Modele liniowe do klasyfikacji

Wojciech Wojnar

21 kwietnia 2021

Wprowadzenie

Dlaczego liniowe?

$$\Pr(G = 1|X = x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}$$

$$\Pr(G = 2|X = x) = \frac{1}{1 + \exp(\beta_0 + \beta^T x)}$$

Odpowiedź: Liniowa granica decyzyjna.

$$\log \frac{\Pr(G = 1|X = x)}{\Pr(G = 2|X = x)} = \beta_0 + \beta^T x$$

Przykłady

Przykłady klasyfikatorów liniowych:

- Liniowa analiza dyskryminacyjna
- Regresja logistyczna
- Hiperpłaszczyzny separujące

Liniowa analiza dyskryminacyjna (Linear Discriminant Analysis)

Szukamy: $\Pr(G|X)$
Niech $f_k(x)$ będzie gęstością X w grupie $G = k$ oraz niech π_k będzie prawdopodobieństwem, że obserwacja pochodzi z klasy k i $\sum_{k=1}^{K} \pi_k = 1$.
Z twierdzenia Bayesa otrzymujemy:

$$\Pr(G = k|X = x) = \frac{\Pr(G \cap X)}{\Pr(X)} = \frac{\Pr(X|G)\Pr(G)}{\Pr(X)} = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

Wyprowadzenie LDA

Załóżmy, że w każdej klasie zmienne mają rozkład normalny:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k))$$

LDA dodatkowo zakłada, że w każdej z grup mamy taką samą strukturę kowariancji: $\Sigma_k = \Sigma \ \forall k$. Wówczas możemy wyprowadzić iloraz log-wiarogodności:

$$\begin{aligned} \log \frac{\Pr(G=k|X=x)}{\Pr(G=l|X=x)} &= \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l} \\ &= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)\Sigma^{-1}(\mu_k-\mu_l) \\ &+ x^T\Sigma^{-1}(\mu_k-\mu_l) \end{aligned} \quad (1)$$

Funkcja decyzyjna

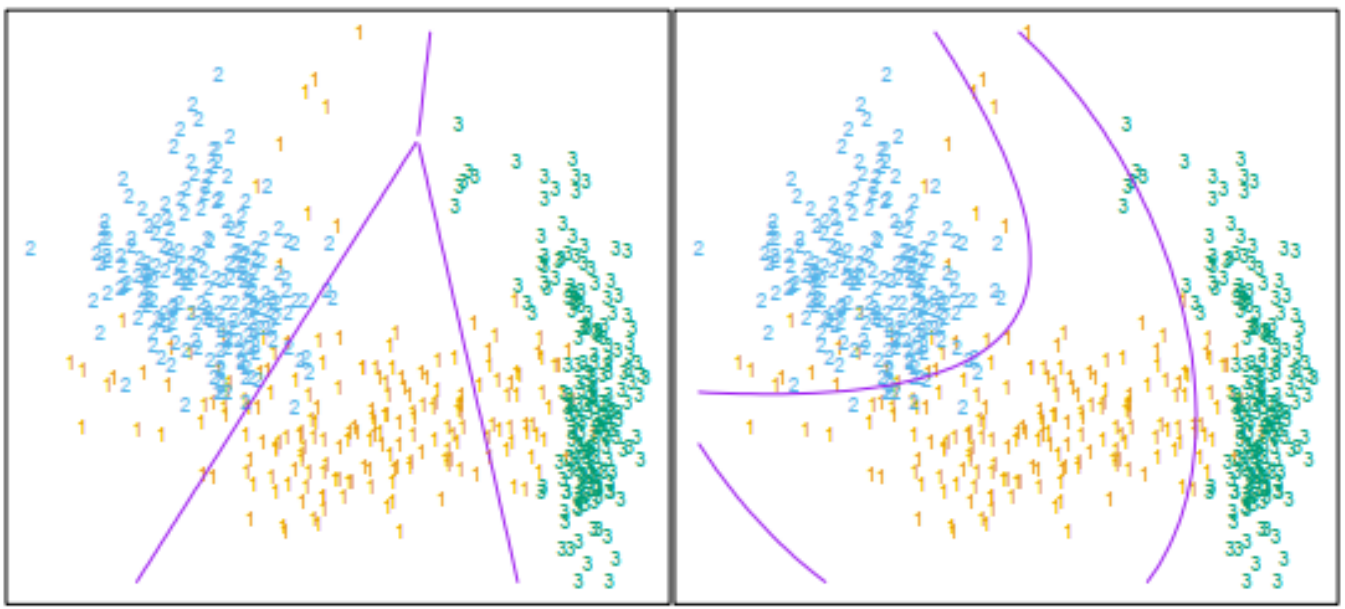
Funkcja granicy decyzyjnej przyjmuje zatem liniową formę (linear discriminant function):

