Regresja logistyczna. Rozdzielanie hiperpłaszczyzną.

Justyna Fijałkowska

Uniwersytet Wrocławski

26 maja 2023

Plan prezentacji

Regresja logistyczna

Postać modelu regresji logistycznej:

$$\log \frac{P(G=1|X=x)}{P(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$\log \frac{P(G=2|X=x)}{P(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$\log \frac{P(G=K-1|X=x)}{P(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x.$$

Regresja logistyczna

Zakładamy, że model regresji logistycznej jest następującej postaci:

$$\log \frac{P(G=1|X=x)}{P(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$\log \frac{P(G=2|X=x)}{P(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$\log \frac{P(G=K-1|X=x)}{P(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x.$$

Stąd zachodzą poniższe równości:

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

Dopasowanie modelu regresji logistycznej

Dopasowania dokonuje się metodą największej wiarogodności z funkcją wiarogodności postaci:

$$L(\theta) = \prod_{i=1}^{N} P(G = g_i | X = x_i; \theta)$$

Dopasowanie modelu regresji logistycznej

Dopasowania dokonuje się metodą największej wiarogodności z funkcją wiarogodności postaci:

$$L(\theta) = \prod_{i=1}^{N} P(G = g_i | X = x_i; \theta)$$

lub z funkcją log-wiarogodności:

$$l(\theta) = \sum_{i=1}^{N} \log P(G = g_i | X = x_i; \theta).$$

Dopasowanie modelu regresji logistycznej dla $K = 2$

Załóżmy, iż liczba klas jest równa $K = 2$. Przypisujemy y_i wartość 1, jeżeli $g_i = 1$ lub 0, jeżeli $g_i = 2$. Wprowadźmy oznaczenia:

$$p(x; \beta) = P(G = 1|X = x_i; \beta) \quad \text{oraz} \quad 1 - p(x; \theta) = P(G = 2|X = x_i; \beta),$$

Dopasowanie modelu regresji logistycznej dla $K = 2$

Załóżmy, iż liczba klas jest równa $K = 2$. Przypisujemy y_i wartość 1, jeżeli $g_i = 1$ lub 0, jeżeli $g_i = 2$. Wprowadźmy oznaczenia:

$$p(x; \beta) = P(G = 1|X = x_i; \beta) \quad \text{oraz} \quad 1 - p(x; \theta) = P(G = 2|X = x_i; \beta),$$

wówczas log-wiarogodność jest postaci:

$$l(\beta) = \sum_{i=1}^{N} \{y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))\} = \sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\},$$

gdzie $\beta = \{\beta_{10}, \beta_1\}$.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Cel: Maksymalizacja funkcji log-wiarogodności.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Cel: Maksymalizacja funkcji log-wiarogodności.

Pomysł: Wyznaczenie miejsca zerowania się pochodnych cząstkowych.

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

Dopasowanie modelu regresji logistycznej dla $K = 2$

Cel: Maksymalizacja funkcji log-wiarogodności.

Pomysł: Wyznaczenie miejsca zerowania się pochodnych.

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

Problem: $p + 1$ równości nieliniowych ze względu na β.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Cel: Maksymalizacja funkcji log-wiarogodności.

Pomysł: Wyznaczenie miejsca zerowania się pochodnych.

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

Problem: $p + 1$ równości nieliniowych ze względu na β.

Rozwiązanie: Zastosowanie algorytmu Newtona-Raphsona.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Cel: Maksymalizacja funkcji log-wiarogodności.

Pomysł: Wyznaczenie miejsca zerowania się pochodnych.

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

Problem: $p + 1$ równości nieliniowych ze względu na β.

Rozwiązanie: Zastosowanie algorytmu Newtona-Raphsona.

