CS142: Machine Learning
Lecture 4: Generative Learning & Inference, Maximum Likelihood Estimation, Naïve Bayes

Instructor: Erik Sudderth
Brown University Computer Science
September 24, 2015

Figure credits:
PRML: Pattern Recognition & Machine Learning, Bishop 2007
ESL: Elements of Statistical Learning, Hastie & Tibshirani & Friedman 2009
Homework 1
Due tonight at 11:59pm!
Be sure to read the collaboration & grading policy document!

Electronic Submission of Homeworks
Submit Matlab code plus a single pdf with all other answers & plots.
Include your Banner ID (B01234567), not your name, in files.
To submit from outside the CIT you must setup ssh in advance!
CS142 ML: Lecture 4 Outline

- Generative models: learning & inference
- Maximum Likelihood (ML) parameter estimation
- ML estimation for the Naïve Bayes model
Directed Graphical Models

Each node in directed graph represents random variable:

\[ x = \{ x_s \mid s = 1, \ldots, N \} \]

Directed graphical model encodes factorization in edges:

\[
p(x) = \prod_{s=1}^{N} p(x_s \mid x_{\Gamma(s)})
\]

\[ \Gamma(s) \rightarrow \text{set of parents of node } s, \text{possibly empty} \]

Valid for any directed acyclic graph (DAG): equivalent to dropping conditional dependencies in chain rule

\[
p(x_{1:5}) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_2, x_3, x_4)
= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)p(x_5|x_3)
\]
**Terminology: Inference versus Learning**

- **Inference (test):** Given a model with known parameters, estimate the “hidden” variables for some data instance, or find their marginal distribution.

- **Learning (train):** Given multiple data instances, estimate parameters for a probabilistic model: *Maximum Likelihood, Bayesian estimators, …*

**Example: Expert systems for medical diagnosis**

- **Inference:** Given observed symptoms for a particular patient, infer probabilities that they have contracted various diseases.

- **Learning:** Given a database of many patient diagnoses, learn the relationships between diseases and symptoms.

**Example: Object detection in images**

- **Inference:** What object category instances are depicted in some image?

- **Learning:** Across a database of images, in which some example objects have been labeled, how do those labels relate to low-level image features?
**Classification:** Choose one of $K$ labels $t$ based on features $x$

$$
\mathcal{D} \rightarrow \text{Training Data} \quad \mathcal{D} = \{(x_1, t_1), \ldots, (x_N, t_N)\}
$$

**Models:** Finite set of $M$ candidates with tunable parameters

$$
\theta_m \rightarrow \text{parameters for model } m = 1, \ldots, M
$$

$$
p_m(x_n, t_n | \theta_m)
$$

**Example:** $M=2$ candidate models for 2 binary features $x_n = (x_{n1}, x_{n2})$

- $m=1$: $t_n \rightarrow x_{n1} \rightarrow x_{n2}$
- $m=2$: $t_n \rightarrow x_{n1} \rightarrow x_{n2}$

**Evaluation:** Loss for decisions made on

Cannot tune methods on this data!

Test Data
Probabilistic Machine Learning

**Classification:** Choose one of $K$ labels $t$ based on features $x$

$$\mathcal{D} \rightarrow \text{Training Data} \quad \mathcal{D} = \{ (x_1, t_1), \ldots, (x_N, t_N) \}$$

**Models:** Finite set of $M$ candidates with tunable parameters

$$\theta_m \rightarrow \text{parameters for model } m = 1, \ldots, M \quad p_m(x_n, t_n \mid \theta_m)$$

**Learning:** Estimate parameters of each model from training data. **Coming next!!!**

- **Maximum Likelihood (ML):** Find model that maximizes likelihood of data
- **Bayesian:** Find model with large posterior probability given data

**Model Selection:** Which model fits training data best?

- **Invalid:** Pick model which performs best on test data. Overfitting!
- **Invalid:** Pick model where learned parameters give highest likelihood. Overfitting!
- **Consistent:** Validation or cross-validation by splitting training data.

**Evaluation:** Loss for decisions made on

Cannot tune methods on this data!

Test Data
Probabilistic Machine Learning

**Classification:** Choose one of $K$ labels $t$ based on features $x$

\[ \mathcal{D} \rightarrow \text{Training Data} \quad \mathcal{D} = \{(x_1, t_1), \ldots, (x_N, t_N)\} \]

**Models:** Finite set of $M$ candidates with tunable parameters

\[ \theta_m \rightarrow \text{parameters for model } m = 1, \ldots, M \quad p_m(x_n, t_n \mid \theta_m) \]

**Learning:** Estimate parameters of each model from training data. **Coming next!!!**

- **Maximum Likelihood (ML):** Find model that maximizes likelihood of data
- **Bayesian:** Find model with large posterior probability given data

**Model Selection:** Which model fits training data best?

- **Consistent:** Validation or cross-validation by splitting training data.

