第8章 广义线性模型：分类

中山大学人工智能学院
毛旭东

Email：maoxd3@mail.sysu.edu.cn

15.4 广义线性模型

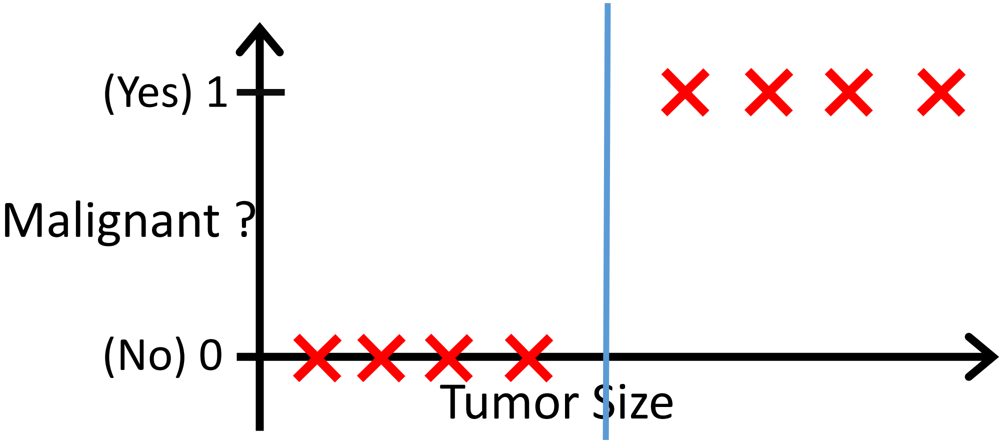
- 广义线性模型（Generalized Linear Model，GLM）：

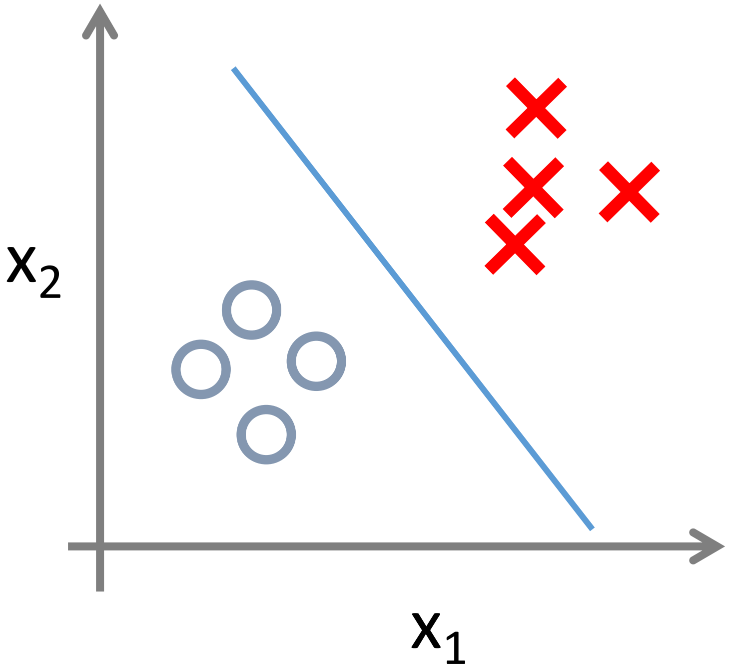
$$\mu = f\left(\mathrm{lin}(x)\right)$$

$$y \sim \mathrm{pdf}(\mu, [\text{其他参数}])$$

- 其中，lin()是线性函数。
- 自变量类型：
  - 度量值
  - 类别值
- 因变量类型：
  - 度量值
  - 类别值
  - 顺序值
  - 计数值

分类问题（Classification）





- 分类问题：因变量是类别值。

(图来源：Andrew Ng's Machine Learning)

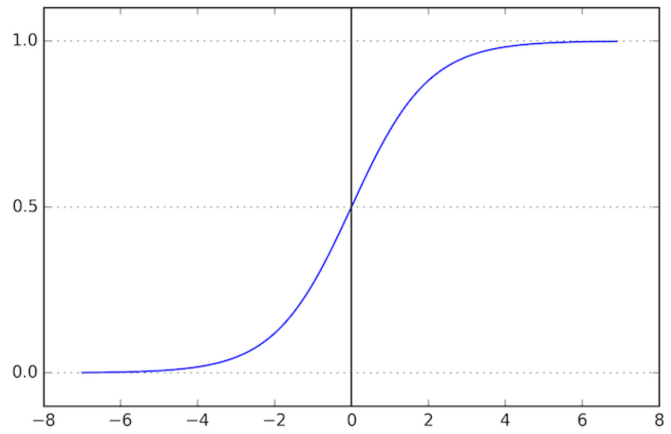
15.3 从自变量的线性组合到含噪声的数据
15.3.1 从自变量到因变量的集中趋势

- 我们通过反向链接函数（inverse link function）将自变量的线性组合映射到因变量：

$$y = f(\mathrm{lin}(x))$$

- 对于回归问题，f是恒等函数，即$f\big(\mathrm{lin}(x)\big) = \mathrm{lin}(x)$， $f(\mathrm{lin}(x))$是高斯分布的均值μ。
- 对于二分类问题，采用伯努利分布，$f\big(\mathrm{lin}(x)\big)$表示伯努利分布的均值$\mu$（即，参数$\theta$）。
- 需要找一个f，使得$f\big(\mathrm{lin}(x)\big)$的值域是[0,1]。

15.3.1.1 逻辑函数（logistic function）



- 逻辑函数：

$$y = \text{logistic}(x) = \frac{1}{1 + e^{-x}}$$

- 逻辑函数是常用的反向链接函数：

$$y = \text{logistic}\big(\text{lin}(x)\big) = \frac{1}{1 + e^{-\text{lin}(x)}}$$

- $\text{logistic}\big(\text{lin}(x)\big)$用于表示伯努利分布的均值，即参数$\theta$。

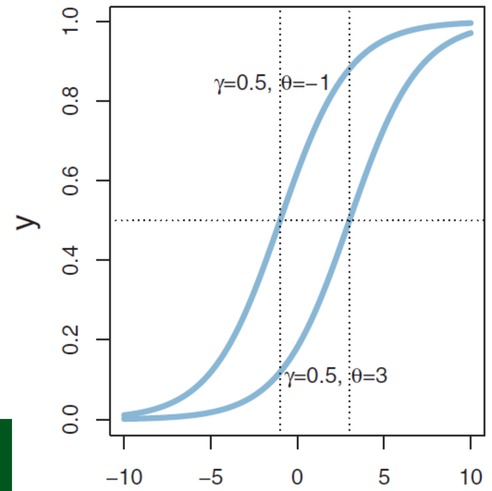
- 逻辑函数又被称为sigmoid function。

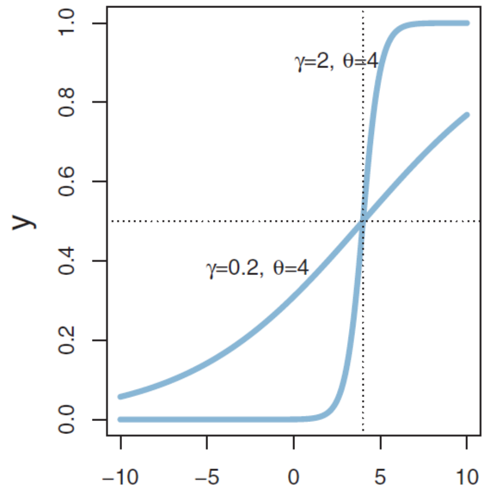
逻辑函数

$$y = \text{logistic}(\mathrm{x}; \beta_0, \beta_1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

$$= \text{logistic}(\mathrm{x}; \gamma, \theta\ ) = \frac{1}{1+e^{-\gamma(x-\theta)}}$$

- γ（即β_1）控制图形的陡峭程度。
- θ（即$-\frac{\beta_0}{\beta_1}$）控制图形的$y=0.5$的位置。





logit函数

- logit函数是逻辑函数的反函数：

$$\text{logit}(x) = \log(\frac{x}{1-x})$$

- 其中，$0 < x < 1$。
- 可得：$\text{logit}(\text{logistic}(x)) = x$
- 广义线性模型中，可得两种等价写法：

$$\mu = \text{logistic}\big(\text{lin}(x)\big)$$
$$\text{logit}(\mu) = \text{lin}(x)$$

- logit名字的来源是“**log unit**”。

21.1 二分类

- 逻辑回归（logistic regression）：

$$\mu = \text{logistic}\left(\sum_k \beta_k x_k + \beta_0\right)$$

$$y \sim \text{Bernoulli}(\mu)$$

- 其中，$k = \{1,2 \ldots, K\}$。
- 参数:
- β_k， $k = \{1,2 \ldots, K\}$
- β_0

数据标准化

- 只对度量值的自变量做标准化：

$$z_j = \frac{x_j - \mu_{x_j}}{\sigma_{x_j}}$$

$$\mu = \operatorname{logistic}\left(\sum_j \zeta_j z_j + \zeta_0\right)$$

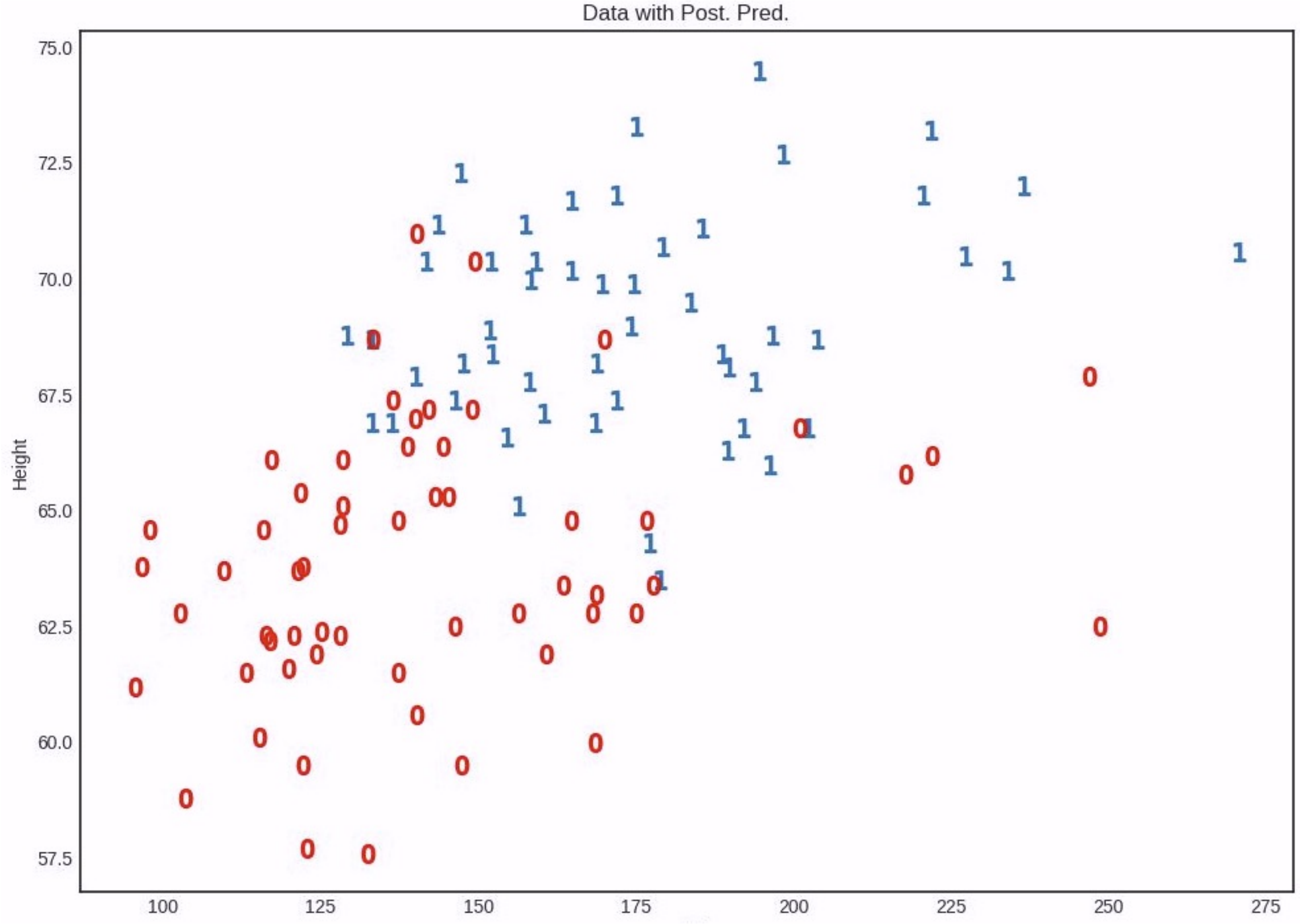
- 还原：

$$\operatorname{logit}(\mu) = \sum_j \zeta_j z_j + \zeta_0$$

$$= \sum_j \frac{\zeta_j}{\sigma_{x_j}} x_j + \zeta_0 - \sum_j \frac{\zeta_j}{\sigma_{x_j}} \mu_{x_j}$$

