Generative and Discriminative Models

CSci 5525: Advanced Machine Learning

Instructor: Nicholas Johnson

September 14, 2023

Announcements

- HW1 posted (due next Tue 9/19)
- Project proposals due in 1 week (9/21)

Project Proposal

- Must be done individually
- Must include programming component
- No more than 1 page and should include:
  - High-level problem/motivation
  - Why use machine learning?
  - Initial solution idea/approach
  - What data/tools/resources are needed
- Submit via Canvas

Problem

Suppose you work at a fruit company and you want to design a system which can determine whether a piece of fruit is good or bad. Let's say you have data from the past month which consists of the mass and label such as 'good' or 'bad' for each piece of fruit. For example:

Mass (g)	Label
70.2	Good
93.2	Good
40.9	Bad
82.3	Good
68.1	Bad
87.6	Bad
96.8	Good

How would you design the system?

Classification

- Dataset: $\mathcal{D} = \{(\text{Mass}_i, \text{Good}/\text{Bad}_i)\}_{i=1}^{n} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
- $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$, $y \in \mathcal{Y}$ (discrete set)
- Mostly focus on binary classification $\mathcal{Y} = \{0, 1\}$
- Goal: find prediction function $f : \mathcal{X} \to \mathcal{Y}$

Generative Models and Bayes Rule

- Bayes rule: $p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$
- For 2-class problem, posterior probability for C_1

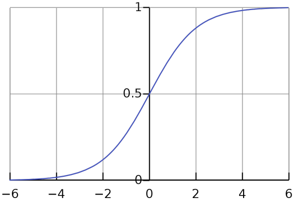
$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} = \frac{\exp(a)}{\exp(a)+1}$$

- Here a is the log-odds ratio:

$$a = \log\left(\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}\right)$$

- The class posterior can be written as

$$p(C_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a) \qquad \text{(Logistic Function)}$$



Continuous Inputs: Multi-variate Gaussians

- Estimate $p(C_j)$ (prior) and $p(\mathbf{x}|C_j)$ (conditional) for $j = 1, 2$
- Assume class conditionals are Gaussian: different μ_j, same Σ

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \mu_k)^\top \Sigma^{-1} (\mathbf{x} - \mu_k) \right\}$$

- Class labels: $y_i \in \{0, 1\}$, classes C_1, C_2:

$$\mathbf{x}_i \in C_1 \Rightarrow y_i = 1, \qquad \mathbf{x}_i \in C_2 \Rightarrow y_i = 0$$

- Class priors: $P(y_i = 1) = \pi, P(y_i = 0) = 1 - \pi$
- Likelihood of one data point $(y_i, \mathbf{x}_i)$

$$\begin{aligned} p(y_i, \mathbf{x}_i) &= p(y_i) p(\mathbf{x}_i | y_i) \\ &= \left\{ \pi p(\mathbf{x}_i | \mu_1, \Sigma) \right\}^{y_i} \left\{ (1 - \pi) p(\mathbf{x}_i | \mu_2, \Sigma) \right\}^{1 - y_i} \end{aligned}$$

Continuous Inputs: Multi-variate Gaussians (cont.)

- Likelihood of the data $\mathcal{D}$, assuming independence

$$
\begin{aligned}
p(\mathcal{D}|\pi,\mu_1,\mu_2,\Sigma) &= p((y_1,\mathbf{x}_1),\ldots,(y_n,\mathbf{x}_n)|\pi,\mu_1,\mu_2,\Sigma) \\
&= \prod_{i=1}^{n} p(y_i,\mathbf{x}_i|\pi,\mu_1,\mu_2,\Sigma) \\
&= \prod_{i=1}^{n} \left\{\pi p(\mathbf{x}_i|\mu_1,\Sigma)\right\}^{y_i} \left\{(1-\pi)p(\mathbf{x}_i|\mu_2,\Sigma)\right\}^{1-y_i}
\end{aligned}
$$

