Homework 4: Diffusion of Tetracycline

ywx 3220101739

We continue examining the diffusion of tetracycline among doctors in Illinois in the early 1950s, building on our work in lab 6. You will need the data sets `ckm_nodes.csv` and `ckm_network.dat` from the labs.

1. Clean the data to eliminate doctors for whom we have no adoption-date information, as in the labs. Only use this cleaned data in the rest of the assignment.

```r
# 读取节点数据并清理缺失值
ckm_nodes <- read_csv('data/ckm_nodes.csv')
noinfor <- which(is.na(ckm_nodes$adoption_date))
ckm_nodes <- ckm_nodes[-noinfor, ]
# 读取网络数据并同步删除无效节点
ckm_network <- read.table('data/ckm_network.dat')
ckm_network <- ckm_network[-noinfor, -noinfor]
# 验证数据维度一致性
cat("节点数:", nrow(ckm_nodes), "网络矩阵维度:", dim(ckm_network))
```

```
## 节点数: 125 网络矩阵维度: 125 125
```

2. Create a new data frame which records, for every doctor, for every month, whether that doctor began prescribing tetracycline that month, whether they had adopted tetracycline before that month, the number of their contacts who began prescribing strictly *before* that month, and the number of their contacts who began prescribing in that month or earlier. Explain why the dataframe should have 6 columns, and 2125 rows.

```r
# 添加医生 ID 列（使用行号作为唯一标识）
ckm_nodes$doctor_id <- 1:nrow(ckm_nodes)
n_doctors <- nrow(ckm_nodes)
adoption_dates <- ckm_nodes$adoption_date
# 创建所有医生-月份组合 (125 医生 × 17 个月 = 2125 行)
doctor_months <- expand.grid(
  doctor_id = ckm_nodes$doctor_id,
  month = 1:17,
  stringsAsFactors = FALSE
)
# 添加关键指标列
doctor_months$adopted_this_month <- with(doctor_months, {
  as.integer(adoption_dates[doctor_id] == month)
})
doctor_months$already_adopted <- with(doctor_months, {
  as.integer(adoption_dates[doctor_id] < month)
})
# 预计算每个医生的邻居索引
neighbor_indices <- lapply(1:n_doctors, function(i) {
  which(ckm_network[i, ] == 1)
})
# 计算邻居采用指标
doctor_months$n_contacts_adopted_before <- mapply(function(doc_id, t) {
  neighbors <- neighbor_indices[[doc_id]]
  if(length(neighbors) > 0) {
    sum(adoption_dates[neighbors] < t, na.rm = TRUE)
  } else {
    0
  }
}, doctor_months$doctor_id, doctor_months$month)
doctor_months$n_contacts_adopted_by_now <- mapply(function(doc_id, t) {
  neighbors <- neighbor_indices[[doc_id]]
  if(length(neighbors) > 0) {
    sum(adoption_dates[neighbors] <= t, na.rm = TRUE)
  } else {
    0
  }
}, doctor_months$doctor_id, doctor_months$month)
# 验证结果
cat("行数:", nrow(doctor_months), "预期: 125 医生 × 17 个月 = 2125\n")
```

```
## 行数: 2125 预期: 125医生 × 17个月 = 2125
```

```r
cat("列数:", ncol(doctor_months), "预期: 6 列\n（医生 id 和月份两列，其余要求信息四列）")
```

```
## 列数: 6 预期: 6列
## （医生id和月份两列，其余要求信息四列）
```

3. Let

$$\begin{aligned} p_k = \Pr(&\text{A doctor starts prescribing tetracycline this month} \mid \\ &\text{Number of doctor's contacts prescribing before this month} = k) \end{aligned} \tag{1}$$

and

$$\begin{aligned} q_k = \Pr(&\text{A doctor starts prescribing tetracycline this month} \mid \\ &\text{Number of doctor's contacts prescribing this month} = k) \end{aligned} \tag{2}$$

We suppose that p_k and q_k are the same for all months.

a. Explain why there should be no more than 21 values of k for which we can estimate p_k and q_k directly from the data.

```r
max_degree <- max(rowSums(ckm_network))
cat(" 最大邻居数量:", max_degree)
```

