CS 165B: Machine Learning
Max-Likelihood Principle, Naïve Bayes Model

CS 165B

Nov 7, 2023

Instructor: Prof. Yu-Xiang Wang
Announcement

• Get your midterm after the lecture today!
  • 25 Percentile, Median, 75 Percentile: 80.5, 92.5, 96.5

• My office hour happens today. I will teach from remotely on Thursday.
MP3: Predict Credit Card Approval

• Still binary classification but more open ended
  • you can use off-the-shelf tool “sklearn” for feature processing, defining and training classifiers.

• You will compete in a leaderboard. Top 10 students (in the final test result) get a bonus.

• During competition, leaderboard shows your result on public test set (your “dev” set). After the deadline, we will reveal results on a clean private data.
Last lecture

• Max-margin linear separator --- hinge loss + L2 regularization
  • Also known as the notorious “support vector machines”

• Probability and statistics review
  • A fundamental problem of statistics: estimation!
  • How to estimate quantities about the population using sampled data?

  • Example: Polls.
Today

- Probabilistic models and max-likelihood estimation
- Derive square loss and logistic loss
- Naïve Bayes models
Recap: Bernoulli Distribution \( X \sim \text{Ber}(p) \)

\[
P(X = x) = \begin{cases} 
    p & \text{if } x = 1, \\
    1 - p & \text{if } x = 0.
\end{cases}
\]
Recap: Gaussian distribution \( X \sim \mathcal{N}(\mu, \sigma^2) \)

- Probability density:
  \[
  f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right)
  \]
Maximum likelihood estimation

- Used since Gauss, Laplace, etc…. Popularized / carefully analyzed by Ronald Fisher.

- Which distribution is more *likely* to have produced the data?

\[
\max_{\mathcal{P} \in \Pi} \mathcal{P}(\text{Data})
\]

- Observation 1: If the data is i.i.d. the by independence the density factorizes

\[
f_{\mathcal{P}}(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_{\mathcal{P}}(x_i)
\]

- Observation 2: Taking log does not change the solution.

\[
\text{arg max likelihood} = \text{arg max log likelihood} = \text{arg min \text{ - log likelihood}}
\]

\[
\log f_{\mathcal{P}}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \log f_{\mathcal{P}}(x_i)
\]
What is the difference between probability and likelihood?

- $P(\text{Data}; \text{Parameter})$
  - If it is a function of the data, then it’s probability.
  - If it is a function of the parameter while the data is fixed, then it is likelihood.
What do we know about MLE? I won’t prove these facts for you but... it’s useful to know them.

- The error in parametric estimation goes to 0
  \[ \mathbb{E}\| \hat{\theta}_{MLE} - \theta^* \|_2 = O(1/\sqrt{n}) \]
  - MLE is equivalent to minimizing the KL-divergence
    \[ \min_{P \in \Pi} D_{KL}(P^*(\text{Data}) \| P(\text{Data})) \]
    - It is asymptotically (as \( n \to \infty \)) the most sample-efficient estimator (in almost every statistical estimation problem... )
Exercise: Estimating Bernoulli Mean using MLE

• Data: \( X_1, \ldots, X_n \overset{i.i.d.}{\sim} \text{Bernoulli}(p) \)
Exercise: Estimating Bernoulli Mean using MLE

• Data: \[ X_1, \ldots, X_n \overset{i.i.d.}{\sim} \text{Bernoulli}(p) \]

• Likelihood: \[ \mathcal{L}(X_i; p) = p^{X_i} (1 - p)^{1-X_i} \]
Exercise: Estimating Bernoulli Mean using MLE

- **Data:** \( X_1, \ldots, X_n \) i.i.d. \( \sim \) Bernoulli\( (p) \)

- **Likelihood:** \( \mathcal{L}(X_i; p) = p^{X_i} (1 - p)^{1 - X_i} \)

- **The MLE problem:**
  \[
  \hat{p} = \arg \max_{p \in [0,1]} p^{\sum_{i} X_i} (1 - p)^{n - \sum_{i} X_i}
  \]

