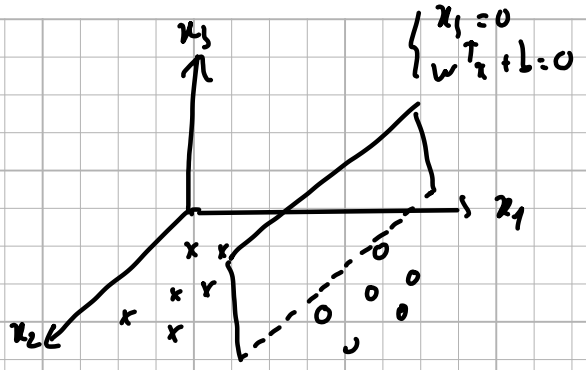
→ Gradient Descent : $\theta_i := 0$

$$\theta := \theta - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

(Intuition #2)



$g(w^T x + b)$



- $w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$

$\min_{w,b} \frac{1}{2} \|w\|^2 \rightarrow w^T w$

$$s.t \; y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad i = 1 \dots m$$

$\langle x, z \rangle = x^T z$ is the inner products

$$\rightarrow \max \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y^{(i)} y^{(j)} \underbrace{x^{(i)T} x^{(j)}}_{\langle x^{(i)}, x^{(j)} \rangle}$$

$$s.t \; y^{(i)}\left(\left(\sum_j \alpha_j y^{(j)} \underbrace{x^{(j)T} x^{(i)}}_{\langle x^{(j)}, x^{(i)} \rangle} + b\right)\right) \geq 1$$

"Dual optimization problem"

$h_{w,b}(x) = g(w^T x + b)$

$= g\left(\sum_i \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b\right)$

(*) Kernel trick

1) Write algo in terms of $\langle x^{(i)}, x^{(j)} \rangle$
2) Let there be mapping $x \in \mathbb{R}^2 \mapsto \phi(x) \in \mathbb{R}^\infty$
3) Find way to compute $K(x, z) = \phi(x)^T \phi(z)$
4) Replace $\langle x, z \rangle$ in algo with $K(x, z)$

$x \in \mathbb{R}^n \rightarrow \phi(x) \in \mathbb{R}^{n^2}$

$$K(x,z) = \phi(x)^T \phi(z) \underset{\uparrow \text{prove}}{=} (x^T z)^2 = \underbrace{\left(\sum_{i=1}^{n} x_i z_i\right)}_{x^T z} \underbrace{\left(\sum_{j=1}^{n} x_j z_j\right)}_{x^T z}$$

($\mathbb{R}^n$, $\mathbb{R}^N$)

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i x_j)(z_i z_j)$$

$\hookrightarrow \in \mathbb{R}^N$

$$K(x,z) = (x^T z + c)^2$$

$\hookrightarrow$
$$x = \begin{bmatrix} x_1 x_1 \\ \cdots \\ \cdots \\ x_3 x_3 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ \sqrt{2c}\, x_3 \\ c \end{bmatrix} \qquad z = \begin{bmatrix} z_1 z_1 \\ \cdots \\ \cdots \\ z_3 z_3 \\ \sqrt{2c}\, z_1 \\ \sqrt{2c}\, z_2 \\ \sqrt{2c}\, z_3 \\ c \end{bmatrix}$$

$K(x,z) = (x^T z + c)^d \rightarrow O(n)$ time

ϕ^T has all $\binom{n+d}{d}$ feature of monomial up to order d

→ SVM = optimal margin classifier + Kernel trick

- How to make kernels

if x, z are "similar" $K(x,z) = \phi(x)^T \phi(z)$ is large and otherwise

$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$$ → Gaussian Kernel $\phi(x) \in \mathbb{R}^\infty$

↳ when to apply?

Does that exists ϕ s.t $K(x,z) = \phi(x)^T \phi(z)$

and $K(x,x) = \phi(x)^T \phi(x) \geq 0$

Theorem (Mercer): K is a valid Kernel func (i.e $\exists \phi$ s.t $K(x,z) = \phi(x)^T \phi(z)$) if and only if for any d points, the corresponding matrix $K \geq 0$

Let $\{x^{(1)} \dots x^{(d)}\}$ be d points $\quad K_{ij} = K(x^{(i)}, x^{(j)})$

$K \in \mathbb{R}^{d \times d}$ "Kernel matrix"