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

Dopasowanie modelu regresji logistycznej dla $K = 2$

Przyjmijmy oznaczenia:

- y - wektor wartości y_i,
- X - macierz rozmiaru $N \times (p + 1)$ wartości x_i,
- p - wektor dopasowanych prawdopodobieństw, gdzie i-ty element oznacza $p(x_i; \beta^{old})$,
- W - diagonalna macierz rozmiaru $N \times N$, gdzie i-ty element na diagonali jest postaci $p(x_i; \beta^{old})(1 - p(x_i; \beta^{old}))$.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Wówczas stosując zapis macierzowy mamy:

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

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

Dopasowanie modelu regresji logistycznej dla $K = 2$

Wówczas stosując zapis macierzowy mamy:

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

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

natomiast pojedynczy krok algorytmu Newtona-Raphsona daje się zapisać w postaci:

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

gdzie:

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

Dopasowanie modelu regresji logistycznej dla $K = 2$

Skoro z_i i w_i zdefiniowane są w poniższy sposób:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)}, \quad w_i = \hat{p}_i(1 - \hat{p}_i),$$

to można pokazać, że:

Dopasowanie modelu regresji logistycznej dla $K = 2$

Skoro z_i i w_i zdefiniowane są w poniższy sposób:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)}, \quad w_i = \hat{p}_i(1 - \hat{p}_i),$$

to można pokazać, że:

- Ważona residualna suma kwadratów jest znaną statystyką chi-kwadrat Pearsona, która stanowi kwadratowe przybliżenie odchylenia.

$$\sum_{i=1}^{N} \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$

Dopasowanie modelu regresji logistycznej dla $K = 2$

Skoro z_i i w_i zdefiniowane są w poniższy sposób:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)}, \quad w_i = \hat{p}_i(1 - \hat{p}_i),$$

to można pokazać, że:

- Ważona residualna suma kwadratów jest znaną statystyką chi-kwadrat Pearsona, która stanowi kwadratowe przybliżenie odchylenia.

$$\sum_{i=1}^{N} \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$

- Jeżeli model jest poprawny, to $\hat{\beta}$ jest estymatorem zgodnym.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Skoro z_i i w_i zdefiniowane są w poniższy sposób:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)}, \quad w_i = \hat{p}_i(1 - \hat{p}_i),$$

to można pokazać, że:

- Ważona residualna suma kwadratów jest znaną statystyką chi-kwadrat Pearsona, która stanowi kwadratowe przybliżenie odchylenia.

$$\sum_{i=1}^{N} \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$

- Jeżeli model jest poprawny, to $\hat{\beta}$ jest estymatorem zgodnym.
- Rozkład $\hat{\beta}$ zbiega do rozkładu normalnego, mianowicie $\hat{\beta} \to N(\beta, (\mathrm{X}^T \mathrm{W} \mathrm{X})^{-1})$.

Dopasowanie modelu regresji logistycznej dla $K = 2$

Skoro z_i i w_i zdefiniowane są w poniższy sposób:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)}, \quad w_i = \hat{p}_i(1 - \hat{p}_i),$$

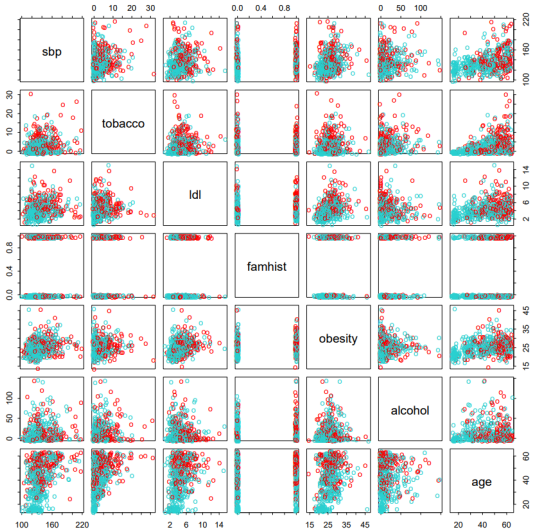
to można pokazać, że:

- Ważona residualna suma kwadratów jest znaną statystyką chi-kwadrat Pearsona, która stanowi kwadratowe przybliżenie odchylenia.

$$\sum_{i=1}^{N} \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$

- Jeżeli model jest poprawny, to $\hat{\beta}$ jest estymatorem zgodnym.
- Rozkład $\hat{\beta}$ zbiega do rozkładu normalnego, mianowicie $\hat{\beta} \to N(\beta, (\mathrm{X}^T \mathrm{W} \mathrm{X})^{-1})$.
- Tworzenie modeli może być kosztowne w przypadku modeli regresji logistycznej, ponieważ każdy dopasowany model wymaga iteracji.