- Estimate parameters by maximizing log-likelihood

$$
\begin{aligned}
&\log p(\mathcal{D}|\pi,\mu_1,\mu_2,\Sigma) \\
&\quad = \sum_{i=1}^{n} \left\{ y_i \log(\pi p(\mathbf{x}_i|\mu_1,\Sigma)) + (1-y_i)\log((1-\pi)p(\mathbf{x}_i|\mu_2,\Sigma)) \right\}
\end{aligned}
$$

Maximum Likelihood Estimation

- Log-likelihood of the data

$$\log p(\mathcal{D}|\pi, \mu_1, \mu_2, \Sigma)$$
$$= \sum_{i=1}^{n} \left\{ y_i \log(\pi p(\mathbf{x}_i|\mu_1, \Sigma)) + (1 - y_i) \log((1 - \pi) p(\mathbf{x}_i|\mu_2, \Sigma)) \right\}$$

- Optimizing over the parameters $(\pi, \{\mu_1, \mu_2\}, \Sigma)$

$$\pi = \frac{1}{n} \sum_{i=1}^{n} y_i = \frac{n_1}{n_1 + n_2}$$

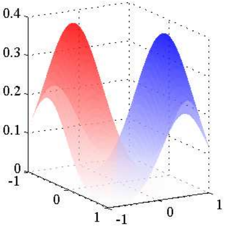
$$\mu_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i \ , k = 1, 2$$

$$\Sigma = \sum_{k=1}^{2} \frac{n_k}{n} \left(\frac{1}{n_k} \sum_{\mathbf{x} \in C_k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top \right)$$

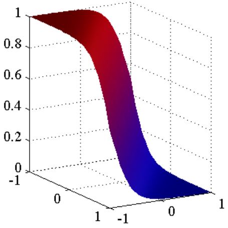
Prediction: 2-class problems

- For 2-class problem:

$$
\begin{aligned}
p(C_1|\mathbf{x}) &= \sigma(\mathbf{w}^\top \mathbf{x} + w_0) \\
\mathbf{w} &= \Sigma^{-1}(\mu_1 - \mu_2) \\
w_0 &= -\frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1}\mu_2 + \log \frac{p(C_1)}{p(C_2)}
\end{aligned}
$$



Class conditionals



Class posteriors

Generative Models and Bayes Rule: K-class

- Recall that $p(\mathbf{x}) = \sum_{j=1}^{K} p(\mathbf{x}, C_j) = \sum_{j=1}^{K} p(C_j)p(\mathbf{x}|C_j)$
- For K-class problem, posterior probability for C_k

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_{j=1}^{K} p(\mathbf{x}|C_j)p(C_j)} = \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)}$$

- Here, a_k is given by

$$a_k = \log p(\mathbf{x}|C_k)p(C_k) = \log p(\mathbf{x}|C_k) + \log p(C_k)$$

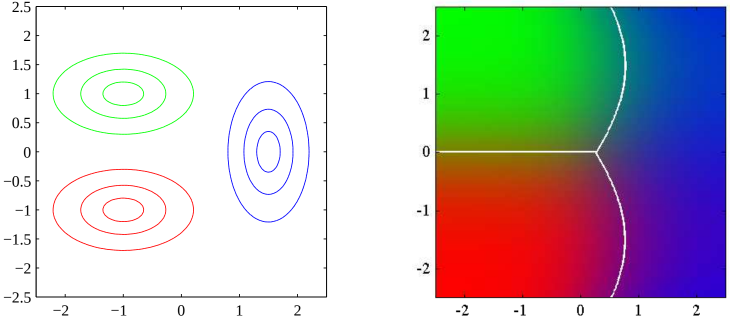
- Estimate $p(C_j)$ (prior) and $p(\mathbf{x}|C_j)$ (conditional) for $j = 1, \ldots, K$
- Make "parametric" assumptions about the conditional $p(\mathbf{x}|C_j)$