  Take log,
  \[
  \hat{p} = \arg \max_{p \in [0,1]} \left( \sum_{i} X_i \right) \log p + \left( n - \sum_{i} X_i \right) \log (1 - p)
  \]

  \[
  \frac{\sum_{i} X_i}{n} = \frac{1}{p} + \frac{1}{1-p} (-1) = 0
  \]

  \[
  (1-p) \cdot \sum_{i} X_i + p \cdot (n - \sum_{i} X_i) \cdot (-1) = 0
  \]

  \[
  \sum_{i} X_i - \frac{\sum_{i} X_i}{n} \cdot p \cdot n \cdot p + \sum_{i} X_i \cdot p = 0
  \]

  \[
  p = \frac{\sum_{i} X_i}{n}
  \]
Exercise: Estimating the mean parameter of a Gaussian distribution

• Data \( X_1, \ldots, X_n \) i.i.d. \( \sim \mathcal{N}(\mu, \sigma^2) \)

• Likelihood:

\[
f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \frac{x - \mu}{\sigma} \right)^2 \right)
\]

• The MLE problem:

\[
\hat{\mu} = \arg \max_{\mu} \prod_{i=1}^{n} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}
\]

\[
= \frac{1}{\sigma \sqrt{2\pi}} \exp \left( \sum_{i=1}^{n} \log \frac{1}{\sigma \sqrt{2\pi}} - \frac{(x_i - \mu)^2}{2\sigma^2} \right)
\]

\[
= \frac{1}{\sigma \sqrt{2\pi}} \exp \left( \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{2\sigma^2} \right)
\]

\[
f'(\mu) = -\frac{\sum_{i=1}^{n} 2(x_i - \mu)}{2\sigma^2} = -\text{var} \mu + \frac{\sum_{i=1}^{n} x_i}{\sigma^2} = 0
\]

\[\mu = \frac{\sum_{i=1}^{n} x_i}{n}\]
Exercise: Linear regression

\[ y_i \sim \mathcal{N}(x_i^T \theta^*, \sigma^2) \]

• P(y|x) is modeled by “Linear Gaussian model”

\[ y_i = x_i^T \theta^* + \epsilon_i \quad \text{where} \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d.} \]

• Data: \((x_1, y_1), \ldots, (x_n, y_n)\)

• Work out the optimization problem to solve for the MLE for \(\theta^*\).
Exercise: Logistic regression

- \( P(y|x) \) is modeled by a Logit model

\[
\begin{align*}
  y &\sim \text{Bernoulli}(\text{Sigmoid}(x^T \theta^*)) \\
  \text{where } \text{Sigmoid}(t) &= \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}
\end{align*}
\]

- Data: \((x_1, y_1), \ldots, (x_n, y_n)\)

- Work out the optimization problem to solve for the MLE for \( \theta^* \).
After we fit the MLE, how to make predictions? The idea is to just “Plug-In”

• For classification problems

\[ h^*(x) = \max_y p_\theta(y|x) \]

Bayes Optimal

• For regression problems

\[ h^*(x) = \mathbb{E}_\theta [y|x] \]
After we fit the MLE, how to make predictions? The idea is to just “Plug-In”

- For classification problems
  \[ h^*(x) = \max_y \hat{p}_\theta(y|x) \]

- For regression problems
  \[ h^*(x) = \mathbb{E}_\theta[y|x] \]
After we fit the MLE, how to make predictions? The idea is to just “Plug-In”

• For classification problems

\[
h^*(x) = \max_y p_{\theta}(y|x) \quad \Rightarrow \quad \hat{h}(x) = \max_y p_{\hat{\theta}}(y|x)
\]

• For regression problems

\[
h^*(x) = \mathbb{E}_\theta[y|x] \quad \Rightarrow \quad \hat{h}(x) = \mathbb{E}_{\hat{\theta}}[y|x]
\]
Checkpoint

• We have shown that

  • Square loss minimization is MLE under a Gaussian noise model

  • Logistic loss minimization is MLE under a Bernoulli model for binary classification.
Recap: We learned about directly modelling the predictive functions. There is another way… called “Probabilistic modelling”

• We can model how the data is generated in the first place.

  • Model the labeling process via a conditional distribution \( P(y | x) \). This is known as a *(probabilistic) discriminative model.*
    • Specifying decision-trees / linear classifiers / shapes of decision boundaries should be considered non-probabilistic discriminative models.

