Supervised Learning

Generalized linear models (GLMs)

June 22nd, 2021

Probability distributions

A **distribution** is a mathematical function $f(x|\theta)$ where

- x may take on continuous or discrete values over the *domain* (i.e. all possible inputs) of $f(x|\theta)$
- θ is a set of parameters governing the shape of the distribution
    - e.g. $\theta = \{\mu, \sigma^2\}$ for a Normal / Gaussian distribution)
- the $|$ symbol means that the shape of the distribution is *conditional* on the values of θ
- $f(x|\theta) \geq 0$ for all x
- $\sum_x f(x|\theta) = 1$ or $\int_x f(x|\theta) = 1$.

We use f to denote the distribution for its:

- **probability density function (PDF)** if x is continuous
- **probability mass function (PMF)** if x is discrete

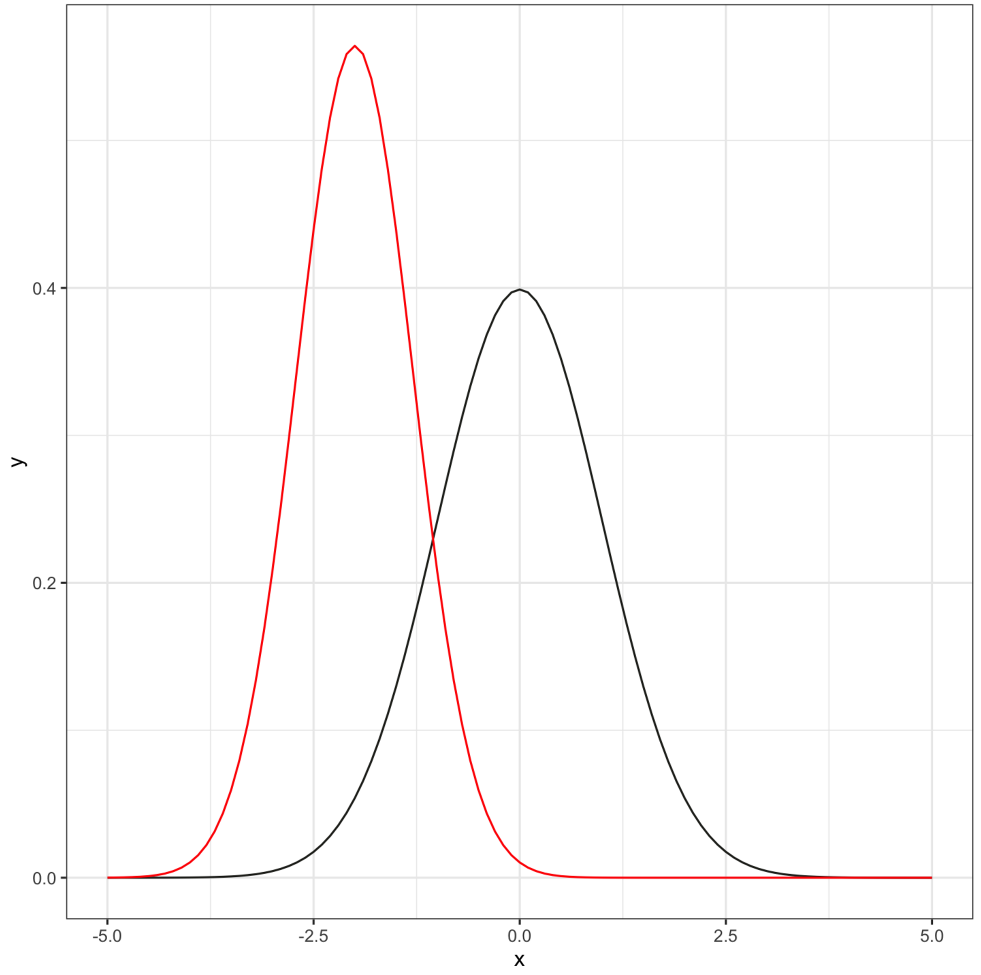
Probability distribution examples: Normal distribution

