Лабораторна робота №4
з непараметричної статистики

Горбунов Даніел Денисович
1 курс магістратури
група "Прикладна та теоретична статистика"
Варіант №4

8 травня 2022 р.

Вступ.

Дана робота є продовженням третьої самостійної роботи. Побудовано класифікатори для визначення на якому винограднику вироблено вино, базуючись на змінних "Proanthocyanins", "Magnesium". Класифікатори побудовані двома способами, а саме: безпосередньо побудова емпірично-баєсового класифікатора, побудова класифікатора з проекцією на оптимальний напрямок.

Підготовча робота.

Узагальнення крос-валідації для двовимірної щільності.

Нехай ми маємо оцінку двовимірної щільності вигляду:

$$\hat{f}_n(x^1, x^2; h_1, h_2) = \frac{1}{nh_1h_2} \sum_{j=1}^{n} K\left(\frac{x^1 - X_j^1}{h_1}\right) K\left(\frac{x^2 - X_j^2}{h_2}\right) \tag{1}$$

Де потрібно добре (в певному сенсі) підібрати значення параметрів згладжування $h_j > 0$, $j = 1, 2$. Для цього розглянемо проінтегровану квадратичну похибку:

$$\mathrm{ISE}(h_1, h_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (\hat{f}_n(x^1, x^2; h_1, h_2) - f(x_1, x_2))^2 dx^1 dx^2$$

Розкривши квадрат в інтегралі, $\mathrm{ISE}(h_1, h_2)$ спрощується до вигляду:

$$\mathrm{ISE}(h_1, h_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (\hat{f}_n(x^1, x^2; h_1, h_2))^2 dx^1 dx^2 - 2\int_{-\infty}^{+\infty} \hat{f}_n(x^1, x^2; h_1, h_2) f(x_1, x_2) dx^1 dx^2 + \\ + \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (f(x^1, x^2))^2 dx^1 dx^2$$

Третій доданок не впливає на вибір точки мінімуму функціонала, тому його можна відкинути. Потрібно якось оцінити другий доданок в $\mathrm{ISE}(h_1, h_2)$: це можна зробити, побачивши наступне:

$$\int\limits_{-\infty}^{+\infty} \hat{f}_n(x^1, x^2; h_1, h_2) f(x_1, x_2) dx^1 dx^2 = \mathbb{E}\left[\hat{f}_n(X_0^1, X_0^2; h_1, h_2) \mid X^1, X^2\right]$$

де (X_0^1, X_0^2) – нове спостереження, що не залежить від (X^1, X^2). Тоді це умовне математичне сподівання можна оцінити відповідним середнім:

$$\mathbb{E}\left[\hat{f}_n(X_0^1, X_0^2; h_1, h_2) \mid X^1, X^2\right] \approx \frac{1}{n} \sum_{j=1}^{n} \hat{f}_n^{-j}(X_j^1, X_j^2; h_1, h_2)$$

де $\hat{f}_n^{-j}$ – ядерна оцінка щільності, побудована за всіма спостереженнями окрім j-го. Вводимо функціонал крос-валідації вигляду:

$$\begin{aligned}
\mathrm{CV}(h_1, h_2) &= J_1(h_1, h_2) - 2J(h_1, h_2), \\
J_1(h_1, h_2) &= \int\limits_{-\infty}^{+\infty} \int\limits_{-\infty}^{+\infty} (\hat{f}_n(x^1, x^2; h_1, h_2))^2 dx^1 dx^2 \\
J_2(h_1, h_2) &= \frac{1}{n} \sum_{j=1}^{n} \hat{f}_n^{-j}(X_j^1, X_j^2; h_1, h_2)
\end{aligned}$$

Оцінкою параметрів згладжування (h_1, h_2) буде точка мінімуму вищенаведеного функціонала. Тепер припустимо, що $K(t) = 0.75 \cdot (1 - t^2) \cdot \mathbb{1}\{|t| < 1\}$ – ядро Єпанєчнікова. Спробуємо спростити вигляд CV-функціоналу:

$$J_1(h_1, h_2) = \int\limits_{-\infty}^{+\infty} \int\limits_{-\infty}^{+\infty} (\hat{f}_n(x^1, x^2; h_1, h_2))^2 dx^1 dx^2 =$$

$$= \frac{1}{n^2 h_1^2 h_2^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \int\limits_{-\infty}^{+\infty} \int\limits_{-\infty}^{+\infty} \left(K\left(\frac{x^1 - X_i^1}{h_1}\right) K\left(\frac{x^1 - X_j^1}{h_1}\right)\right) \left(K\left(\frac{x^2 - X_i^2}{h_2}\right) K\left(\frac{x^2 - X_j^2}{h_2}\right)\right) dx^1 dx^2 =$$

$$= \frac{1}{n^2 h_1^2 h_2^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \int\limits_{-\infty}^{+\infty} \left(K\left(\frac{x^1 - X_i^1}{h_1}\right) K\left(\frac{x^1 - X_j^1}{h_1}\right)\right) dx^1 \int\limits_{-\infty}^{+\infty} \left(K\left(\frac{x^2 - X_i^2}{h_2}\right) K\left(\frac{x^2 - X_j^2}{h_2}\right)\right) dx^2 =$$

$$= \frac{1}{n^2 h_1^2 h_2^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{K}_{ij}(h_1, X^1) \mathbf{K}_{ij}(h_2, X^2),\ \mathbf{K}_{ij}(h_k, X^k) = \int\limits_{-\infty}^{+\infty} \left(K\left(\frac{u - X_i^k}{h_k}\right) K\left(\frac{u - X_j^k}{h_k}\right)\right) du$$

