Yuho Kida Follow
Sep 23, 2019 · 6 min read · Listen

Generalized linear models

Introduction to advanced statistical modeling

In this article, I'd like to explain generalized linear model (GLM), which is a good starting point for learning more advanced statistical modeling. Learning GLM lets you understand how we can use probability distributions as building blocks for modeling. I assume you are familiar with linear regression and normal distribution.

Linear regression revisited

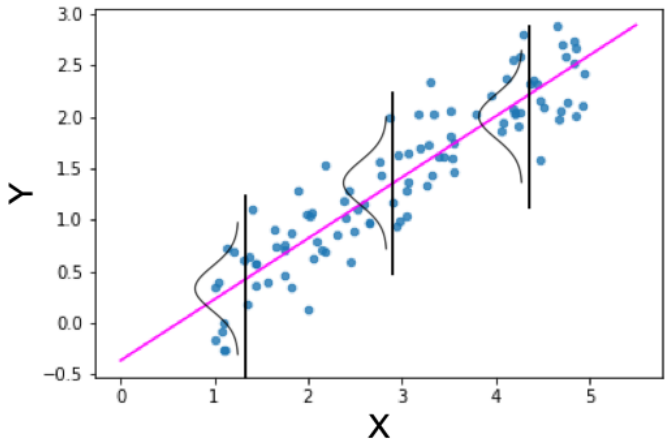
Linear regression is used to predict the value of continuous variable y by the linear combination of explanatory variables X.

In the univariate case, linear regression can be expressed as follows;

$$\mu_i = b_0 + b_1 x_i$$
$$y_i \sim \mathcal{N}(\mu_i, \varepsilon)$$

Linear regression

Here, i indicates the index of each sample. Notice this model assumes normal distribution for the noise term. The model can be illustrated as follows;



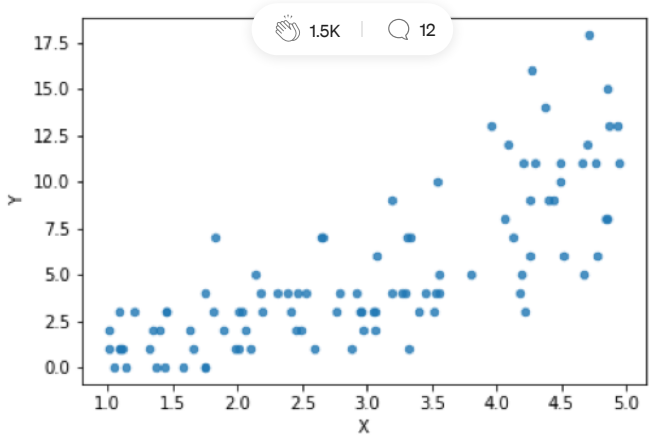
Linear regression illustrated

By the three normal PDF (probability density function) plots, I'm trying to show that the data follow a normal distribution with a fixed variance.

Poisson regression

So linear regression is all you need to know? Definitely not. If you'd like to apply statistical modeling in real problems, you must know more than that.

For example, assume you need to predict the number of defect products (Y) with a sensor value (x) as the explanatory variable. The scatter plot looks like this.



Do you use linear regression for this data?

There are several problems if you try to apply linear regression for this kind of data.

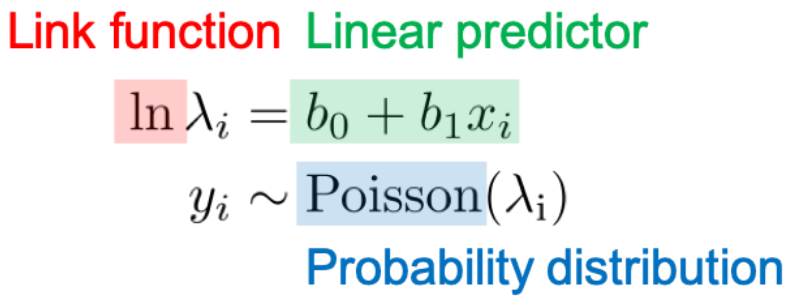
1. The relationship between X and Y **does not look linear**. It's more likely to be exponential.
2. **The variance of Y does not look constant** with regard to X. Here, the variance of Y seems to increase when X increases.
3. As Y represents the number of products, it always has to be a positive integer. In other words, Y is a **discrete variable**. However, the normal distribution used for linear regression assumes continuous variables. This also means the prediction by linear regression can be negative. It's not appropriate for this kind of count data.