Normal distribution PDF (`dnorm`):

$$f(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- we write $X \sim N(\mu, \sigma^2)$
- **standard Normal**: $N(0, 1)$
- can plot density curves with **stat_function()**

```
tibble(x = c(-5, 5)) %>%
  ggplot(aes(x)) +
  stat_function(fun = dnorm, n = 101,
                args = list(mean = 0, sd = 1))
  stat_function(fun = dnorm, color = "red",
                args = list(mean = -2,
                            sd = sqrt(0.5)))
  theme_bw()
```



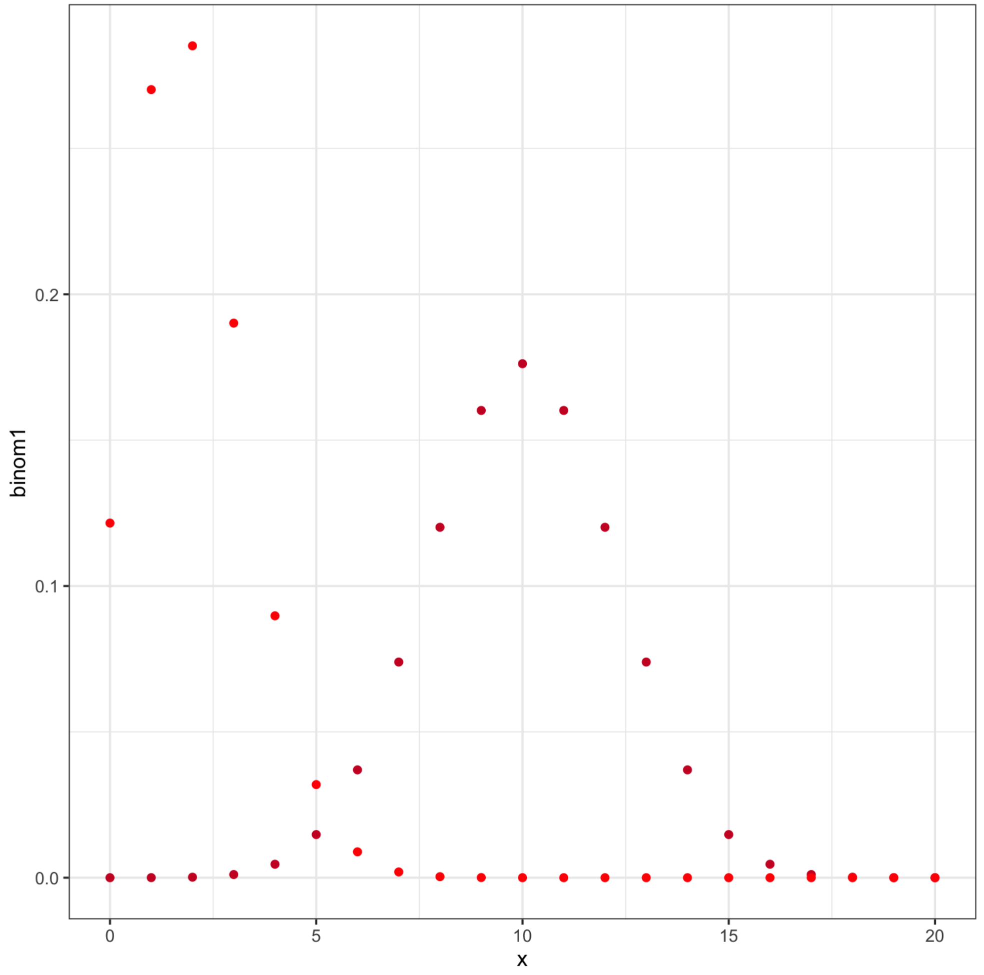
Probability distribution examples: binomial distribution

Binomial distribution PMF (`dbinom`):

$$f(x|n,p) = \binom{n}{x} p^x (1-p)^{n-x}$$

- model for the probability of x successes in n independent trials (`size`), each with success probability of p (`prob`)
- we write $X \sim \text{Binomial}(n, p)$
- `R` uses `d` for both PDFs and PMFs

```
tibble(x = 0:20) %>%
  mutate(binom1 = dbinom(x, size = 20,
                         prob = 0.5),
         binom2 = dbinom(x, size = 20,
                         prob = 0.1)) %>%
  ggplot(aes(x)) + geom_point(aes(y = binom1)
  geom_point(aes(y = binom2), color = "red")
  theme_bw()
```



Distributions and regression

Why does this matter?