例子

- 用身高和体重，来预测性别。



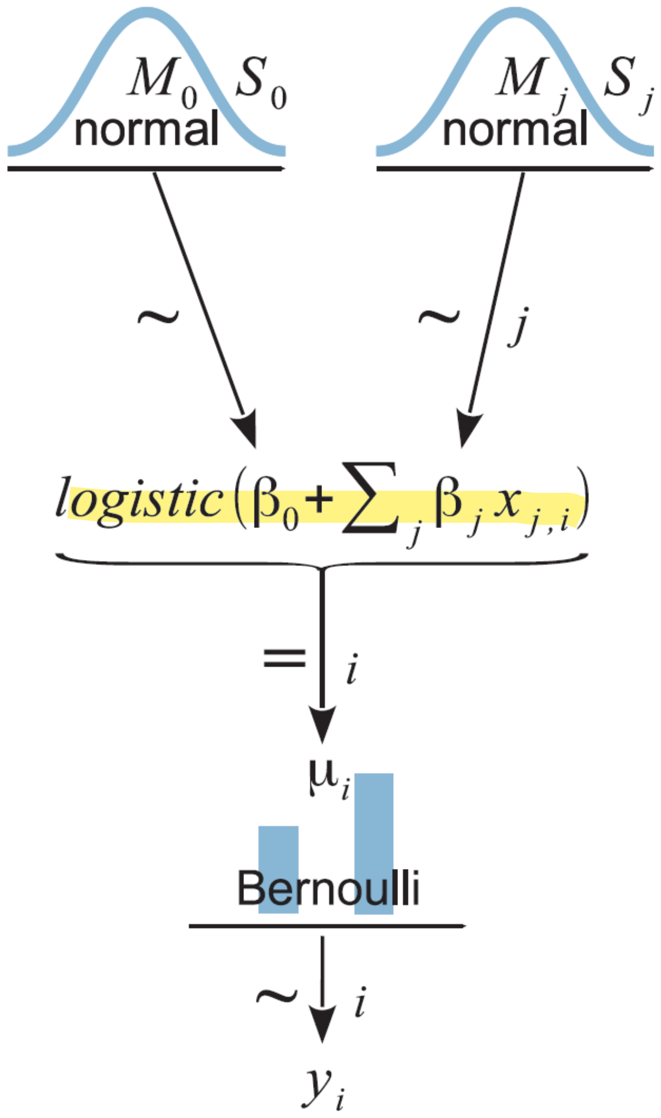
例子：只用体重预测性别

- 模型（似然）：

$$\mu = \text{logistic}(\beta_1 x + \beta_0)$$
$$p(y|\beta_1, \beta_0) = \text{bernoulli}(y|\mu)$$

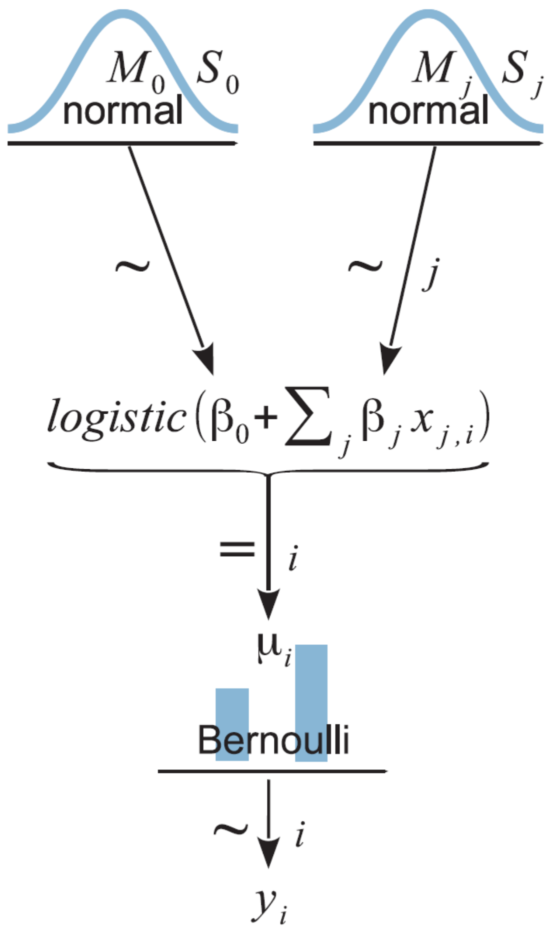
- 先验：

$$\beta_0 \sim \text{normal}(\mu_0, \sigma_0)$$
$$\beta_1 \sim \text{normal}(\mu_1, \sigma_1)$$



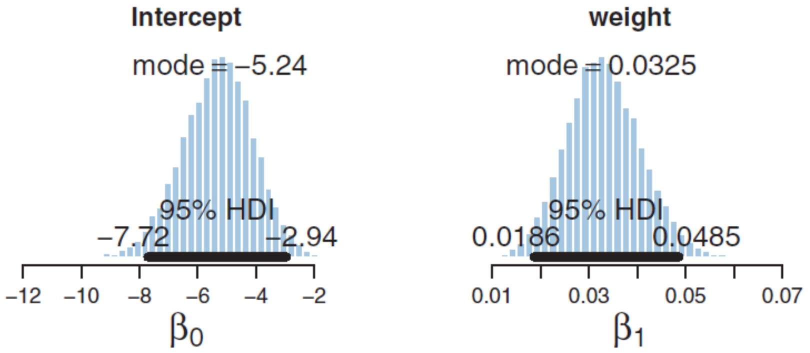
代码

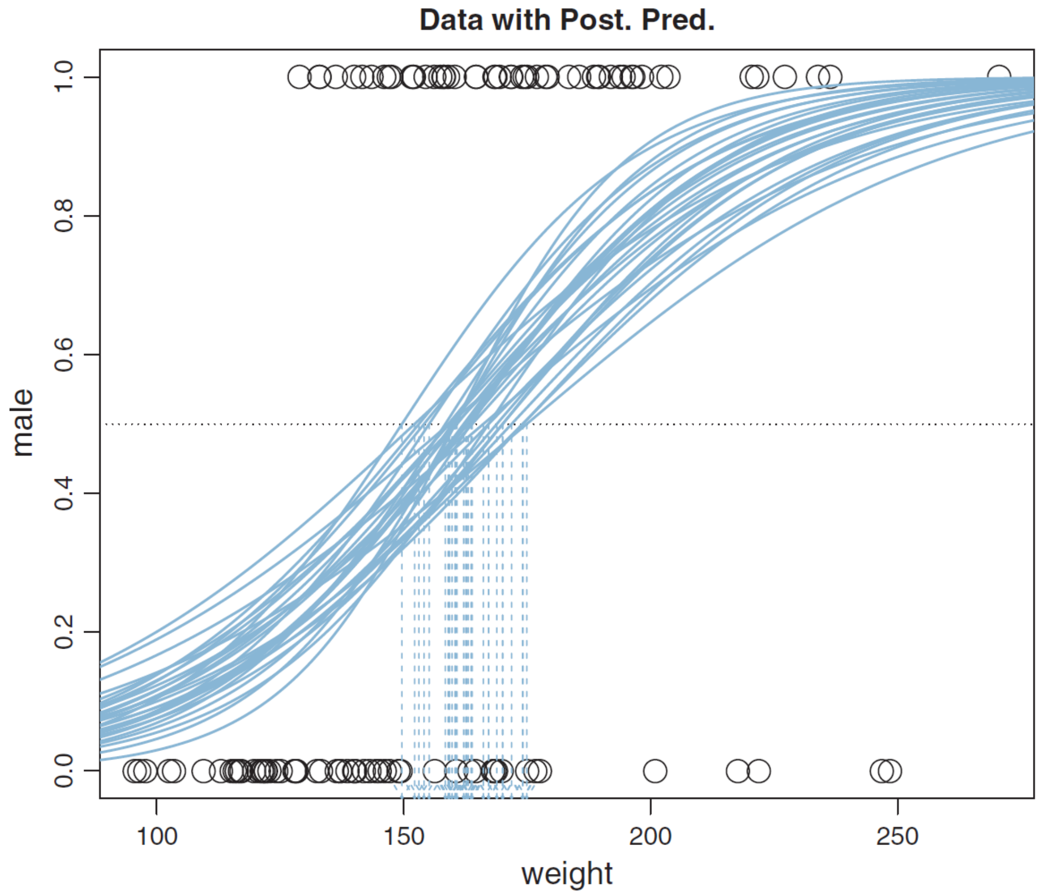
```
X = df['weight']
y = df['male']

meanx = X.mean()
scalex = X.std()
zX = ((X-meanx)/scalex).values

with pm.Model() as model_weight:

    zbeta0 = pm.Normal('zbeta0', mu=0, sd=2)
    zbetaj = pm.Normal('zbetaj', mu=0, sd=2)

    p = pm.invlogit(zbeta0 + zbetaj*zX)

    likelihood = pm.Bernoulli('likelihood', p, observed=y.values)

with model_weight:
    trace = pm.sample(3000, cores=4)

beta0 = trace['zbeta0'] - trace['zbetaj']*meanx/scalex
betaj = (trace['zbetaj']/scalex)
```



结果

- 随着体重增加，是男性的概率增加。
- β_1的HDI是大于0。
  - 随着体重增加，更可能是男性。
- 曲线不是很陡。
  - 没有很确定的阈值区分男女的体重。





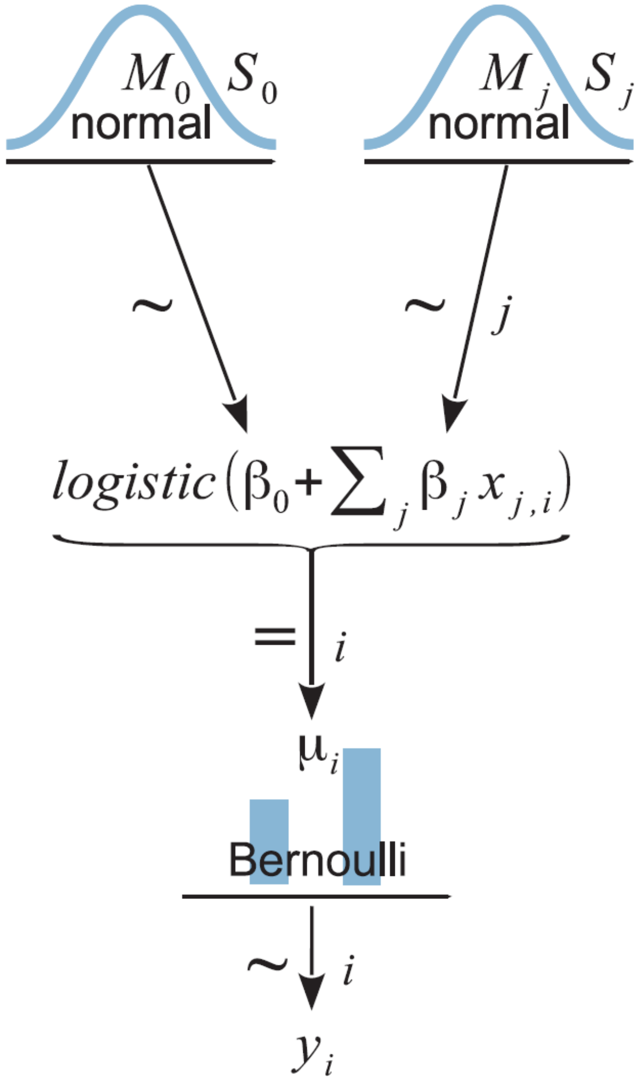
例子：用身高和体重预测性别

- 模型：

$$\mu = \text{logistic}(\beta_2 x_2 + \beta_1 x_1 + \beta_0)$$
$$p(y|\beta_2,\beta_1,\beta_0) = \text{bernoulli}(y|\mu)$$

