Project

- Do you have a project idea?
  - Yes, then prepare “pitch” to present in-class on Thursday 2/4.
  - No, then join one of the pitched ideas.

- How to prepare your pitch?
  - Make one slide following the template on the next slide (either PDF or Powerpoint)
  - Email me the slide by 10:00am on Thursday 2/4.
  - Give a 2-minute explanation why the project is
    • interesting,
    • significant,
    • relevant to the CS6784, and
    • feasible.

- Form project groups
  - Before 2/11, matchmaking in class or online
  - Jointly prepare project proposal

Paper Assignments

- Papers are on course homepage
- Bid on papers:
  - Deadline: Monday, Feb 1, 11:59pm
  - Bidding online
- First three papers:
  - Feb 11
  - Feb 16

Machine Learning Tasks Relevant for CS6784

- Supervised Learning
  - Data: (x,y) ~ iid P(X,Y)
  - x: Input
  - y: Label / output
  - Learn: h: X → Y

- Unsupervised Learning
  - Data: (x) ~ iid P(X)
  - x: Observation
  - Learn: structure of P(X)

- Reinforcement Learning
  - Data: Markov decision Process P(S|A,S'), P(R|S)
  - (s,a,r)*: Sequence of state/action/reward triples
  - Learn: policy π: S → A that maximizes reward

Supervised Learning

- Learning Task: P(X,Y) = P(X) P(Y|X)
  - Input Space: X (e.g. feature vectors, word sequence, etc.)
  - Output Space: Y (e.g. class label 1..k)
  - Training Data: S_{train} = {(x1,y1), ..., (x_n,y_n)} ~ iid P(X,Y)
  - Goal: Find h: X → Y with low prediction error Err_P(h)
Generalization Error and Sample Error

Definition: The prediction error/generalization error/true error/expected loss/risk $Err_P(h)$ of a hypothesis $h$ for a learning task $P(X,Y)$ is

$$Err_P(h) = \sum_{z \in X,y \in Y} \Delta(h(z), y)P(X = z, Y = y).$$

Definition: $\Delta(a, b)$ is a loss function that measures the cost of making a wrong prediction. A commonly used loss function is the 0/1-loss

$$\Delta(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{else} \end{cases}$$

Definition: The error on sample $S$ $Err_S(h)$ of a hypothesis $h$ is $Err_S(h) = \frac{1}{n} \sum_{i=1}^{n} \Delta(h(x_i), y_i).$

Classifying Examples

• Bayes’ Decision Rule: Optimal decision is

$$h(x) = \arg \max_{y} \sum_{y'} \Delta(h(x), y')P(Y = y | X = x).$$

• Equivalent Reformulation: For 0/1-Loss

$$h(x) = \arg \min_{y} \sum_{y'} \Delta(h(x), y')P(Y = y | X = x).$$

Generative vs. Discriminative Models

Learning Task:
- Generator: Generate descriptions according to distribution $P(X)$.
- Teacher: Assigns a value to each description based on $P(Y|X)$.

Training Examples

Discriminative Model
- Model $P(Y|X)$ with $P(Y|X,a)$
- Find $a$ e.g. via MLE
- Examples: Log. Reg., CRF
- Model discriminant functions
- Find $h \in H$ with low train loss (e.g. Emp. Risk Min.)
- Examples: SVM, Dec. Tree

Generative Model
- Model $P(X,Y)$ with distributions $P(Y|X,a)$
- Find $a$ that best matches $P(X,Y)$ on training data (e.g. MLE)
- Examples: Naive Bayes, HMM

Naïve Bayes’ Classifier (Multivariate)

• Input Space $X$: Feature Vector
• Output Space $Y$: {1,-1}
• Model:
  - Prior class probabilities
    $$P(Y = 1) \quad P(Y = -1)$$
  - Class conditional model (one for each class)
    $$P(X = x | Y = 1) = \prod_a P(X_a = x_a | Y = 1)$$
    $$P(X = x | Y = -1) = \prod_a P(X_a = x_a | Y = -1)$$
• Classification rule:
  $$h_{Naive}(x) = \arg \max_{y} \sum_a \left( P(Y = a) \prod_a P(X_a = x_a | Y = y) \right)$$

Estimating the Parameters of Naïve Bayes

• Count frequencies in training data
  - $n$: number of training examples
  - $n_a$: number of pos/neg examples
  - $x^a$: number of times feature $X^a$ takes value $x^a$ for examples in class $y$
  - $|X^a|$: number of values of attribute $X^a$
• Estimating $a$: \begin{align*}
\hat{\theta}_a &= \frac{\sum_{y} \prod_{x} P(x | y) + \alpha}{\sum_{y} \prod_{x} P(x | y) + \alpha |X^a|}, \\
\hat{\phi}_a &= \frac{\sum_{y} \prod_{x} x^a |y| + \alpha}{\sum_{y} \prod_{x} x^a |y| + \alpha |X^a|} \end{align*}
• $P(Y)$: Maximum Likelihood Estimate
• $P(X|Y)$: Maximum Likelihood Estimate
• $P(X|Y)$: Smoothing with Laplace estimate
Generative vs. Discriminative Models

Learning Task:
- Teacher: Assigns a value to each description based on distribution \( P(Y|X) \).