→ Linear Kernel $K(x,z) = x^T z$, $\phi(x) = x$

→ L_1 norm soft margin svm

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \quad \text{where} \quad \xi_i \geq 0$$

$$\text{s.t } y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i$$

→ Protein sequence classifier $\quad \alpha_i$ in lagrangian

$$\max_a W(a) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t} \quad 0 \leq \alpha_i \leq C, \quad i = 1 \dots m$$

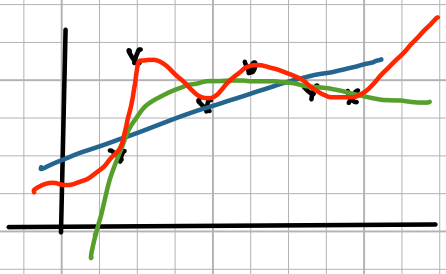
$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

$$L(w, b, \xi, \alpha, r) = \frac{1}{2} w^T w + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i [y^{(i)}(x^T w + b) - 1 + \xi_i] - \sum_{i=1}^{m} r_i \xi_i$$

Lecture 08 Learning theory

- Bias / Variance
- Regularization
- Train / dev / test splits
- Model selection / Cross-Validation

① Bias / Variance



- underfit → high bias (strong bias / prejudice)
- "just right"
- overfit → high variance (totally different res)

(2) Regularization

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{m} \| y^{(i)} - \theta^T x^{(i)} \|^2 + \underbrace{\frac{\lambda}{2} \|\theta\|^2}_{\text{regularization term}}$$

(choose lambda)

$$\arg\max_{\theta} \sum_{i=1}^{m} \log p(y^{(i)} | x^{(i)}; \theta) - \lambda \|\theta\|^2 \rightsquigarrow \|w\|^2$$

- Text classification

$n = 100$

$n = 10000$

$$x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \begin{matrix} a \\ aa \\ \vdots \\ \vdots \end{matrix}$$

- $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

$$P(\theta | S) = \frac{P(S|\theta) P(\theta)}{P(S)}$$

$$\arg\max_{\theta} P(\theta|S) = \arg\max_{\theta} P(S|\theta) P(\theta)$$

Assume: $\theta \sim \mathcal{N}(0, \tau^2 I)$

$$= \arg\max_{\theta} \prod_{i=1}^{m} P(y^{(i)} | x^{(i)}; \theta) \cdot P(\theta)$$

$$P(\theta) = \frac{1}{\sqrt{2\pi} (\tau^2 I)^{n/2}} \exp\left(-\frac{1}{2} \theta^T (\tau^2 I)^{-1} \theta \right)$$

Logistic regression

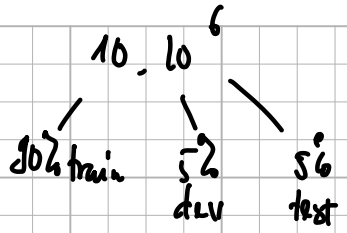
Frequentist → $\arg\max P(S|\theta)$ - MLE

Bayesians → prior $P(\theta)$ → distribution → $\arg\max_{\theta} P(\theta|S)$ - MAP (maximum a posterior)

(3) Train | dev | test sets

- Train each model i (option for degree of polynomial) on S_{train}. Get some hypothesis h_i
- Measure error S_{dev} ⇒ pick lowest S_{dev}
- Evaluate on seperate S_{test} (optional)

CIFAR



60% train | 20% dev | 20% test

(4) Hold-out cross validation

"development set" = "cross validation set"

- Small data sets

  $m = 100$

  70 S_{train}, 30 S_{dev}

  $m \in [20, 100]$

  ↳ $K = m$

K-fold CV

- K=10 typical
- ↳ expensive

```
for d = 1 .. 5 {
  for i = 1 ... K
    Train K-1 pieces
    Test on remaining 1 pieces
  Average
}
```

CS230
ml.yearning.org

- feature selection

  Start with $\mathcal{F} = \emptyset$

  Repeat {

  1) Try add each feature i to $\mathcal{F}$, and see which single-feature addition most improves dev set performance.
  2) Add that feature to $\mathcal{F}$

  }

Lecture 09 - Approx / Estimation Error & ERM

- Setup / Assumptions
- Bias / Variance
- Approx Esti
- Empirical Risk Minimizer
- Uniform Convergence
- VC dimension

Assumptions:

1. Data distribution D

$(x, y) \sim D$ — train / Test

2. Independent Samples

$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\} \sim D \Rightarrow$ [Learning Alg] $\Rightarrow \hat{h}$ or $\hat{\theta} \sim$ Sampling distr

