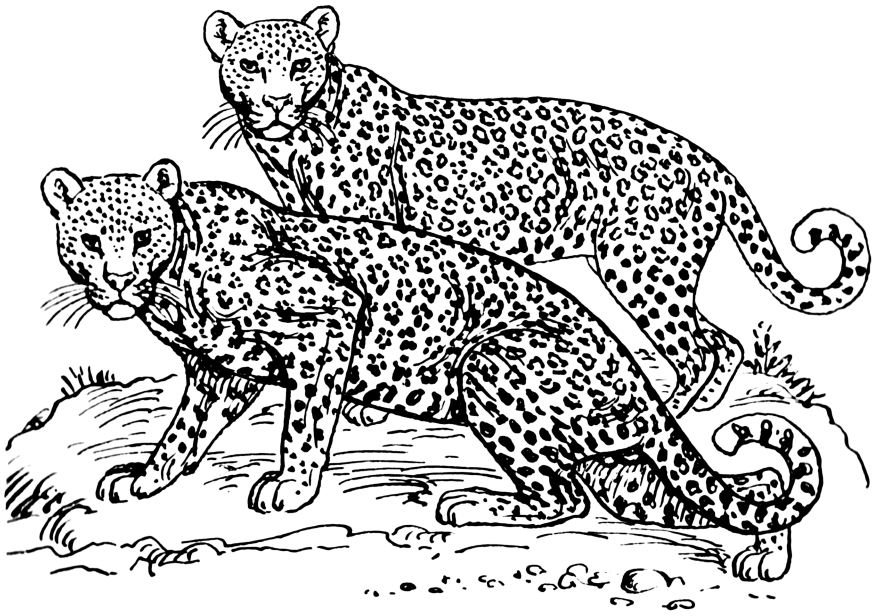


Metody Probabilistyczne w Uczeniu Maszynowym

Zagadnienia Egzaminacyjne

„Gdzie błąd?”

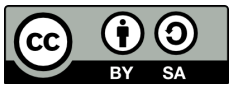
Popełnione przez

Załatany Ponton

Kraków
Anno Domini 2023

Spis treści

Licencja



Rozdział 1

Metody estymacji parametrów

1.1 Definicje

1.1.1 Modele parametryczne

Definicja 1.1.1. Model parametryczny z parametrami $\theta \in \Theta$ to dowolny zbiór rozkładów prawdopodobieństwa $\{p(y \mid \theta) : \theta \in \Theta\}$

Przykładowe modele parametryczne to:

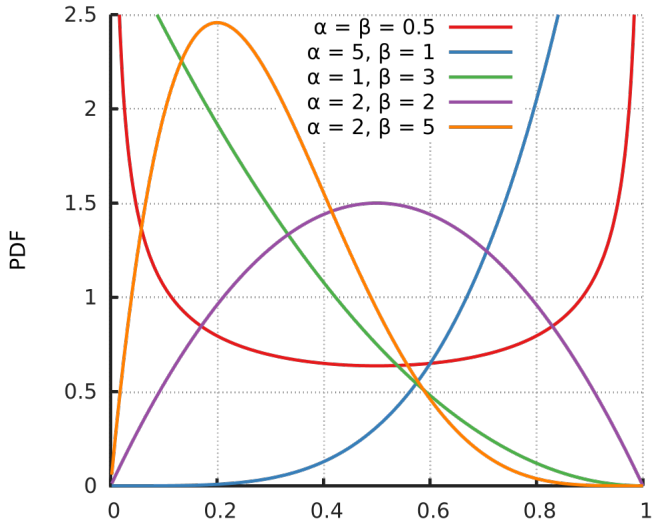
- Bern(p) – pojedyncza próba z sukcesem p
- Binom(n, p) – rozkład dwumianowy, n prób z sukcesem p
- Uni(a, b) – rozkład jednostajny na przedziale (a, b)
- Poi(λ) – rozkład Poissona z parametrem λ
- $\mathcal{N}(\mu, \sigma^2)$ – rozkład normalny
- Beta$(\alpha, \beta$ – rozkład beta

1.1.2 Rozkład Beta

Nowością w stosunku do MPI jest tzw. rozkład beta Beta(α, β) Jego funkcja gęstości wynosi

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot x^{\alpha-1} \cdot (1-x)^{\beta-1}$$

Na rysunku przedstawia on się następująco:



Rysunek 1.1: Gęstość rozkładu beta dla różnych parametrów

Na ćwiczeniach pokazywaliśmy fakt, który warto znać, a mianowicie

$$\mathbb{E}[\text{Beta}(\alpha, \beta)] = \frac{\alpha}{\alpha+\beta}$$

Ponadto mamy jeszcze jedną własność, którą można zauważyć na rysunku

- dla $\alpha, \beta < 1$ rozkład jest bimodalny
- dla $\alpha, \beta \geq 1$ rozkład jest unimodalny
- dla $\alpha = \beta$ rozkład jest symetryczny względem $x = \frac{1}{2}$

1.1.3 Rozkłady sprzężone

Aby nam się łatwiej liczyło wprowadzamy coś takiego jak **rozkład sprzężony**, czyli taki rozkład aprioryczny dla rozkładu próby, dla którego rozkład a posteriori i rozkład a priori jest taki sam.

rozkład próby	a priori	a posteriori
dwumianowy	beta	beta
Poissona	gamma	gamma
geometryczny	beta	beta
wielomianowy	Dirichleta	Dirichleta
wykładniczy	gamma	gamma
normalny	normalny	normalny

Jest to głównie potrzebne po to aby nam się prościej liczyło.