- 先验：

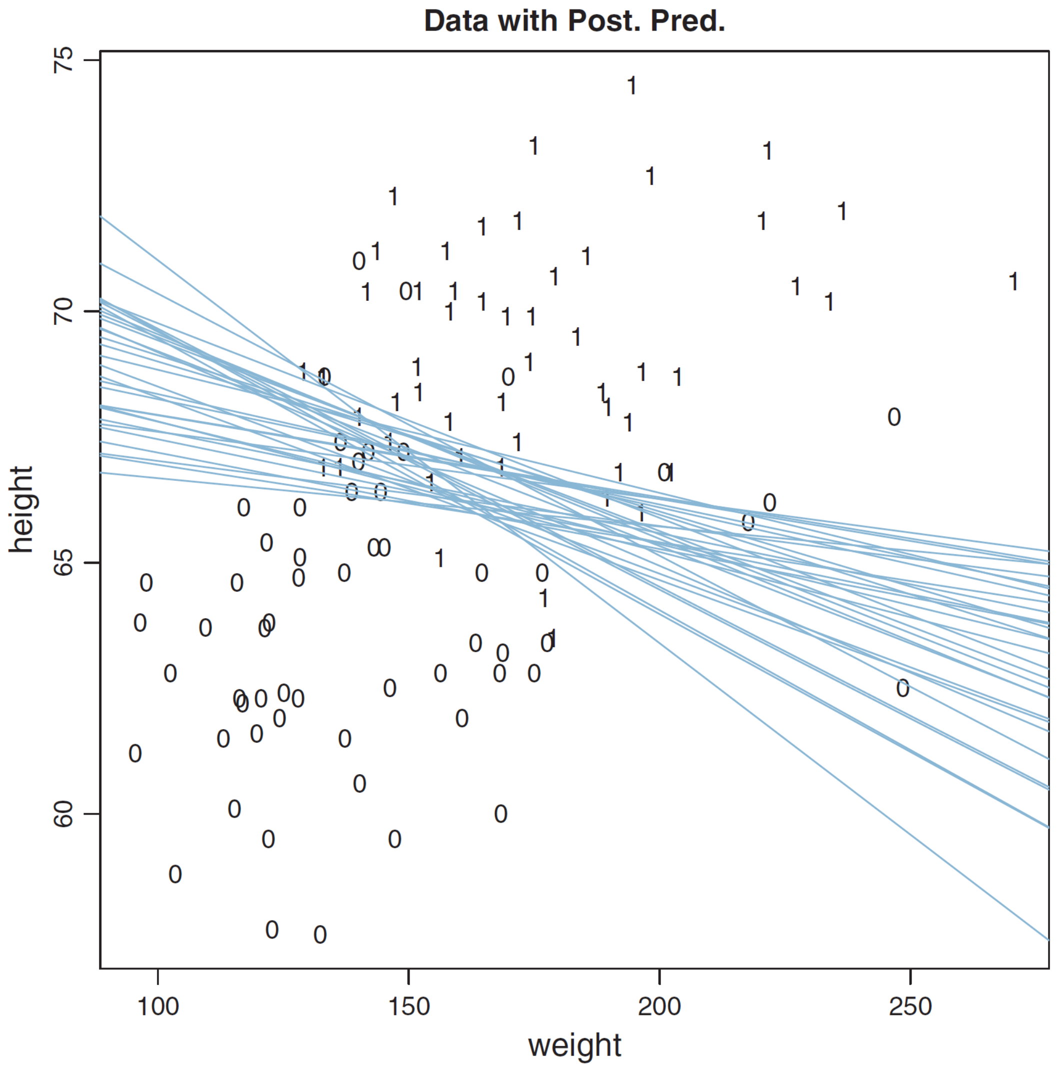
$$\beta_0 \sim \text{normal}(\mu_0,\sigma_0)$$
$$\beta_1 \sim \text{normal}(\mu_1,\sigma_1)$$
$$\beta_2 \sim \text{normal}(\mu_2,\sigma_2)$$



结果

$$y = \text{logistic}(\text{lin}(x)) = \frac{1}{1 + e^{-\text{lin}(x)}}$$

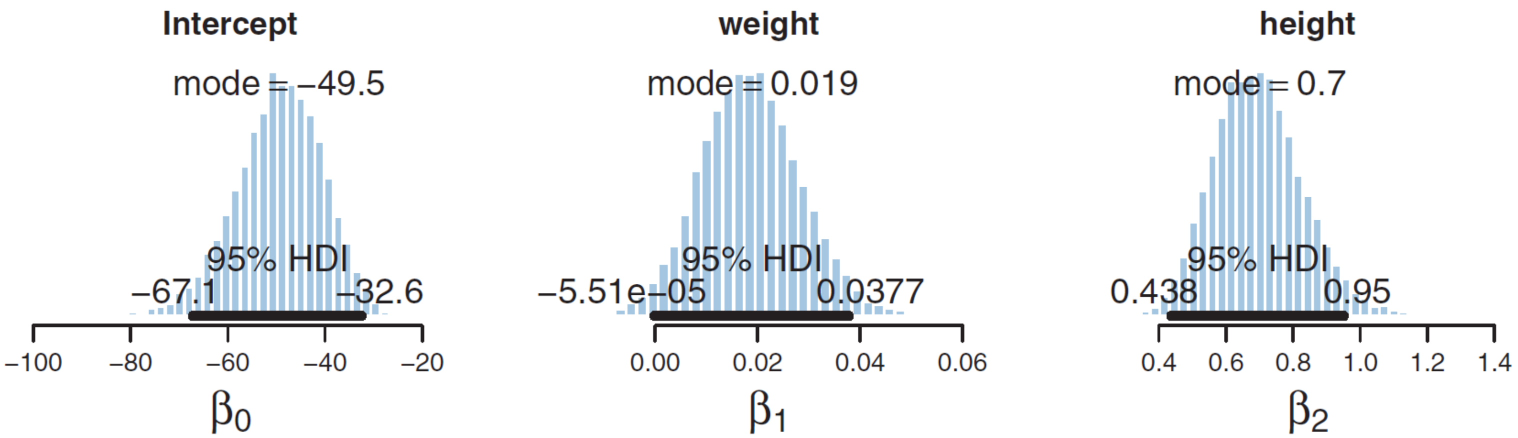
- 直线表示$\mu = 0.5$。
  - $\beta_2 x_2 + \beta_1 x_1 + \beta_0 = 0$
- 直线的分布显示了参数估计的确定性。
  - 直线越集中，越确定。
- 垂直直线的方向，表示概率值变化最快的方向。
- 直线的角度表明概率值变化速度，身高的方向比体重快。



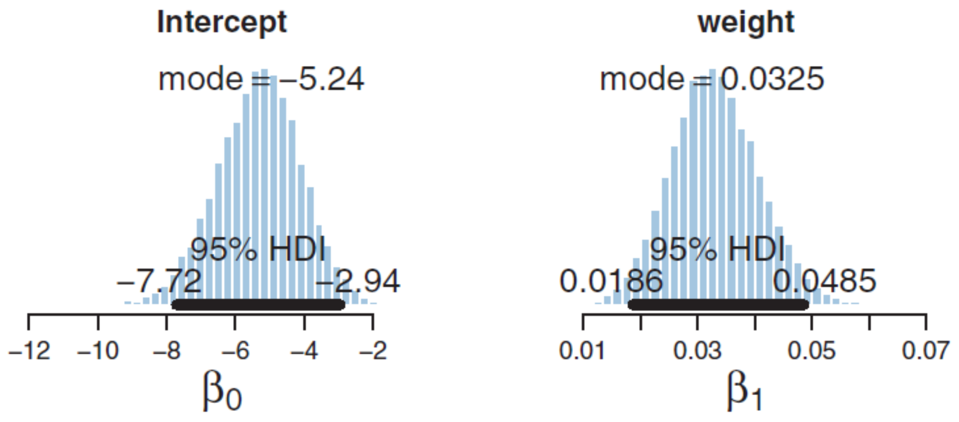
结果

- β_2大于β_1，也是说明概率值变化速度，身高的方向比体重快。
- β_1比“只用体重”的β_1小。
  - 因为身高和体重是有相关性的，身高把体重的信息包含了。

用体重和身高：



只用体重：



21.2 回归系数解释

$$\text{logit}(\mu) = \beta_2 x_2 + \beta_1 x_1 + \beta_0$$

- x_2每增加1，右式增加β_2，含义是什么?
- 在Bernoulli分布中，$\mu = p(y = 1)$，可得:

$$\text{logit}(\mu) = \log\frac{\mu}{1-\mu} = \log\frac{p(y=1)}{p(y=0)}$$

- $\frac{p(y=1)}{p(y=0)}$称为$y = 1$ 对$y = 0$ 的概率比值（odds）。
- $\log\frac{p(y=1)}{p(y=0)}$称为log odds。
  - “**log** unit”（logit）的unit指的就是odds。

21.2 回归系数解释

$$\mathrm{logit}(\mu) = \beta_2 x_2 + \beta_1 x_1 + \beta_0$$

- 因此，x_2每增加1，右式增加β_2，表示log odds增加β_2。
  - 也就是log($y = 1$ 对$y = 0$ 的概率比值)增加β_2。
- 例子：$\mu = \mathrm{logistic}(0.7x_2 + 0.02x_1 - 50)$
- 对于$x_2 = 63$，$x_1 = 160$：
- $\mu = 0.063$，log odds$= \log\frac{0.063}{1-0.063} = -2.7$
- 对于$x_2 = 64$，$x_1 = 160$：
- $\mu = 0.119$，log odds$= \log\frac{0.119}{1-0.119} = -2$