Prediction: K-class problems

- For K-class problem

$$a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$
$$\mathbf{w}_k = \Sigma^{-1}\mu_k$$
$$w_{k0} = -\frac{1}{2}\mu_k^T \Sigma^{-1}\mu_k + \log p(C_k)$$

- If Σ is the same for each class: Linear Discriminant
- If Σ is not the same for each class: Quadratic Discriminant



Naive Bayes: Conditional Independence of Features

- Generative models need to specify $p(\mathbf{x}|C_k)$
- Assume conditional independence (CI) of features given class
- CI simplifies the specification

$$p(\mathbf{x}|C_k) = p(x_1, \ldots, x_p|C_k) = \prod_{i=1}^{p} p(x_i|C_k)$$

- Factorized form for $p(\mathbf{x}|C_k)$
- Sufficient to specify marginal distributions $p(x_i|C_k)$
- Examples:
  - Binary $x_i \in \{0, 1\}$, Bernoulli distribution $p(x_i|C_k) = \mu_{ik} \in [0, 1]$
  - Count $x_i \in \{0, 1, 2, \ldots\}$, multinomial, Poisson, etc.
  - Real $x_i \in \mathbb{R}$, univariate Gaussian $p(x_i|C_k) = \mathcal{N}(\mu_{ik}, \sigma_{ik}^2)$

Naive Bayes: Binary Features, Bernoulli Marginals

- Assume binary features $x_i \in \{0, 1\}$
- Bernoulli marginals: for feature i, class k, $\mu_{ik} \in [0, 1]$

$$p(x_i = 1 | C_k) = \mu_{ik} \qquad p(x_i = 0 | C_k) = 1 - \mu_{ik}$$

- Assume conditional independence of features

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{p} p(x_i|C_k) = \prod_{i=1}^{p} \mu_{ik}^{x_i}(1 - \mu_{ik})^{1-x_i}$$

- For K-classes, we have

$$a_k(\mathbf{x}) = \sum_{i=1}^{p} \{x_i \log \mu_{ik} + (1 - x_i) \log(1 - \mu_{ik})\} + \log p(C_k)$$

Discriminative Models and Bayes Rule

- Bayes rule states that $p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$
- Generative models make assumptions about $p(\mathbf{x}|y)$
- Discriminative models
  - Make assumptions about $p(y|\mathbf{x})$
  - There is no attempt to model $p(\mathbf{x})$
  - Does not solve a more general problem

Logistic Regression (2 Class)

- Assume a 2 class problem with $\mathbf{x} \in \mathbb{R}^p$ and $y \in \{0, 1\}$
- Logistic Regression assumes

$$\log\left(\frac{P(1|\mathbf{x})}{P(0|\mathbf{x})}\right) = \mathbf{w}^\top \mathbf{x}$$

- The log-odds ratio is linear in $\mathbf{x}$

- A direct calculation gives

$$\begin{aligned} P(1|\mathbf{x}) &= \frac{\exp(\mathbf{w}^\top \mathbf{x})}{1 + \exp(\mathbf{w}^\top \mathbf{x})} = \sigma(\mathbf{w}^\top \mathbf{x}) \\ P(0|\mathbf{x}) &= \frac{1}{1 + \exp(\mathbf{w}^\top \mathbf{x})} = 1 - \sigma(\mathbf{w}^\top \mathbf{x}) \end{aligned}$$

Logistic Regression as a Bernoulli Model

- Labels: $y_i \in \mathcal{Y} := \{0, 1\}$
- Signed version: $2y_i - 1 \in \{-1, 1\}$
- Assume $y_i | \mathbf{x}_i \sim \text{Bern}(\sigma(\mathbf{w}^\top \mathbf{x}_i))$
- MLE aims to find model parameters which maximize $P(\text{observed data}|\text{model parameters})$
- For logistic regression this means finding $\mathbf{w}$ that maximizes $P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n | \mathbf{w})$

