

INTRODUCTION TO GENERATIVE SUPERVISED LEARNING MODELS

■ ISE – 364 / 464

■ DEPT. OF INDUSTRIAL & SYSTEMS
ENGINEERING

■ GRIFFIN DEAN KENT



LEHIGH
UNIVERSITY.

OVERVIEW OF DISCRIMINATIVE VS. GENERATIVE MODELS

Discriminative Learning

- Models that directly aim at learning (or approximating) the **posterior probability distribution** $\mathbb{P}(Y|X)$.
- The posterior distribution $\mathbb{P}(Y|X)$ is the distribution of the target variable Y conditioned on observing a set of data features X .
- This distribution is given the name “**posterior**” distribution in reference to the probability of the target variable Y “**post**” (or after) observing the data X .
- Alternatively, we refer to $\mathbb{P}(Y)$ as the **prior distribution** since it is meant to represent our knowledge or belief in an outcome of interest Y before observing any data X , hence the name “**prior**”.

Generative Learning

- Models that *still* predict posterior probabilities, but instead of learning the posterior distribution directly, generative models focus on learning the **underlying joint probability distribution** $\mathbb{P}(X, Y)$, after which **Baye’s Theorem** can be applied to “**generate**” probabilities from the posterior distribution.
- Once the joint probability distribution $\mathbb{P}(X, Y)$ is learned, using **Baye’s Theorem**, posterior probabilities are generated according to the equation

$$\mathbb{P}(Y|X) = \frac{\mathbb{P}(Y)\mathbb{P}(X|Y)}{\mathbb{P}(X)}.$$

GENERATIVE MODELS & BAYES' THEOREM

- As stated, the difference between discriminative models and generative models in supervised learning is that discriminative models aim at learning the posterior distribution $\mathbb{P}(y|x)$ directly whereas generative models aim at learning the joint distribution $\mathbb{P}(x, y)$ from which one can generate posterior probabilities via Bayes' Theorem.
- Recall from the *Probability & Statistics Slides* that conditional probability is defined as

$$\mathbb{P}(y|x) = \frac{\mathbb{P}(x, y)}{\mathbb{P}(x)}.$$

- However, in machine learning problems, we **do not typically have access to “exact” information of the joint probability distribution $\mathbb{P}(x, y)$** (recall from the *Linear Regression Slides* that this is a direct approximation of the true underlying distribution that the data is drawn from, denoted by $\mathcal{P}(x, y)$).
- On the other hand, we do typically have information of the marginal “**prior**” $\mathbb{P}(y)$, the marginal $\mathbb{P}(x)$, and the conditional distribution $\mathbb{P}(x|y)$, which can be modeled by again using the definition of conditional probability as

$$\mathbb{P}(x|y) = \frac{\mathbb{P}(x, y)}{\mathbb{P}(y)}.$$

- Using this, we can solve for the joint distribution in terms of quantities that we do know

$$\mathbb{P}(x, y) = \mathbb{P}(y)\mathbb{P}(x|y).$$

- Lastly, substituting this into the equation for the posterior distribution above, we obtain Bayes' Theorem

$$\mathbb{P}(y|x) = \frac{\mathbb{P}(y)\mathbb{P}(x|y)}{\mathbb{P}(x)}.$$

Therefore, **if one can correctly model** the probability distributions $\mathbb{P}(y)$ and $\mathbb{P}(x|y)$, then one can generate posterior probabilities $\mathbb{P}(y|x)$ via Bayes' Theorem. **This is the idea behind Generative Models.**

WHAT MAKES GENERATIVE MODELS USEFUL?

- Although generative models “can” be used to predict a target variable y in a supervised learning context, they are typically only used for this purpose in scenarios when there is **not much data available** to train on.
- If relatively large amounts of data are available, then one would typically use a discriminative model which will almost always have higher predictive power.

So, then what are generative models used for?

- The reason that these models are given the moniker “**generative**” is because they are very useful for **generating synthetic data that mimics the original “true” distribution** $\mathcal{P}(x, y)$ (with the approximation obtained by $\mathbb{P}(x, y) = \mathbb{P}(y)\mathbb{P}(x|y)$).
- As a result, once the approximate joint distribution $\mathbb{P}(x, y)$ has been learned, one can **sample new synthetic data from that distribution**.
- This can be very useful for a variety of problems, such as **class-imbalance problems** (i.e., artificially increasing the number of instances for an underrepresented class), **data cleaning** (i.e., replacing missing data), and others.

GAUSSIAN DISCRIMINANT ANALYSIS

The first generative model that we will introduce is the **Gaussian Discriminant Analysis** model. This model (and all other generative models for that matter) assumes that the features and target variable of a dataset follow certain distributions, from which we can model relatively well.