Przykład: Południowoafrykańska choroba serca



Przykład: Południowoafrykańska choroba serca

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Przykład: Południowoafrykańska choroba serca

	Coefficient	Std. Error	Z score
`(Intercept)`	-4.204	0.498	-8.45
`tobacco`	0.081	0.026	3.16
`ldl`	0.168	0.054	3.09
`famhist`	0.924	0.223	4.14
`age`	0.044	0.010	4.52

Regresja logistyczna z karą L_1

Funkcja log-wiarogodności w regresji logistycznej:

$$\sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}.$$

Regresja logistyczna z karą L_1

Funkcja log-wiarogodności w regresji logistycznej:

$$\sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}.$$

Problem maksymalizacji po nałożeniu kary L_1 :

$$\max_{\beta_0, \beta} \left\{ \sum_{i=1}^{N} [y_i(\beta_0 + \beta^T x_i) - \log(1 + e^{\beta_0 + \beta^T x_i})] - \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

Regresja logistyczna czy LDA?

Iloraz prawdopodobieństwa posteriori między klasami k i K w przypadku LDA:

$$\log \frac{P(G=k|X=x)}{P(G=K|X=x)} = \log \frac{\pi_k}{\pi_K} - \frac{1}{2}(\mu_k+\mu_K)^T\Sigma^{-1}(\mu_k-\mu_K)+x^T\Sigma^{-1}(\mu_k-\mu_K) = \alpha_{k0}+\alpha_k^T x.$$

Regresja logistyczna czy LDA?

Iloraz prawdopodobieństwa posteriori między klasami k i K w przypadku LDA:

$$\log \frac{P(G=k|X=x)}{P(G=K|X=x)} = \log \frac{\pi_k}{\pi_K} - \frac{1}{2}(\mu_k+\mu_K)^T\Sigma^{-1}(\mu_k-\mu_K) + x^T\Sigma^{-1}(\mu_k-\mu_K) = \alpha_{k0}+\alpha_k^T x$$

i w przypadku regresji logistycznej:

$$\log \frac{P(G=k|X=x)}{P(G=K|X=x)} = \beta_{k0} + \beta_k^T x.$$

Regresja logistyczna czy LDA?

Iloraz prawdopodobieństwa posteriori między klasami k i K w przypadku LDA:

$$\log \frac{P(G=k|X=x)}{P(G=K|X=x)} = \log \frac{\pi_k}{\pi_K} - \frac{1}{2}(\mu_k+\mu_K)^T\Sigma^{-1}(\mu_k-\mu_K)+x^T\Sigma^{-1}(\mu_k-\mu_K) = \alpha_{k0}+\alpha_k^T x$$

i w przypadku regresji logistycznej:

$$\log \frac{P(G=k|X=x)}{P(G=K|X=x)} = \beta_{k0} + \beta_k^T x.$$

Gęstość łączną X i G możemy zapisać jako:

$$P(X, G=k) = P(X)P(G=k|X),$$

gdzie dla LDA i regresji logistycznej:

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

Regresja logistyczna czy LDA?

Regresja liniowa maksymalizuje warunkową log-wiarogogdność, natomiast LDA maksymalizując pełne prawdopodobieństwo logarytmiczne.

Regresja logistyczna czy LDA?

Regresja liniowa maksymalizuje warunkową log-wiarogogdność, natomiast LDA maksymalizując pełne prawdopodobieństwo logarytmiczne.