**Inference:** Analyze test examples with best learned model $m$. **Use decision theory!**

\[ p(t_{N+1} \mid x_{N+1}, \hat{\theta}_m) \propto p(t_{N+1} \mid \hat{\theta}_m)p(x_{N+1} \mid t_{N+1}, \hat{\theta}_m) \]

**Evaluation:** Loss for decisions made on

*Cannot tune methods on this data!*
Graphical Models for Supervised Learning

Convention: Shaded nodes are observed.

TRAINING DATA:

\[ p(t, x) = \prod_{n=1}^{N} p(t_n)p(x_n \mid t_n) \]

- Labels & features observed
- Learning: Must estimate good numeric probability models

TEST DATA:

\[ p(t \mid x = \bar{x}) = \frac{p(\bar{x} \mid t)p(t)}{p(\bar{x})} \]

- Probabilities fixed from training
- Inference: Given novel features, apply Bayes rule & decision theory

Labels: \( t_n \in \{0, 1\} \)

or \( t_n \in \{0, \ldots, K - 1\} \)

or \( t_n \in \mathbb{R} \)

Features: \( x_n \in \mathbb{R}^d \)
Graphical Shorthand: Plate Notation

Convention: Shaded nodes are observed.

TRAINING DATA:

\[
p(t, x) = \prod_{n=1}^{N} p(t_n) p(x_n | t_n)
\]

The rectangular plate is defined to replicate the enclosed variables, giving a set of (conditionally) independent variables with the same distribution.
Learning: Parameter Estimation

**Training Data:**

- Prior model: \( p(t \mid \mu) \)
- Likelihood model: \( p(x \mid t, \theta) \)

\[
p(t \mid x = \bar{x}, \mu = \hat{\mu}, \theta = \hat{\theta}) = \frac{p(\bar{x} \mid t, \hat{\theta})p(t \mid \hat{\mu})}{p(\bar{x} \mid \hat{\theta}, \hat{\mu})}
\]

\[
p(t, x \mid \mu, \theta) = \prod_{n=1}^{N} p(t_n \mid \mu)p(x_n \mid t_n, \theta)
\]

**Test Data:**

- Parameter estimates: \( \hat{\mu}, \hat{\theta} \)
Generative models: learning & inference
Maximum Likelihood (ML) parameter estimation
ML estimation for the Naïve Bayes model
Frequentist Parameter Estimation

- Training data has $N$ observations: $x = \{x_1, x_2, \ldots, x_N\}$
- Assume independent, identically distributed (i.i.d.) samples from an unknown member of some family of distributions:

$$p(X = x \mid \theta), \quad \theta \in \mathbb{R}^d$$

Example: Data is Gaussian, $\theta$ is unknown mean

- Think of parameter vector $\theta$ as fixed, non-random constant
- An estimator guesses (learns) parameters from data

$$\hat{\theta}_N(x) = f(x_1, x_2, \ldots, x_N), \quad \hat{\theta}_N(x) \in \mathbb{R}^d$$

Example: $\theta$ is unknown mean, and

$$\hat{\theta}_N(x) = \frac{1}{N} \sum_{i=1}^{N} x_i$$

- GOAL: Estimators with good properties, uniformly across $\theta$
- Analyze estimators via expected performance on hypothetical data (many trials of drawing $N$ i.i.d. samples)
Properties of Estimators

\[ p(x \mid \theta) = \prod_{i=1}^{N} p(x_i \mid \theta) \]

\[ \hat{\theta}_N(x) = f(x_1, x_2, \ldots, x_N), \quad \hat{\theta}_N(x) \in \mathbb{R}^d \]

- **Bias:**
  \[ \mathbb{E}[\hat{\theta}_N(X)] - \theta = \int_{\mathcal{X}^N} \hat{\theta}(x_1, \ldots, x_N)p(x \mid \theta) \, dx - \theta \]

- For an **unbiased** estimator, the bias equals zero for any true \( \theta \)

- An **asymptotically unbiased** estimator has decreasing bias:
  \[ \|\mathbb{E}[\hat{\theta}_N(X)] - \theta\| \to 0 \quad \text{as} \quad N \to \infty \]

- A **consistent** estimator **converges in probability** to the true \( \theta \)
  \[ \lim_{N \to \infty} \Pr(\|\mathbb{E}[\hat{\theta}_N(X)] - \theta\| \geq \epsilon) = 0 \quad \text{for any} \quad \epsilon > 0 \]

