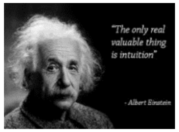
Generalized Linear Methods (GLM)

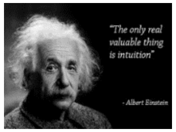


Maximum Likelihood: Intuition

- Maximum Likelihood Estimation (**MLE**) seeks to maximize the **likelihood** that the regression coefficients and the respective predictors **predict** Y **accurately** for all observations – i.e., **"agreement"** between the model predicts and the actual data
- The **Likelihood function** provides the probability that a given observation's predictor values, **predicts** accurately the **outcome** of an observation – e.g., correct **value** or correct **classification**
- **MLE** finds the set of **β** coefficients that **maximizes** this **likelihood** (see math in the textbook.
- As such MLE provides more precise estimates, especially with large samples and when OLS assumptions are not met, e.g., outcome is not normally distributed (e.g., Poisson, Logit), outcome is not continuous (e.g., binary, count).
- However, mathematically, **MLE** and **OLS** yield the **same** results when the **OLS assumptions** are met → OLS is a special case of MLE

Generalized Linear Models (GLM)

- **OLS** models are fit by finding the line that minimizes the **SSE**
- **R fits OLS** regressions with the "Linear Model" **lm()** function, which should only be used when **OLS assumptions** are met
- **Other linear models** are fit with the **GLM**
- In **R**, the Generalized Linear Model **glm()** function estimates several types of **linear models** using **MLE** , (e.g., **logistic** regression)
- **GLM** is a **family** of linear estimation methods that are more generally applicable to **any linear model**, OLS or otherwise
- If **OLS assumptions** are met, GLM results are **identical** to OLS
- Process: many regression **lines** are identified and **fit** (through complex **algorithms**) and a likelihood function is calculated for each → **GLM picks** the line that **maximizes** the **likelihood** function – i.e., **minimizes** the $2LL = -2 * Log\ Likelihood$



-2 Log Likelihood (2LL): Intuition

- Again, the **MLE** method finds the linear model in which the **likelihood** function (i.e., the probability that the model predicts the correct value of every point in the data set) is **maximized**
- **Why the Log?** It is a mathematical **convenience**. A **likelihood** is bound between **0** and **1** → its **log** will be between **-∞** and **1**; It also makes it easy to **compare** likelihoods (of competing models) – i.e., $Log\left(\frac{L_1}{L_2}\right) = Log(L_1) - Log(L_2)$
- **Why the (-)?** 2 reasons: (1) Log likelihoods are **negative**; (2) We want to **maximize** the **Log-Likelihood = minimizing** its **(-)** value, making its **interpretation** similar to **SSE** → we want to **minimize** the $2LL$ → the **model** with **smaller** $2LL$ is best
- **Why the 2?** Another mathematical **convenience**. It has to do with the fact that $Log(X^2) = 2 * Log(X)$ and that $2 * Log - Likelihood$ has a X^2 (Chi-Square) **distribution**, which is convenient for model comparison and testing.

2LL (Deviance) in Predictive Models

- **2LL** is a popular **fit statistic** used in many **GLM** models referred to as **"Deviance"**
- **MLE** and **GLM** models produce various fit statistics; many of them are **based** on **MLE** and **2LL** statistics
- **2LL** can be used to evaluate:
  - **A Single Model** → **Small Deviance** is better
  - Compare **2 nested models** → Log-Likelihood **ratio** test $2LL_R$:

$$2LL_R = -2 * Log\left(\frac{L_{Small}}{L_{Full}}\right) = Deviance_{Small} - Deviance_{Full}$$

    - ✓ A **small** $2LL_R$ means that the 2 models are **similar**, so the **Reduced Model** is preferred (i.e., simpler)
    - ✓ A **large** $2LL_R$ means that the **Full Model** is preferred; it **reduces** the **deviance** substantially
- **OLS** and **MLE** select the same model when OLS assumptions hold

R Tips

`glm{stats}` → The `glm()` function works exactly like the `lm()` function, except that it accepts more error distributions than just normal distribution. It is widely used for logistic regression and other non-OLS models

```
glm.fit = glm(y~x1+x2+etc., data=dataName,
              family=binomial(link="logit"))
```

→ a `glm()` function to run a logistic model; `family=binomial` is for binomial logistic and `link="logit"` is the link function – i.e., the function used to transform the predicted variable y.

`logLik(glm.fit)` → Displays the log-likelihood

`deviance(glm.fit)` or `-2*logLik(glm.fit)` → deviance fit statistic; small deviance is better

`AIC(glm.fit)` → Akaike Information Criterion = deviance + penalty for model size (like an adjusted R^2)

NOTE: the `glm()` default is `family=gaussian(link="identity")`, which yields an OLS fit (`gaussian` is a normal distribution and `"identity"` means no transformation). Therefore, `glm(y~x1+x2+etc, data=etc.)` yields the **exact** same results at the OLS linear model `lm(y~x1+x2+etc, data=etc.)`, except that the former give you `2LL` stats and the latter give you R^2.

KOGOD SCHOOL
of
BUSINESS