Використовуючи попередні обчислення з третьої лабораторної роботи, маємо:

$$\mathbf{K}_{ij}(h_1, X^1) \mathbf{K}_{ij}(h_2, X^2) = \begin{cases} h_1 \cdot h_2 \cdot d^4,\ i = j \\ h_1 \cdot h_2 \cdot (K * K)\left(\dfrac{X_i^1 - X_j^1}{h_1}\right) \cdot (K * K)\left(\dfrac{X_i^2 - X_j^2}{h_2}\right),\ i \neq j \end{cases}$$

де згортка ядер була попередньо обчислена на папері. Також зауважимо, що $J_2(h_1, h_2)$ можна подати у вигляді:

$$J_2(h_1, h_2) = \frac{1}{n(n-1) h_1 h_2} \sum_{i=1}^{n} \sum_{j \neq i} K\left(\frac{X_i^1 - X_j^1}{h_1}\right) K\left(\frac{X_i^2 - X_j^2}{h_2}\right)$$

Отже $\mathrm{CV}(h_1, h_2)$ можна подати у більш зручному вигляді:

$$\mathrm{CV}(h_1,h_2) = \frac{1}{n^2 h_1 h_2}\left(nd^4 + 2\sum_{i=1}^{n}\sum_{j<i}(K*K)\left(\frac{X_i^1 - X_j^1}{h_1}\right)\cdot(K*K)\left(\frac{X_i^2 - X_j^2}{h_2}\right)\right) -$$
$$- \frac{4}{n(n-1)h_1h_2}\sum_{i=1}^{n}\sum_{j<i}K\left(\frac{X_i^1 - X_j^1}{h_1}\right)\cdot K\left(\frac{X_i^2 - X_j^2}{h_2}\right) =$$
$$= \frac{1}{nh_1h_2}\left(d^4 + 2\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{j<i}(K*K)\left(\frac{X_i^1 - X_j^1}{h_1}\right)\cdot(K*K)\left(\frac{X_i^2 - X_j^2}{h_2}\right) -\right.\right.$$
$$\left.\left. - \frac{2}{n-1}\sum_{j<i}K\left(\frac{X_i^1 - X_j^1}{h_1}\right)\cdot K\left(\frac{X_i^2 - X_j^2}{h_2}\right)\right)\right)$$

Програмна реалізація підрахунку наближеної оцінки, отриманої за допомогою крос-валідації (пошук мінімуму робився за допомогою пошуку на гратці, рівномірно розбивши проміжки для h_1, h_2 відповідно):

```
CV.h.2d <- function(h1, h2, x1, x2)
{
  n <- length(x); idx <- 1:n
  double.sum <- sum(sapply(idx, function(j) {
    delta1 <- (x1[idx < j] - x1[j])/h1
    delta2 <- (x2[idx < j] - x2[j])/h2
    A <- sum(epan.kernel.conv(delta1) * epan.kernel.conv(delta2)) * (9/16)^2
    B <- sum(epan.kernel(delta1) * epan.kernel(delta2))
    A / n - 2 * B / (n - 1)
  }))
  (ep.d.sq^2 + 2 * double.sum) / (n * h1 * h2)
}
h.crossvalid.2d <- function(x1, x2, h1_0, h2_0, D = 200)
{
  vals.h1 <- 0.1 * h1_0 * ((D-1):0)/(D-1) + 10 * h1_0 * (0:(D-1))/(D-1)
  vals.h2 <- 0.1 * h2_0 * ((D-1):0)/(D-1) + 10 * h2_0 * (0:(D-1))/(D-1)
  CV.grid <- outer(vals.h1, vals.h2, function(h1, h2) {
    apply(cbind(h1, h2), 1,
          function(u) { CV.h.2d(u[1], u[2], x1, x2) }
          )
    })
  h1.opt <- vals.h1[which.min(apply(CV.grid, 1, min))]
  h2.opt <- vals.h2[which.min(apply(CV.grid, 2, min))]
  c(h1.opt, h2.opt)
}
```

Двовимірна щільність.

Програмна реалізація двовимірної щільності має вигляд:

```
f.est.2d <- function(h1, h2, x1, x2, n, K)
{ function(y1, y2) { sum(K((y1 - x1)/h1) * K((y2 - x2)/h2)) / (n * h1 * h2) } }
```

Хід роботи.

Короткий огляд даних.

Спочатку треба зрозуміти з чим доведеться працювати. Зазвичай початковий аналіз даних може нести в собі вагому інформацію про доцільність використання тих чи інших моделей. Зобразимо загальну діаграму розсіювання даних, тобто пронумерувавши точки, що відповідають конкретним спостереженням, номерами виноградників, де вироблено вина.