- Would like a small **mean squared error**:
  \[ \mathbb{E}[(\hat{\theta}_N(X) - \theta)^2] = \mathbb{V}[\hat{\theta}_N(X)] + (\mathbb{E}[\hat{\theta}_N(X)] - \theta)^2 \]
Natural Logarithms and Exponentials

\[ e^x = \exp(x) : \mathbb{R} \rightarrow \mathbb{R}^+ \]
\[ \ln(x) = \log(x) : \mathbb{R}^+ \rightarrow \mathbb{R} \]

\[ \exp(\log(x)) = x \]
\[ \log(xyz) = \log(x) + \log(y) + \log(z) \]
\[ \log(x^a) = a \log(x) \]
Maximum Likelihood (ML) Estimators

\[ \hat{\theta}_N(x) = \arg \max_{\theta} \prod_{i=1}^{N} p(x_i \mid \theta) = \arg \max_{\theta} \sum_{i=1}^{N} \log p(x_i \mid \theta) \]

**Maximum Likelihood (ML)** is a recipe for building estimators by solving optimization problems. Given a regular (smooth) density, ML estimators always have good asymptotic properties:

- ML estimator is **consistent** and **asymptotically unbiased**.  *It may be biased for finite N.*
- It satisfies a **central limit theorem**, so that \((\hat{\theta}_N(X) - \theta)\) is asymptotically Gaussian (normal), with zero mean and standard deviation proportional to \(N^{-0.5}\).
- The estimator is asymptotically **efficient**: The error variance (and thus mean squared error) are no larger than any other possible estimator (meets **Cramer-Rao bound**).
Bernoulli Distribution: Single toss of a (possibly biased) coin

\[ \text{Ber}(x \mid \theta) = \theta^x (1 - \theta)^{1-x}, \quad 0 \leq \theta \leq 1 \]
\[ x \in \{0, 1\} \]

- Observe \(N\) samples from Bernoulli with unknown mean:

\[ X_i \sim \text{Ber}(\theta), \ i = 1, \ldots, N \]

- What is the maximum likelihood (ML) parameter estimate?

\[ p(x_1, \ldots, x_N \mid \theta) = \theta^{N_1} (1 - \theta)^{N_0} \]
\[ N_1 = \sum_{i=1}^{N} x_i \]
\[ N_0 = \sum_{i=1}^{N} (1 - x_i) = N - N_1 \]
\[ \hat{\theta} = \arg \max_{\theta} \log p(x \mid \theta) = \frac{N_1}{N} \]
**Learning Categorical Distributions**

**Categorical Distribution:** Single roll of a (possibly biased) die

\[ \text{Cat}(x \mid \theta) = \prod_{k=1}^{K} \theta_{k}^{x_{k}} \]

\[ x = \{0, 1\}^{K}, \sum_{k=1}^{K} x_{k} = 1 \]

\[ 0 \leq \theta_{k} \leq 1, \sum_{k=1}^{K} \theta_{k} = 1 \]

\[ \theta_{k} = \mathbb{E}[x_{k}] \]
A Constrained Optimization Lemma

Objective:

\[ \hat{\theta} = \arg \max_{\theta} \sum_{k=1}^{K} a_k \log \theta_k \]

subject to

\[ \sum_{k=1}^{K} \theta_k = 1 \]

\[ 0 \leq \theta_k \leq 1 \]

Solution:

\[ \hat{\theta}_k = \frac{a_k}{a_0} \]

\[ a_0 = \sum_{k=1}^{K} a_k \]

Proof for \( K=2 \): Change of variables to unconstrained problem

Proof for general \( K \): Lagrange multipliers

\[ a_k \geq 0 \]
Learning Categorical Distributions

Categorical Distribution: Single roll of a (possibly biased) die

\[ \text{Cat}(x_i \mid \theta) = \prod_{k=1}^{K} \theta_{k}^{x_{ik}} \quad x_i \in \{0, 1\}^K, \sum_{k=1}^{K} x_{ik} = 1 \]

- If we have \( N_k \) observations of outcome \( k \) in \( N \) trials:

\[ p(x_1, \ldots, x_N \mid \theta) = \prod_{k=1}^{K} \theta_{k}^{N_k} \quad N_k = \sum_{i=1}^{N} x_{ik} \]

\[ \log p(x_1, \ldots, x_N \mid \theta) = \sum_{k=1}^{K} N_k \log \theta_{k} \]

- From the previous optimization lemma:

\[ \hat{\theta} = \arg \max_{\theta} \log p(x \mid \theta) \quad \hat{\theta}_k = \frac{N_k}{N} \]

- Will this produce sensible predictions when \( K \) is large or \( N \) is small?
Generative models: learning & inference
Maximum Likelihood (ML) parameter estimation
ML estimation for the Naïve Bayes model
ML Learning for Classification