Random var

Deterministic func

S

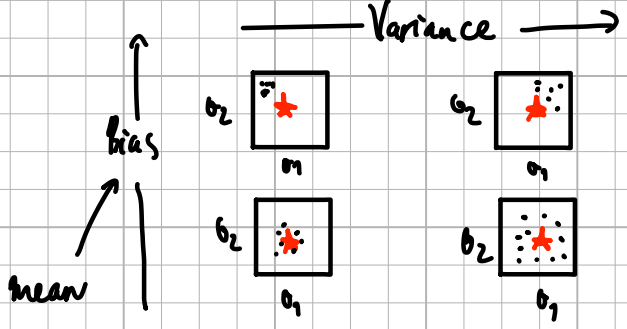
Random var

$\hookrightarrow \theta^*$ or h^* "True" parameter - Not random, Const distribution

Bias / Variance

→ Under fit "just right" overfit

→ Parameter view



$m \to +\infty$

$\mathrm{Var}[\hat{\theta}] \to 0$

↳ "stats efficiency"

$\hat{\theta} \to \theta^*$ $m \to \infty$: Consistent

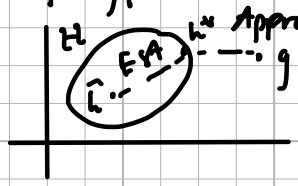
$E[\hat{\theta}] = \theta^*$ for all m

- fighting variance

(i) $m \to \infty$

(ii) Regularization L_1, L_2 → Small Bias, Low Var

Space of Hypothesis



- where

g - Best possible Hypothesis

h^* - Best in class $\mathcal{H}$

$\hat{h}$ - Learn from finite data

$\varepsilon(g)$ = Bayes Error / Irreducible Error

$\varepsilon(h^*) - \varepsilon(g)$: Approx Error } Class

$\varepsilon(\hat{h}) - \varepsilon(h^*)$: Estimation Error } Data

$\varepsilon(h)$ - Risk / Generalization Error

$= E_{(x,y) \sim D}[1\{h(x) \neq y\}]$

$\hat{\varepsilon}_S(h)$: Empirical Risk

$= \frac{1}{m} \sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$

$\varepsilon(\hat{h})$ = Estimation Error + Approx Error + Irreducible Error

Est Var + Est Bias + Approx E + Irreducible Error

Var + Bias + Irredu

- fight high bias
  - make $\mathcal{H}$ bigger → Bias / Var trade-off

Empirical Risk minimizer (ERM)

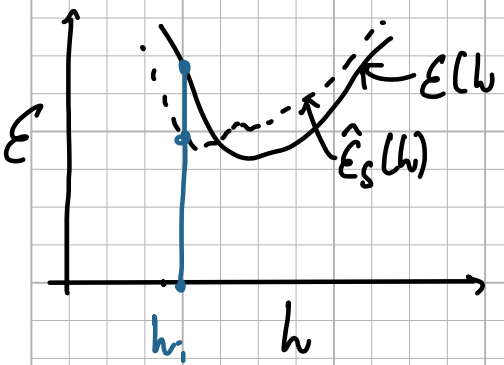
(x, y) ⋮ ⇒ [Learning Alg. ERM] ⇒ $\hat{h}_{ERM}$

$$\hat{h}_{ERM} = \arg\min_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$$

Uniform Convergence → tools

① $\hat{\varepsilon}(h)$ vs $\varepsilon(h)$

② $\varepsilon(\hat{h})$ vs $\varepsilon(h^*)$



i/ Union Bound

$A_1, A_2 \dots A_k$ (need not be independent)

$P(A_1 \cup A_2 \dots \cup A_k) \leq P(A_1) + P(A_2) + \dots + P(A_k)$

ii/ Hoeffding inequality

Let $z_1, z_2 \dots z_m \sim$ Bernoulli (ϕ)

$$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} z_i$$

Let $\gamma > 0$ [margin]

$$Pr\left[|\underbrace{\hat{\phi} - \phi}_{\text{deviation error}}| > \underbrace{\gamma}_{\text{margin}}\right] \leq 2\exp(-2\gamma^2 m)$$

$E[\hat{\varepsilon}(h_i)] = \varepsilon(h_i)$

- Finite Hypotheses

$|\mathcal{H}| = k$

$Pr[\exists h \in \mathcal{H} : |\hat{\varepsilon}(h) - \varepsilon(h)| > \gamma] \leq k \cdot 2\exp(-2\gamma^2 m)$

