Machine Learning

"Field of study that gives computers the ability to learn with out being explicitly programmed" - Arthur Samuel

1. Exponential family

In supervised learning, the exponential family *EF* is a probability density function expressed as follows

$$P(y\,|\,\eta) = b(y)\exp\left\{\eta^T T(y) - a(\eta)\right\}$$

This model *generalizes* the probability density function of many ML models (*generalization*).

- η *natural parameter*
- $b(y)$ *base measure*
- $T(y) = y$ *sufficient statistic*
- $a(\eta)$ *log position function*

Starting from the known probability density function of a model, we can work our way out to a new form of those that enlightens the *canonical parameters*.

Logistic regression - Bernoulli distribution

$$\begin{aligned} P(y\,|\,\phi) &= \phi^y(1-\phi)^{1-y} \\ &= \exp\left\{\log\left(\phi^y(1-\phi)^{1-y}\right)\right\} \\ &= \exp\left\{y\log\phi + (1-y)\,\log(1-\phi)\right\} \\ &= \exp\left\{y\log\frac{\phi}{1-\phi} + \log(1-\phi)\right\} \end{aligned}$$

Where

- $T(y) = y$
- $\eta = \log\dfrac{\phi}{1-\phi}$ (g^{-1})
- $a(\eta) = -\log(1-\phi)$
- $b(y) = 1$

And just like this we demonstrated that Bernoulli's is part of the EF.

Linear regression - Gaussian distribution

$$\begin{aligned} P(y \mid \mu, \sigma = 1) &= \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}(y-\mu)^2 \right\} \\ &= k \exp\left\{ -\frac{y^2}{2} + y\mu - \frac{\mu^2}{2} \right\} \\ &= k \exp\left\{ -\frac{y^2}{2} \right\} \exp\left\{ y\mu - \frac{\mu^2}{2} \right\} \end{aligned}$$

Where

- $T(y) = y$
- $b(y) = \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{y^2}{2} \right\}$
- $\eta = \mu$ (g^{-1})
- $a(\eta) = \frac{\mu^2}{2}$

The normal gaussian distribution is part of the EF.

We can also revert the process and obtain a ϕ, μ as functions of η (g)

$$\mu = \eta$$

$$\phi = \frac{1}{1 + \exp(-\eta)}$$

2. Generalized linear models

In order to generalize a linear model we encounter 3 main steps, as well as some assumptions.

I. $y(x, \theta) \sim$ EF
II. $h_\theta(x) = E[y \,|\, x]$ (random variable)
III. $\eta = \theta^T x$ (assuming linear relationship between θ and x)

As previously demonstrated, we can describe linear regression and logistic regression as *consequences* of choosing gaussian, bernoulli distribution respectively ($h_\theta(x)$ directly comes by this choice, $\theta^T x$ remains fixed).

In particular, $h_\theta(x) = g(\eta)$ and g (*canonical response function, CRF*) is nothing but ϕ, μ functions of η, that as for linear and logistic regressions coincide with the expression derived in the previous paragraph.

—$g(\eta)\longrightarrow$ $\longleftarrow \eta = \theta^T x \longrightarrow$

Canonical parameters	**Natural parameters**	**Model parameters**
ϕ (Bernoulli)	η	$\theta \in \mathbb{R}^{d+1} \eta$
μ, σ^2 (Gaussian)		

$\longleftarrow g^{-1}(\eta)$---

Worth noting, the *dimension* of η (which is the variable that *constantly changes* in our model) depends on the *chosen distribution*

- Bernoulli: $\dim \eta = 1$
- Gaussian:
  - fixed σ: $\dim \eta = 1$
  - non-fixed σ: $\dim \eta = 2$

In general, the number of *parameters* the distribution has, equals to the number of dimensions of $\vec{\eta}$.

Summary of some of the linear models generalized with the EF

Type	**EF**	**GLM**
$\mathbb{R}$	Gaussian $\left(\mu, \sigma^2\right)$	Linear regression
$\{0, 1\}$	Bernoulli (ϕ)	Logistic regression
$\mathbb{N}$	Poisson (λ)	Poisson regression
$\{1, \ldots, \mathbb{R}\}$	Categorical	Softmax regression
$\mathbb{R}^2$	Exponential	Exponential regression

We can benefit from the *GLM*, and encapsulate the data in the most fitting model by making assumptions. Three main steps:

