Discriminative models and generative models

Jae Yun JUN KIM[*]

February 22, 2020

Sources: Wikipedia and Andrew Ng's lecture notes.

1 Discriminative models

Discriminative models, also called conditional models, are a class of models used in machine learning for modeling the dependence of unobserved (target) variables y on observed variables x. Within a probabilistic framework, this is done by modeling the conditional probability distribution $P(y|x)$, which can be used for predicting y from x.

Discriminative models, as opposed to generative models, do not allow one to generate samples from the joint distribution of observed and target variables. However, for tasks such as classification and regression that do not require the joint distribution, discriminative models can yield superior performance (in part because they have fewer variables to compute). On the other hand, generative models are typically more flexible than discriminative models in expressing dependencies in complex learning tasks. In addition, most discriminative models are inherently supervised and cannot easily support unsupervised learning. Application-specific details ultimately dictate the suitability of selecting a discriminative versus generative model.

1.1 Examples of discriminative models used in machine learning

- Logistic regression
- Support vector machines
- Linear regression
- Neural networks
- Random forests
- Boosting
- Conditional random fields

[*] ECE Paris Graduate School of Engineering, 37 quai de Grenelle 75015 Paris, France; jae-yun.jun-kim@ece.fr

2 Generative models

In probability and statistics, a generative model is a model for generating all values for a phenomenon, both those that can be observed in the world and "target" variables that can only be computed from those observed. By contrast, discriminative models provide a model only for the target variable(s), generating them by analyzing the observed variables. In simple terms, discriminative models infer outputs based on inputs, while generative models generate both inputs and outputs, typically given some hidden parameters.

Generative models are used in machine learning for either modeling data directly (i.e., modeling observations drawn from a probability density function), or as an intermediate step to forming a conditional probability density function. Generative models are typically probabilistic, specifying a joint probability distribution over observation and target (label) values. A conditional distribution can be formed from a generative model through Bayes' rule.

Shannon (1948) gives an example in which a table of frequencies of English word pairs is used to generate a sentence beginning with "representing and speedily is an good"; which is not proper English but which will increasingly approximate it as the table is moved from word pairs to word triplets etc.

Despite the fact that discriminative models do not need to model the distribution of the observed variables, they cannot generally express complex relationships between the observed and target variables. They do not necessarily perform better than generative models at classification and regression tasks. The two classes are seen as complementary or as different views of the same procedure.

2.1 Examples of generative models used in machine learning

- Gaussian discriminant analysis
- Naive Bayes
- Gaussian mixture model
- Restricted Boltzmann machine
- Latent Dirichlet allocation
- Probabilistic context-free grammar
- Generative adversarial networks

3 Comparison between discriminative models and generative models

A generative algorithm models how the data was generated in order to categorize a signal. It asks the question: based on my generation assumptions, which category is most likely to generate this signal? A discriminative algorithm does not care about how the data was generated, it simply categorizes a given signal.

Suppose the input data is x and the set of labels for x is y. A generative model learns the joint probability distribution $p(x, y)$, while a discriminative model learns the conditional probability distribution $p(y|x)$, "probability of y given x".

Let us try to understand this with an example. Consider the following 4 data points: $(x, y) = \{(0, 0), (0, 0), (1, 0), (1, 1)\}$

For above data, $p(x, y)$ will be following:

	$y = 0$	$y = 1$
$x = 0$	$1/2$	0
$x = 1$	$1/4$	$1/4$

while $p(y|x)$ will be following:

	$y = 0$	$y = 1$
$x = 0$	1	0
$x = 1$	$1/2$	$1/2$

So, discriminative algorithms try to learn $p(y|x)$ directly from the data and then try to classify data. On the other hand, generative algorithms try to learn $p(x, y)$ which can be transformed into $p(y|x)$ later to classify the data. One of the advantages of generative algorithms is that you can use $p(x, y)$ to generate new data similar to existing data. On the other hand, discriminative algorithms generally give better performance in classification tasks.

If the observed data are truly sampled from the generative model, then fitting the parameters of the generative model to maximize the data likelihood is a common method. However, since most statistical models are only approximations to the true distribution, if the model's application is to infer about a subset of variables conditional on known values of others, then it can be argued that the approximation makes more assumptions than are necessary to solve the problem at hand. In such cases, it can be more accurate to model the conditional density functions directly using a discriminative model, although application-specific details will ultimately dictate which approach is most suitable in any particular case.