The Gaussian Discriminant Analysis (GDA) Model

Consider some dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^m$, where $x^{(i)} \in \mathbb{R}^n$ and the target variable is **binary** $y \in \{0,1\}$.

- Therefore, the target variable y can be modeled as a **Bernoulli random variable** with probability parameter given by $\phi \in [0,1]$ such that $\mathbb{P}(y = 1) = \phi$.
- Further, this model assumes that each feature vector $x \in \mathbb{R}^n$ comes from an **n -dimensional multivariate Gaussian distribution when conditioned on y** . More specifically, this means that given some mean vector $\mu_y \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$, then it follows that $\mathbb{P}(x|y) \sim \mathcal{N}(\mu_y, \Sigma_y)$ for $y \in \{0,1\}$. Thus, we have that

$$\mathbb{P}(x|y; \mu_y, \Sigma_y) = \frac{1}{(2\pi)^{n/2} |\Sigma_y|^{1/2}} e^{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1} (x-\mu_y)}.$$

Therefore, when trying to classify a new datapoint (x, y) , the GDA model h_θ (where the parameters $\theta = [\mu_0, \mu_1, \Sigma_0, \Sigma_1]$) utilizes Bayes' Theorem and chooses the value of $y \in \{0,1\}$ that maximizes the posterior probability $\mathbb{P}(y|x)$, i.e.,

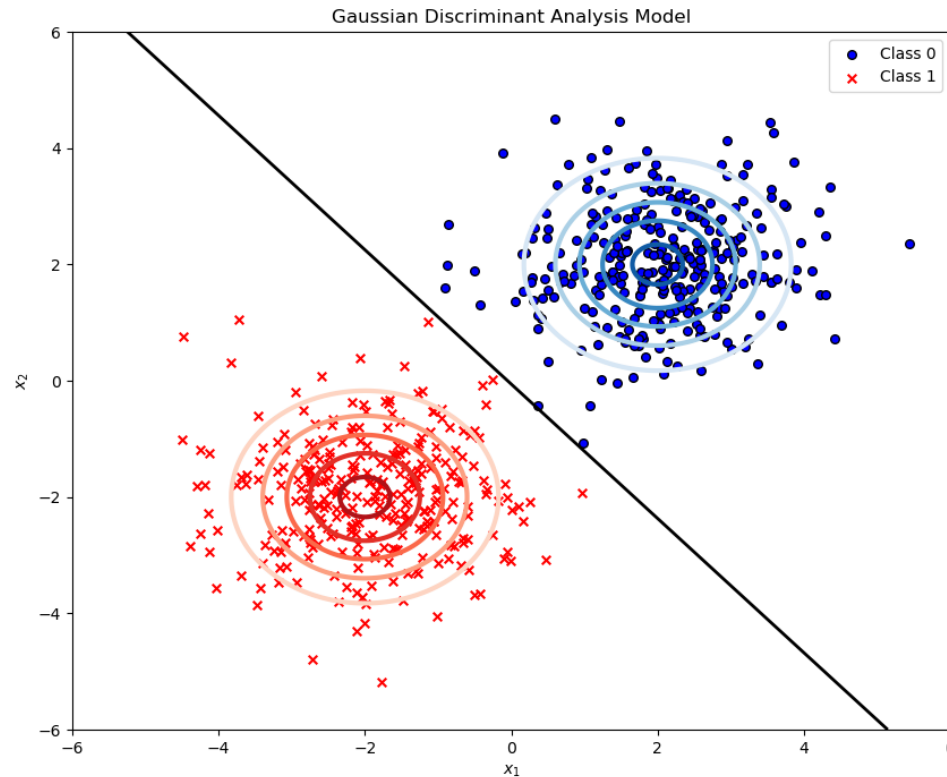
$$h_\theta(x) := \operatorname{argmax}_{y \in \{0,1\}} \mathbb{P}(y|x) = \operatorname{argmax}_{y \in \{0,1\}} \frac{\mathbb{P}(y)\mathbb{P}(x|y)}{\mathbb{P}(x)} = \operatorname{argmax}_{y \in \{0,1\}} \mathbb{P}(y)\mathbb{P}(x|y).$$

Since $\mathbb{P}(x)$ will only be a function in x and not y , we can omit it in the maximization over y .

- For the remainder of these slides, we will make the simplifying assumption that $\Sigma_0 = \Sigma_1 = \Sigma$ (i.e., equal covariances).