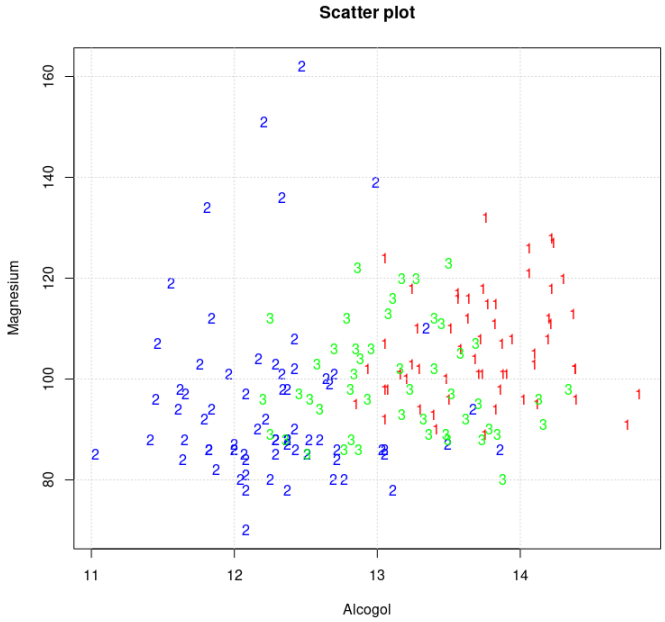

Рис. 1: Загальна діаграма розсіювання.

З діаграми розсіювання видно, що хмарини різних класів сильно перемішуються, що ускладнює можливість побудови хорошого класифікатора. Якщо нашою задачею була б класифікація походження вин до першого чи другого виноградників, то змішування не було такою катастрофічною проблемою (в тому сенсі, що відповідні хмарини для цих виноградників розділяються нормально, на відміну від хмарини зі спостереженнями з третього виноградника). Від перемішування тут не втечеш, бо якщо будувати складніші моделі, які б намагалися розкласифікувати максимально всі точки коректно, то це може призвести до ефекту перепідгонки. В такому випадку, якщо є можливіть, то варто дослідити поведінку даних використовуючи інші пари змінних (де, наприклад, розмежування між класами видно чіткіше). Однак нам потрібно робити класифікацію саме на цих даних, тому зайві кроки не будемо робити.

Формування навчальної та тестової вибірок.

Дані складаються з $n = 178$ спостережень. Обсяг тренувальної вибірки обрано так, аби частка відбору становила $f \approx 0.8$. Тобто обсяг тренувальної вибірки буде становити $n_{\text{train}} = 142$. Цього має бути достатньо для нормальної підгонки нескладних класифікаційних моделей.

```
wine.data <- read.csv2("wine.csv", header = T)[,c("Alcogol","Magnesium","Site")]
N <- nrow(wine.data)
...
```

```
...
set.seed(0)
p <- 0.8
N.train <- floor(p * N)
idx.train <- sample(1:N, N.train)
# Навчальна вибірка
wine.data.train <- wine.data[idx.train,]
# Контрольна вибірка
wine.data.test <- wine.data[-idx.train,]
```

У сформований тренувальній вибірці частки спостережень з відповідних класів (виноградників) становить:

$$(\hat{\pi}_1, \hat{\pi}_2, \hat{\pi}_3) = (0.3169014, 0.4014085, 0.2816901)$$

Тобто маємо на кожний клас по $\approx 40 \sim 50$ спостережень, що здається достатнім для оцінювання. Зокрема отримані частки можна використовувати в якості оцінки апріорного розподілу номера виноградника, що власне і знадобиться при побудові емпірично-баєсового класифікатора.

Наївний баєсів класифікатор.

Коментар про дані.

Наївний баєсів класифікатор базується на припущенні, що досліджувані змінні для кожного спостереження є незалежними в сукупності, тобто щільність спостереження розпадається на добуток щільностей за кожною змінною:

$$f(x_1, \ldots, x_n) = \prod_{j=1}^{n} f_j(x_j)$$

Роботу цієї моделі ми продемонструємо лише для ознайомлення, оскільки наші дані не задовольняють висунутому припущенню. Дійсно, наприклад, це можна перевірити за допомогою кореляційного тесту Спірмена про корельованість змінних:

$$\mathbf{H}_0 : \rho_{spearman}(Alcogol, Magnesium) = 0, \ \mathbf{H}_1 : \rho_{spearman}(Alcogol, Magnesium) \neq 0$$

```
test.spearman <- cor.test(wine.data$Alcogol, wine.data$Magnesium,
                          method = "spearman", exact = F)
```

Обчислений досягнутий рівень значущості становить $\approx 5.256 \cdot 10^{-07}$, що є досить маленьким значенням для довільно адекватно взятого рівня значущості. Тому приймаємо гіпотезу про те, що досліджувані змінні є корельованими. Тобто розглянути модель можна, однак теоретично її застосування у даному разі є недоцільним.

Оцінювання одновимірних щільностей.

Для побудови наївного баєсового класифікатора потрібно мати якісь попередньо обчислені оцінки щільностей для досліджуваних змінних. Ми скористаємося ядерними оцінками щільності з ядром Єпанєчнікова та параметром згладжування, який є приблизною точкою мінімуму функціонала крос-валідації.

```
# Будуємо оцінки щільностей для кожної групи
# Пілотні оцінки для згладжування
h.j.pilot.alc <- sapply(wine.sites, function(j) {
  h.silv.improved(ep.d.sq, ep.D,
                  wine.data.train$Alcogol[wine.data.train$Site == j])
  })
h.j.pilot.mag <- sapply(wine.sites, function(j) {
  h.silv.improved(ep.d.sq, ep.D,
                  wine.data.train$Magnesium[wine.data.train$Site == j])
})

# CV-оцінки згладжування
h.j.cv.alc <- sapply(wine.sites, function(j) {
  h.crossvalid(wine.data.train$Alcogol[wine.data.train$Site == j],
               0.1 * h.j.pilot.alc[j], 10 * h.j.pilot.alc[j])
})
h.j.cv.mag <- sapply(wine.sites, function(j) {
  h.crossvalid(wine.data.train$Magnesium[wine.data.train$Site == j],
               0.1 * h.j.pilot.mag[j], 10 * h.j.pilot.mag[j])
})

# Ядерні оцінки щільності на основі CV-оцінки згладжування
alc.dens <- sapply(wine.sites, function(j) {
  dens.estim(wine.data.train$Alcogol[wine.data.train$Site == j],
             h.j.cv.alc[j], epan.kernel)
})
mag.dens <- sapply(wine.sites, function(j) {
  dens.estim(wine.data.train$Magnesium[wine.data.train$Site == j],
             h.j.cv.mag[j], epan.kernel)
})
```