21.2.2 数据不均衡问题

- 前面的例子中，$y=1$和$y=0$的样本数量差不多刚好是各50%。

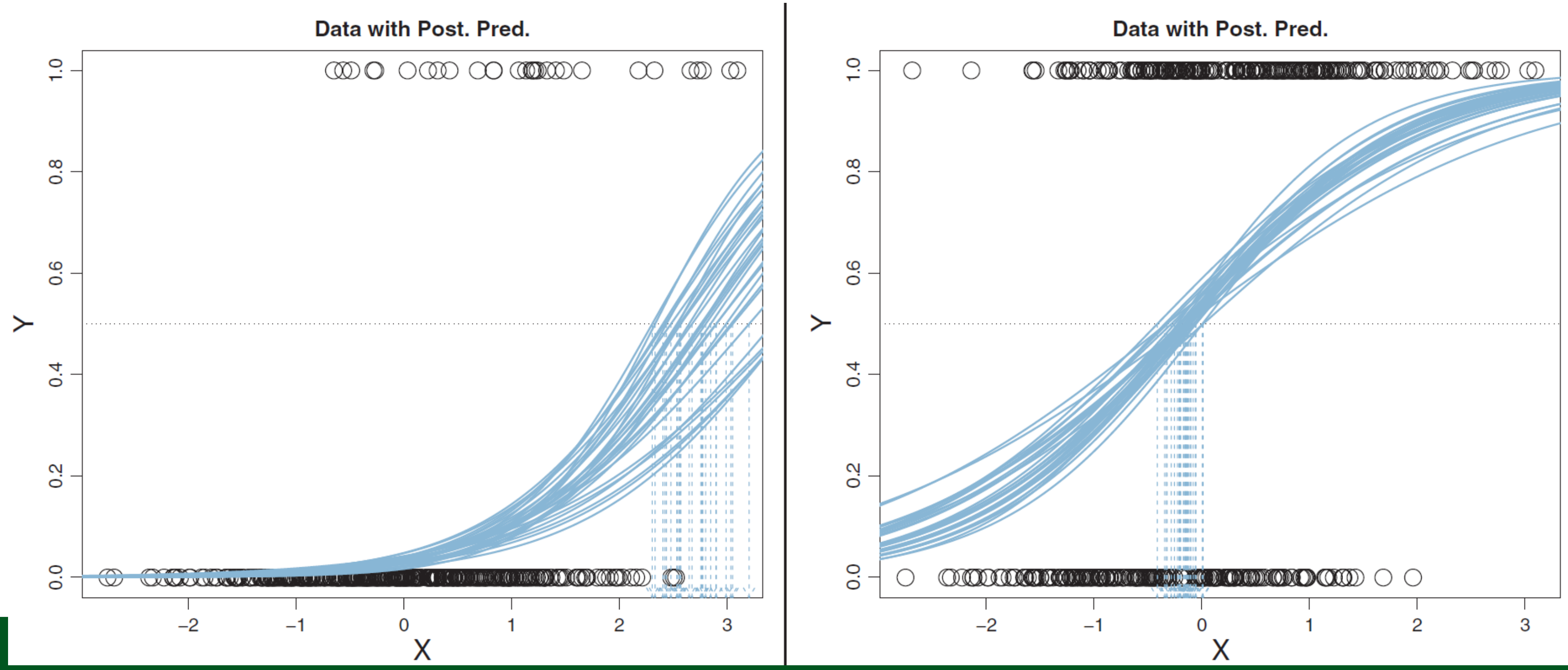
- 实际问题中，经常会出现$y=1$和$y=0$的样本数量不均衡的情况。

- 比如，用血压预测发生心脏病的概率。
  - 收集的数据中，发生心脏病（$y=1$）的样本占比很小。

- 对于不同类别的样本数量不均衡的情况，分类结果往往会不准确。

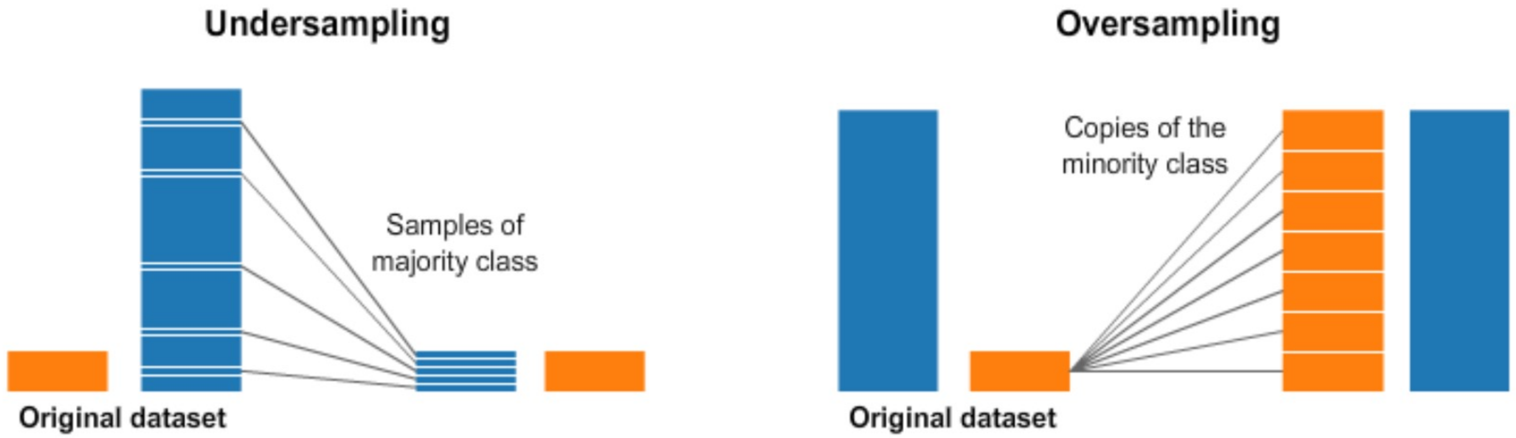
例子：

- 左图是类别不均衡的情况，右图是均衡的情况。
- 左图的阈值（$y = 0.5$对应的x值），明显偏右。
  - 因为$y = 0$的样本多，模型会使$p(y = 0|D)$的概率大。
    - 图中纵坐标表示$y = 1$的概率。



解决办法

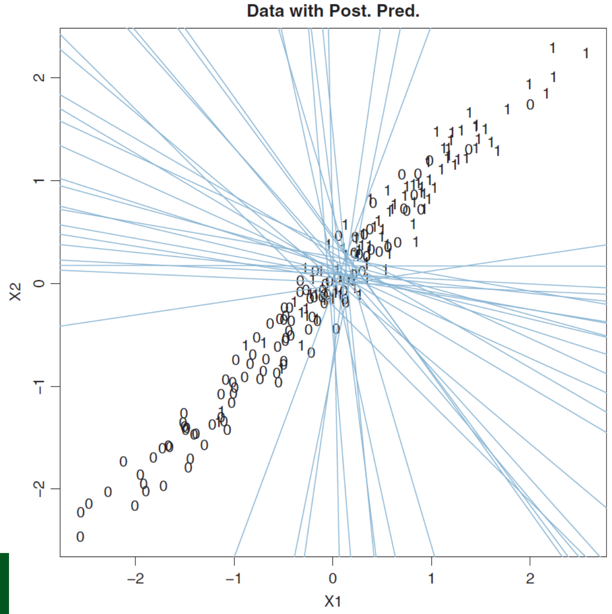
1. 收集数据阶段，就控制好不同类别的样本数量的均衡性。
2. 过采样（Over-sampling）
   - 重复一些数量少的类别的样本。
3. 欠采样（Under-sampling）
   - 去除一些数量多的类别的样本。

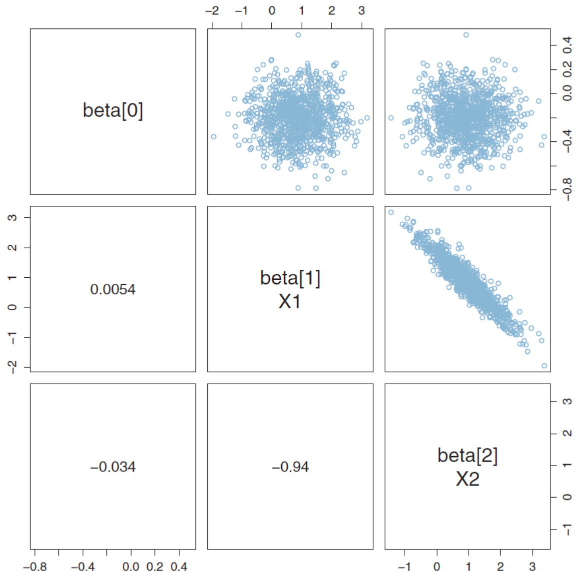


(图片来源： https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook)

21.2.3 冗余自变量

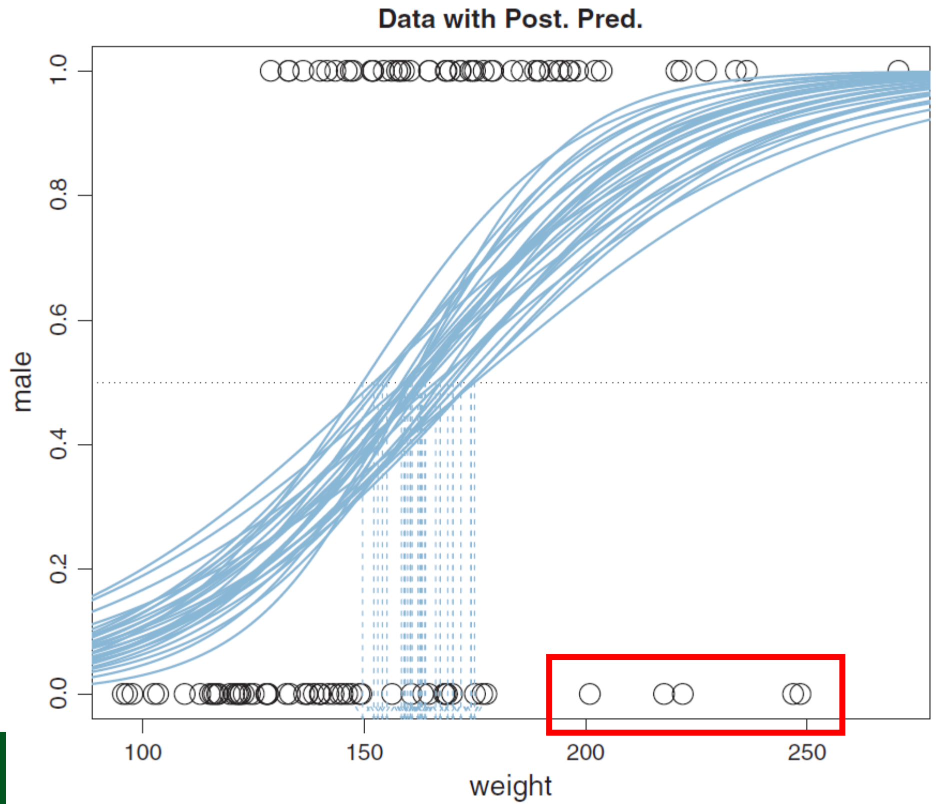
- 和回归类似，强相关的自变量的系数可以互相调节，导致回归系数的不确定性很大。
  - 比如：对于$\beta_2 x_2+\beta_1 x_1$，当$x_2=x_1$时，β_2加1并且β_1减1，结果不变。
- 例子：下图$y=0.5$的线不确定性很大，很分散。
- 解决办法：类似线性回归，先计算自变量之间的相关系数。





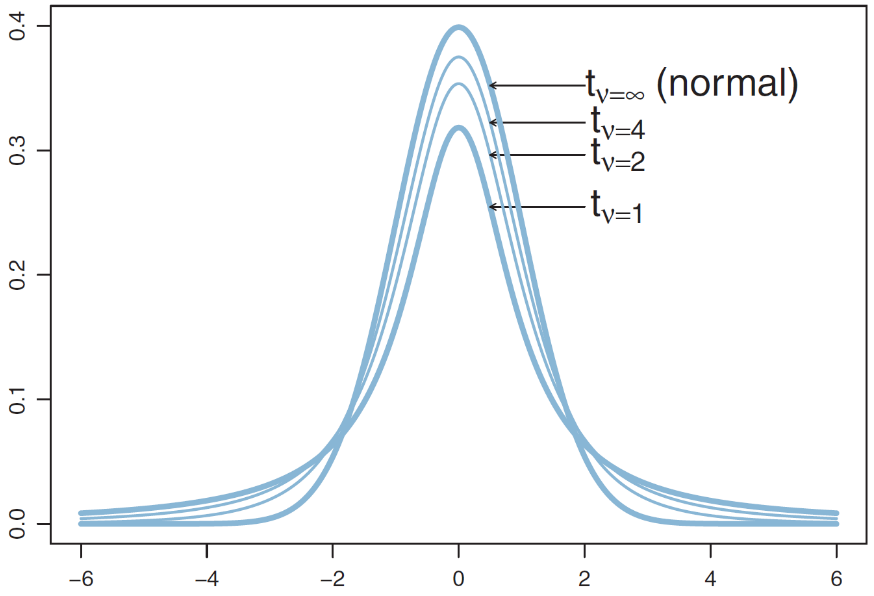
21.3. 鲁棒逻辑回归（Robust Logistic Regression）

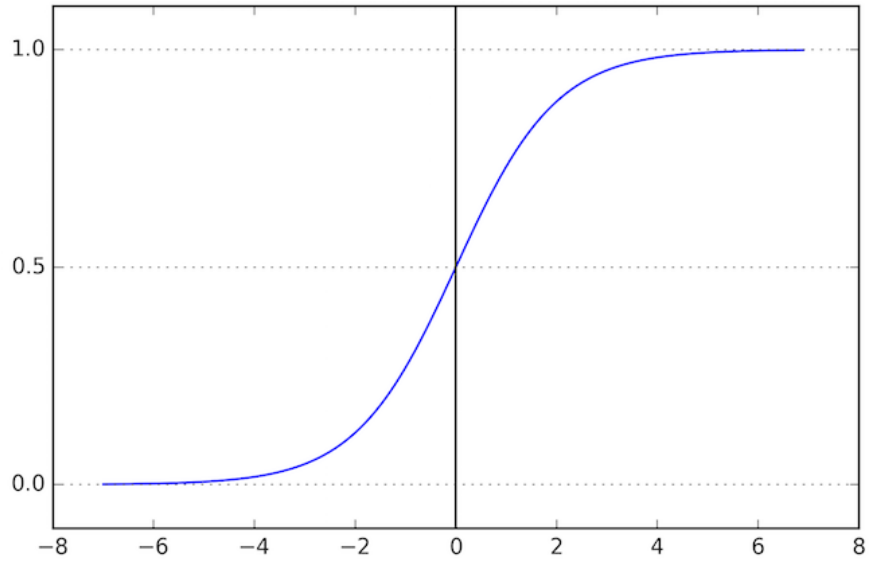
- 下图红框内是异常值（outlier）。
- 由于这些异常值的存在，使得曲线不能太陡，即β_1不能太大。
  - 因为不能使得这些点的$p(y=1)$太大。