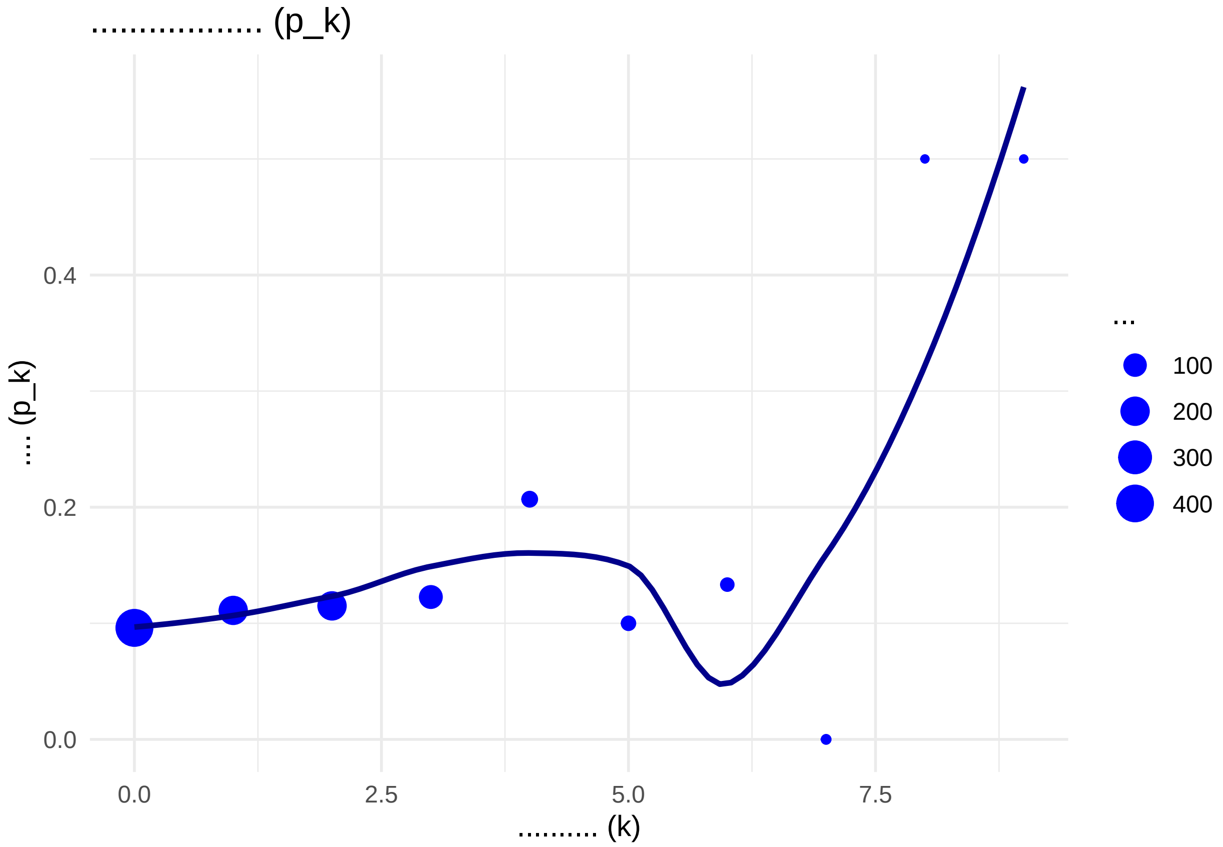
```
## 最大邻居数量: 20
```

说明：医生的最大邻居数量（网络度数）决定了 k 的最大可能值。对于 k > 20 的情况，数据中没有医生拥有超过 20 个邻居，因此无法估计 b. Create a vector of estimated p_k probabilities, using the data frame from (2). Plot the probabilities against the number of prior-adoptee contacts k.

```r
# 计算 p_k
p_data <- doctor_months %>%
  filter(already_adopted == 0) %>%  # 只考虑尚未采用的医生
  group_by(n_contacts_adopted_before) %>%
  summarise(
    p_k = mean(adopted_this_month),
    count = n()
  ) %>%
  rename(k = n_contacts_adopted_before)