- Because linear regression, and generalized variants, **make assumptions** about how observed data are distributed around the true regression line, conditional on a value of x

For simple linear regression, our goal is to estimate $E[Y|x]$, assuming that for every value of x...

- the distribution governing the possible values of Y is a **Normal distribution**
  - *Note:* capitalize Y because values are **random variables** (random samples from distribution)
- the **mean** of the Normal distribution is $E[Y|x] = \mu(y|x) = \beta_0 + \beta_1 x$
- the **variance** of the Normal distribution is σ^2, which is a constant (i.e., does not vary with x)
- $\Rightarrow Y|x \sim N(\beta_0 + \beta_1 x, \sigma^2)$, same as yesterday: $Y = \beta_0 + \beta_1 x + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$

However, just because these assumptions are made in simple linear regression doesn't mean that all linear regression-related models utilize the same assumptions. **They don't**. When we step back from these assumptions, we enter the realm of **generalized linear models (GLMs)**.

Maximum likelihood estimation

In generalized regression, we

1. assume a (family of) distribution(s) that govern observed response values Y, and
2. estimate the parameters θ of that distribution.

Estimation is done by maximizing the **likelihood function**:

$$\mathcal{L} = \prod_{i=1}^{n} f(Y_i|\theta)$$

to find the **maximum likelihood estimators (MLEs)** (typically maximize $l = \log \mathcal{L}$, the **log-likelihood**)

Leaving many details under the rug:

- the maximum is the point at which the derivative of the likelihood function is zero
- you don't need to check the second derivative: wherever the derivative equals zero, it's a maximum value, not a minimum value

MLE for regression

Determining the value of θ that achieves the maximum likelihood can be difficult

It may require **numerical optimization**

- wherein the computer, using an algorithm, searches over possible values of θ to find the optimal one

For linear regression, $\mathcal{L}$ can be maximized analytically:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

- the $\hat{\boldsymbol{\beta}}$ estimates that minimize the **residual sum of squares (RSS)** are the MLEs!
- Unbiased estimate for $\hat{\sigma}^2$ is $= \frac{RSS}{n-(p+1)}$
- This enables us to perform statistical inference:
    - Hypothesis testing for coefficients from yesterday
    - Confidence intervals and prediction intervals

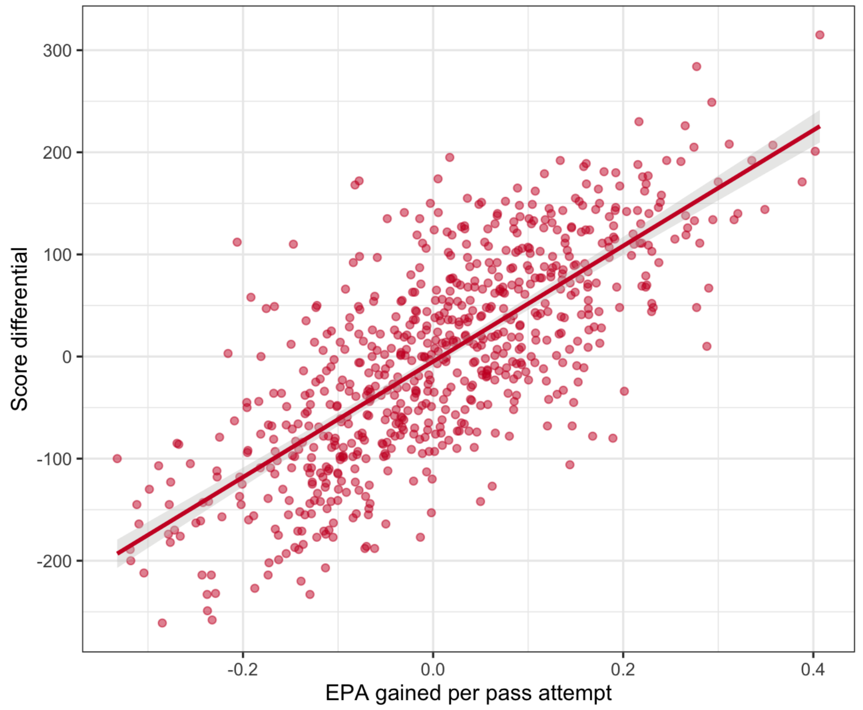
Example: NFL teams summary and modeling score differential