鲁棒逻辑回归

- 回顾：鲁棒线性回归
- 用student-t分布替换高斯分布，使得图形上有“尾巴”，也就是让原来概率密度值接近0的区域，增加概率值。



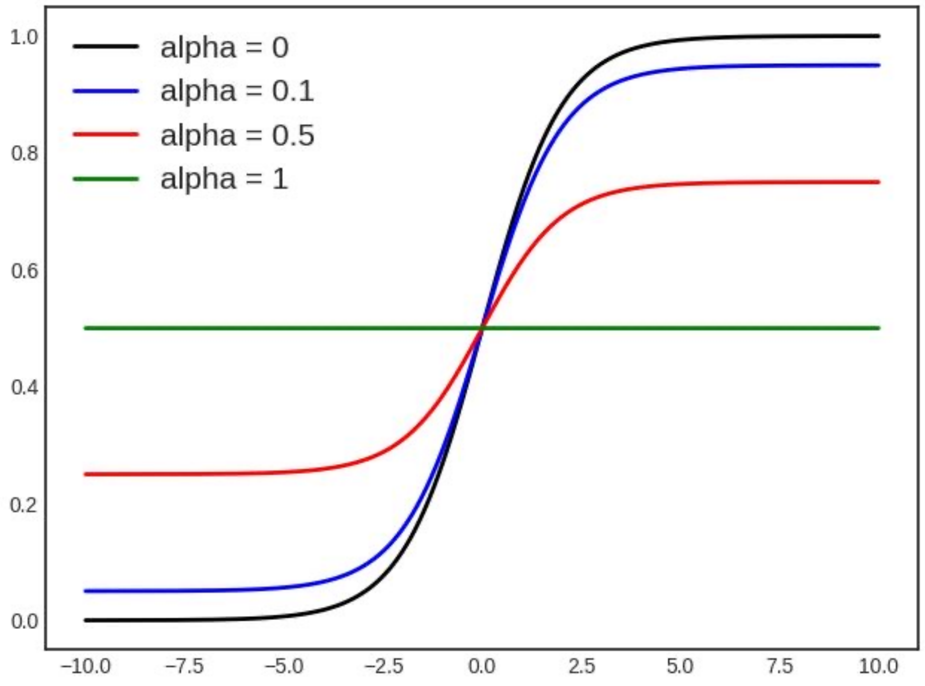


鲁棒逻辑回归

- 鲁棒逻辑回归:

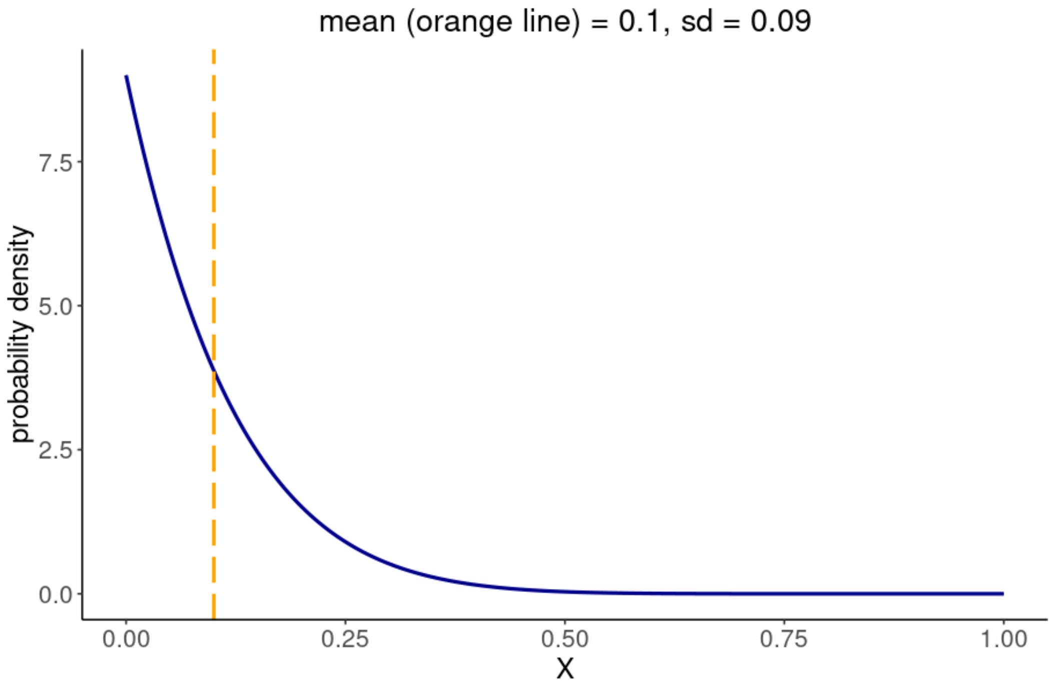
$$\mu = \alpha \cdot \frac{1}{2} + (1-\alpha)\mathrm{logistic}(\beta_2 x_2 + \beta_1 x_1 + \beta_0)$$

- 图形上，让原来概率密度值接近0或者1的区域，增加或减少概率值。
- 可以理解为：任何样本有α的概率是猜的。
  - α称为猜测系数。
- α也是需要估计的参数。



α的先验

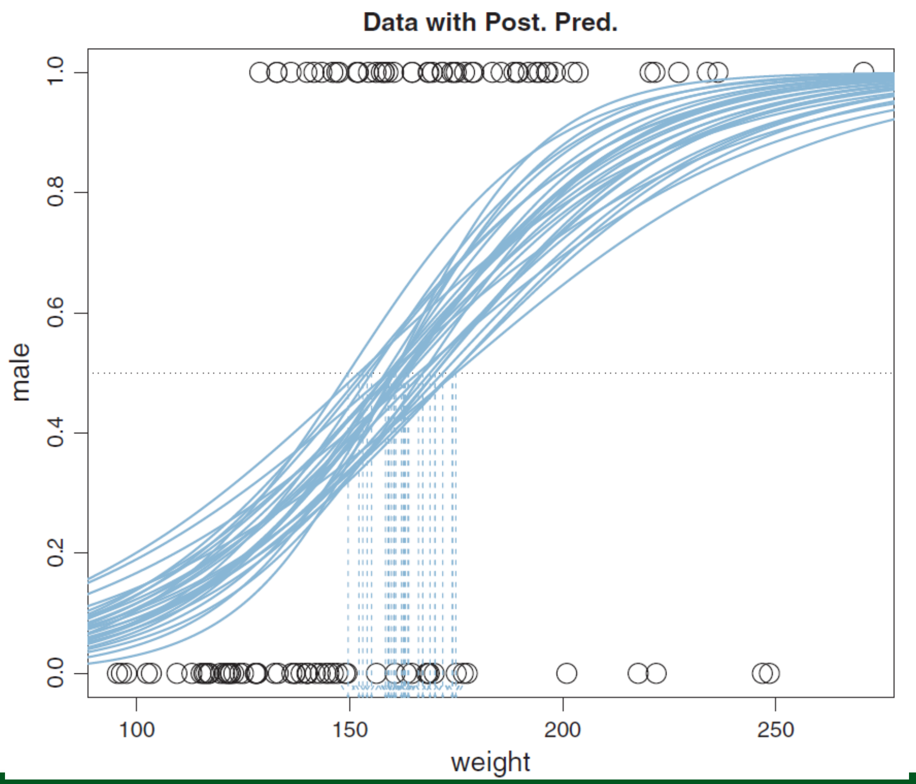
- α的取值范围是[0,1]。
- 可以使用Beta分布作为先验。
- 大部分情况下，异常值会比较少。因此我们期望α是比较小的。
  - 让大部分的概率值在α比较小的区域。
- 比如，可以设置$\text{beta}(\alpha|1,9)$。
  - [0.5,1]的概率很小。



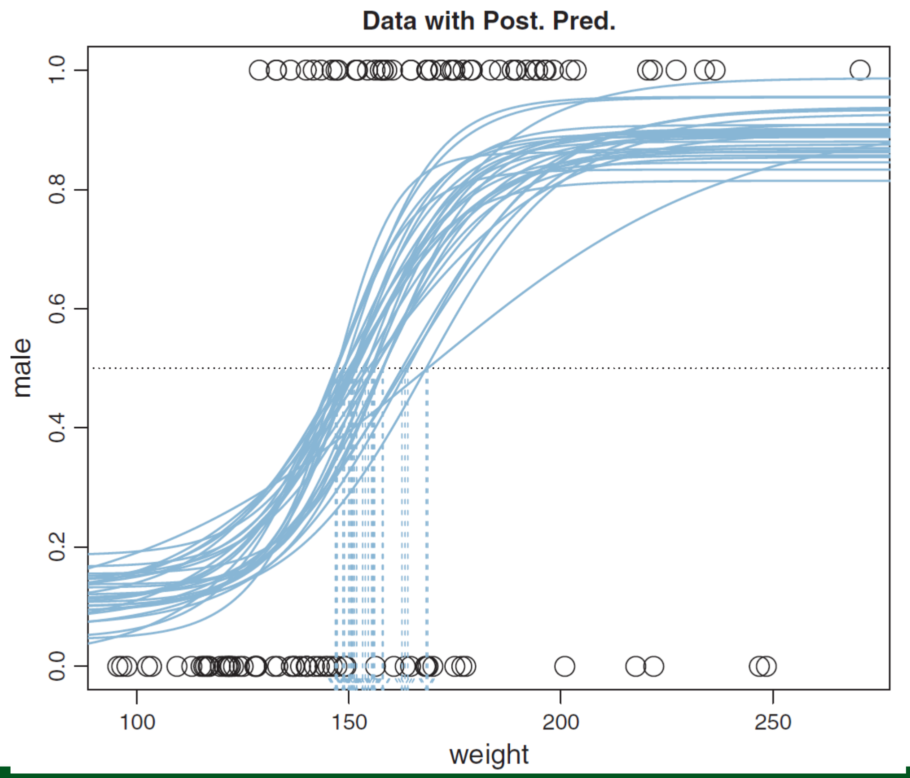
结果

- 右图的曲线更陡。
- 右图的阈值更小。

逻辑回归:



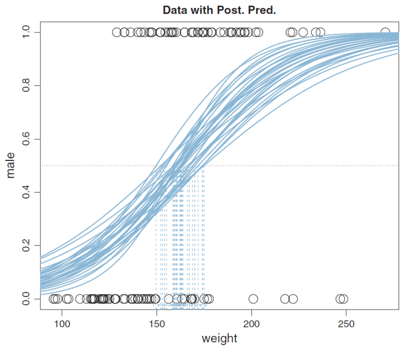
鲁棒逻辑回归:

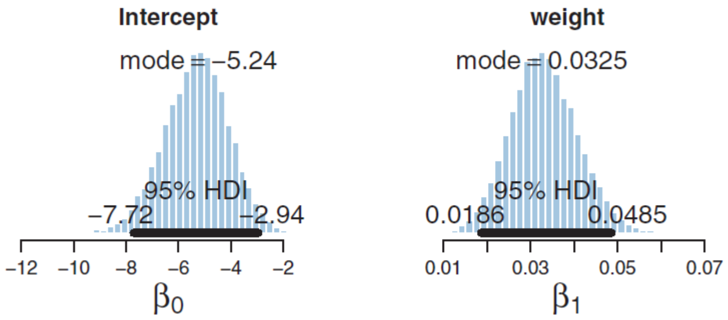


结果

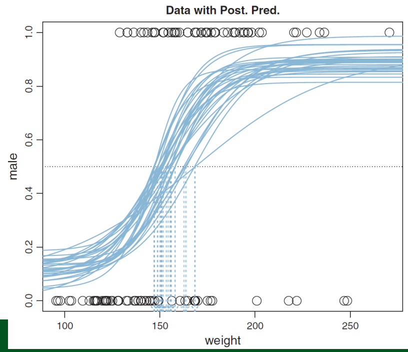
- α的峰值接近0.2，说明数据中有异常值。
- β_1比原来大，对应于曲线更陡。