# 绘图
ggplot(p_data, aes(x = k, y = p_k)) +
  geom_point(aes(size = count), color = "blue") +
  geom_smooth(method = "loess", se = FALSE, color = "darkblue") +
  labs(title = " 采用概率与本月前已采用邻居数量的关系 (p_k)",
       x = " 本月前已采用邻居数量 (k)",
       y = " 采用概率 (p_k)") +
  scale_size_continuous(name = " 样本量") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

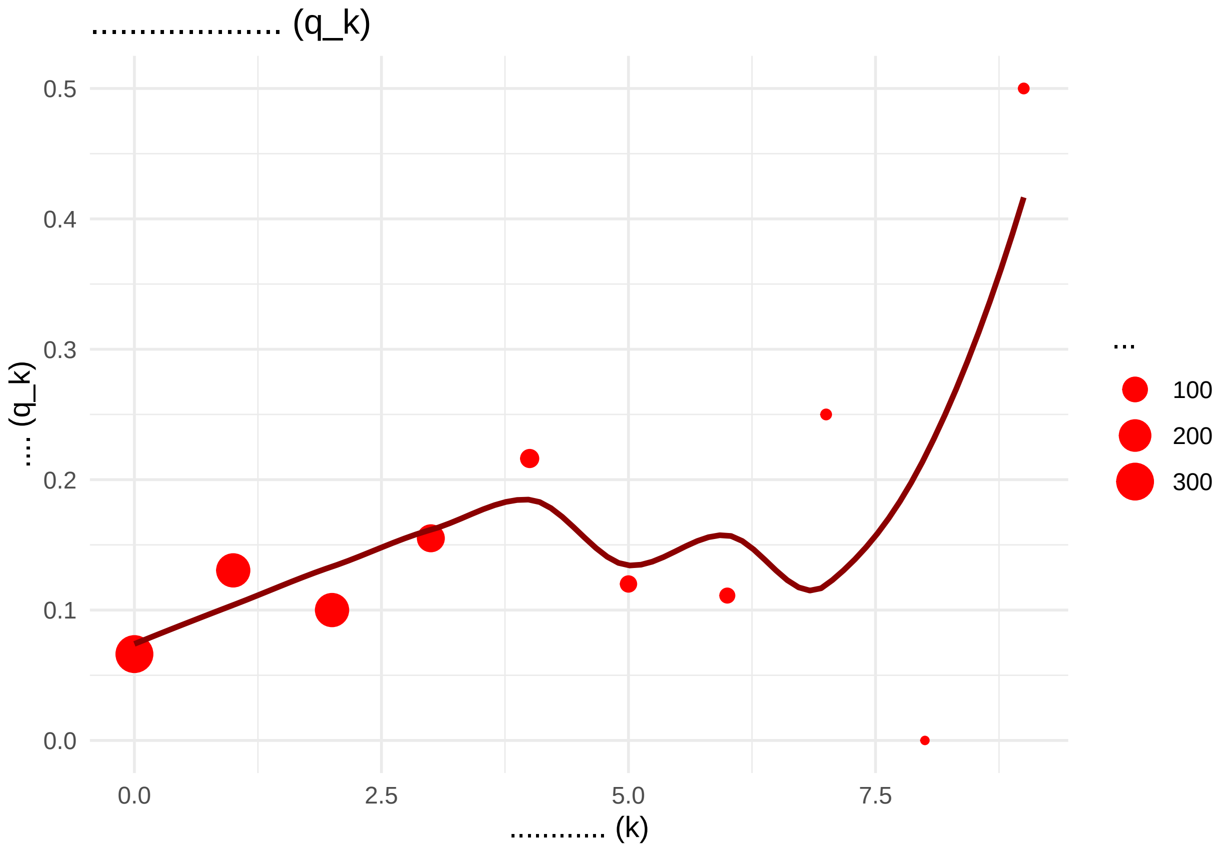


c. Create a vector of estimated q_k probabilities, using the data frame from (2). Plot the probabilities against the number of prior-or-contemporary-adoptee contacts k.

