Cases of GLMs -

1) Normal

2) Poisson

3) Binomial
   a) $h(\mu) = \log \frac{p}{1-p}$
      $\theta = \frac{e^{x'\beta}}{1 + e^{x'\beta}}$  logit link
      
      "logistic regression"
      
   b) $h(\mu) = \Phi^{-1}(\frac{\mu}{\sigma}) = \Phi^{-1}(p)$
      $p = \Phi(x'\beta)$  "probit link"
      
   c) $h(\mu) = \log(-\log(1-\frac{\mu}{\sigma}))$
      $\log(-\log(1-p))$  "complementary log-log link"
      
      gives  $p = 1 - e^{-e^{x'\beta}}$

"Canonical" (mathematically most natural/convenient) link for
a GLM is $\theta = x'\beta$ i.e. is $h = b^{-1}$

---

One last General Point (about inference in all models to date) --- made in Section 7.3.2 of trying outline

Big model with parametric vector $\tilde{\Theta} = \begin{pmatrix} \Theta_1 \\ px1 \\ \Theta_2 \\ (r-p)x1 \end{pmatrix}$
Smaller/reduced model with parameter vector \( \beta \). This could be any fixed/known vector.

To test \( H_0: \beta_1 = 0 \) in the big model (i.e. test if the big model that the sub-model that the smaller one is adequate)

I can use a LRT and thus the statistic

\[
X^2 = 2 \left( \ln(\hat{\theta}_{\text{MLE}}) - \ln(\theta_0, \hat{\beta}_2(\theta_0)) \right)
\]

(maximized log-likelihood for big model in small model)

and an approximate \( \chi^2 \) null dist. (for estimating p-values)

For the LM, this statistic (for \( H_0: \beta_1 = \beta_2 = \ldots = \beta_p = 0 \))

can be shown to reduce to the Full model/Reduced model

F statistic (and I don't need to resort to the \( \chi^2 \) approximation)

In other models (non-linear regression, MLM, GLM, etc.) this statistic and \( \chi^2 \) approximation provides a way of comparing nested models - so I can use this to test all main effects = 0 in a GLM, e.g.
"Smoothing" / Model - Free Curve Fitting / Nonparametric Regression

(Chill at Faraway Book II)

(\(x_i, y_i\))

There are many applications where the "obvious" ways of modeling how \(y\) changes with \(x\) just fail to be flexible enough. I need a flexible way of "fitting curves" - Chill discusses a number of such means.

Bin "Smoothing"

\[ y \]

Partition the \(x\)-axis into "bins" and coak up a step function to approximate the relationship of \(y\) to \(x\):

\[ g(x) = \text{mean} \text{ or median} \text{ of } y_i \text{'s with } x_i \text{'s in the same bin as } x \]

or trimmed mean.
Running Smoothers use a different "bin" for each $x$

$k$ Nearest Neighbor Versions

A symmetric version would be to for $\hat{y}(x)$ use

- the median
- the mean
- the trimmed mean

of $\frac{k}{2}$ $y_i$'s corresponding to $x_i < x$

and $\frac{k}{2}$ $y_i$'s corresponding to nearest $x_i > x$

- a possibly non-symmetric version is to apply (mean or median or trimmed mean)

- to $k$ $y_i$'s with $x_i$'s closest to $x$ of interest (without regard to how they split in terms of being above or below $x$)

The $\hat{y}$ could be the value of a fitted regression function (based on $k$ nearest neighbors) - i.e. I could fit each $x$

1) identify a set of $k$ neighbors
2) fit $y = b_0 + b_1 x \Rightarrow \text{to } k \text{ points by } OLS$
3) use $\hat{y}(x) = b_0 + b_1 x$
This "local regression" idea one version of the list

\[ \text{mean} \quad \text{or} \quad \text{median} \]

- this treats all points in the neighborhood of \( x \)

at which I'm trying to predict equally (and I completely ignore all points just outside the neighborhood) - because of how least squares works it is the neighbors furthest from \( x \) at which I'm predicting that get most influence