Here, the more proper model you can think of is the **Poisson regression** model. Poisson regression is an example of **generalized linear models (GLM)**.

There are three components in generalized linear models.

1. **Linear predictor**
2. **Link function**
3. **Probability distribution**

In the case of Poisson regression, it's formulated like this.



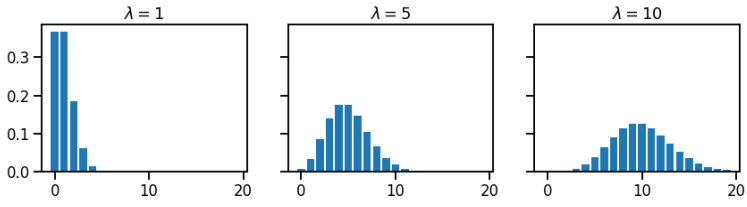
Poisson regression

Linear predictor is just a linear combination of parameter (b) and explanatory variable (x).

Link function literally "links" the linear predictor and the parameter for probability distribution. In the case of Poisson regression, the typical link function is the log link function. This is because the parameter for Poisson regression must be positive (explained later).

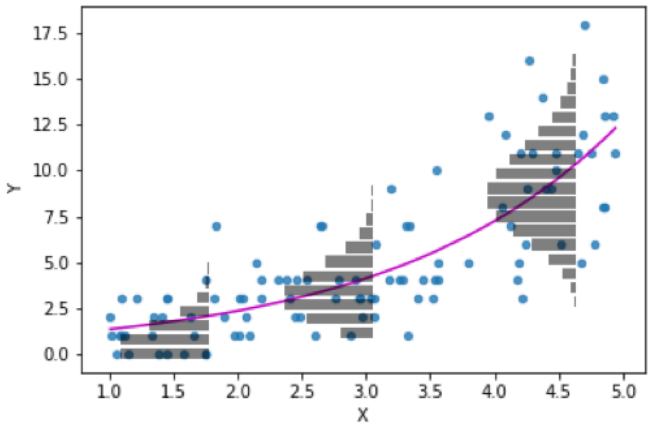
The last component is the **probability distribution** which generates the observed variable y. As we use Poisson distribution here, the model is called Poisson regression.

Poisson distribution is used to model count data. It has only one parameter which stands for both mean and standard deviation of the distribution. This means the larger the mean, the larger the standard deviation. See below.



Poisson distribution with mean=1, 5, 10

Now, let's apply Poisson regression to our data. The result should look like this.



Poisson regression illustrated

The magenta curve is the prediction by Poisson regression. I added the bar plot of the probability mass function of Poisson distribution to make the difference from linear regression clear.

The prediction curve is exponential as the inverse of the log link function is an exponential function. From this, it is also clear that the parameter for Poisson regression calculated by the linear predictor guaranteed to be positive.

$$\ln \lambda_i = b_0 + b_1 x_i$$
$$\Leftrightarrow \lambda_i = \exp(b_0 + b_1 x_i)$$

Inverse of log link function

If you use Python, statsmodels library can be used for GLM. The code for Poisson regression is pretty simple.

```
# Poisson regression code
import statsmodels.api as sm
exog, endog = sm.add_constant(x), y
mod = sm.GLM(endog, exog,
             family=sm.families.Poisson(link=sm.families.links.log))
res = mod.fit()
```

endog (endogenous) and exog (exogenous) are how you call *y* and *X* in statsmodels. Notice you need to add the constant term to *X*. Without this, your linear predictor will be just *b_1*x_i*.

Actually, you don't need to supply link argument here as log link is the default for the Poisson family.

The full code I used to create all the figures is in my Github repository.

Other typical GLM

Linear regression is also an example of GLM. It just uses **identity link function** (the linear predictor and the parameter for the probability distribution are identical) and **normal distribution** as the probability distribution.

$$\mu_i = b_0 + b_1 x_i$$
$$y_i \sim \mathcal{N}(\mu_i, \varepsilon)$$

Linear regression

If you use **logit function** as the link function and **binomial / Bernoulli distribution** as the probability distribution, the model is called **logistic regression**.

$$
\begin{aligned}
z_i &= b_0 + b_1 x_i \\
q_i &= \frac{1}{1 + \exp(-z_i)} \\
y_i &\sim \text{Bern}(q_i)
\end{aligned}
$$

logistic regression

If you represent the linear predictor with z, the above equation is equivalent to the following.