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

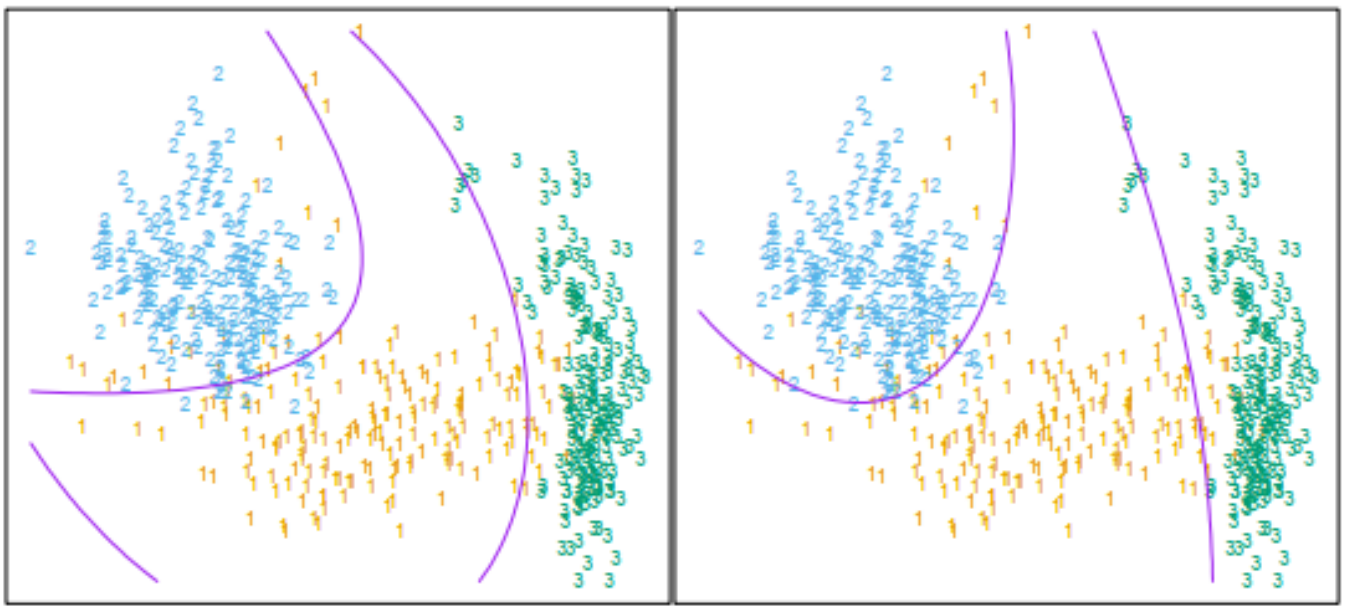
Natomiast gdy nie mamy spełnionego założenia $\Sigma_k = \Sigma \ \forall k$, wówczas otrzymujemy kwadratową funkcję decyzyjną (quadratic discriminant function):

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log \pi_k$$

LDA vs QDA



LDA vs QDA



Regularyzowana Analiza Dyskryminacyjna

Friedman (1989) zaproponował kompromis pomiędzy LDA i QDA:

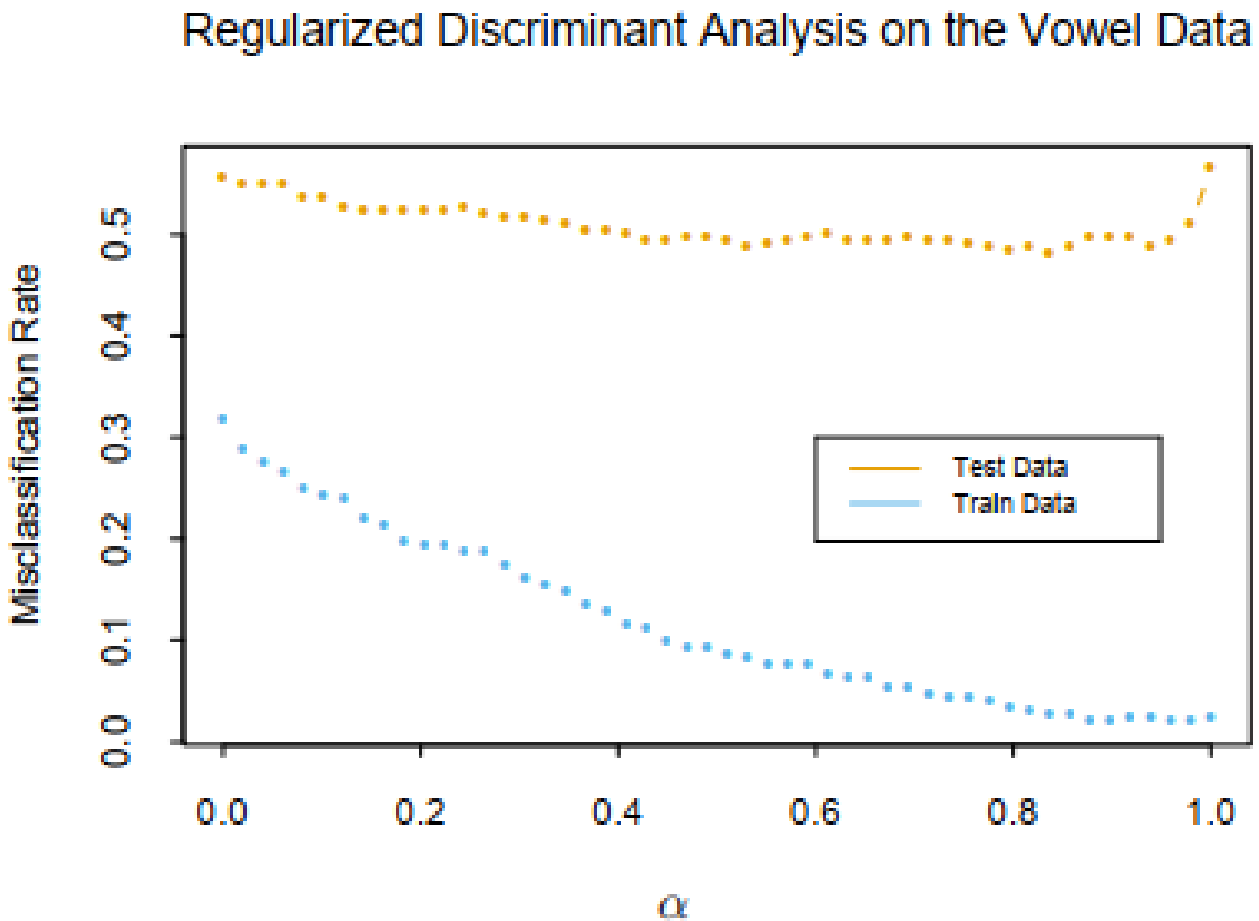
$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}$$

Dodatkowo można wprowadzić dodatkową modyfikację Σ:

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1-\gamma) + \hat{\sigma}^2\mathrm{I}$$

Połączenie tych dwóch modyfikacji prowadzi do powstania ogólniejszej rodziny macierzy kowariancji $\hat{\Sigma}(\alpha, \gamma)$.

Regularyzowana Analiza Dyskryminacyjna



Regresja logistyczna

Celem regresji logistycznej jest modelowanie prawdopodobieństwa zdarzenia, że obserwacja należy do danej klasy w taki sposób, aby prawdopodobieństwa sumowały się do 1, model przyjmuje następującą formę:

$$\log \frac{\Pr(G = k|X = x)}{\Pr(G = K|X = x)} = \beta_{k0} + \beta_k^T x, \text{ dla k} = 1,...,\text{K-1}$$

Wówczas szukane prawdopodobieństwa mają następującą postać:

$$\Pr(G = k|X = x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}, \text{ dla k} = 1,...,\text{K-1}$$

$$\Pr(G = K|X = x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}$$

Uczenie modelu

Model regresji logistycznej uczymy z wykorzystaniem metody największej wiarogodności (MLE), gdzie log-wiarogodność dla próby N-elementowej przyjmuje postać:

$$l(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta),$$

gdzie $p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$ oraz
$\theta = (\beta_{10}, \beta_1^T, ..., \beta_{(K-1)0}, \beta_{K-1}^T)$
Najczęściej regresję logistyczną stosuje się w problemie klasyfikacji zmiennej zero-jedynkowej, wówczas możemy również zapisać $p_1 = p(x; \theta)$ i $p_0 = 1 - p(x; \theta)$.