$\Rightarrow Pr[\forall h \in \mathcal{H} : |\hat{\varepsilon}(h) - \varepsilon(h)| < \gamma] \geq 1 - \underbrace{2k\exp(-2\gamma^2 m)}_{\delta}$

let $\delta = 2|\mathcal{H}| \exp(-2\gamma^2 m)$

δ : prob of error

γ : margin of error

m : Sample size

fixed $\delta, \gamma > 0$

$$m \geq \frac{1}{2\gamma^2} \log \frac{2|\mathcal{H}|}{\delta} \Rightarrow \text{Sample Complexity}$$

$$\mathcal{E}(\hat{h}) \leq \hat{\mathcal{E}}(\hat{h}) + \gamma \leq \hat{\mathcal{E}}(h^*) + \gamma \leq \hat{\mathcal{E}}(h^*) + 2\gamma$$

$\Rightarrow$ with prob $1-\delta$, traing size m

$$\mathcal{E}(\hat{h}) \leq \mathcal{E}(h^*) + 2\sqrt{\frac{1}{2m} + \log \frac{2|\mathcal{H}|}{\delta}}$$

- VC dimension

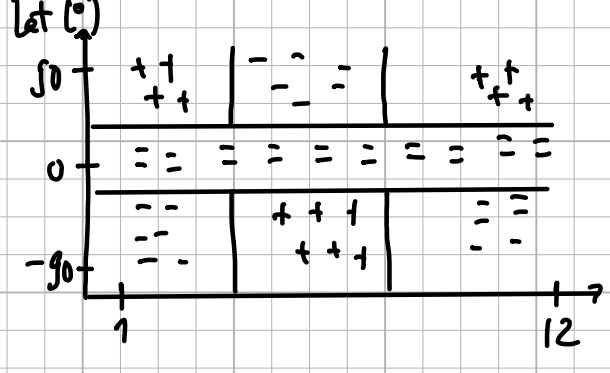
$VC(\mathcal{H}) = |\mathcal{H}|$

$$\mathcal{E}(\hat{h}) \leq \mathcal{E}(h^*) + O\left(\sqrt{\frac{VC(\mathcal{H})}{m} + \log\left[\frac{m}{|\mathcal{H}(\mathcal{H})|}\right] + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

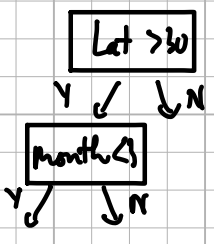
Lecture 10 Decision Trees and Ensemble Methods

* Decision Trees
* Ensemble Methods
* Bagging
* Random Forests
* Boosting

Decision Trees - Non linear Model



Greedy, Top-Down, Recursive Partitioning



- Region R_p

looking for a split S_p

$S_p(j,t) = (\{X \mid X_j < t, X \in R_p\}, \{X \mid X_j \geq t, X \in R_p\})$

R_1 R_2

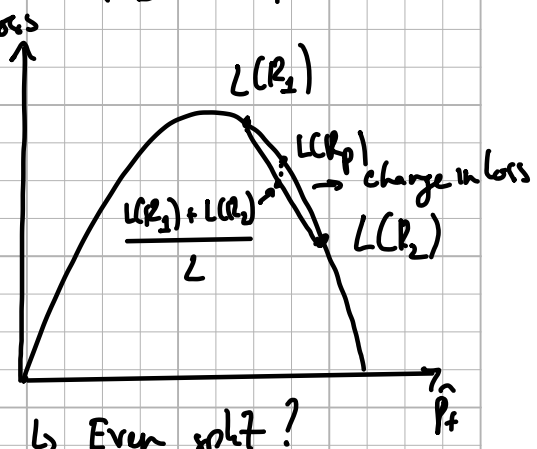
How to choose Splits?

Define $L(R)$: loss on R $\Rightarrow$ $L_{misclass} = 1 - \max_c \hat{p}_c$

Given C classes, define
$\hat{p}_c$ to be proportion of examples in R that are of class c

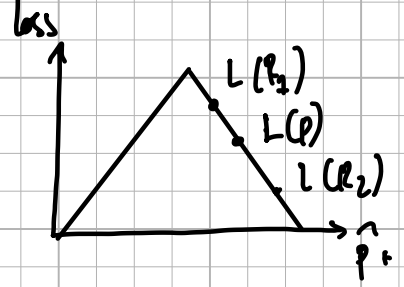
max $L(R_p) - (L(R_1) + L(R_2))$
just (parent loss, children loss)

$\rightarrow L(R_p) - \frac{|R_1| L(R_1) + |R_2| L(R_2)}{|R_1| + |R_2|}$



↳ Even split?