4 Examples of discriminative model: linear regression and logistic regression

4.1 Linear regression

Assume

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $i = \{1, \cdots, I\}$, and I is the number of data examples. Hence,

$$P\left(\epsilon^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\epsilon^{(i)}\right)^2}{2\sigma^2}\right). \tag{2}$$

This implies that

$$P\left(y^{(i)}|x^{(i)}; \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right). \tag{3}$$

Hence,

$$y^{(i)}|x^{(i)}; \theta \sim \mathcal{N}\left(\theta^T x^{(i)}, \sigma^2\right). \tag{4}$$

Assume that $\epsilon^{(i)}$'s are independently identically distributed (i.i.d.). Then, define the **likelihood function** as

$$L(\theta) = p(y|x;\theta) = \prod_{i=1}^{I} P\left(y^{(i)}|x^{(i)};\theta\right) = \prod_{i=1}^{I} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right). \tag{5}$$

Notice that the likelihood function is the same as the probability function, but it is used to emphasize that it is in fact a function of the parameters having the variables fixed. Hence, we say that the likelihood of the parameters and the probability of the data (and not, the likelihood of the data or the probability of the parameters).

Now, from some given data, the optimal parameter values can be estimated by the **principle of the maximum likelihood estimation**.

4.1.1 The principle of the maximum likelihood estimation

- Choose θ that make the data as probable as possible (i.e., choose θ that make the parameters as likely as possible). That is, choose θ that maximize $L(\theta) = P\left(y|x;\theta\right)$.
- For a mathematical convenience, define

$$\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log\left(\prod_{i=1}^{I} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)\right) \\
&= \sum_{i=1}^{I} \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)\right) \\
&= I \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \sum_{i=1}^{I} -\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}.
\end{aligned} \tag{6}$$

So, maximizing $\ell(\theta)$ is the same as minimizing

$$\sum_{i=1}^{I} \frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2}. \tag{7}$$

Notice that this is the cost function (the sum of squared errors (SSE)) that we considered for the ordinary least squares (OLS). On the other hand, no matter what σ^2 is, since we should get the same parameters.

Consequently, the OLS is just the MLE, assuming that the errors are i.i.d. Gaussian random variables.

4.2 Logistic regression

A binary classification problem may be resolved using the logistic regression for which $y \in \{0, 1\}$.

One can approach this classification problem as a regression problem with $h_\theta(x) \in [0, 1]$.

Because a linear regression can give bad results for classification problem, let us choose instead the logistic function (also known as the sigmoid function)

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + \exp\left(-\theta^T x\right)}. \tag{8}$$

Assume now that the hypothesis function is used to estimated the probability of $y^{(i)} = 1$. Then,

$$P(y = 1|x; \theta) = h_\theta(x). \tag{9}$$

And, therefore,

$$P(y = 0|x; \theta) = 1 - h_\theta(x). \tag{10}$$

Now, we can compactly write the two above equations as

$$P(y|x; \theta) = \left(h_\theta(x)\right)^y \left(1 - h_\theta(x)\right)^{(1-y)}. \tag{11}$$

Then, define the likelihood of the parameters as

$$L(\theta) = P(y|x; \theta) = \prod_{i=1}^{I} P(y^{(i)}|x^{(i)}; \theta) = \prod_{i=1}^{I} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{(1-y^{(i)})}. \tag{12}$$

Then, define the log-likelihood as

$$\ell(\theta) = \log\left(L(\theta)\right) = \sum_{i=1}^{I} y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right). \tag{13}$$

Now, how can we maximize $\ell(\theta)$?

We can maximize it using the gradient ascent algorithm:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \ell(\theta). \tag{14}$$

If one actually computes the gradient of $\ell(\theta)$, one obtains

$$\frac{\partial \ell(\theta)}{\partial \theta_n} = \sum_{i=1}^{I} \left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right) x_n^{(i)}, \tag{15}$$

for $n \in \{0, 1, \cdots, N\}$. Hence,

$$\theta_n \leftarrow \theta_n + \alpha \sum_{i=1}^{I} \left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right) x_n^{(i)}. \tag{16}$$

The last expression looks exactly the same as the batch gradient descent for the OLS. But, in fact, they are not the same because the hypothesis function for this problem is different from the one used for the OLS.