  • Model the **joint distribution** \( P(x,y) \). Often one models the label distribution \( P(y) \) and a generative process \( P(x|y) \). This is known as a *generative model.*

• The natural prediction would be
  • \( h(x) = \text{argmax}_y P(y | x) \)
  • If the data generative process is indeed \( P(y|x) \), then this is “Bayes optimal”.
Discriminative models vs generative models

Image Credit: Dr. Roi Yehoshua
Generative model builds a world. How do we model the joint distribution \( p(x,y) \)?

- Prior: \( p(y) \)

- Per-class generative distribution \( p(x|y) \)

- Then how do we make inference about the label given feature?
  - By Bayes rule

- How to we determine \( p(y) \) and \( p(x|y) \)?
  - Fit them using data by MLE!
Modeling $p(x|y)$ is challenging

Consider a dataset with 16 attributes (let's assume they are all binary).
How many parameters to we need to estimate to fully determine $p(X|Y)$?

Learning the values for the full conditional probability table would require
enormous amounts of data.
Simple example

1. What is the number of parameters required to determine $p(X|y)$?

2. What happens if there are $d$ Boolean features?
Naïve Bayes assumption

- Naïve Bayes classifiers assume that given the class label $Y$ the features are *conditional independent* of each other

$$p(X|y) = \prod_j p_j(x^j|Y)$$

- $p_j$: specific model for attribute $j$. 
Simple example with naïve Bayes assumption

1. What is the number of parameters required to determine $p(X|y)$?

2. What happens if there are $d$ Boolean features?
Example: Text classification

Machine learning is a subset of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. In other words, it's a process of data analysis that automates analytical model building. Machine learning involves the creation and use of algorithms that can learn from and make decisions or predictions based on data...

X is a sequence of words
y is “Human” or “Machine”

Naïve Bayes Assumption:
P(X | “Machine”) = P(Word 1 | “Machine”) \cdots P(Word N | “machine”)
MLE for fitting the Naïve Bayes Model for text classification

All machine generated texts. 
(m documents
ith document with \(N_i\)
words.)

\[
\hat{p}(y = \text{“machine”}) = \frac{m}{m + n}
\]

\[
\hat{p}(\text{Word} = w | y = \text{“machine”}) = \frac{\# \text{ of } w}{\sum_{i=1}^{m} N_i}
\]

All human written text. 
(n documents. jth
document with \(N_j\) words.)

\[
\hat{p}(y = \text{“human”}) = \frac{n}{m + n}
\]

\[
\hat{p}(\text{Word} = w | y = \text{“human”}) = \frac{\# \text{ of } w}{\sum_{j=1}^{n} N_j}
\]
2 min exercise: Naïve Bayes Model for text classification

Sentence 1: “Why do you cry?”
Sentence 2: “Yes.”
Sentence 3: “Hasta La Vista, Baby!”
Sentence 4: “I will be back.”

Sentence 1: “You mean people?”
Sentence 2 “I don't know. We just cry.”
Sentence 3: “No, no, no, no. You gotta listen to the way people talk.”

• What is our estimate of P(y)?

• What is our estimate for P(word = “No” | y) and P(word = “You” | y)?
Prediction with Naïve Bayes Model: “Just plug in”

Once we computed all parameters for attributes in both classes we can easily decide on the label of a new sample $x$:

$$\hat{y} = \arg\max_{y_k} p(y = y_k | x)$$

$$= \arg\max_{y_k} \frac{p(x|y = y_k)p(y = y_k)}{p(x)}$$

$$= \arg\max_{y_k} \prod_j p(x^j|y = y_k)p(y = y_k)$$

Try it on this example:
New sentence: “I know now why you cry”

- Perform this computation for both class 1 and class 2 and select the class that leads to a higher probability as your decision
Naïve Bayes model with continuous variables

- So far we assumed a binomial or discrete distribution for the data given the model ($p(x^i|y)$)
- However, in many cases the data contains continuous features:
  - Height, weight, Levels of genes in cells, Brain activity

- Gaussian Naïve Bayes model:

$$x_i | y \sim \mathcal{N}(\mu_y, \sigma^2_y)$$

You will learn more about Gaussian Naïve Bayes models in HW3!
Next Lecture

• Error decomposition

• Decision Tree and Boosting

• Gradient Boosting