Gęstość łączna X, G w przypadku LDA:

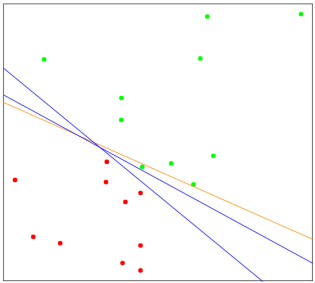
$$P(X, G = k) = \phi(X; \mu_k, \Sigma)\pi_k,$$

gdzie ϕ oznacza gęstość rozkładu normalnego. Wówczas gęstość brzegowa X jest postaci:

$$P(X) = \sum_{k=1}^{K} \pi_k \phi(X; \mu_k, \Sigma).$$

Rozdzielanie hiperpłaszczyzną - przykład

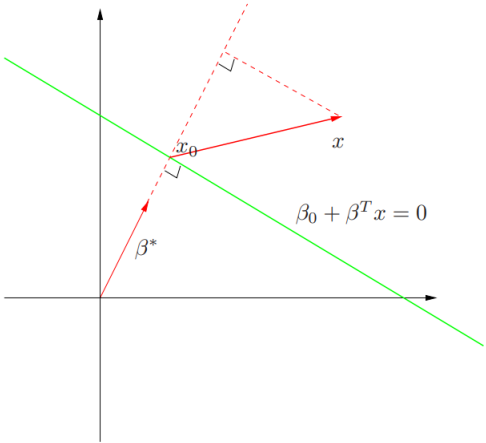
Zbiór danych składa się z 20 punktów na płaszczyźnie.



Pomarańczowa linia została wyznaczona metodą najmniejszych kwadratów, a punkty na niej leżące opisuje zbiór:

$$\{x : \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 = 0\}.$$

Kilka faktów z algebry



Kilka faktów z algebry

Niech L definiuje równanie afiniczny $f(x) = \beta_0 + \beta^T x = 0$, wówczas:

- Dla dowolnych dwóch punktów $x_1, x_2 \in L$ zachodzi $\beta^T(x_1 - x_2) = 0$, zatem $\beta^* = \beta/\|\beta\|$ jest wektorem normalnym do powierzchni L.

Kilka faktów z algebry

Niech L definiuje równanie afiniczny $f(x) = \beta_0 + \beta^T x = 0$, wówczas:

- Dla dowolnych dwóch punktów $x_1, x_2 \in L$ zachodzi $\beta^T(x_1 - x_2) = 0$, zatem $\beta^* = \beta/\|\beta\|$ jest wektorem normalnym do powierzchni L.
- Dla dowolnego punktu $x_0 \in L$ zachodzi $\beta^T x_0 = -\beta_0$.

Kilka faktów z algebry

Niech L definiuje równanie afiniczny $f(x) = \beta_0 + \beta^T x = 0$, wówczas:

- Dla dowolnych dwóch punktów $x_1, x_2 \in L$ zachodzi $\beta^T(x_1 - x_2) = 0$, zatem $\beta^* = \beta/\|\beta\|$ jest wektorem normalnym do powierzchni L.
- Dla dowolnego punktu $x_0 \in L$ zachodzi $\beta^T x_0 = -\beta_0$.
- Odległość dowolnego punktu x od L zadana jest wzorem:

$$\beta^{*T}(x - x_0) = \frac{1}{\|\beta\|}(\beta^T x + \beta_0) = \frac{1}{\|f'(x)\|} f(x).$$

Algorytm uczenia perceptronowego

Jeżeli $y_i = 1$ jest źle sklasyfikowane, to $x_i^T \beta + \beta_0 < 0$ i odwrotnie, jeżeli $y_i = -1$ zostanie źle sklasyfikowane, to $x_i^T \beta + \beta_0 > 0$.

Algorytm uczenia perceptronowego

Jeżeli $y_i = 1$ jest źle sklasyfikowane, to $x_i^T \beta + \beta_0 < 0$ i odwrotnie, jeżeli $y_i = -1$ zostanie źle sklasyfikowane, to $x_i^T \beta + \beta_0 > 0$.

Cel: Minimalizacja wyrażenia:

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

gdzie $\mathcal{M}$ - zbiór indeksów źle sklasyfikowanych punktów.