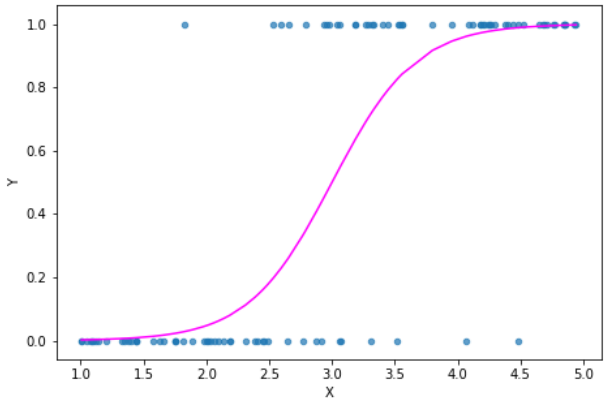
$$
\begin{aligned}
z_i &= b_0 + b_1 x_i \\
q_i &= \frac{1}{1 + \exp(-z_i)}
\end{aligned}
$$

Logistic function

The right-hand side of the second equation is called logistic function. Therefore, this model is called logistic regression.

As the logistic function returns values between 0 and 1 for arbitrary inputs, it is a proper link function for the binomial distribution.

Logistic regression is used mostly for binary classification problems. Below is an example to fit logistic regression to some data.



Logistic regression illustrated

Custom GLM

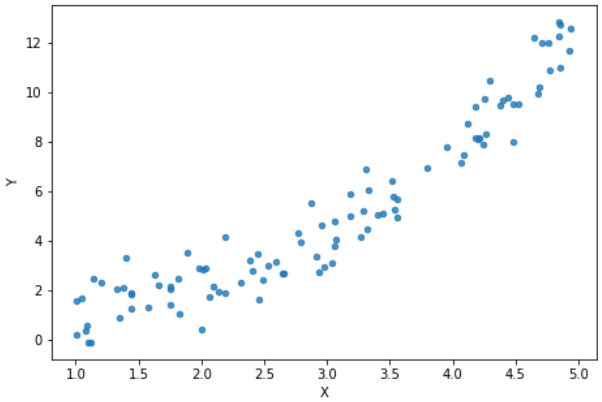
The models I've explained so far uses a typical combination of probability distribution and link function. In other words, all the models above use the **canonical link function.**

This is the list of probability distributions and their canonical link functions.

- Normal distribution: identity function
- Poisson distribution: log function
- Binomial distribution: logit function

However, you don't necessarily use the canonical link function. Rather, the advantage of statistical modeling is that you can make any kind of model that fits well with your data.

For example, let's consider the following data.



This looks similar to the data I prepared for Poisson regression. However, if you see the data carefully, it seems the variance of y is constant with regard to X. Besides, y is continuous, not discrete.

Therefore, it's appropriate to use normal distribution here. As the relationship between X and y looks exponential, you had better choose the log link function.

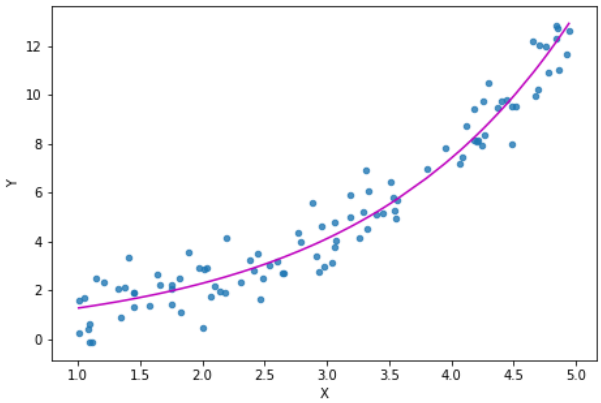
$$\ln \mu_i = b_0 + b_1 x_i$$
$$y_i \sim \mathcal{N}(\mu_i, \varepsilon)$$

GLM with non-canonical link function

With statsmodels you can code like this.

```
mod = sm.GLM(endog, exog,
             family=sm.families.Gaussian(sm.families.links.log))
res = mod.fit()
```

Notice you need to specify the link function here as the default link for Gaussian distribution is the identity link function. The prediction result of the model looks like this.



Various link functions are implemented in statsmodels. However, if you need to use more complex link functions, you have to write models yourself.

For this purpose, probabilistic programming frameworks such as Stan, PyMC3 and TensorFlow Probability would be a good choice. This might be the topic of my future work.