Created dataset using `nflfastR` summarizing NFL team performances from 1999 to 2020

```r
library(tidyverse)
nfl_teams_data <- read_csv("http://www.stat.cmu.edu/cmsac/sure/2021/materials/data/regression_pro
  mutate(score_diff = points_scored - points_allowed)
init_lm <- lm(score_diff ~ offense_ave_epa_pass, data = nfl_teams_data)
```

- `geom_smooth()` displays **confidence intervals** for the regression line

```r
nfl_plot <- nfl_teams_data %>%
  ggplot(aes(x = offense_ave_epa_pass,
             y = score_diff)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  theme_bw() +
  labs(x = "EPA gained per pass attempt",
       y = "Score differential")
nfl_plot
```



Confidence intervals versus prediction intervals

Regression **confidence intervals** are based on standard errors for the estimated regression line at x^*:

$$SE_{\text{line}}\left(x^*\right) = \hat{\sigma} \cdot \sqrt{\frac{1}{n} + \frac{\left(x^* - \bar{x}\right)^2}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}}$$

Regression **prediction intervals** add the variance of a **single predicted value** σ^2:

$$SE_{\text{pred}}\left(x^*\right) = \hat{\sigma} \cdot \sqrt{1 + \frac{1}{n} + \frac{\left(x^* - \bar{x}\right)^2}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}}$$

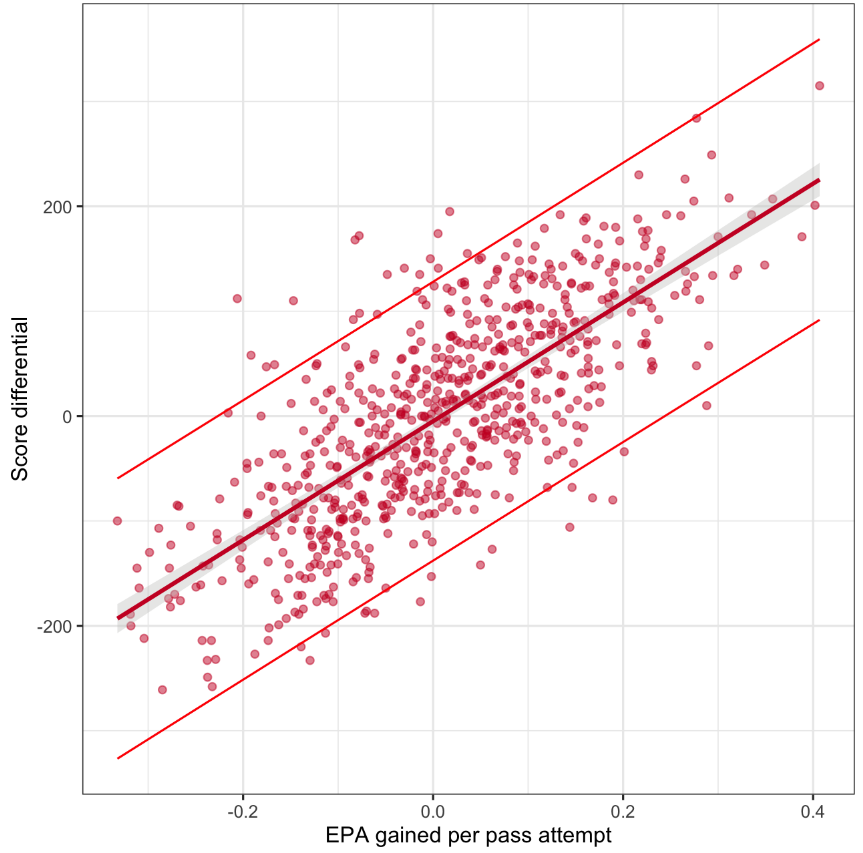
- $SE_{\text{line}}\left(x^*\right) \Rightarrow$ std error for the predicted AVERAGE $\hat{\beta}_0 + \hat{\beta}_1 x^*$
- $SE_{\text{pred}}\left(x^*\right) \Rightarrow$ std error for the prediction of an observation $\hat{\beta}_0 + \hat{\beta}_1 x^* + \epsilon$
- Why does the standard error for a prediction never go to 0 as n goes to ∞?