Отримані значення оцінок методом крос-валідації для параметрів згладжування такі:

	Site = 1	Site = 2	Site = 3
Alcogol, h_1	0.7552105	0.4845489	0.7739326
Magnesium, h_2	13.357422	6.475023	11.641866

Табл. 1: Оцінки параметрів згладжування методом крос-валідації.

Було б цікаво подивитися на графіки оцінок щільностей для кожної з змінних та в залежності від номера виноградника. Це ми продемонструємо далі:

```
# Будуємо графіки оцінок щільностей
# Для Alcogol
eps <- 1
min.alc <- min(wine.data.train$Alcogol); max.alc <- max(wine.data.train$Alcogol)
I.alc <- seq(min.alc - eps, max.alc + eps, 0.01)
...
```

```
...
plot(c(I.alc, I.alc, I.alc), as.numeric(sapply(wine.sites, function(j) {
  (alc.dens[[j]])(I.alc)
})), type = "n", xlab = "Alcogol", ylab = "Density",
main = "Графіки оцінок щільностей за Alcogol"); grid()
for(j in wine.sites)
{
  lines(I.alc, (alc.dens[[j]])(I.alc), col = c("red", "blue", "green")[j])
}
legend("topleft", legend = c("Site = 1", "Site = 2", "Site = 3"),
       col = c("red", "blue", "green"), lwd = 1)
# Аналогічно для Magnesium
```

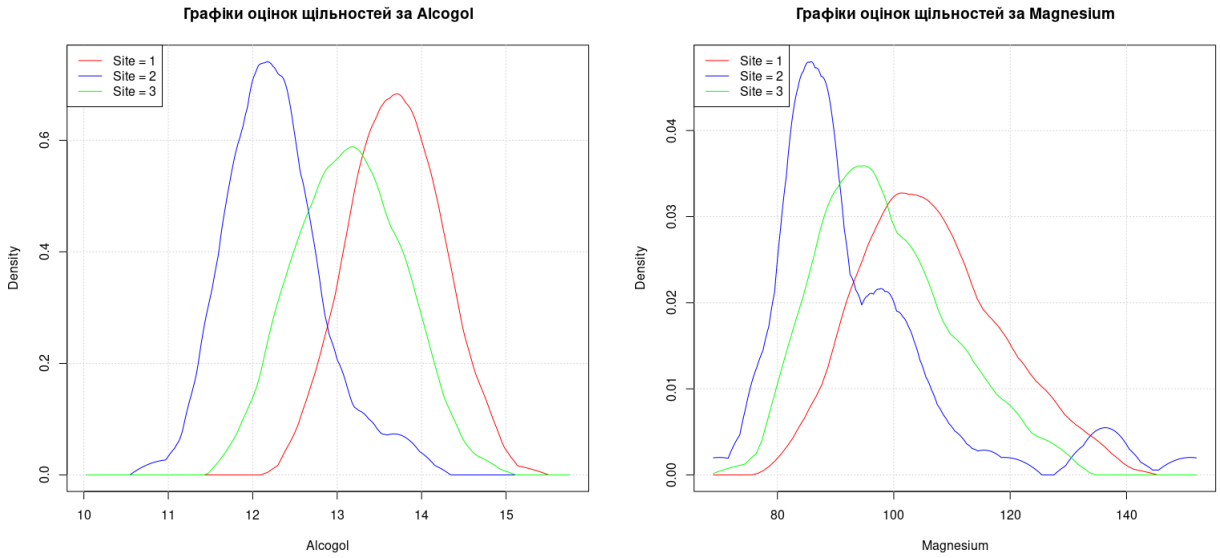

Рис. 2: Графіки оцінок щільностей для Alcogol та Magnesium.

З графіків можна прикинути на око, що для першого та другого виноградників ймовірність хибно віднести спостереження є не настільки великим, яке може вийти, взявши до уваги спостереження з третього виноградника.

Побудова класифікатора: застосування, перевірка якості.