Uczenie modelu cd.

Funkcję wiarogodności możemy zatem zapisać następująco:

$$\begin{aligned} l(\beta) &= \sum_{i=1}^{N} \{y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta))\} \\ &= \sum_{i=1}^{N} \{y_i \beta^T x_i - \log(1+\exp \beta^T x_i)\} \end{aligned} \tag{2}$$

Tutaj $\beta = (\beta_{10}, \beta_1)$ i zakładamy, że obserwacja x_i zawiera 1 odpowiadającą wyrazowi wolnemu (intercept).

Maksymalizacja funkcji log-wiarogodności

W celu maksymalizacji funkcji log-wiarogodności przyrównujemy pochodną do 0:

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i (y_i - p(x_i; \beta)) = 0$$

Do rozwiązania powyższego problemu używa się np. algorytmu Newtona - Raphsona, który potrzebuje Hesjan:

$$\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i; \beta)(1 - p(x_i; \beta))$$

Algorytm Newtona-Raphsona

Zaczynając od β^{old}, pojedyczy krok algorytmu wygląda następująco:

$$\beta^{new} = \beta^{old} - (\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T})^{-1}\frac{\partial l(\beta)}{\partial\beta},$$

pochodne liczone są w punkcie β^{old}.
Gradient i Hesjan możemy zapisać w formie macierzowej:

$$\frac{\partial l(\beta)}{\partial\beta} = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{p})$$

$$\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T} = -\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$$

Algorytm Newtona-Raphsona

Krok algorytmu możemy zatem zapisać:

$$\begin{aligned}\beta^{new} &= \beta^{old} + (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T (\boldsymbol{y} - \boldsymbol{p}) \\ &= (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} (\boldsymbol{X} \beta^{old} + \boldsymbol{W}^{-1} (\boldsymbol{y} - \boldsymbol{p})) \qquad (3) \\ &= (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{z}\end{aligned}$$

W drugiej i trzeciej linijce zapisaliśmy krok algorytmu jako metodę "ważonych najmniejszych kwadratów" ze "skorygowaną" zmienną odpowiedzi:

$$\boldsymbol{z} = (\boldsymbol{X} \beta^{old} + \boldsymbol{W}^{-1} (\boldsymbol{y} - \boldsymbol{p})$$

Podsumowanie

W każdym kroku algorytmu rozwiązywany jest problem:

$$\beta^{new} \leftarrow \arg\min_{\beta}(\boldsymbol{z} - \boldsymbol{X}\beta)\boldsymbol{W}(\boldsymbol{z} - \boldsymbol{X}\beta),$$

punktem startowym dla metody iteracyjnej może być $\beta = 0$, algorytm z reguły zbiega.
Biblioteki w R z implementacjami regresji logistycznej:

- stats (funkcja: glm - podstawowa, zmienna odpowiedzi zero-jedynkowa)
- glmnet (rozbudowana i efektywna nawet dla dużej liczby obserwacji i parametrów, dostępna wieloklasowa zmienna odpowiedzi)

Zaletą regresji logistycznej w analizie danych jest naturalna interpretowalność parametrów.

Perceptron Rosenblatta

Cel: minimalizacja odległości źle sklasyfikowanych punktów od granicy decyzyjnej.
Jeśli $y_i = 1$ jest źle sklasyfikowane, wówczas $x_i^T \beta + \beta_0 < 0$, a gdy $y_i = -1$, jest źle sklasyfikowane mamy: $x_i^T \beta + \beta_0 > 0$.
Naszym celem jest minimalizacja następującego wyrażenia:

$$D(\beta, \beta_0) = -\sum_{i \in \mathcal{M}} y_i (x_i^T \beta + \beta_0),$$

gdzie $\mathcal{M}$ jest zbiorem indeksów źle sklasyfikowanych punktów.

Perceptron Rosenblatta c.d.

Gradienty względem β i β_0 wyglądają następująco:

$$\frac{\partial D(\beta, \beta_0)}{\partial \beta} = -\sum_{i \in \mathcal{M}} y_i x_i$$

$$\frac{\partial D(\beta, \beta_0)}{\partial \beta_0} = -\sum_{i \in \mathcal{M}} y_i$$

Zatem parametry są aktualizowane w następujący sposób:

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i x_i \\ y_i \end{pmatrix}, \tag{4}$$

gdzie ρ jest wielkością kroku.

Perceptron Rosenblatta c.d.