MLE for Logistic Regression

$$\begin{aligned}
\mathbf{w}^* &= \text{argmax}_{\mathbf{w}} \, P(y_1, \mathbf{x}_1, \ldots, y_n, \mathbf{x}_n | \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} P(y_i, \mathbf{x}_i | \mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \prod_{i=1}^{n} \sigma(\mathbf{w}^\top \mathbf{x}_i)^{y_i} (1 - \sigma(\mathbf{w}^\top \mathbf{x}_i))^{1-y_i} \\
&= \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} y_i \log(\sigma(\mathbf{w}^\top \mathbf{x}_i)) + (1 - y_i) \log((1 - \sigma(\mathbf{w}^\top \mathbf{x}_i))) \\
&= \text{argmax}_{\mathbf{w}} - \sum_{i=1}^{n} y_i \log(1 + \exp(-\mathbf{w}^\top \mathbf{x}_i)) + (1 - y_i) \log(1 + \exp(\mathbf{w}^\top \mathbf{x}_i)) \\
&= \text{argmax}_{\mathbf{w}} - \sum_{i=1}^{n} \log(1 + \exp(-(2y_i - 1)\mathbf{w}^\top \mathbf{x}_i)) \\
&= \text{argmin}_{\mathbf{w}} \sum_{i=1}^{n} \log(1 + \exp(-(2y_i - 1)\mathbf{w}^\top \mathbf{x}_i))
\end{aligned}$$

ERM for Logistic Regression

- Labels: $y_i \in \mathcal{Y} := \{-1, 1\}$
- Here we are considering the class of linear functions $\mathbf{w}^\top \mathbf{x}$
- Recall last lecture we gave several surrogate loss functions to replace 0-1 loss
- We will use the logistic loss: $\log(1 + \exp(-y\mathbf{w}^\top \mathbf{x}))$
- ERM problem:

$$\text{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i))$$

- This ERM problem is the same as the MLE problem!

Training Logistic Regression

- Try finding solutions of

$$\nabla_{\mathbf{w}} E(\mathbf{w}_t) = \nabla_{\mathbf{w}} \sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)) = 0$$

- This has no closed-form solution
- Must use an iterative algorithm to compute solution
- PRML gives detailed use of iteratively reweighted least squares (IRLS)
- Can also use gradient descent to find global minima (why?)
  - $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \nabla E(\mathbf{w}_t)$
  - step size $\alpha_t \in \mathbb{R}_+$

Multi-class Logistic Regression

- The class posteriors are given by:

$$p(C_k|\mathbf{x}) = \pi_k(\mathbf{w}_k; \mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}\ , \quad a_k = \mathbf{w}_k^\top \mathbf{x}$$

- The likelihood can be written using $\mathbf{y}_i$ (1-of-K coding)

$$p(\mathbf{y}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = \prod_{i=1}^{n}\prod_{k=1}^{K} p(C_k|\mathbf{x}_i)^{y_{ik}} = \prod_{i=1}^{n}\prod_{k=1}^{K} \pi_{ik}^{y_{ik}}$$

$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\log p(\mathbf{y}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log \pi_{ik}$$

Generative vs Discriminative

- Generative models make explicit assumptions on $p(\mathbf{x}|y)$
  - Solves a more general problem, finds $p(\mathbf{x})$
  - Has higher "bias" (focuses on a smaller set of models)
  - Converges faster to asymptotic performance
  - There are consistent estimation algorithms
  - True error rate may be high if assumptions are not appropriate
- Logistic regression makes assumptions on $p(y|\mathbf{x})$
  - Does not solve a more general problem
  - Has "lower bias" (focuses on a bigger set of models)
  - Convergence to asymptotic performance is slower
  - Careful consistency analysis is required
  - True error rate may be lower due to low bias