Наївний баєсівський класифікатор визначається за формулою:

$$g^{NB}(x^{alc}, x^{mag}) = \arg\max_{m=1,2,3} \hat{\pi}_m \cdot \hat{f}^m_{alc}(x^{alc}) \cdot \hat{f}^m_{mag}(x^{mag})$$

де $\{\hat{\pi}_m\}_{m=1}^3$ – раніше згадані оцінки апріорного розподілу номера виноградника:

```
# Оцінюємо апріорний розподіл номера виноградника
pi.j <- sapply(wine.sites, function(j) { mean(wine.data.train$Site == j) })
```

Підрахувавши все необхідне, можна реалізувати бажаний класифікатор:

```
# alc.dens, mag.dens - списки з функціями, що реалізують оцінки щільностей
g.emp.nbayes <- function(x.alc, x.mag)
{
  fj.alc.emp.x <- c(
      (alc.dens[[1]])(x.alc), (alc.dens[[2]])(x.alc), (alc.dens[[3]])(x.alc))
  fj.mag.emp.x <- c(
      (mag.dens[[1]])(x.mag), (mag.dens[[2]])(x.mag), (mag.dens[[3]])(x.mag))
  which.max(pi.j * fj.alc.emp.x * fj.mag.emp.x)
}
```

Підгонка класифікатора по навчальній вибірці дає наступні відомості про якість класифікації (на тих самих даних): частота неправильних класифікацій становить $L^{NB}_{\text{train}}(X) \approx 0.2535211$, тобто майже чверть спостережень з тренувального набору вгадано неправильно. На контрольній вибірці похибка погіршилася несуттєво: $L^{NB}_{\text{test}}(X) \approx 0.2777778$. Числові характеристики, звісно, певну інформацію дають, однак не можна обійтися таблицями спряженості, побудованих за пронозом класифікатора та значеннями відгука (справжніх номерів виноградників):

Прогноз \ Відгук	1	2	3
1	37	1	13
2	0	52	10
3	8	4	17

Прогноз \ Відгук	1	2	3
1	13	2	4
2	0	10	1
3	1	2	3

Табл. 2: Таблиці спряженості для прогнозу і відгука: зліва для навчальної вибірки, справа для тестової.

Класифікатор погано відносить спостереження з третього виноградника до відповідного класу, що можна було очікувати. Для інших спостережень все вгадується відносно не так і погано, як очікувалося. Зобразимо множини рішень класифікатора:

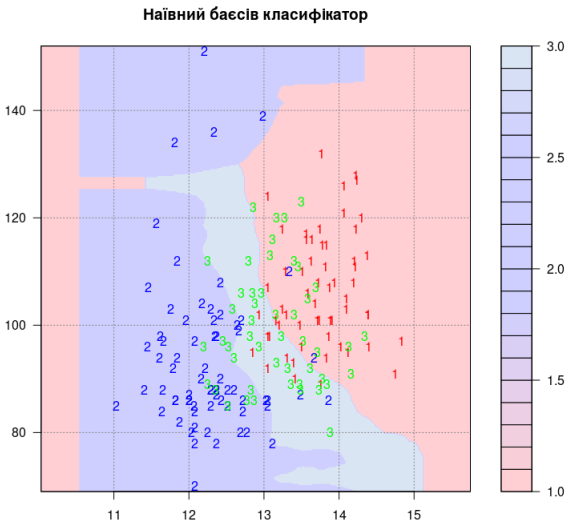

Рис. 3: Візуалізація роботи класифікатора.

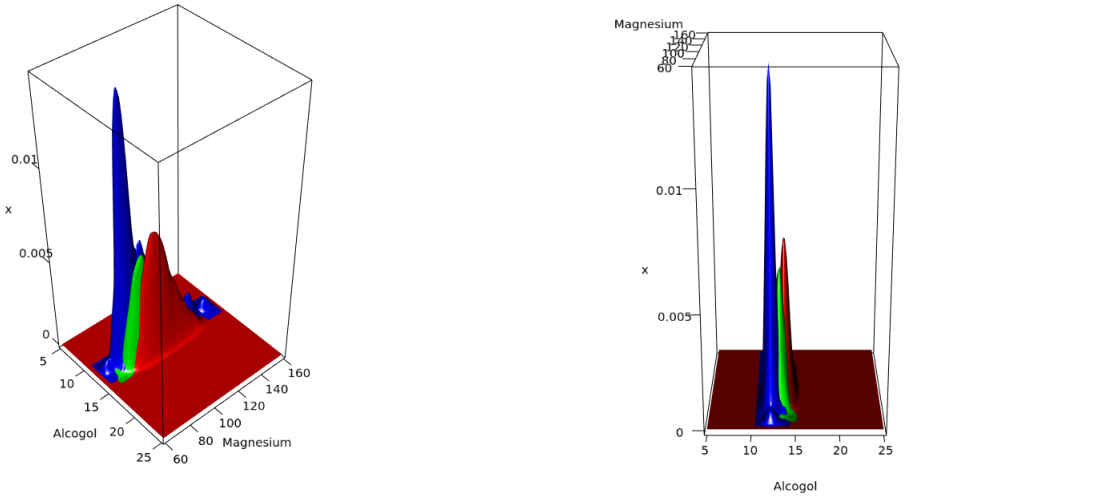

Рис. 4: Візуалізація роботи класифікатора. Графіки $\hat{\pi}_m \cdot \hat{f}^m_{alc}(x^{alc}) \cdot \hat{f}^m_{mag}(x^{mag})$.

З рисунка видно, що області рішень більш-менш виглядать природньо. Різке затухання значень "апостеріорної" ймовірності для другого виноградника пояснюється вибором носія для побудованої оцінки. В цілому видно, що класифікатор непогано може розмежувати спостереження з першого та другого виноградників, втім маємо оочевидні проблеми з класифікацією вин з третього виноградника внаслідок сильного перемішування. Даний класифікатор начебто нормальний, однак слід пам'ятати, що він побудований на хибному припущенні про незалежність змінних. Спробуємо перейти до більш обгрунтованих моделей.

Емпірично-баєсів класифікатор.

Оцінювання двовимірних щільностей.