Confidence intervals versus prediction intervals

Generate 95% intervals with $\hat{Y}^* +/- 2 \cdot SE_{type}(x^*)$

```r
pred_int_data <-
  predict(init_lm, data = nfl_teams_data,
                interval = "prediction",
                level = .95) %>%
  as_tibble()
nfl_plot +
  geom_ribbon(data =
                bind_cols(nfl_teams_data,
                          pred_int_data),
              aes(ymin = lwr, ymax = upr),
              color = "red", fill = NA)
```

Subtle point: both are **confidence intervals**...



Generalization example

In typical linear regression, the distribution is Normal and the domain of $Y|x$ is $(-\infty, \infty)$.

What, however, happens if we know that

1. the domain of observed values of the response is actually $[0, \infty]$? and
2. the observed values are **discrete**, with possible values 0, 1, 2, ...

The Normal distribution doesn't hold here

- Any idea of what distribution could possibly govern $Y|x$?
- Remember, we might not know truly how $Y|x$ is distributed, but any assumption we make has to fit with the limitations imposed by points 1 and 2 above

Generalization: Poisson regression

A distribution that fulfills the conditions imposed on the last slide is the **Poisson** distribution,

$$f(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{ where } x = 0, 1, 2, \ldots$$

- has a single parameter λ, which is **both** the mean **AND** variance of the distribution
  - in general the variance governs the distribution's shape
- distribution of independent event occurences in an interval, e.g. soccer goals in a match
- λ is the average number of the events in an interval

So, when we apply generalized linear regression in this context, we would identify the family as Poisson.

But there's another step in generalization…

Generalization: link Function

Start with one predictor, linear function: $\beta_0 + \beta_1 x$

Range of this function is $(-\infty, \infty)$ - but for Poisson regression example, we know that Y **cannot be negative**,

- **We need to transform the linear function** to be $[0, \infty)$! (We could punt and use simple linear regression, but results may not be meaningful, e.g., we predict $\hat{Y}$ to be negative!)

There is usually no unique transformation, but rather conventional ones

- e.g., for Poisson we use the $log()$ function as the **link function** $g()$:

$$g(\lambda|x) = \log(\lambda|x) = \beta_0 + \beta_1 x$$

Given Y with values limited to being either 0 or positive integers, with no upper bound, we

1. assume $Y|x \sim \text{Poisson}(\lambda)$
2. assume $\lambda|x = e^{\beta_0 + \beta_1 x}$
3. use optimization to estimate β_0 and β_1 by maximizing the likelihood function

Exercises

What family might be appropriate for…

1. $Y|x$ continuous, but bounded between 0 and 1? Beta distribution
2. $Y|x$ continuous, but bounded between 0 and ∞? Gamma distribution
3. $Y|x$ discrete, but can only take on the values 0 and 1? Bernoulli distribution

Focus on Case 3 for Thursday!