Algorytm uczenia perceptronowego

Jeżeli $y_i = 1$ jest źle sklasyfikowane, to $x_i^T \beta + \beta_0 < 0$ i odwrotnie, jeżeli $y_i = -1$ zostanie źle sklasyfikowane, to $x_i^T \beta + \beta_0 > 0$.

Cel: Minimalizacja wyrażenia:

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

gdzie $\mathcal{M}$ - zbiór indeksów źle sklasyfikowanych punktów.

$D(\beta, \beta_0) \geqslant 0$ i proporcjonalne do odległości błędnie sklasyfikowanych punktów od granicy $\beta^T x + \beta_0 = 0$.

Algorytm uczenia perceptronowego

Jeżeli $y_i = 1$ jest źle sklasyfikowane, to $x_i^T \beta + \beta_0 < 0$ i odwrotnie, jeżeli $y_i = -1$ zostanie źle sklasyfikowane, to $x_i^T \beta + \beta_0 > 0$.

Cel: Minimalizacja wyrażenia:

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

gdzie $\mathcal{M}$ - zbiór indeksów źle sklasyfikowanych punktów.

$D(\beta, \beta_0) \geqslant 0$ i proporcjonalne do odległości błędnie sklasyfikowanych punktów od granicy $\beta^T x + \beta_0 = 0$.

Załóżmy, że $\mathcal{M}$ jest ustalone, wtedy:

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

Algorytm uczenia perceptronowego

Pomysł: Użycie stochastic gradient descent, wtedy β jest aktualizowane poprzez podstawienie:

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

gdzie ρ oznacza współczynnik uczenia.

Algorytm uczenia perceptronowego

Pomysł: Użycie stochastic gradient descent, wtedy β jest aktualizowane poprzez podstawienie:

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

gdzie ρ oznacza współczynnik uczenia.

Problemy algorytmu:

- Jeśli dane są rozłączne, to istnieje wiele rozwiązań.

Algorytm uczenia perceptronowego

Pomysł: Użycie stochastic gradient descent, wtedy β jest aktualizowane poprzez podstawienie:

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

gdzie ρ oznacza współczynnik uczenia.

Problemy algorytmu:

- Jeśli dane są rozłączne, to istnieje wiele rozwiązań.
- Skończona liczba kroków może być bardzo duża.

Algorytm uczenia perceptronowego

Pomysł: Użycie stochastic gradient descent, wtedy β jest aktualizowane poprzez podstawienie:

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

gdzie ρ oznacza współczynnik uczenia.

Problemy algorytmu:

- Jeśli dane są rozłączne, to istnieje wiele rozwiązań.
- Skończona liczba kroków może być bardzo duża.
- Jeśli dane nie są rozłączne, algorytm nie będzie zbieżny i powstają cykle.

Algorytm uczenia perceptronowego

Pomysł: Użycie stochastic gradient descent, wtedy β jest aktualizowane poprzez podstawienie:

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

gdzie ρ oznacza współczynnik uczenia.

Problemy algorytmu:

- Jeśli dane są rozłączne, to istnieje wiele rozwiązań.
- Skończona liczba kroków może być bardzo duża.
- Jeśli dane nie są rozłączne, algorytm nie będzie zbieżny, i powstają cykle.

Rozwiązania:

- Dodanie dodatkowych ograniczeń do hiperpłaszczyzny rozdzielającej.

Algorytm uczenia perceptronowego

Pomysł: Użycie stochastic gradient descent, wtedy β jest aktualizowane poprzez podstawienie:

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

gdzie ρ oznacza współczynnik uczenia.

Problemy algorytmu:

- Jeśli dane są rozłączne, to istnieje wiele rozwiązań.
- Skończona liczba kroków może być bardzo duża.
- Jeśli dane nie są rozłączne, algorytm nie będzie zbieżny, i powstają cykle.

Rozwiązania:

- Dodanie dodatkowych ograniczeń do hiperpłaszczyzny rozdzielającej.
- Poszukiwanie hiperpłaszczyzny w znacznie powiększonej przestrzeni.

Dziękuję za uwagę.