```
# 计算 q_k
q_data <- doctor_months %>%
  filter(already_adopted == 0) %>%  # 只考虑尚未采用的医生
  group_by(n_contacts_adopted_by_now) %>%
  summarise(
    q_k = mean(adopted_this_month),
    count = n()
  ) %>%
  rename(k = n_contacts_adopted_by_now)

# 绘图
ggplot(q_data, aes(x = k, y = q_k)) +
  geom_point(aes(size = count), color = "red") +
  geom_smooth(method = "loess", se = FALSE, color = "darkred") +
  labs(title = " 采用概率与本月或之前已采用邻居数量的关系 (q_k)",
       x = " 本月或之前已采用邻居数量 (k)",
       y = " 采用概率 (q_k)") +
  scale_size_continuous(name = " 样本量") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



4. Because it only conditions on information from the previous month, p_k is a little easier to interpret than q_k. It is the probability per month that a doctor adopts tetracycline, if they have exactly k contacts who had already adopted tetracycline.

a. Suppose $p_k = a + bk$. This would mean that each friend who adopts the new drug increases the probability of adoption by an equal amount. Estimate this model by least squares, using the values you constructed in (3b). Report the parameter estimates.

```r
# 使用加权最小二乘法估计线性模型（权重为样本量）
linear_model <- lm(p_k ~ k, data = p_data, weights = count)
# 获取参数估计
linear_coef <- coef(linear_model)
cat("线性模型估计结果:\n")
```

```
## 线性模型估计结果:
```

```r
cat(sprintf("截距 a = %.4f\n斜率 b = %.4f", linear_coef[1], linear_coef[2]))
```

```
## 截距 a = 0.0947
## 斜率 b = 0.0126
```

b. Suppose $p_k = e^{a+bk}/(1 + e^{a+bk})$. Explain, in words, what this model would imply about the impact of adding one more adoptee friend on a given doctor's probability of adoption. (You can suppose that $b > 0$, if that makes it easier.) Estimate the model by least squares, using the values you constructed in (3b).

```r
# 使用加权非线性最小二乘法估计 logistic 模型
logistic_model <- nls(
  p_k ~ exp(a + b*k)/(1 + exp(a + b*k)),
  data = p_data,
  weights = count,
  start = list(a = -4, b = 0.1) # 初始值
)
# 获取参数估计
logistic_coef <- coef(logistic_model)
cat("\nLogistic 模型估计结果:\n")
```

```
##
## Logistic模型估计结果:
```