Training Examples \( (m_1, y_1), \ldots, (m_n, y_n) \sim P(Y|X) \)

Discriminative Model
- Model \( P(Y|X) \) with \( P(Y|X, \omega) \)
  - Find \( \omega \) e.g. via MLE
  - Examples: Log. Reg., CRF
  - Model discriminant functions
  - Find \( h \in H \) with low train loss (e.g. Emp. Risk Min.)
  - Examples: SVM, Dec. Tree

Generative Model
- Model \( P(X|Y) \) with distributions \( P(Y|X, \omega) \)
- Find \( \omega \) that best matches \( P(X|Y) \) on training data (e.g. MLE)
- Examples: naive Bayes, HMM

Bayes’ Decision Rule:
- General: \( \omega(y|x) = \sum_{\omega} P(\omega|x) y = y(x) \)
- Assume 0/1 Loss \( \Delta(y, y') = 1, \text{if } y \neq y', 0 \text{ else} \)
  \[ \omega(y|x) = \arg \min_{\omega} \{ P(y|X = x, \omega) \} \]

Generative vs. Discriminative Models

Learning Task:
- Teacher: Assigns a value to each description based on distribution \( P(Y|X) \).

Training Examples \( (m_1, y_1), \ldots, (m_n, y_n) \sim P(Y|X) \)

Discriminative Model
- Model \( P(Y|X) \) with \( P(Y|X, \omega) \)
  - Find \( \omega \) e.g. via MLE
  - Examples: Log. Reg., CRF
  - Model discriminant functions
  - Find \( h \in H \) with low train loss (e.g. Emp. Risk Min.)
  - Examples: SVM, Dec. Tree

Generative Model
- Model \( P(X|Y) \) with distributions \( P(Y|X, \omega) \)
- Find \( \omega \) that best matches \( P(X|Y) \) on training data (e.g. MLE)
- Examples: naive Bayes, HMM

Support Vector Machine

Training Examples: \( \{m_1, y_1\}, \ldots, \{m_n, y_n\} \) \( \in \mathbb{R}^T, y \in \{H_y - 1\} \)

Hypothesis Space: \( H = \{ h(x), y \in \mathbb{R}, h(x): \text{linear discriminant} \} \)

Training Loss: \( \text{Err}_{\text{train}}(h) = \sum_{i} \Delta(h(x_i), y_i) \leq \sum_{i} c_i \)

Optimization Problem:
\[ \min_{h, c_i} \frac{1}{2} \| \theta \|^2 + \sum_{i} c_i \]
\[ s.t. \quad \theta^T (\phi(x_i) + b_i) \geq 1 - c_i \]
\[ \theta^T (\phi(x_i) + b_i) \geq 1 - c_i \]

[Vapnik et al.]
Challenges

Problem: Exponentially many classes!

Approach: view as multi-class classification task

Hypothesis Space:

• How to do efficient inference with model (i.e. Crammer & Singer 02)

Training: Find $\omega$ that solve

$$
\min_{\omega} \sum_{i=1}^{m} \sum_{j \neq y_{i}} \omega_{j}^{T} \Phi(x_{i}) + \xi_{i} \quad \forall j \in M, \quad \omega_{j}^{T} \Phi(x_{i}) \geq d_{j}^{i} + 1 - \xi_{i}
$$

• Discriminative Model: $P(Y|X, \cdot)$

• Every complex output $y \in Y$ is one class

So what about Structured Outputs?

• Approach: view as multi-class classification task
  – Every complex output $y \in Y$ is one class

  $\begin{array}{c}
  \text{The bear chased the cat} \\
  \text{Det} \to \text{N} \to \text{V} \to \text{Det} \to \text{N}
  \end{array}$

• Problem: Exponentially many classes!
  – Generative Model: $P(X, Y|\cdot)$
  – Discriminative Model: $P(Y|X, \cdot)$
  – Discriminant functions: $h_{\omega}: X \times Y \to \mathbb{R}$

• Challenges
  – How to compactly represent model?
  – How to do efficient inference with model (i.e. $P(Y|X, \cdot)$)?
  – How to effectively estimate model from data? (e.g. compute $P(Y|X_{t}, \cdot)$)

Types of Learning Methods

• Generative: Joint Model
  – Probabilistic Minimization

• Discriminative
  – Empirical Risk Minimization