I. Choose the distribution and obtain the consequent CRF
II. Optimize the estimation by maximizing the likelihood (minimizing the negative log likelihood, minimizing the loss) with GD, that has the following *fixed* form

$$\theta = \theta - \alpha(y - h_\theta(x))x$$

III. Once the model is trained, make *predictions*

$$h_\theta(x) = g\left(\eta^T x\right) = E[y \mid x;\, \theta]$$

Useful definitions

- *Mean*

$$E[y] = \frac{\mathrm{d}a(\eta)}{\mathrm{d}\eta} = g(\eta)$$

- *Variance (non-centered)*

$$E\left[y^2\right] = \frac{d^2 a(\eta)}{d^2 \eta}$$

3. Gaussian discriminant analysis (GDA)

Discriminative and generative models

In general, taking y as a scalar and x as a vector

- *Discriminative* is a type of model that, starting from its probability density function, can only make distinctions. A discriminative model cannot be generative

$$P(y \mid x) \rightarrow \textbf{\textit{“distinguish”}}$$

- *Generative* are models that can actually generate the dataset starting from a classification (category y)

$$P(x \mid y)P(y) = P(x, y) \rightarrow \textbf{\textit{”generate”}}$$

Generative models can also be discriminative

$$P(x, y) \rightarrow \frac{P(x, y)}{\int_y P(x, y)} = \frac{P(x, y)}{P(x)} = P(y \mid x)$$

We can also affirm that

$$\hat{y} = \arg\max_y P(y \mid x) = \arg\max_y \frac{P(x \mid y)P(y)}{P(x)}$$

But $P(x)$ is constant so we can simplify it (imply it). The argmax iterates the product between the two factors in the numerator and returns the maximum value.

$$\hat{y} = \arg\max_y P(x \mid y)P(y) = P(x, y)$$

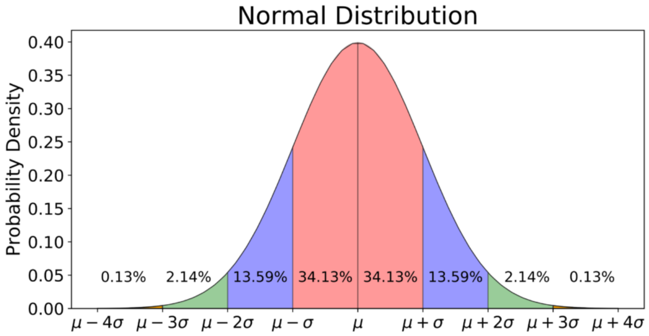
Multivariable normal gaussian distribution

Before diving into GDA, let’s recap the *M-NGD,* a key distribution in machine learning. It extends the concept of the normal gaussian distribution to a d -dimensional vector space.

In $\mathbb{R}$ the NGD looks like this

$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{ -\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2} \right\}$$

Where $\mu \in \mathbb{R}$ *expectancy*, $\sigma^2 \in \mathbb{R}^+$ *variance*



Now, in a multi-dimensional vector space, the M-NGD is expressed by

$$P(x \mid \mu; \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right\}$$

Where $\mu \in \mathbb{R}^d$ *expectancy*, $\Sigma \in \mathbb{R}^{d \times d}$ *covariance*
Note that $(x-\mu)^T \Sigma^{-1} (x-\mu)$ is a *quadratic form* and

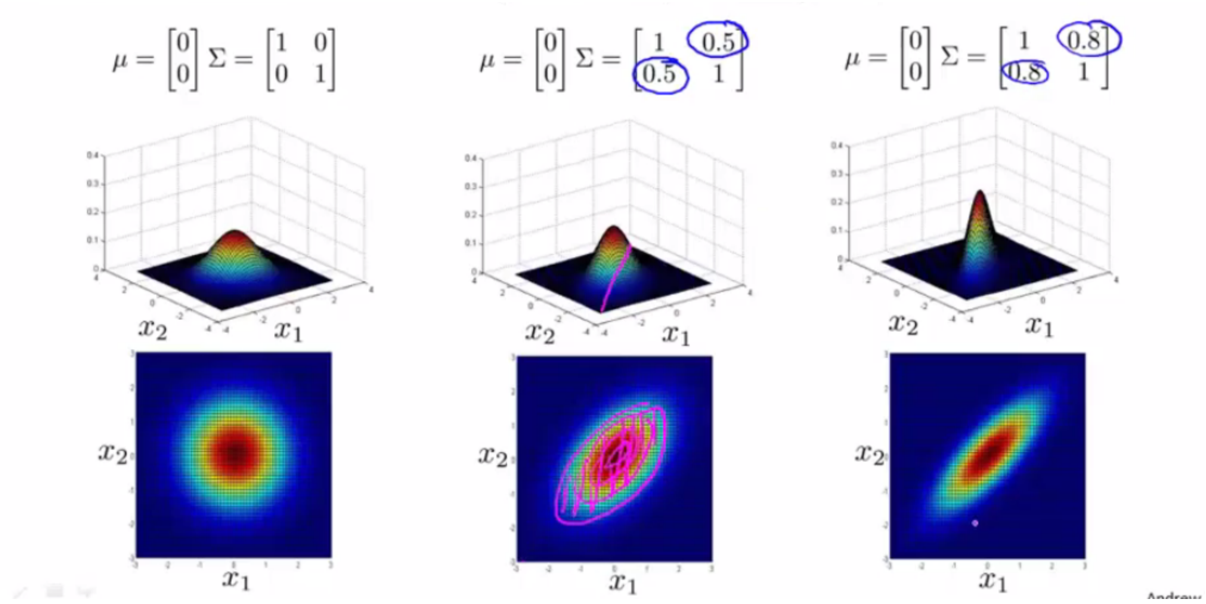
$$S = \Sigma^{\cdot -1} = \begin{pmatrix} \sigma_1^2 & & c \\ & \ddots & \\ c & & \sigma_d^2 \end{pmatrix} \geq 0$$

So Σ^{-1} is *PSD* meaning that $sgn(\Sigma^{-1}) = (1, 0)$.