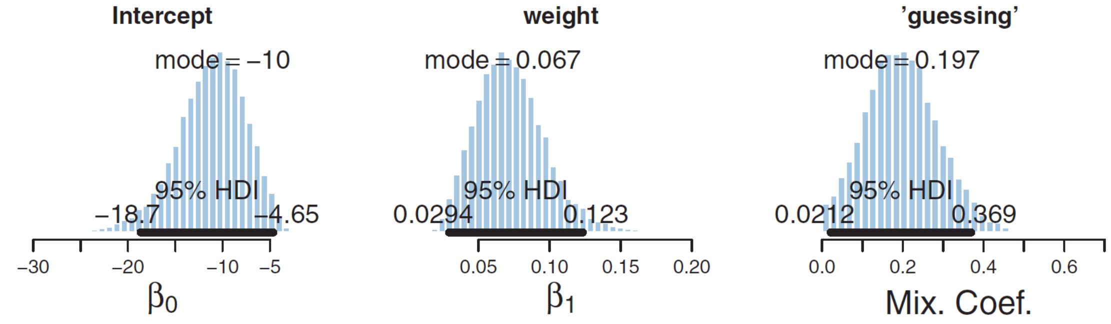
逻辑回归：





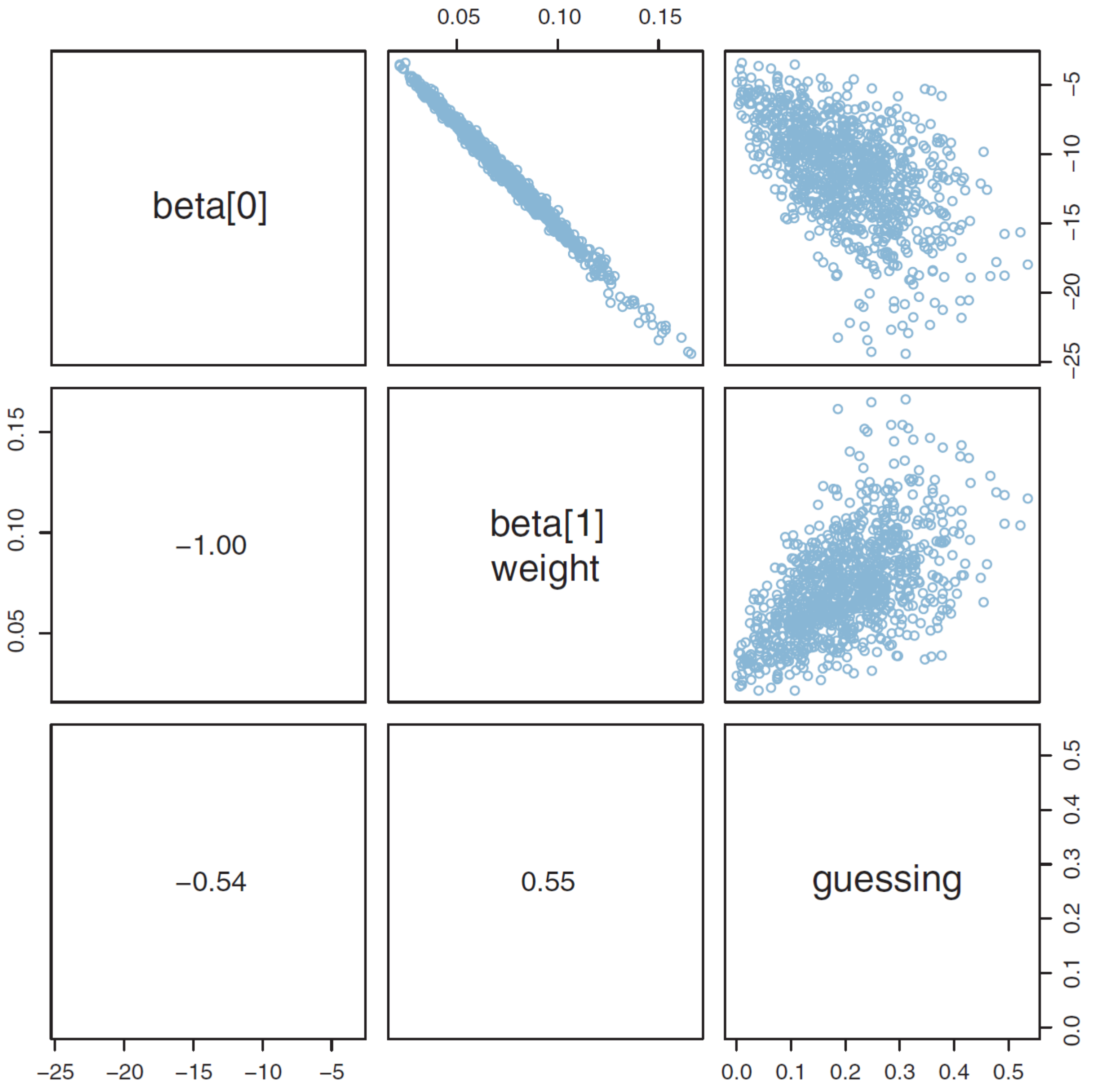
鲁棒逻辑回归：





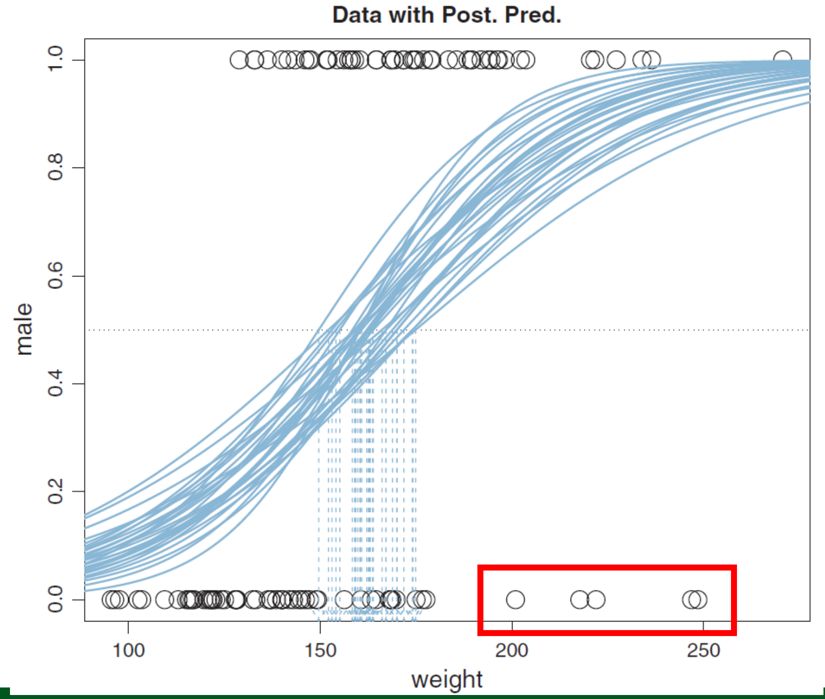
结果

- β_1和α之间有很强的正相关性。
  - 因为α越大，曲线可以越陡。
- β_1和β_0之间有很强的负相关性。
  - 因为$-\frac{\beta_0}{\beta_1}$是$y = 0.5$的位置。



鲁棒逻辑回归

- 对于异常值，另一种解决办法是增加其他的自变量。
- 通过其他自变量来判断类别，模型会让当前有异常值的自变量的系数变小。
- 比如，右图的例子只用了体重，我们可以增加身高作为自变量。



逻辑回归---极大似然估计

- 似然：

$$\hat{y} = \text{logistic}(\text{lin}(x))$$

$$p(y|\beta_k) = \text{bernoulli}(y|\hat{y}) = \hat{y}^y(1-\hat{y})^{1-y}$$

$$p(D|\beta_k) = \prod_i {\hat{y}_i}^{y_i}(1-\hat{y}_i)^{1-y_i}$$

- NLL：

$$-\log p(D|\beta_k) = -\sum_i [y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]$$

- 上式也被称为交叉熵损失（Cross-entropy loss）。

22 多分类问题

- 例子：
- 根据水温和水的盐度，来预测鱼的种类。
- 根据数字图片，来识别图片中的数字（0-9）。
- 根据文字内容，来预测文字中的情感（比如高兴、悲伤、惊讶、生气等）。

22.1 Softmax回归（Softmax Regression）

$$\lambda_1 = \sum_j \beta_{j,1} x_j + \beta_{0,1}$$

$$\lambda_2 = \sum_j \beta_{j,2} x_j + \beta_{0,2}$$

$$\lambda_3 = \sum_j \beta_{j,3} x_j + \beta_{0,3}$$

$$\phi_k = \text{softmax}(\{\lambda_k\}) = \frac{\exp(\lambda_k)}{\sum_{k^*=1}^{K} \exp(\lambda_{k^*})}$$

22.1 Softmax回归（Softmax Regression）

$$\lambda_k = \sum_j \beta_{j,k} x_j + \beta_{0,k}$$

$$\phi_k = \mathrm{softmax}(\{\lambda_k\}) = \frac{\exp(\lambda_k)}{\sum_{k^*=1}^{K} \exp(\lambda_{k^*})}$$

- 其中，K表示一共有K个类别，$k = \{1,2,\ldots,K\}$。
- ϕ_k表示属于第k个类别的概率。
- 上式表示第k个类别的$\exp(\lambda_k)$的占比。

Softmax函数的好处

- 假设λ_k是[1,1,5,3]，对比3种方法：

1. $\max(\lambda_k) = [0,0,1,0]$
2. $\frac{\lambda_k}{\sum_{k^*=1}^{K} \lambda_{k^*}} = [0.1,0.1,0.5,0.3]$
3. $\mathrm{softmax}(\{\lambda_k\}) = [0.02,0.02,0.85,0.11]$

- 相比于方法2，softmax的exp函数扩大了最大项和其他项的概率值差距。
- 相比于方法1，softmax给非最大项一定的概率值。
- 从这个例子也能看出，softmax是soft的max函数。

系数不确定性

$$\phi_k = \text{softmax}(\{\lambda_k\}) = \frac{\exp(\lambda_k)}{\sum_{k^*=1}^{K} \exp(\lambda_{k^*})}$$

- 当$\lambda_1 = 1$，$\lambda_2 = 2$，$\lambda_3 = 3$时，

$$\phi_1 = \frac{e}{e + e^2 + e^3} = \frac{1}{1 + e + e^2}$$

- 当$\lambda_1 = 0$，$\lambda_2 = 1$，$\lambda_3 = 2$时，

$$\phi_1 = \frac{1}{1 + e + e^2}$$

系数不确定性

$$\lambda_k = \sum_j \beta_{j,k} x_j + \beta_{0,k}$$

- 对所有k， $\beta'_{j,k} = \beta_{j,k} + \alpha_j$， $\beta'_{0,k} = \beta_{0,k} + \alpha_0$：

$$\frac{\exp(\sum_j (\beta_{j,k} + \alpha_j) x_j + \beta_{0,k} + \alpha_0)}{\sum_{k^*=1}^{K} \exp(\sum_j (\beta_{j,k^*} + \alpha_j) x_j + \beta_{0,k^*} + \alpha_0)}$$

$$= \frac{\exp\left(\sum_j \beta_{j,k} x_j + \beta_{0,k}\right) \exp(\sum_j \alpha_j x_j + \alpha_0)}{\sum_{k^*=1}^{K} \exp(\sum_j \beta_{j,k^*} x_j + \beta_{0,k^*}) \exp(\sum_j \alpha_j x_j + \alpha_0)}$$

$$= \frac{\exp\left(\sum_j \beta_{j,k} x_j + \beta_{0,k}\right)}{\sum_{k^*=1}^{K} \exp(\sum_j \beta_{j,k^*} x_j + \beta_{0,k^*})} = \phi_k$$