For efficiency, one can use the stochastic gradient descent (SGD) instead:

$$\theta_n \leftarrow \theta_n + \alpha \left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right) x_n^{(i)}. \tag{17}$$

5 Examples of generative model: Gaussian discriminant analysis and naive Bayes

Recall that in generative models we compute $P(x|y)$ and $P(y)$ to compute in turn $P(y|x)$, which can be obtained from the Bayes rule.

Example:

Suppose that x is a continuous-valued random variable and y is a discrete random variable taking only values $\{0, 1\}$. Then, using the Bayes rule we have:

$$P(y = 1|x) = \frac{P(x|y = 1)P(y = 1)}{P(x)}, \tag{18}$$

where $P(x) = P(x|y = 0)P(y = 0) + P(x|y = 1)P(y = 1)$.

5.1 Gaussian discriminant analysis model

Assume $x \in \mathbb{R}^n$ is continuous valued and assume that $P(x|y)$ is Gaussian. Therefore, $x|y$ is a multivariate Gaussian random variable. If we suppose that y is a Bernoulli random variable, then

$$\begin{aligned} P(y;\phi) &= \phi^y (1-\phi)^{(1-y)}, \\ P(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^{\mathrm{T}}\,\Sigma^{-1}\,(x-\mu_0)\right), \\ P(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^{\mathrm{T}}\,\Sigma^{-1}\,(x-\mu_1)\right). \end{aligned} \tag{19}$$

Then, the **joint log-likelihood of the parameters** is

$$\ell(\theta) = \log\left(\prod_{i=1}^{I} P\left(x^{(i)}, y^{(i)}\right)\right) = \log\left(\prod_{i=1}^{I} P\left(x^{(i)}|y^{(i)}\right) P\left(y^{(i)}\right)\right), \tag{20}$$

where $\theta = (\phi, \mu_0, \mu_1, \Sigma)$.
Now when we maximize ℓ with respect to $\phi, \mu_0, \mu_1, \Sigma$, we get

$$\begin{aligned} \phi &= \frac{1}{I}\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\}, \\ \mu_0 &= \frac{\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 0\right\} x^{(i)}}{\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 0\right\}}, \\ \mu_1 &= \frac{\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\} x^{(i)}}{\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\}}, \\ \Sigma &= \frac{1}{I}\sum_{i=1}^{I} \left(x^{(i)} - \mu_{y^{(i)}}\right)\left(x^{(i)} - \mu_{y^{(i)}}\right)^{\mathrm{T}}. \end{aligned} \tag{21}$$

Finally, we predict an output using the following criterion:

$$\arg\max_y P(y|x) = \arg\max_y \frac{P(x|y)P(y)}{P(x)} = \arg\max_y P(x|y)P(y). \tag{22}$$

5.2 Naive Bayes

We will describe the Naive Bayes using an example. This example consists of classifying whether a given email is a spam.

To solve this problem, we first construct a dictionary and generate a feature vector that corresponds to the given email. The dictionary can be constructed, for instance, by listing all the words contained in a collection of emails. Then, we identify the words contained in a given email by generating the feature vector, which can have the following shape:

$$x = [1, 0, 0, \cdots, 0, 1, 0, \cdots, 0]^{\mathrm{T}}. \tag{23}$$

Let us build a generative learning model for this problem. In other words, let us compute $P(x|y)$, where $x \in \{0, 1\}^N$ (where $N = 50000$, for instance).

For simplicity, let us assume that x_n's are conditionally independent given y (a *naive* assumption). That is,

$$\begin{aligned} P(x_1, \cdots, x_{50000}|y) &= P(x_1|y)P(x_2|y, x_1)P(x_3|y, x_1, x_2)\cdots \\ &= P(x_1|y)P(x_2|y)P(x_3|y)\cdots P(x_{50000}|y) \\ &= \prod_{n=1}^{N} P(x_n|y). \end{aligned} \tag{24}$$

The above expression says that once we know that an email is Spam (or not), the presence of some words in this email does not help to know whether some other words are also present in that email.

This (*naive*) assumption is obviously false. But, despite this incongruence, the Naive Bayes algorithm works well for classifying texts. On the other hand, without this assumption the problem to be solved would be much harder.