The expectancy μ determines the center of the gaussian curve

The covariance Σ^{-1} determines the overall shape, orientation and spread of the distribution

The *correlation* term c indicates the correlation between the x_is components of the vectored random variable x. If $c = 0$ there is no correlation so the curve will be ball shaped. If $c > 0$ the more an x_i grows, the more an x_j also does. If $c < 0$ the relationships between the components are inversely proportional.



Gaussian discriminant analysis
GDA is a similar model to logistic regression, but performs better with poor datasets, since it is a generative process. In order to build it, we need to make some assumptions

- $y \sim$ Bernoulli
- $x \,|\, y = 0 \sim \; N(\mu_o, \, \Sigma)$
- $x \,|\, y = 1 \sim \; N(\mu_1, \, \Sigma)$

Those assumptions altogether, constitute the so-called *data generation process*.
We chose to sample x with two M-NGDs but could be any distribution, because $P(y \,|\, x)$ will always be in the form of logistic regression. By sampling x, row by row, a dataset will be constructed.

I. Choosing the distributions

a. $P(y) = \phi^y (1 - \phi)^{1-y}$

b. $P(x \,|\, y = 0) = \dfrac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{ -\dfrac{1}{2} (x - \mu_0)^T \Sigma^{-1} (x - \mu_0) \right\}$

c. $P(x \,|\, y = 1) = \dfrac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{ -\dfrac{1}{2} (x - \mu_1)^T \Sigma^{-1} (x - \mu_1) \right\}$

II. Estimating the parameter

Now we are considering the probability over the entire dataset X.
Let's define the likelihood function L

$$L(\phi, \mu_0, \mu_1, \Sigma) = \log P(X, \; \vec{y}; \; \theta)$$

$$= \log \prod_{i=1}^{n} P\left(x^{(i)}, y^{(i)}; \theta\right) = \sum_{i=1}^{n} \log P\left(x^{(i)}, y^{(i)}\right) = \sum_{i=1}^{n} \log P\left(y^{(i)}\right) + \log P\left(x^{(i)} \,\middle|\, y^{(i)}\right)$$

Now we need to optimize the obtained L by taking the gradient and nulling it

$$\nabla_\theta L(\phi, \mu_0, \mu_1, \Sigma) = 0$$

Coming to the following closed form expressions (I the indicator, same as if) for the parameters

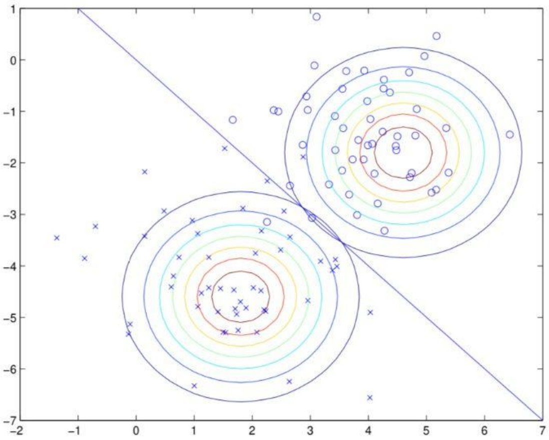
$$\hat{\phi} = \frac{1}{n} \sum_{i=1}^{n} I\left\{ y^{(i)} = 1 \right\}$$

$$\hat{\mu_0} = \frac{\sum_{i=1}^{n} I\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{n} I\{y^{(i)} = 0\}}$$

$$\hat{\mu_1} = \frac{\sum_{i=1}^{n} I\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{n} I\{y^{(i)} = 1\}}$$

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \left(x^{(i)} - \mu_{y^{(i)}}\right) \left(x^{(i)} - \mu_{y^{(i)}}\right)^T$$

Pictorially, what the algorithm is doing can be seen in as follows:



Shown in the figure are the training set, as well as the contours of the two Gaussian distributions that have been fit to the data in each of the two classes. Note that the two Gaussians have contours that are the same shape and orientation, since they share a covariance matrix Σ, but they have different means μ_0 and μ_1. Also shown in the figure is the straight line giving the decision boundary at which $p(y = 1|x) = 0.5$. On one side of the boundary, we'll predict $y = 1$ to be the most likely outcome, and on the other side, we'll predict $y = 0$.