Deep Generative Models

Chapter 2: Data Generation Problem

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Summer 2025

Old Classification Problem

The most basic generative model is

Naive Bayes

which was originally developed for classification

Say we have single-channel $k \times k$ images belonging to two classes

$$y = \begin{cases} 1 & \textit{cat} \\ 0 & \textit{dog} \end{cases}$$

We have access to a dataset $\mathbb{D}$

$$\mathbb{D} = \left\{ (x_i, y_i) \, x_i \in \{0, \ldots, 255\}^{k \times k}, y_i \in \{0, 1\} : i = 1, \ldots, n \right\}$$

and intend to learn a classifier

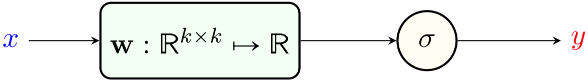
Conventional Solution: *Discriminative Modeling*

We all know the conventional solution: *we consider a discriminative model*

$$P_{\mathbf{w}}\left(y|x\right)$$

and train it on the data

In good old days, we were very limited in computation: it was typical to think of simplest possible model,i.e., a linear model



- What if we could go even simpler?
+ *What do you mean? It's linear, we cannot think of simpler one!*
- Well, we could think tabular!

Tabular *Discriminative Modeling*

In tabular approach, we estimate distribution directly from dataset, e.g.,

$$\hat{P}(y = 1) = \frac{\text{\# of } y = 1\text{'s in } \mathbb{D}}{n}$$

This approach *obviously does not work for discriminative learning*

+ *And how is that obvious?*

Noting that *typically* $n \ll 256^{k^2} \equiv$ *size of data space: it is impossible to have enough samples to get a reliable tabular estimate of the posterior*

$$\hat{P}(y = 1|x) = \frac{\text{\# of } (x, 1)\text{'s in } \mathbb{D}}{\text{\# of samples with image } x \text{ in } \mathbb{D}}$$

Well, most of the time $\rightsquigarrow$ *we are to classify new unseen image*

both nominator and denominator are zero!

Tabular *Generative Modeling*

Unlike discriminative case, tabular generative learning seems possible!

In tabular generative learning: *we need* $P(x|y)P(y)$

1. *An estimate of prior belief on label* $\rightsquigarrow$ *this is easy to estimate*

$$\hat{P}(y) = \frac{\text{\# of } y\text{'s in } \mathbb{D}}{n}$$

↳ *For sure, we have enough samples of both labels*

2. *We need an estimate of the class generative distribution*

$$\hat{P}(x|y) = \frac{\text{\# of } (x,y)\text{'s in } \mathbb{D}}{\text{\# of samples with label } y \text{ in } \mathbb{D}}$$

↳ *For sure, the denominator is non-zero*

↳ *The nominator is however most of the tine zero!*

Tabular *Generative Modeling*

+ *Well! We ended up with the same issue! Right?!*
– Yes! But there is a way out of it

We reduce the complexity by postulating a simple assumption $\equiv$ tabular model

Naive Bayes Model

In naive Bayes, we assume that given y entries of x are mutually independent

$$x = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{kk} \end{bmatrix} \leftrightsquigarrow P(x|y) = \prod_{i,j} P(x_{ij}|y)$$

Naive Bayes Model is *Just a Naive Tabular Model*

In reality, the pixels are not independent $\leftrightsquigarrow$ this is just a simple model

Naive Bayes Model

+ *How does it help then?!*
- Well, each $P(x_{ij}|y)$ is feasible to estimate

Say for instant we want to estimate $P(x_{11}|y)$

$$\hat{P}(x_{11} = 9|y) = \frac{\text{\# of samples in } \mathbb{D} \text{ with label } y \text{ and first entry } 9}{\text{\# of samples with label } y \text{ in } \mathbb{D}}$$

↳ *With large* $\mathbb{D}$*, we could have enough samples with* $x_{11} = 9$

For a new sample x*: though we haven't seen it before, we have seen each of its pixels before* ⇝ *We can compute its class generative distributions as*

$$\hat{P}(x|0) = \prod_{i,j} \hat{P}(x_{ij}|0) \qquad \hat{P}(x|1) = \prod_{i,j} \hat{P}(x_{ij}|1)$$

Naive Bayes Model

To classify the new sample x, we can then use the Bayes rule and write

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

and classify with the label that has higher posterior

Moral of Story

Sometimes generative modeling can offer more degrees of freedom

↳ *It is impossible to think about tabular discriminative learning*

↳ *With generative learning we can use a tabular model*

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad\qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	\$	0
B	\$	0
A	€	0
A	\$	0
B	€	0
B	€	1
B	\$	1

Discriminative Learning

Say, we want to classify a new sample $x = [\mathtt{A}\ , \$]$

$$\hat{P}(1|x) = \frac{\#\,(x, 1)\text{ occurred}}{\#\,x\text{ occurred}} = \frac{0}{1} = 0$$

$$\rightsquigarrow \hat{P}(0|x) = 1$$

which is not reliable with only one training sample!