Задача дещо ускладнюється тим, що потрібно тепер оцінювати векторний параметр згладжування в ядерній оцінці двовимірної щільності (1). Для цього ми скористаємося технікою крос-валідації, узагальненої на багатовимірний випадок для розглянутої оцінки щільності (з попередньої роботи ми переконалися у тому, що якісь здорові результати можна з цього підходу вичепити). Хоча ніхто не гарантує адекватність на даних такого малого обсягу. Не наводячи громіздкий шматок коду, лише покажемо отримані наближені точки мінімуму (якого?) функціоналу крос-валідації:

	Site = 1	Site = 2	Site = 3
Alcogol, h_1	0.751795	1.3501627	1.270961
Magnesium, h_2	12.632497	0.6475023	3.480859

Табл. 3: Оцінки параметрів згладжування методом багатовимірної крос-валідації.

Побудова класифікатора: застосування, перевірка якості.

Емпірично-баєсівський класифікатор визначається за формулою:

$$g^B(x^{alc}, x^{mag}) = \arg\max_{m=1,2,3} \hat{\pi}_m \cdot \hat{f}^m(x^{alc}, x^{mag})$$

Маючи все необхідне, можемо реалізувати даний класифікатор:

```
# fj.est - оцінки двовимірних щільностей в залежності від номера
g.emp.bayes <- function(x.alc, x.mag)
{
  fj.emp.x <- c(f1.est(x.alc, x.mag), f2.est(x.alc, x.mag), f3.est(x.alc, x.mag))
  which.max(pi.j * fj.emp.x)
}
```

Підгонка класифікатора по навчальній вибірці дає наступні відомості про якість класифікації (на тих самих даних): частота неправильних класифікацій становить $L^B_{\text{train}}(X) \approx 0.2112676$, тобто трохи менше чверті спостережень з тренувального набору вгадано неправильно. На контрольній вибірці похибка погіршилася: $L^B_{\text{test}}(X) \approx 0.3333333$. На цій тренувальній вибірці, похибка зменшилася, але маємо погіршення на тестовій вибірці. Щоб зрозуміти що вгадано правильно або неправильно, знову побудуємо таблиці спряженості для цього класифікатора:

Прогноз \ Відгук	1	2	3
1	37	1	13
2	1	56	8
3	7	0	19

Прогноз \ Відгук	1	2	3
1	12	3	5
2	1	11	2
3	1	0	1

Табл. 4: Таблиці спряженості для прогнозу і відгука: зліва для навчальної вибірки, справа для тестової.

І цей класифікатор не зовсім добре відносить класифікує спостереження з третього виноградника (на контрольній вибірці майже нічого не вгадано для цього класу). Візуалізуємо роботу емпірично-баєсового класифікатора:

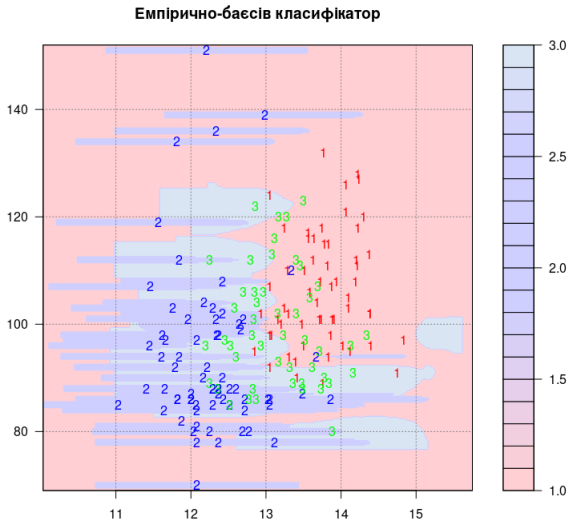

Рис. 5: Візуалізація роботи класифікатора.

З рисунка видно, що класифікатор працює кострубато. Множина для присвоєння другого номера виглядає не зовсім природньо: для віддалених точок відмічено окремі смужки. Трактувати ці точки як викиди ми, очевидно, не можемо (з інтуїтивних міркувань та того, що ми

не настільки добре спеціалізуємося на винах). Область для присвоєння третього номера накладається попередньою. Такий класифікатор навряд чи захочеться використовувати надалі на подібних даних. Хоча з іншого боку такі аномальні графіки для оцінок щільностей можна пояснити невдалою роботою двовимірної крос-валідації внаслідок недостатньо великого обсягу даних і грубістю методу пошуку приблизного мінімуму.