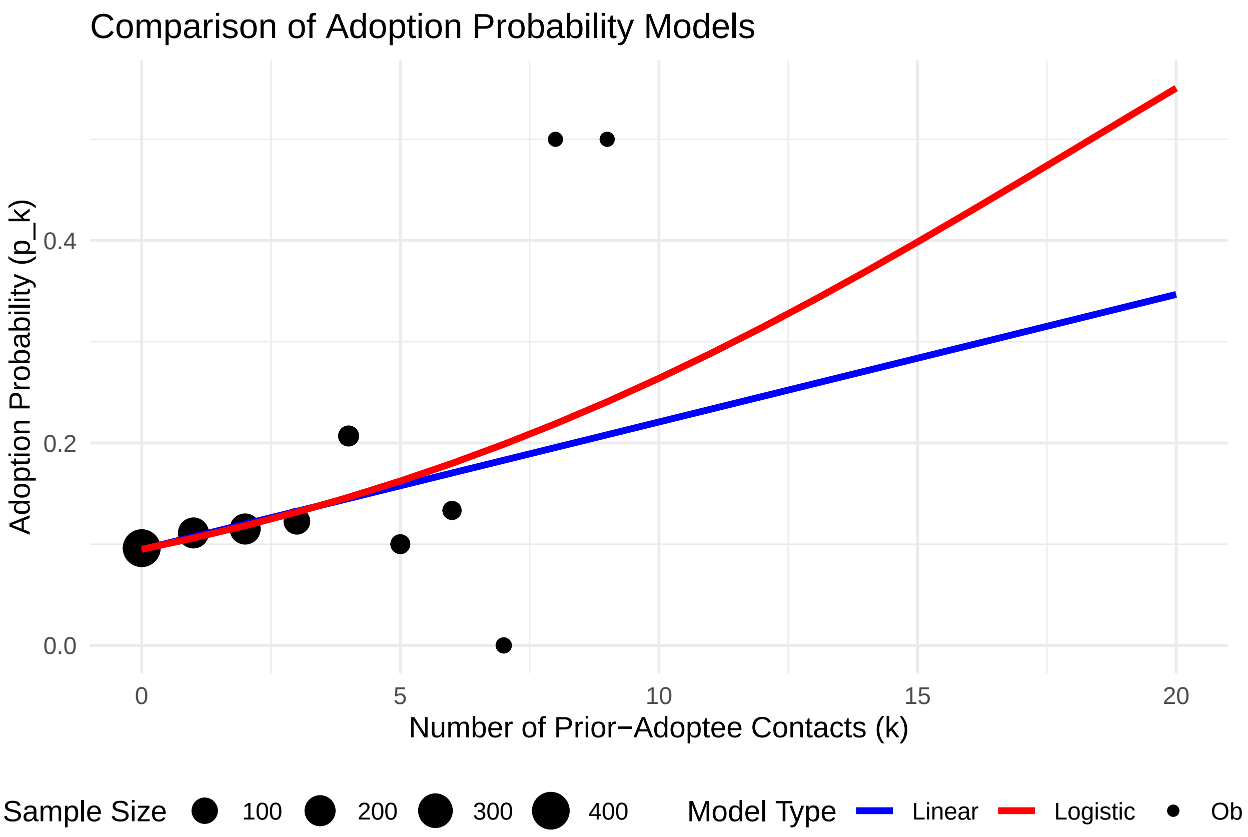
```r
cat(sprintf("参数 a = %.4f\n参数 b = %.4f", logistic_coef[1], logistic_coef[2]))
```

```
## 参数 a = -2.2548
## 参数 b = 0.1229
```

c. Plot the values from (3b) along with the estimated curves from (4a) and (4b). (You should have one plot, with k on the horizontal axis, and probabilities on the vertical axis .) Which model do you prefer, and why?

```r
# 生成预测数据
k_range <- 0:20
linear_pred <- linear_coef[1] + linear_coef[2]*k_range
logistic_pred <- exp(logistic_coef[1] + logistic_coef[2]*k_range)/
                (1 + exp(logistic_coef[1] + logistic_coef[2]*k_range))
# 创建比较数据框
comparison_df <- data.frame(
  k = rep(k_range, 3),
  Probability = c(p_data$p_k[match(k_range, p_data$k)],
                 linear_pred,
                 logistic_pred),
  Type = rep(c("Observed", "Linear", "Logistic"), each = length(k_range))
)

# 绘图
ggplot(comparison_df, aes(x = k, y = Probability, color = Type)) +
  geom_point(data = subset(comparison_df, Type == "Observed"),
             aes(size = p_data$count[match(k_range, p_data$k)])) +
  geom_line(data = subset(comparison_df, Type != "Observed"),
            linewidth = 1.2) +
  scale_color_manual(values = c("Observed" = "black",
                                "Linear" = "blue",
                                "Logistic" = "red")) +
  labs(title = "Comparison of Adoption Probability Models",
       x = "Number of Prior-Adoptee Contacts (k)",
       y = "Adoption Probability (p_k)",
       color = "Model Type") +
  scale_size_continuous(name = "Sample Size", range = c(2, 6)) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



For quibblers, pedants, and idle hands itching for work to do: The p_k values from problem 3 aren't all equally precise, because they come from different numbers of observations. Also, if each doctor with k adoptee contacts is independently deciding whether or not to adopt with probability p_k, then the variance in the number of adoptees will depend on p_k. Say that the actual proportion who decide to adopt is $\hat{p}_k$. A little probability (exercise!) shows that in this situation, $\mathbb{E}[\hat{p}_k] = p_k$, but that $\text{Var}[\hat{p}_k] = p_k(1-p_k)/n_k$, where n_k is the number of doctors in that situation. (We estimate probabilities more precisely when they're really extreme [close to 0 or 1], and/or we have lots of observations.) We can estimate that variance as $\hat{V}_k = \hat{p}_k(1-\hat{p}_k)/n_k$. Find the $\hat{V}_k$, and then re-do the estimation in (4a) and (4b) where the squared error for p_k is divided by $\hat{V}_k$. How much do the parameter estimates change? How much do the plotted curves in (4c) change?