- 即，所有类别的某个自变量参数都加同一个值，softmax函数输出不变。

系数不确定性

从上式可得：

- 对于每个λ_k，任一自变量的系数或者截距，同时加一个数，softmax函数的输出不变。
  - 如果不加限制，参数估计会不确定性很大。

一般的做法是：

- 设置一个参考类别（比如λ_1），使得：

$$\lambda_1 = \sum_j 0 \cdot x_j + 0 = 0$$

Softmax函数和Logistic函数

- 当Softmax函数只有2个类别（1或0）时：

（设置类别0为参考类别，即$\lambda_0 = 0$）

$$\phi_1 = \frac{\exp(\lambda_1)}{\exp(\lambda_1) + \exp(\lambda_0)}$$

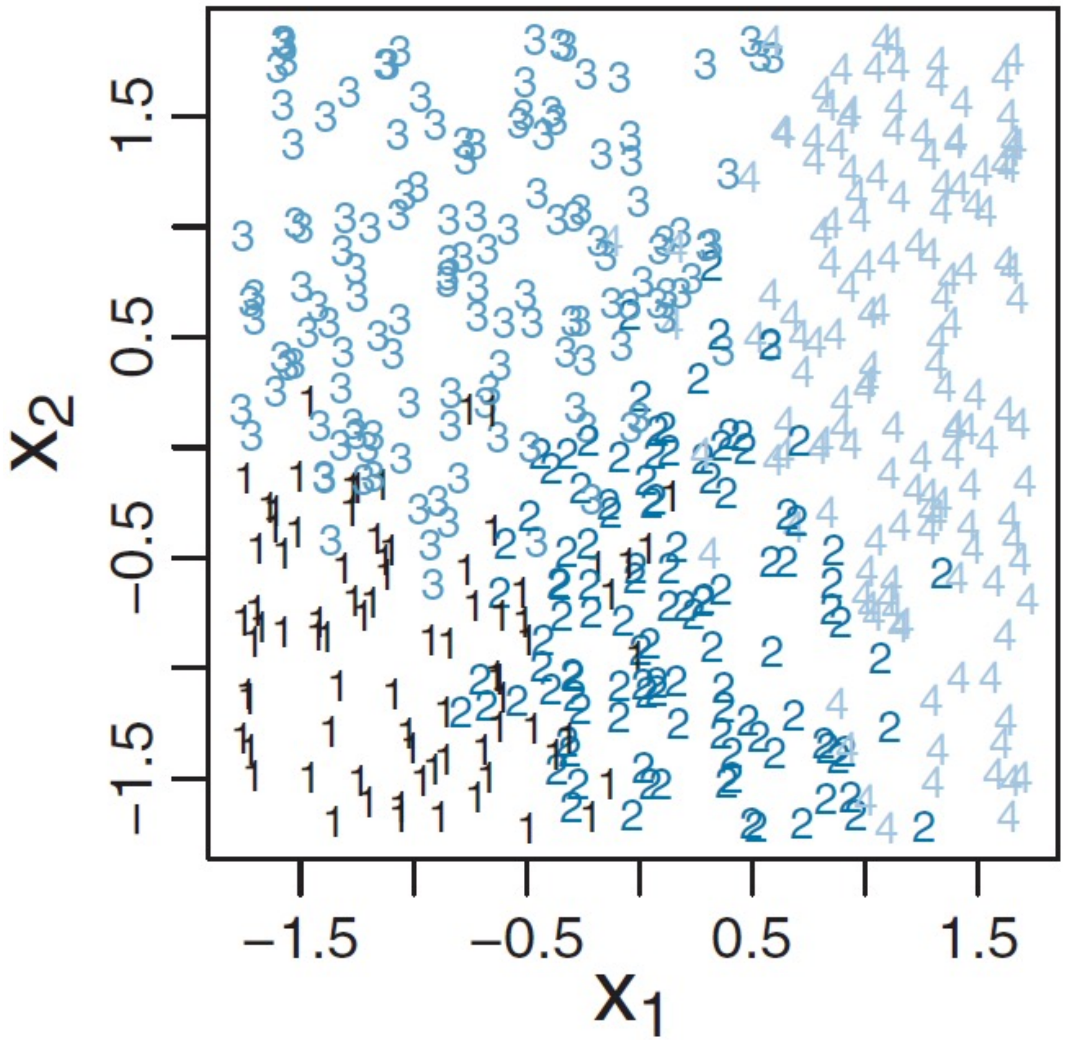
$$= \frac{\exp(\lambda_1)}{\exp(\lambda_1) + 1}$$

$$= \frac{1}{1 + \exp(-\lambda_1)}$$

$$= \text{logistic}(\lambda_1)$$

例子：

- 对如下的二维数据做分类：
- 2个自变量： x_1, x_2
- 4个类别： 1,2,3,4

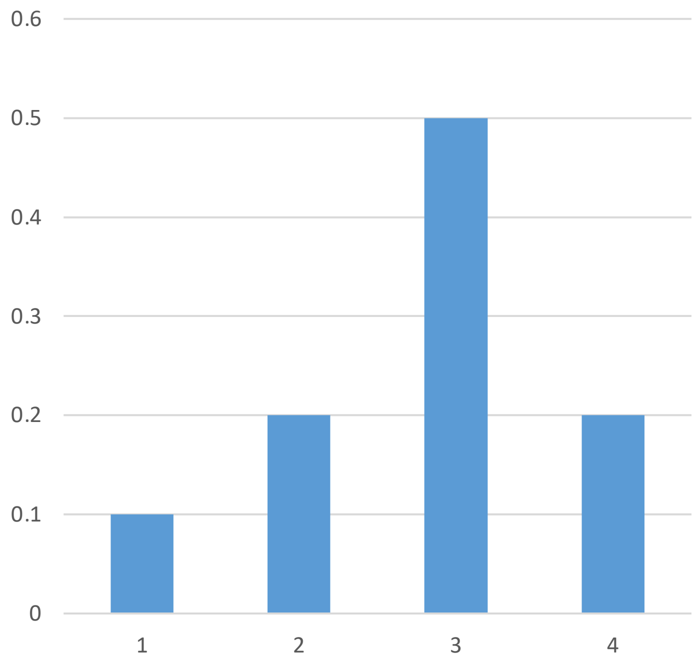


类别分布（Categorical distribution）

- 类别分布是伯努利分布的扩展。
- 对于K个类别的类别分布，结果是第k个类别的概率记为θ_k，即：

$$p(y=k|\boldsymbol{\theta})=\theta_k$$

- $\boldsymbol{\theta}=(\theta_1,\theta_2\dots\theta_K)$，且满足$\sum_k\theta_k=1$。



建模

- 似然:

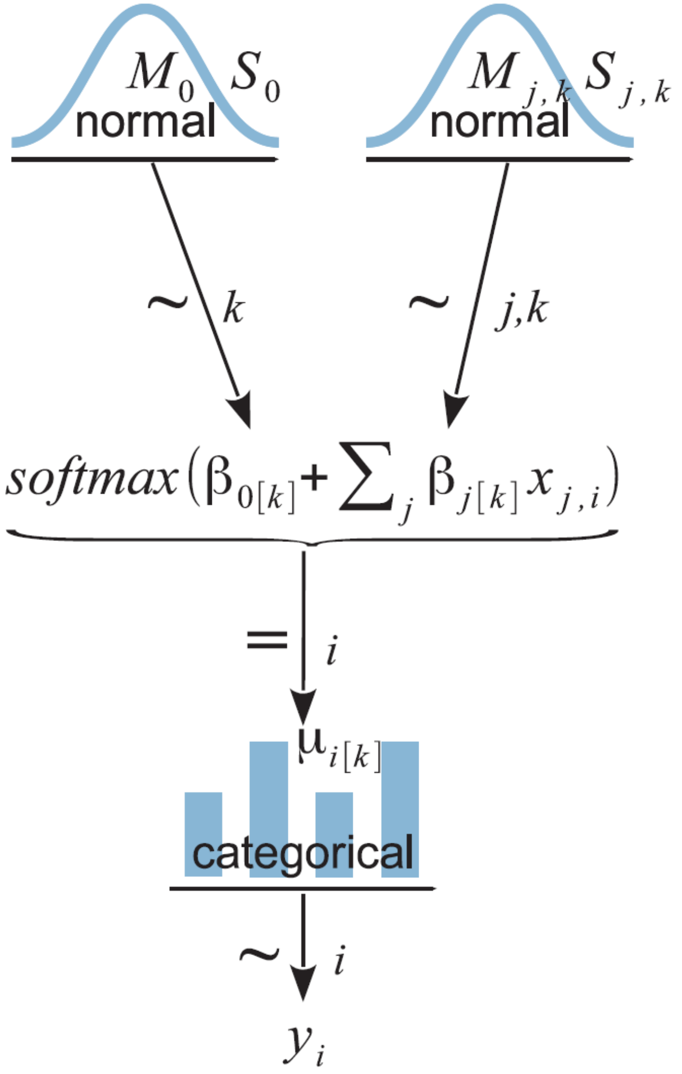
$$\phi_k = \text{softmax}(\beta_{2,k}x_2 + \beta_{1,k}x_1 + \beta_{0,k})$$

$$y \sim \text{categorical}(\{\phi_k\})$$

- 先验:

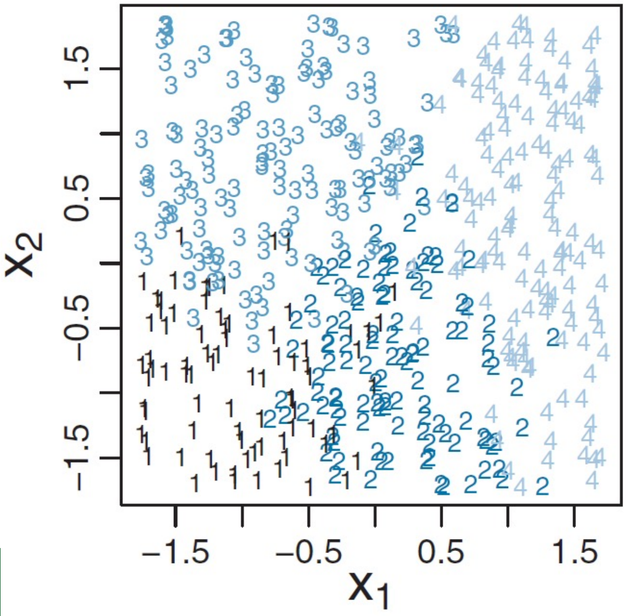
$$\beta_{0,k} \sim \text{normal}(\mu_0, \sigma_0)$$

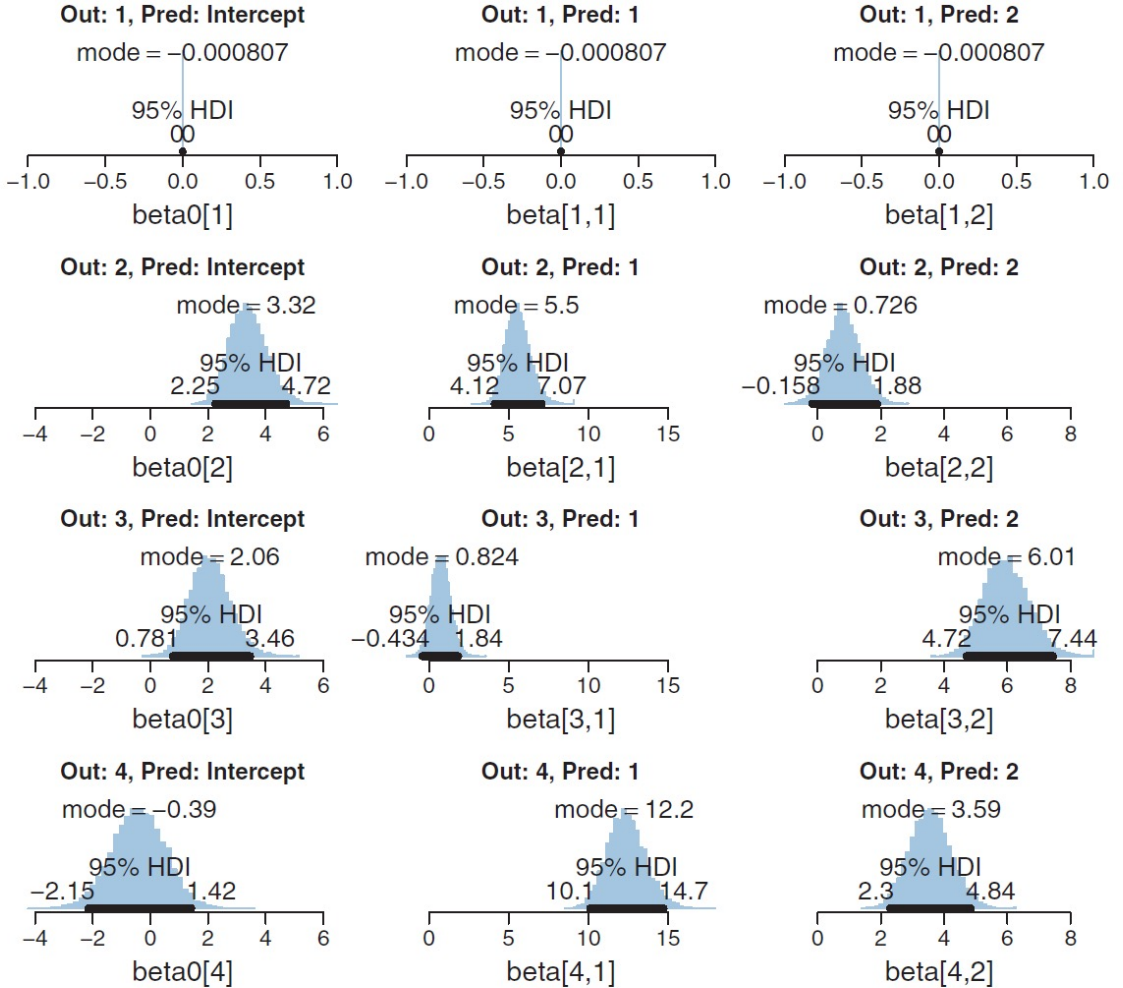
$$\beta_{j,k} \sim \text{normal}(\mu_{j,k}, \sigma_{j,k})$$