ILLUSTRATION OF GDA

This figure illustrates a **Gaussian Discriminative Analysis** (GDA) Model where the conditional distribution is a multivariate Gaussian distribution conditioned on a binary variable $Y \in \{0,1\}$, i.e., $f_{X|Y}(X|Y = y) \sim \mathcal{N}(\mu_y, \Sigma)$.



Recall from Homework Assignment 6 that a GDA model with the **same covariance matrices** (i.e., $\Sigma_0 = \Sigma_1 = \Sigma$) describing each of the Gaussians is **equivalent to a Logistic Regression model**.

LIKELIHOOD FUNCTION OF GDA

As with all probabilistic models that we have seen, we can go about “training” the GDA model by choosing the set of parameters θ that maximize the likelihood function of observing the data that was observed.

Likelihood Function for GDA

Under the assumptions that the datapoints, given by the set $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^m$, are i.i.d. and the prior distribution $\mathbb{P}(y^{(i)})$ is given by the **Bernoulli distribution**

$$\mathbb{P}(y^{(i)}) = \phi^{y^{(i)}}(1 - \phi)^{(1-y^{(i)})},$$

and the conditional distribution $\mathbb{P}(x^{(i)}|y^{(i)})$ is given by the **multivariate Gaussian distribution** $\mathcal{N}(\mu_{y^{(i)}}, \Sigma)$ written as

$$\mathbb{P}(x^{(i)}|y^{(i)}) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}})}.$$

Then, one can define the **likelihood** function of observing the data \mathcal{D} , given the set of model parameters $\theta := [\mu_0, \mu_1, \Sigma]$, as

$$L(\theta; \{(x^{(i)}, y^{(i)})\}_{i=1}^m) = \prod_{i=1}^m \mathbb{P}(x^{(i)}, y^{(i)}) = \prod_{i=1}^m \mathbb{P}(y^{(i)}) \mathbb{P}(x^{(i)}|y^{(i)}).$$

Similarly, the corresponding **log-likelihood** function is given by

$$\ell(\theta; \{(x^{(i)}, y^{(i)})\}_{i=1}^m) = \sum_{i=1}^m [\log \mathbb{P}(y^{(i)}) + \log \mathbb{P}(x^{(i)}|y^{(i)})].$$

MAXIMUM LIKELIHOOD ESTIMATES FOR GDA

Training the GDA model amounts to solving the optimization problem of maximizing the likelihood function, i.e.,

$$\theta^* = \operatorname{argmax}_{\theta} \ell \left(\theta; \{(x^{(i)}, y^{(i)})\}_{i=1}^m \right).$$

After some algebra, the explicit form of the log-likelihood can be written as

$$\ell(\theta) = \sum_{i=1}^m \left[y^{(i)} \log \phi + (1 - y^{(i)}) \log(1 - \phi) - \frac{1}{2} \left(x^{(i)} - \mu_{y^{(i)}} \right)^T \Sigma^{-1} \left(x^{(i)} - \mu_{y^{(i)}} \right) \right] - \frac{mn}{2} \log 2\pi - \frac{m}{2} \log |\Sigma|.$$

- **An important point:** this **log-likelihood function is not concave in general**... However, we can **still solve for closed-form analytical solutions** for the maximum-likelihood estimates (MLE) ϕ^* , μ_0^* , μ_1^* , and Σ^* because the partial derivatives with respect to each of these parameters reduces to solvable linear or quadratic equations, which do enable closed-form solutions. It can be shown that, by setting the partial derivatives of the parameters to zero, their optimal solutions can be derived as

$$\phi^* = \frac{1}{m} \sum_{i=1}^m y^{(i)} \quad \text{and} \quad \mu_y^* = \frac{\sum_{i=1}^m \mathbb{I}[y^{(i)} = y] x^{(i)}}{\sum_{i=1}^m \mathbb{I}[y^{(i)} = y]},$$
$$\Sigma^* = \frac{1}{m} \sum_{i=1}^m \left[[y^{(i)} = 0] x^{(i)} (x^{(i)})^T + [y^{(i)} = 1] x^{(i)} (x^{(i)})^T \right].$$

- Lastly, if the covariance matrices are not equivalent (i.e., $\Sigma_0 \neq \Sigma_1$), the other MLEs of the other parameters are the same, but the solutions Σ_0^* and Σ_1^* will take on different forms; however, they can still be derived by setting the respective partial derivatives of the log-likelihood equal to 0 and solving.

ATTRIBUTION & LICENSE

These lecture slides are part of the "*Introduction to Machine Learning*" course materials created by **Griffin Dean Kent**.

For the latest version, updates, and additional resources, visit the GitHub repository:

<https://github.com/GdKent/Introduction-to-Machine-Learning>.

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). You are free to share and adapt these materials, provided proper attribution is given and any derivatives are shared under the same license.