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	\$	0
B	\$	0
A	€	0
A	\$	0
B	€	0
B	€	1
B	\$	1

Generative Learning

What if we use naive Bayes for $x = [\mathtt{A}, \$]$?

$$\hat{P}(\mathtt{A} \mid 1) = \frac{\#\ (\mathtt{A}, 1) \text{ occurred}}{\#\ 1 \text{ occurred}} = \frac{2}{5} = 0.4$$

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	$	0
B	$	0
A	€	0
A	$	0
B	€	0
B	€	1
B	$	1

Generative Learning

What if we use naive Bayes for $x = [\mathtt{A}\ , \$]$?

$$\hat{P}(\mathtt{A}\ |0) = \frac{\#\ (\mathtt{A}\ , 0) \text{ occurred}}{\#\ 0 \text{ occurred}} = \frac{2}{5} = 0.4$$

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	$	0
B	$	0
A	€	0
A	$	0
B	€	0
B	€	1
B	$	1

Generative Learning

What if we use naive Bayes for $x = [\mathtt{A}, \$]$*?*

$$\hat{P}(\$|1) = \frac{\#\,(\$, 1) \text{ occurred}}{\#\,1 \text{ occurred}} = \frac{1}{5} = 0.2$$

So we can say

$$\hat{P}(x|1) = \hat{P}(\mathtt{A}|1)\hat{P}(\$|1) = 0.08$$

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	\$	0
B	\$	0
A	€	0
A	\$	0
B	€	0
B	€	1
B	\$	1

Generative Learning

What if we use naive Bayes for $x = [\mathtt{A}, \$]$?

$$\hat{P}(\$|0) = \frac{\#\,(\$, 0)\text{ occurred}}{\#\,0\text{ occurred}} = \frac{3}{5} = 0.6$$

So we can say

$$\hat{P}(x|0) = \hat{P}(\mathtt{A}|0)\hat{P}(\$|0) = 0.24$$

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	\$	0
B	\$	0
A	€	0
A	\$	0
B	€	0
B	€	1
B	\$	1

Generative Learning

Prior probability is also readily estimated

$$\hat{P}(1) = \frac{\#\ 1 \text{ occurred}}{10} = \frac{5}{10} = 0.5$$

So, we have

$$\hat{P}(0) = 1 - \hat{P}(1) = 0.5$$

Example: Bayes vs Discriminative

In the old problem, assume we have only two pixels

$$x_1 \in \{\mathtt{A}, \mathtt{B}\} \qquad x_2 \in \{€, \$\}$$

x_1	x_2	y
A	€	1
A	€	1
B	€	1
B	$	0
B	$	0
A	€	0
A	$	0
B	€	0
B	€	1
B	$	1

Generative Learning

What if we use naive Bayes for $x = [\mathtt{A}, \$]$?

$$\hat{P}(1|x) = \frac{\hat{P}(x|1)\hat{P}(1)}{\hat{P}(x|1)\hat{P}(1) + \hat{P}(x|0)\hat{P}(0)}$$
$$= \frac{0.04}{0.16} = 0.25 \rightsquigarrow \hat{P}(0|x) = 0.75$$

Sounds like more reliable!

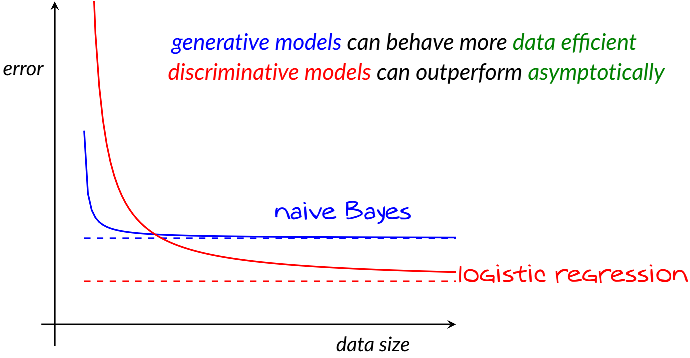
Interesting Old Study

On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes

Andrew Y. Ng
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720

Michael I. Jordan
C.S. Div. & Dept. of Stat.
University of California, Berkeley
Berkeley, CA 94720

generative models can behave more data efficient
discriminative models can outperform asymptotically



Deep Generative Models

Using naive Bayes model, we can sample new data points

$$\text{we sample } y \sim \hat{P}(y) \rightsquigarrow \text{fixing } y \text{ we sample } x \sim \hat{P}(x|y)$$

Nevertheless, we obviously cannot expect meaningful samples

↳ *the tabular model is too simple*

↳ *data distribution is too complex*

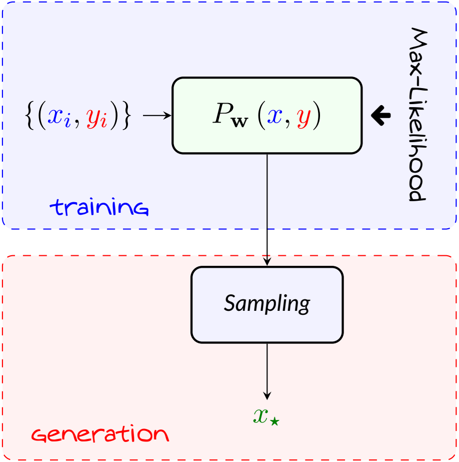
The solution is hence to go for a deep generative model

$$\text{we sample } y \sim \hat{P}(y) \rightsquigarrow \text{fixing } y \text{ we sample } x \sim \hat{P}_{\mathbf{w}}(x|y)$$

where $\hat{P}_{\mathbf{w}}(x|y)$ is a deep model, e.g., a deep NN

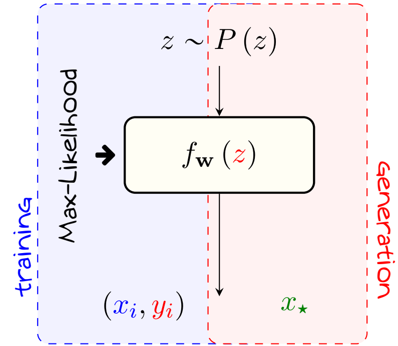
Road Map: *Deep Generative Models*

Conventional Deep Generation



- *LMs, Energy-based* $\rightsquigarrow$ *Chapter 3*

Modern Deep Generation



- *GANs, VAEs, Diffusion* $\rightsquigarrow$ *Chapters 4-6*