→ Misclassification



→ Cross-entropy

→ Gini
$\sum \hat{p}_c (1 - \hat{p}_c)$ $\Rightarrow$ same shape

$L_{cross} = -\sum_c \hat{p}_c \log_2 \hat{p}_c$ → proportion right

Reg trees R_m

→ Predict $\hat{y}_m = \frac{\sum_{i \in R_m} y_i}{|R_m|}$ $\Rightarrow$ $L_{squared} = \frac{\sum_{i \in R_m} (y_i - \hat{y}_m)^2}{|R_m|}$

Categorical Vars

↓

loc $\in \{N\}$

q categories $\Rightarrow$ 2^q possible splits

Regularization of DTs

1) min leaf size
2) max depth
3) max number of nodes
4) minimum decrease in loss
5) Pruning (misclassification with val set)

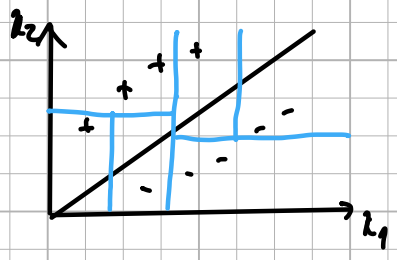
Runtime

n examples
f features
d depth of tree

Test time: $O(d)$, $d < \log_2 n$

Train time:
Each point is part of $O(d)$ nodes
Cost of point at each node is $O(f)$
So total cost $O(nfd)$
Data matrix is of size nf

No additive structure



Recap

⊕ Easy to explain
⊕ Interpretable
⊕ Categorical Vars
⊕ Fast

⊖ High variance
⊖ Bad at additive
⊖ Low predictive acc

Ensembling → Groups to one

Take X_i's which are random variables (RV)
that are independent identically distributed (iid)

$Var(X_i) = \sigma^2$ $\quad Var(\bar{X}) = Var\left(\frac{1}{n}\sum_i X_i\right) = \frac{\sigma^2}{n}$

Drop independence assumption
So now X_i's are i.d, X_i's correlated by ρ (rho)

$Var(\bar{X}) = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2$ ↓

Ways to ensemble

1/ different algos
2/ different train sets
3/ Bagging (Random Forest)
4/ Boosting (Adaboost, xgboost)

bagging - Bootstrap Aggregation

↳ to measure uncertainty of your est

Have a true Population P
Train Set $S \sim P$

Assume $P = S$

Bootstrap samples $Z \sim S$

$\downarrow$

$Z_1 \dots Z_M$

Train model G_m on Z_m

$$G(x) = \frac{\sum_{m=1}^{M} G_m(x)}{M}$$

Bias Variance Analysis

$$\text{Var}(\bar{x}) = \rho\sigma^2 + \frac{1-\rho}{M}\sigma^2$$

Bootstrapping is driving down ρ → has lower bound

More M → less variance

Bias slightly increase because of random subsampling

DTs + Bagging

DT are high variance, low bias

Ideal fit for bagging

Random forests

At each split, consider only a fraction of your total features

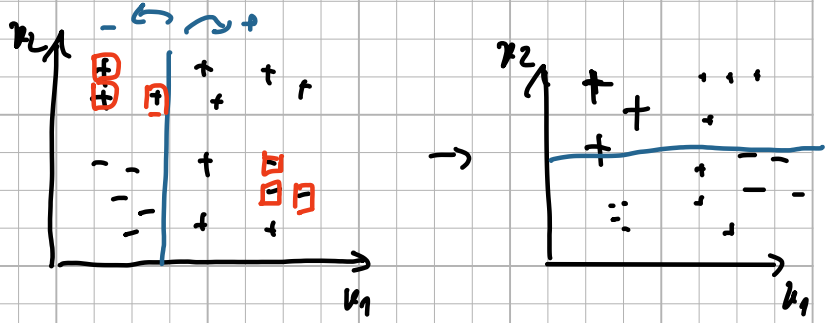
Decrease ρ

Decorrelate Models

Boosting

Increase bias

Additive



Determine for classifier G_m a weight α_m proportional $\log\left(\frac{1-err_m}{err_m}\right)$

$$G(x) = \sum_m \alpha_m G_m$$

each G trained on reweighted training set

Adaboost + Xgboost (Lecture Notes)