```
# 基于问题 3b 的 p_data 计算方差估计
p_data <- p_data %>%
  mutate(
    V_k = p_k * (1 - p_k) / count,  # 方差估计
    weight = ifelse(V_k > 0, 1/V_k, 0)  # 权重（方差的倒数）
  ) %>%
  filter(!is.na(V_k) & is.finite(V_k))  # 移除无效值
```

```r
# 加权线性回归
weighted_linear_model <- lm(p_k ~ k, data = p_data, weights = weight)
weighted_linear_coef <- coef(weighted_linear_model)
# 加权非线性回归
weighted_logistic_model <- nls(
  p_k ~ exp(a + b*k)/(1 + exp(a + b*k)),
  data = p_data,
  weights = weight,
  start = list(a = -4, b = 0.1)
)
weighted_logistic_coef <- coef(weighted_logistic_model)
```

参数比较

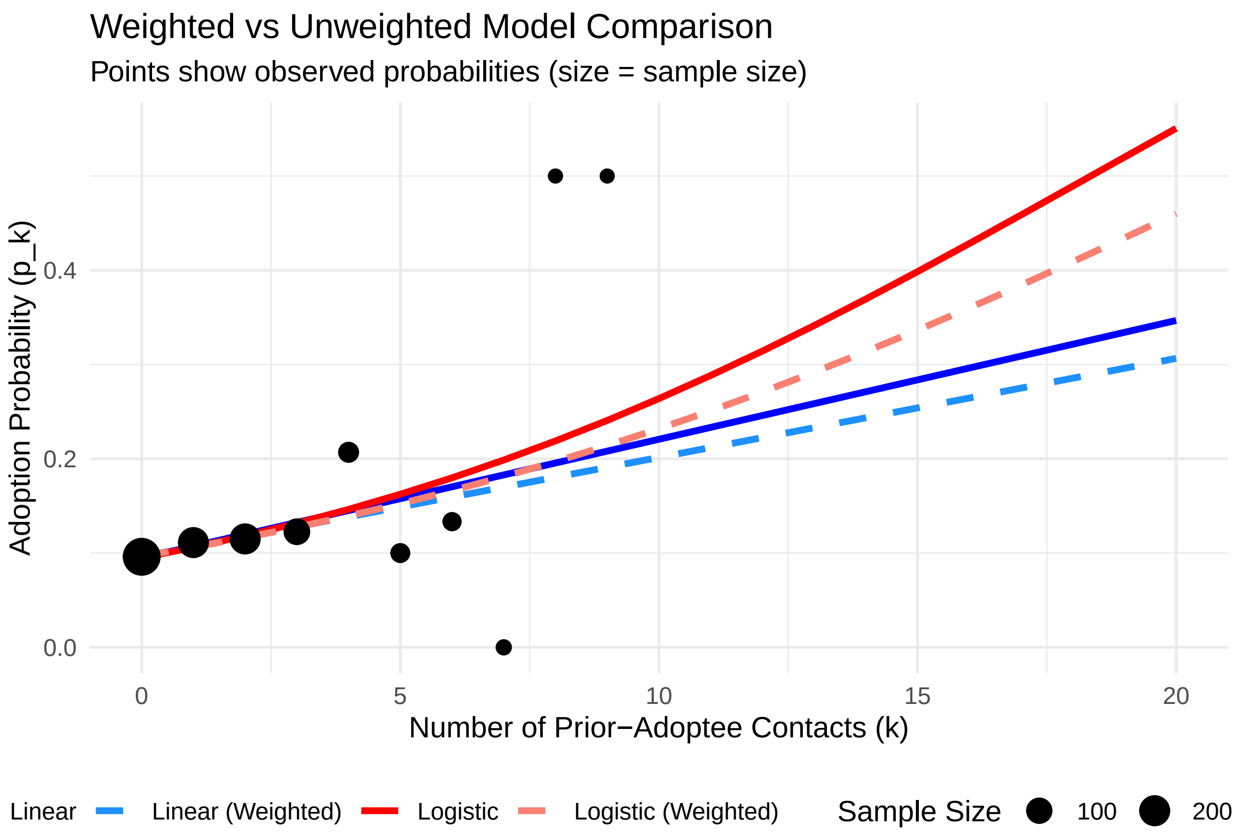
```r
# 创建参数比较表格
param_comparison <- data.frame(
  Model = c("Linear", "Linear (Weighted)", "Logistic", "Logistic (Weighted)"),
  a = c(linear_coef[1], weighted_linear_coef[1],
        logistic_coef[1], weighted_logistic_coef[1]),
  b = c(linear_coef[2], weighted_linear_coef[2],
        logistic_coef[2], weighted_logistic_coef[2])
)
# 计算变化百分比
param_comparison <- param_comparison %>%
  mutate(
    a_change = c(0, 100*(a[2]-a[1])/a[1], 0, 100*(a[4]-a[3])/a[3]),
    b_change = c(0, 100*(b[2]-b[1])/b[1], 0, 100*(b[4]-b[3])/b[3])
  )
print(param_comparison)
```