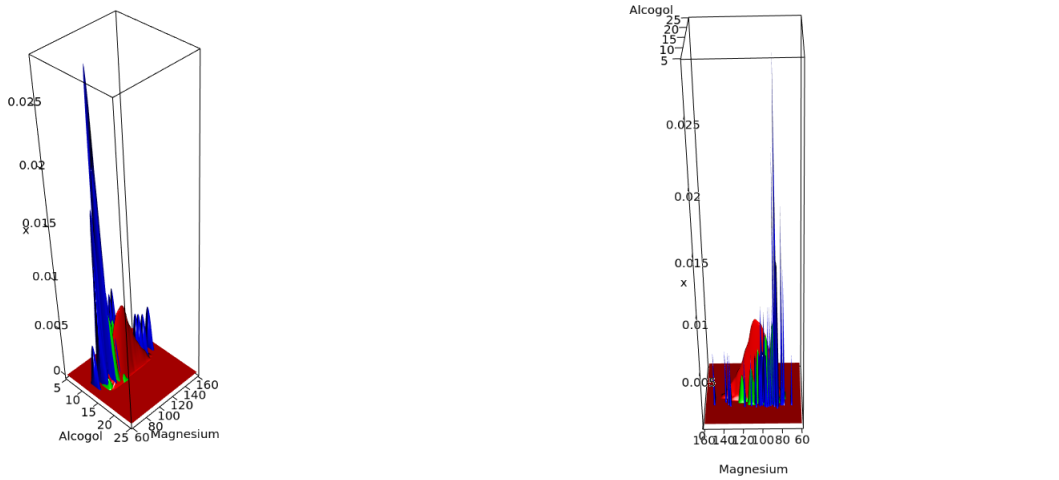

Рис. 6: Візуалізація роботи класифікатора. Графіки $\hat{\pi}_m \cdot \hat{f}^m(x^{alc}, x^{mag})$.

Спробуємо простіше підійти до вибору параметрів згладжування в оцінці двовимірної щільності. Застосуємо просте двовимірне правило Сільвермана:

$$\hat{h}_k^m = \left(\frac{4}{d+2}\right)^{1/(d+4)} n^{-1/(d+4)} \sqrt{\hat{S}_0^2(X_k^m)},\ k = 1, 2,\ m = 1, 2, 3,\ d := 2$$

Принаймані ми виграємо у часі в підрахунку значень векторного параметра. Маємо:

	Site = 1	Site = 2	Site = 3
Alcogol, h_1	0.2359844	0.2933222	0.2869752
Magnesium, h_2	5.7531159	8.4287321	5.6075592

Табл. 5: Оцінки параметрів згладжування за багатовимірним правилом Сільвермана.

Зауважте, що параметри згладжування вийшли іншими, особливо для змінних за другим номером виноградника. За правилом Сільвермана, наприклад, за другим номером виноградника для змінної Magnesium $h_2 \approx 8.4287321$, що набагато більше за те значення, що мали за багатовимірною крос-валідацією. Використовуючи ці значення для згладжування двовіимірних оцінок, маємо такі результати для класифікатора: на навчальній вибірці $L^B_{\text{train}}(X) \approx 0.1619718$, а на контрольній похибка така: $L^B_{\text{test}}(X) \approx 0.25$. Поки що це рекорд серед усіх класифікаторів, які ми встигли побудувати. Зобразимо таблиці спряженості для прогнозу і відгука:

Прогноз \ Відгук	1	2	3
1	40	0	9
2	0	52	4
3	5	5	27

Прогноз \ Відгук	1	2	3
1	11	2	1
2	0	10	1
3	3	2	6

Табл. 6: Таблиці спряженості для прогнозу і відгука: зліва для навчальної вибірки, справа для тестової.

Видно, що поправка на залежність змінних (тобто не розбиваючи двовимірну щільність на добуток одновимірних) дає кращі результати. Зокрема менше помилок вгадування виходить для спостережень з третього виноградника. З графіків це все теж видно:

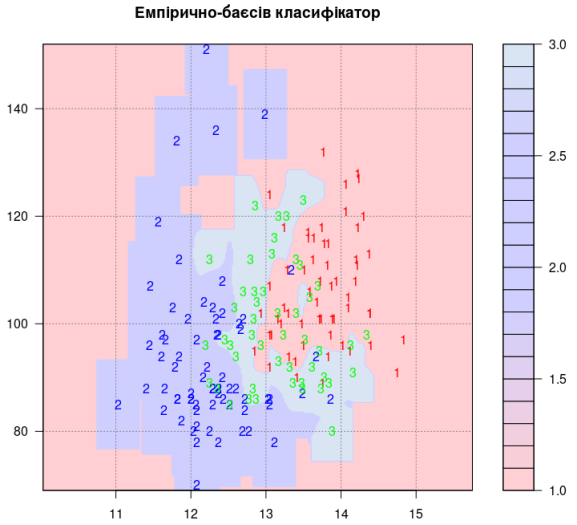

Рис. 7: Візуалізація роботи класифікатора.

Нижче продемонструємо графіки для нормованих щільностей.

Проекційний емпірично-баєсів класифікатор.

Коментар про дані.

Застосування проекційного класифікатора є поганою ідеєю для наших даних внаслідок сильного перемішування. Тому не варто очікувати хороші результати у даному підпункті.

Вибір оптимального напрямку проекції.

Ідея базується на тому, що спочатку береться деякий направляючий вектор одиничної довжини $\vec{b} = (\cos(\beta), \sin(\beta))$, $\beta \in [0, 2\pi)$ і обчислюються коефіцієнти ортогональної проекції жаних на лінійний підпростір, натягнутий на заданий вектор: $c_j = \left\langle X_j, \vec{b} \right\rangle = X_j^1 \cos(\beta) + X_j^2 \sin(\beta)$. Далі класифікація робиться на отриманих коефіцієнтах з використанням раніше згаданого емпірично-баєсового класифікатора. Вибір оптимального напрямку буде робитися з метою мінімізації частоти помилки класифікації:

$$L_{\text{train}}^{PB,\beta}(X) \to \min_{\beta \in [0,2\pi)}$$

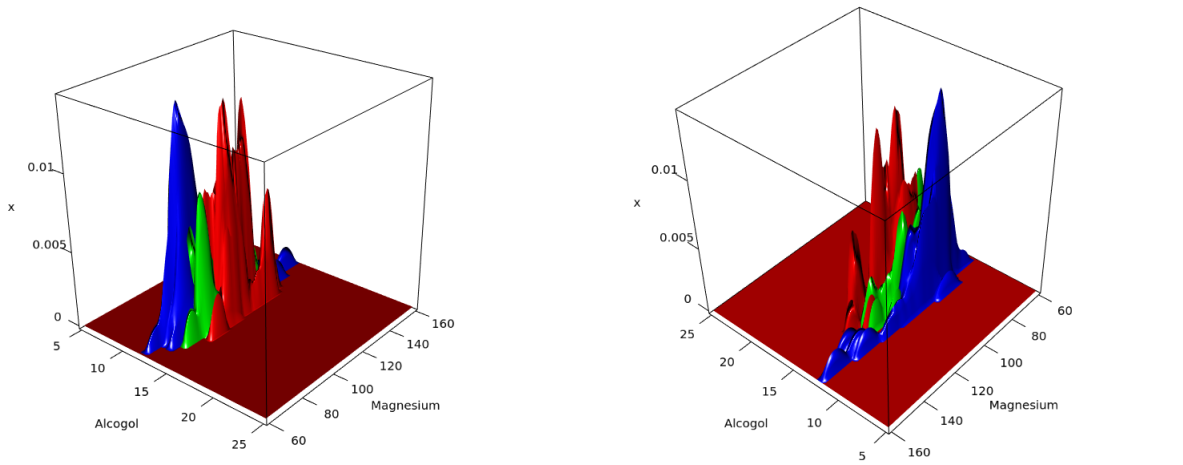

Рис. 8: Візуалізація роботи класифікатора. Графіки $\hat{\pi}_m \cdot \hat{f}^m(x^{alc}, x^{mag})$.

Побудова класифікатора: застосування, перевірка якості.

Виявляється, що оптимальним значенням β, що мінімізує частоту помилки класифікації на тренувальних даних, є $\beta^{opt} \approx 1.94141$ (з використанням якого помилка проекційно-баєсового класифікатора становить $L_{\text{train}}^{PB,\beta^{opt}}(X) \approx 0.3239437$). Помилка на тестових даних дорівнює $L_{\text{test}}^{PB,\beta^{opt}}(X) \approx 0.6944444$, що є повним провалом роботи класифікатора на нових даних (а цього ми і очікували).

Продемонструємо таблиці спряженості:

Прогноз \ Відгук	1	2	3
1	34	10	7
2	4	37	8
3	7	10	25

Прогноз \ Відгук	1	2	3
1	5	4	3
2	2	3	2
3	7	7	3

Табл. 7: Таблиці спряженості для прогнозу і відгука: зліва для навчальної вибірки, справа для тестової.

І зобразимо множини рішень класифікатор нижче.

Висновки.

Найкраще себе продемонстрував емпірично-баєсів класифікатор на цих даних. Ключовим моментом у побудові хорошого класифікатора виявилося підгонка "гіперпараметрів" у ньому (як то кажуть, як корабель назвеш так він і попливе). Для доввимірних оцінок доцільно було використати емпіричні правила на відміну від складних технік на такому невеликому обсязі даних. Використання проекційного методу не має сенсу внаслідок неможливості інтуїтивного підбору підпростору, на якому б дані мали б більш-менш нормальне розділення по класам.

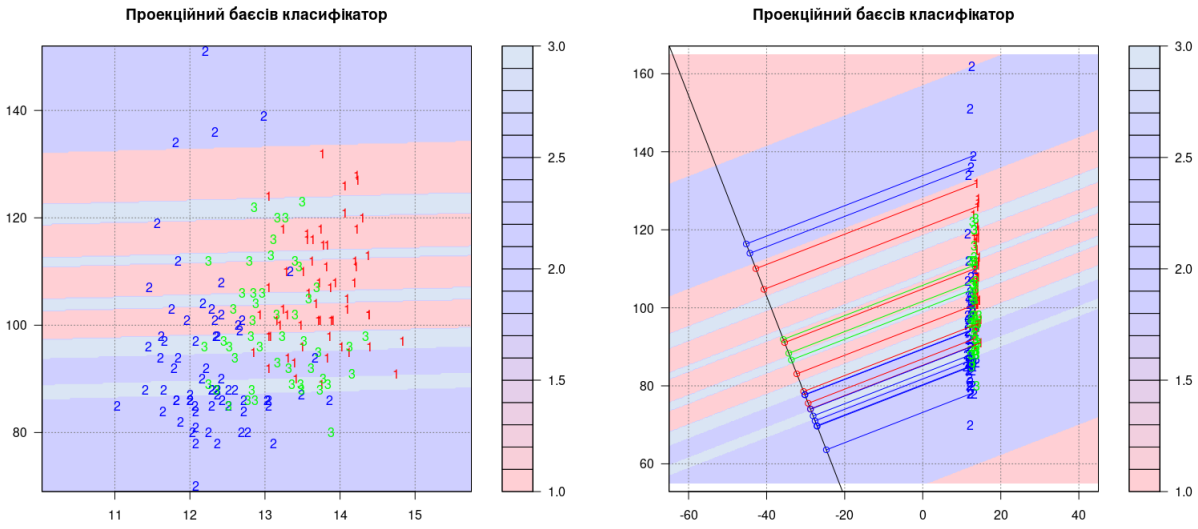

Рис. 9: Візуалізація роботи класифікатора.