Now, let us define the model parameters:

$$\begin{aligned} \phi_{n|y=1} &= P(x_n = 1|y = 1), \\ \phi_{n|y=0} &= P(x_n = 1|y = 0), \\ \phi_y &= P(y = 1). \end{aligned} \tag{25}$$

To fit the parameters of the model, write the *joint likelihood* as

$$\mathcal{L}(\phi_y, \phi_{j|y=1}, \phi_{j|y=0}) = \prod_{i=1}^{I} P\left(x^{(i)}, y^{(i)}\right). \tag{26}$$

When you perform the MLE, then you obtain the following optimal parameter values:

$$\begin{aligned} \phi_y &= \frac{1}{I}\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\}, \\ \phi_{n|y=1} &= \frac{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{x_n^{(i)} = 1, y^{(i)} = 1\right\}}{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\}}, \\ \phi_{n|y=0} &= \frac{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{x_n^{(i)} = 1, y^{(i)} = 0\right\}}{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 0\right\}}. \end{aligned} \tag{27}$$

Once these parameters are computed, one can evaluate whether a new email is spam or not by computing the corresponding $P(y|x)$ using the Bayes rule, since these parameters give $P(x|y)$ and $P(y)$.

That is, we predict an output using the following criterion:

$$\arg\max_y P(y|x) = \arg\max_y \frac{P(x|y)P(y)}{P(x)} = \arg\max_y P(x|y)P(y). \tag{28}$$

However, if, for instance, we encounter a case where

$$P(x_{30000}|y = 1) = P(x_{30000}|y = 0) = 0 \tag{29}$$

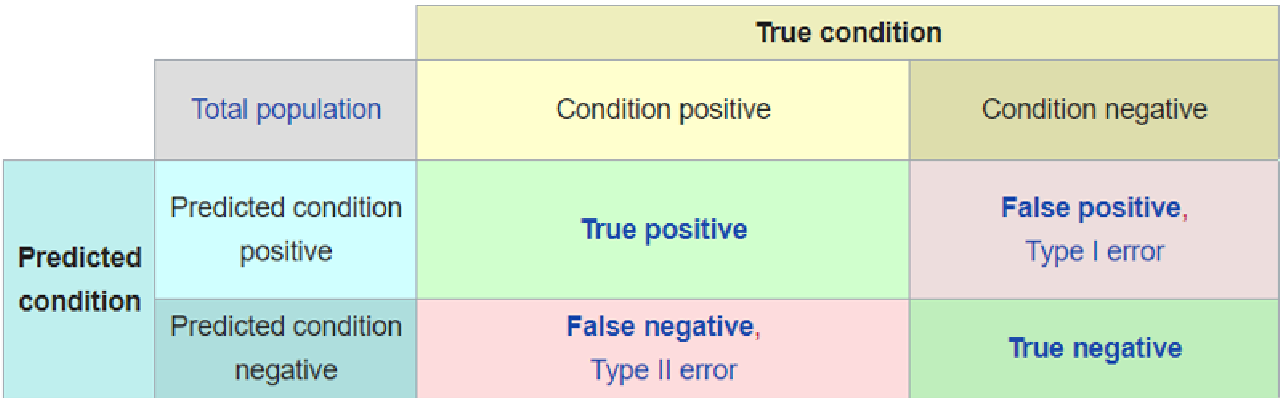
because such a word was never seen before. On the other hand, we know that

$$P(y = 1|x) = \frac{P(x|y = 1)P(y = 1)}{P(x|y = 1)P(y = 1) + P(x|y = 0)P(y = 0)} = \frac{0}{0 + 0} = \frac{0}{0}. \tag{30}$$

We can fix this with the **Laplace smoothing technique**. Therefore,

$$\begin{aligned} \phi_y = P(y^{(i)} = 1) &= \frac{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\} + 1}{I + 2}, \\ \phi_{n|y=1} = P(x_n = 1|y = 1) &= \frac{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{x_n^{(i)} = 1, y^{(i)} = 1\right\} + 1}{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 1\right\} + 2}, \\ \phi_{n|y=0} = P(x_n = 1|y = 0) &= \frac{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{x_n^{(i)} = 1, y^{(i)} = 0\right\} + 1}{\displaystyle\sum_{i=1}^{I} \mathbb{I}\left\{y^{(i)} = 0\right\} + 2}. \end{aligned} \tag{31}$$

6 Confusion matrix



		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

Figure 1: Confusion matrix