- Joint distribution of training data given parameters:

\[
p(t, x | \mu, \theta) = \prod_{n=1}^{N} p(t_n | \mu) p(x_n | t_n, \theta)
\]

\[
\log p(t, x | \mu, \theta) = \sum_{n=1}^{N} \left[ \log p(t_n | \mu) + \log p(x_n | t_n, \theta) \right]
\]

- The maximum likelihood learning objective is decomposable:

\[
(\hat{\mu}, \hat{\theta}) = \arg \max_{\mu, \theta} \log p(t, x | \mu, \theta)
\]

\[
\hat{\mu} = \arg \max_{\mu} \sum_{n=1}^{N} \log p(t_n | \mu) \\
\hat{\theta} = \arg \max_{\theta} \sum_{n=1}^{N} \log p(x_n | t_n, \theta)
\]
**ML Learning: Class Priors**

\[ \hat{\mu} = \arg \max_{\mu} \sum_{n=1}^{N} \log p(t_n \mid \mu) \]

- **Binary classification problems** \((K=2)\): \(t_n \in \{0, 1\}\)
  \[ p(t_n \mid \mu) = \mu^{t_n} (1 - \mu)^{1-t_n} \quad 0 \leq \mu \leq 1 \]
  \[ \hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} t_n \]

- **General classification problems** \((K>2)\): \(t_{nk} = 1\) if training data \(n\) is of class \(k\), otherwise \(t_{nk} = 0\)
  \[ p(t_n \mid \mu) = \prod_{k=1}^{K} \mu_{k}^{t_{nk}} \]
  \[ \hat{\mu}_k = \frac{1}{N} \sum_{n=1}^{N} t_{nk} \]

*Estimates are normalized counts of training frequencies!*
ML Learning: Class Likelihoods

\[ t_{nk} = 1 \text{ if training data } n \text{ is of class } k, \text{ otherwise } t_{nk} = 0 \]

\[ \hat{\theta} = \arg\max_{\theta} \sum_{n=1}^{N} \log p(x_n \mid t_n, \theta) \]

Assume likelihoods take the following form:

\[ p(x_n \mid t_{nk} = 1, \theta) = f(x_n \mid \theta_k) \]

\[ f(x \mid \theta) \to \text{some family of probability distributions: Categorical, Gaussian, ...} \]

\[ \theta_k \to \text{parameters for class } k \]

\[ \theta = \{\theta_1, \ldots, \theta_K\} \]

ML learning then decomposes across different classes:

\[ \hat{\theta} = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \log f(x_n \mid \theta_k) \]

\[ \hat{\theta}_k = \arg\max_{\theta_k} \sum_{n \mid t_{nk} = 1} \log f(x_n \mid \theta_k) \]

Divide features into \( K \) class-specific datasets, learn independently from each!
The Naïve Bayes Assumption

$t_{nk} = 1$ if training data $n$ is of class $k$, otherwise $t_{nk} = 0$

- Suppose we measure a large number $D$ of features:
  \[ x_n = \{ x_{n1}, x_{n2}, \ldots, x_{nD} \} \]

- Learning their relationships could be hard! A simple, naïve approximation is to assume they are conditionally independent given the class label:
  \[
  p(x_n \mid t_{nk} = 1, \theta) = \prod_{d=1}^{D} f(x_{nd} \mid \theta_{kd})
  \]
  \[
  \theta_{kd} \rightarrow \text{parameters for feature } d \text{ of class } k
  \]

- ML learning then decomposes across both features and classes:
  \[
  \hat{\theta} = \arg \max_{\theta} \sum_{n=1}^{N} \sum_{d=1}^{D} \sum_{k=1}^{K} t_{nk} \log f(x_{nd} \mid \theta_{kd})
  \]
  \[
  \hat{\theta}_{kd} = \arg \max_{\theta_{kd}} \sum_{n \mid t_{nk} = 1} \log f(x_{nd} \mid \theta_{kd})
  \]
Naïve Bayes for Binary Features

\( N \) digits, \( D \) pixels, \( x_{nd} \in \{0, 1\} \)

\( K = 10 \)

\( N \) emails, \( D \) words, \( x_{nd} \in \{0, 1\} \)

\( K = 2 \)

\( t_{nk} = 1 \) if training data \( n \) is of class \( k \), otherwise \( t_{nk} = 0 \)

\( \theta_{kd} = p(x_{nd} = 1 \mid t_{nk} = 1) \)

- The maximum likelihood objective is then:

\[
\hat{\theta}_{kd} = \arg \max_{\theta_{kd}} \sum_{n \mid t_{nk} = 1} \log f(x_{nd} \mid \theta_{kd}) = \frac{1}{N_k} \sum_{n=1}^{N} y_{nk} x_{nd}
\]

\[
N_k = \sum_{n=1}^{N} y_{nk}
\]