```
##                    Model           a          b  a_change  b_change
## 1                 Linear  0.09470668 0.01260200  0.000000   0.00000
## 2      Linear (Weighted)  0.09637512 0.01050772  1.761692 -16.61857
## 3               Logistic -2.25484997 0.12288660  0.000000   0.00000
## 4 Logistic (Weighted) -2.23476722 0.10372101 -0.890647 -15.59616
```

曲线比较

```
# 生成预测数据
k_range <- 0:20
orig_linear <- linear_coef[1] + linear_coef[2]*k_range
weighted_linear <- weighted_linear_coef[1] + weighted_linear_coef[2]*k_range
orig_logistic <- exp(logistic_coef[1] + logistic_coef[2]*k_range)/
                 (1 + exp(logistic_coef[1] + logistic_coef[2]*k_range))
weighted_logistic <- exp(weighted_logistic_coef[1] + weighted_logistic_coef[2]*k_range)/
                     (1 + exp(weighted_logistic_coef[1] + weighted_logistic_coef[2]*k_range))
# 创建比较数据框
curve_comparison <- data.frame(
  k = rep(k_range, 4),
  Probability = c(orig_linear, weighted_linear, orig_logistic, weighted_logistic),
  Model = rep(c("Linear", "Linear (Weighted)", "Logistic", "Logistic (Weighted)"),
              each = length(k_range))
)

# 绘图
ggplot(curve_comparison, aes(x = k, y = Probability, color = Model, linetype = Model)) +
  geom_line(linewidth = 1.2) +
  geom_point(data = p_data, aes(x = k, y = p_k, size = count),
             color = "black", inherit.aes = FALSE) +
  scale_color_manual(values = c("Linear" = "blue", "Linear (Weighted)" = "dodgerblue",
                                "Logistic" = "red", "Logistic (Weighted)" = "salmon")) +
  scale_linetype_manual(values = c("Linear" = "solid", "Linear (Weighted)" = "dashed",
                                   "Logistic" = "solid", "Logistic (Weighted)" = "dashed")) +
  labs(title = "Weighted vs Unweighted Model Comparison",
       subtitle = "Points show observed probabilities (size = sample size)",
       x = "Number of Prior-Adoptee Contacts (k)",
       y = "Adoption Probability (p_k)") +
  scale_size_continuous(name = "Sample Size", range = c(2, 6)) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



说明：通过比较可以发现，Logistic 模型参数变化较小，更稳健，受加权影响较小；加权估计降低了对样本量小、高方差数据点的敏感性，更准确地反映了高精度估计（大样本量点）的影响。因此，加权 Logistic 模型平衡了拟合优度和稳定性。