结果

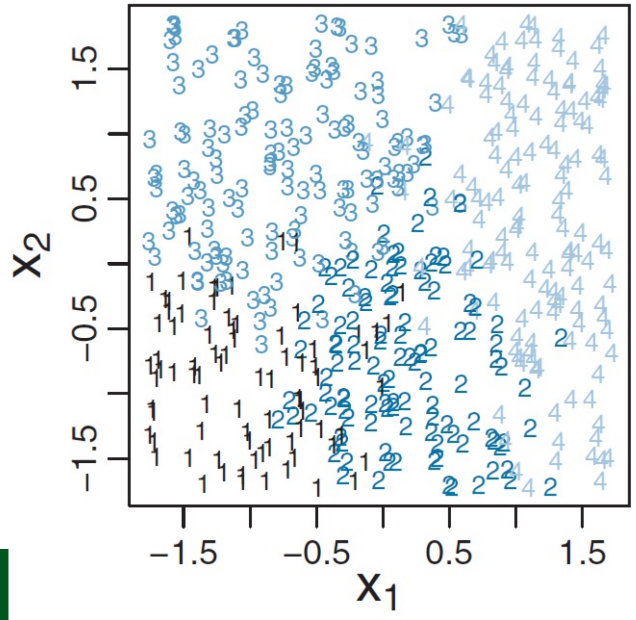
- 第1类的β都是设置为0。
- 如果$\beta_{j,k}$比较大，说明自变量x_j对第k个类别影响比较大。x_j越大，越可能是第k个类别。
- $\beta_{j,k}$的绝对值意义不大，要看β之间的相对大小。

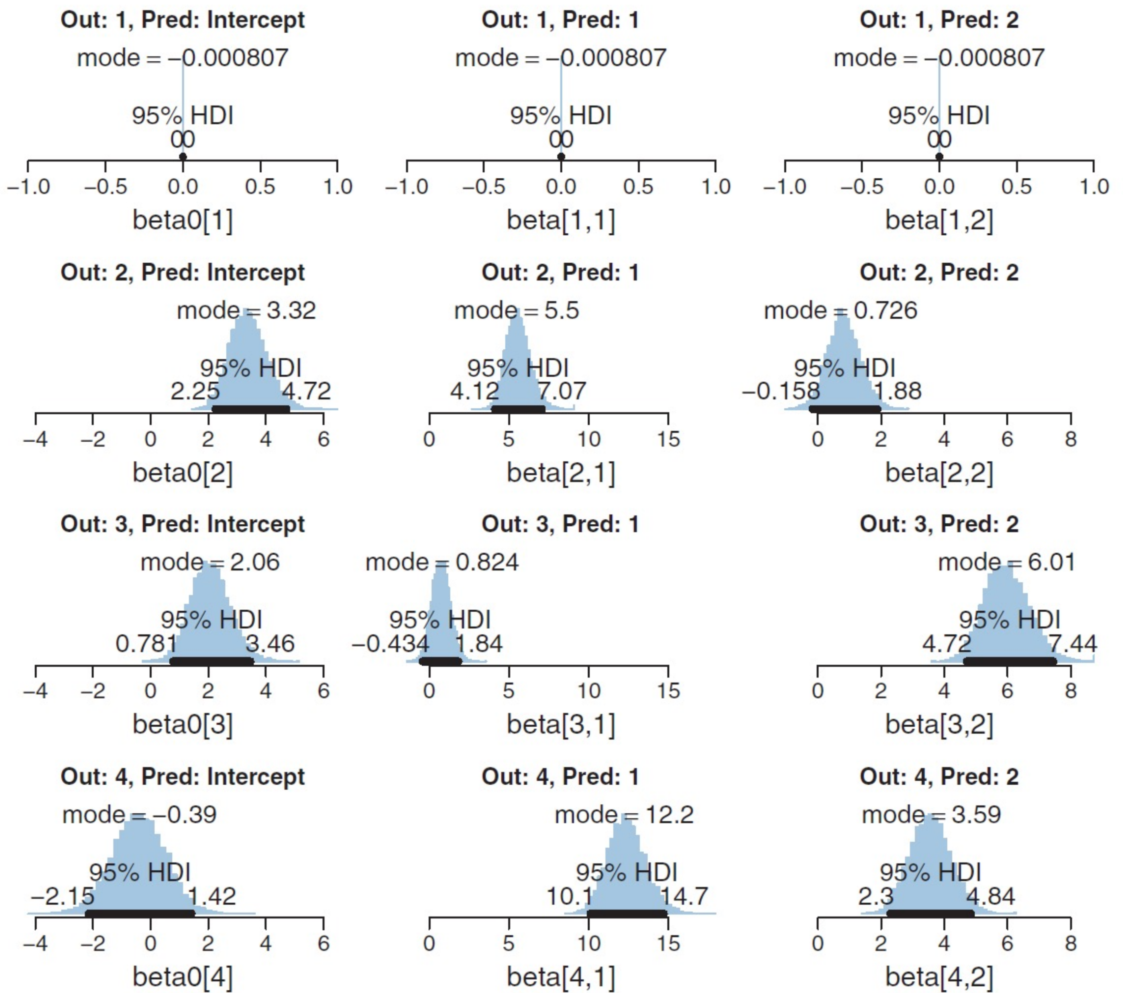




结果

- 比如，类别3的β_2最大，类别4的β_1最大。
- 对应于，在左图中：
- 只要x_1够大，就属于类别4。
- 只要x_2够大，且x_1不是太大，就属于类别3。
  - 类别3的β_1很小。





生成模型（以下内容为阅读材料）

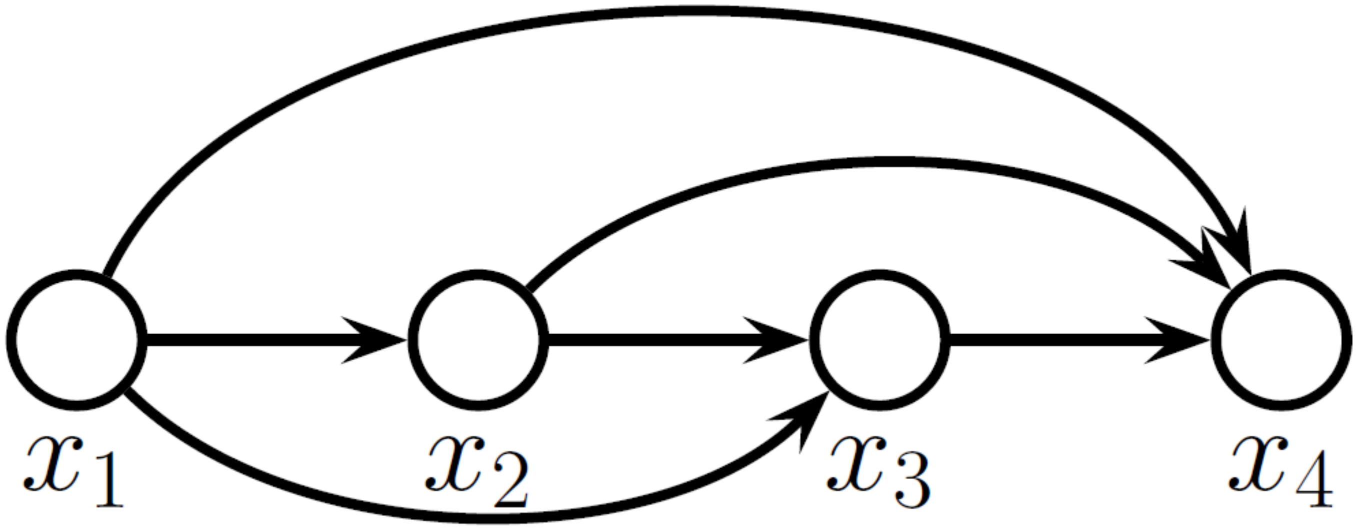
- 对于生成模型，给定一些数据x，常见的目标是最大化边缘似然$p(x)$（marginal likelihood）。

- 最大化边缘似然和最大化似然的区别：
  - 似然$p(x|\theta)$是在特定模型下找使$p(x|\theta)$最大化的θ。
  - 边缘似然$p(x)$是不限定特定模型，可以是任意模型，找使$p(x)$最大化的模型。

自回归模型（Auto-regressive models）

- 自回归模型利用概率链式法则将T个变量的联合概率分布建模为：

$$p(\boldsymbol{x}_{1:T}) = p(\boldsymbol{x}_1)p(\boldsymbol{x}_2|\boldsymbol{x}_1)p(\boldsymbol{x}_3|\boldsymbol{x}_2,\boldsymbol{x}_1)p(\boldsymbol{x}_4|\boldsymbol{x}_3,\boldsymbol{x}_2,\boldsymbol{x}_1)\ldots = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1})$$



Neural autoregressive density estimators (NADE)

$$p(x_t|x_{1:t-1}) = \mathrm{Bernoulli}(x_t|\theta)$$

- 假设$p(x_t|x_{1:t-1})$服从伯努利分布。
- $\theta = f(x_{1:t-1})$

NADE生成的图像

(Larochelle et. al., 2011)

Real-valued Neural Autoregressive Density Estimator (RNADE)

- 假设$p(x_t|x_{1:t-1})$服从混合高斯分布：

$$p(x_t|x_{1:t-1}) = \sum_k \pi_{t,k} N\left(x_t \middle| \mu_{t,k}, \sigma_{t,k}^2\right)$$

- $(\pi_{t,k}, \mu_{t,k}, \sigma_{t,k}) = f(x_{1:t-1})$

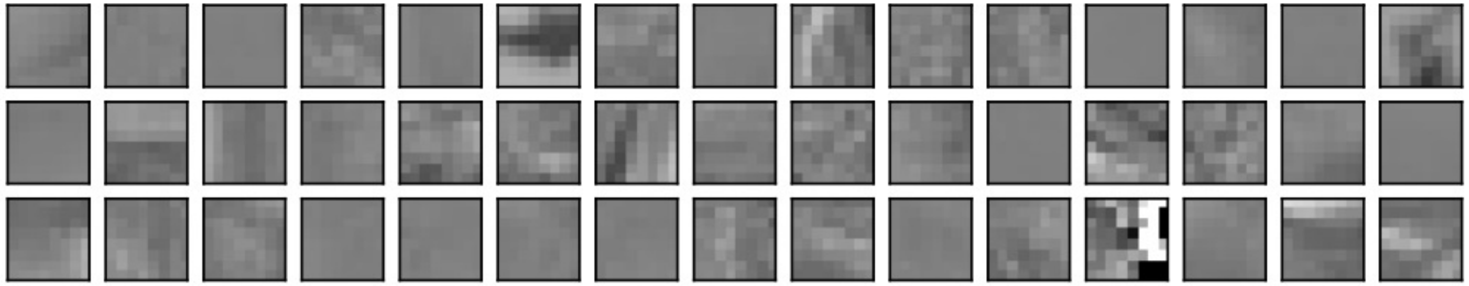


Figure 1: **Top:** 15 8x8 patches from the BSDS test set. **Center:** 15 samples from Zoran and Weiss's MoG model with 200 components. **Bottom:** 15 samples from an RNADE with 512 hidden units and 10 output components per dimension. All data and samples were drawn randomly.

(Uria et. al., 2013)

序列生成（例如GPT）

- 在步骤t，对前t个输入$y_{1:t}$用masked self-attention（权重为a_t），计算得到:

$$z_t = \sum a_t y_t$$

- 用MLP网络映射为：

$$h_t = \mathrm{MLP}(z_t)$$

- 最终：

$$p(y_{t+1}|y_{1:t}) = \mathrm{Cat}(\mathrm{Softmax}(W h_t))$$