1.1.4 Estymator

Definicja 1.1.2. Estymator to dowolna wartość szacowanego parametru obliczona na podstawie jakiejś próbki. Dla parametru θ będziemy oznaczać estymator przez $\widehat{\theta}$

Definicja 1.1.3. Obciążenie estymatora $\widehat{\theta}$ parametru θ definiujemy jako

$$\text{bias}(\widehat{\theta}) = \mathbb{E}\left[\widehat{\theta}\right] - \theta$$

Mówimy, że estymator jest **obciążony** jeśli $\text{bias}(\theta) \neq 0$

1.1.5 Twierdzenie Bayesa

Ustalmy dowolny model parametryczny. Niech θ będzie zmienną opisującą parametr modelu. Niech D będzie zmienną losową, która opisuje obserwacje zwracane przez ten model. Zachodzi wtedy wzór Bayesa na prawdopodobieństwo warunkowe:

$$\mathrm{P}(\theta = t \mid D = d) = \frac{P(D = d \mid \theta = t) \cdot P(\theta = t)}{P(D = d)}$$

Dzięki temu jesteśmy w stanie na podstawie obserwacji szacować parametr naszego modelu.

Podobnie jeśli θ i D są zmiennymi ciągłymi to

$$f_{\theta \mid D=d} = \frac{f_{D \mid \theta=t}(d) \cdot f_\theta(t)}{f_D(d)}$$

Nadużywając notacji będziemy zapisywać obie te sytuacje jednocześnie, przyjmując przy tym że $p(X)$ jest rozkładem zmiennej X

$$p(\theta \mid D) = \frac{p(D \mid \theta) \cdot p(\theta)}{p(D)}$$

Nazywamy elementy tego wzoru:

- $p(\theta)$ to **prawdopodobieństwo a priori**, czyli co wiemy (zakładamy) o θ jeśli nie mamy żadnych obserwacji
- $p(D \mid \theta)$ to **wiarygodność**, czyli jak dobrze model z parametrem θ przewiduje dane z obserwacji
- $p(\theta \mid D)$ to **prawdopodobieństwo a posteriori**, czyli co powinniśmy zakładać o θ jeśli poczyniliśmy już obserwacje
- $p(D)$ to **wiarygodność brzegowa**, czyli jakie w ogólności jest prawdopodobieństwo uzyskania naszych obserwacji

$$p(D) = \int_{\text{im}\,\theta} p(D \mid \theta) p(\theta) d\theta$$

Warto tutaj zauważyć, że $p(D)$ jest niezależne od θ tj. zależy jedynie od samego modelu, a nie od konkretnego parametru.

1.2 Metoda największej wiarygodności

Zwana także MLE (maximum likelihood estimation).

Szukamy estymatora, który maksymalizuje wiarygodność, czyli $p(D \mid \theta)$

$$\widehat{\theta}_{MLE} = \arg\max_{\theta} p(D \mid \theta) = \arg\max_{\theta}(\log p(D \mid \theta))$$

Zamiast logarytmu możemy wstawić dowolną inną monotoniczną funkcję (możemy też nie brać logarytmu), ważne żeby się łatwo liczyło.

1.3 Metoda maksymalnego a posteriori

Zwana także MAP (maximum a posteriori).

Tym razem maksymalizujemy $p(\theta \mid D)$

$$\widehat{\theta}_{MAP} = \arg\max_{\theta} p(\theta \mid D) = \arg\max_{\theta} \frac{p(D \mid \theta)p(\theta)}{p(D)} = \arg\max_{\theta}(p(D \mid \theta)p(\theta))$$

1.4 Wnioskownie bayesowskie

Idea jest podobna jak w poprzednich dwóch przypadkach, tyle że explicite obliczamy rozkład $p(\theta \mid D)$.

Rozdział 2

Podstawy teorii decyzji

2.1 Definicje

Definiujemy trzy przestrzenie, którymi będziemy się zajmować:

- X – wejścia
- A – akcje, czyli nasze przewidywania
- Y – oczekiwane przewidywania

Definicja 2.1.1. Funkcja decyzyjna to dowolna funkcja $f : X \to A$

Definicja 2.1.2. Funkcja straty (kosztu) to dowolna funkcja $\ell : A \times Y \to \mathbb{R}$

Będziemy ustalali funkcję straty ℓ, która opisuje jak dobry jest wynik i starali się dopasować funkcję f, która podejmuje dobre decyzje.

Definicja 2.1.3. Ryzyko funkcji decyzyjnej f przy stracie ℓ definiujemy jako

$$R(f) = \mathbb{E}[\ell(f(X), Y)]$$

gdzie X, Y to zmienne losowe opisujące dane.

Definicja 2.1.4. Bayesowska funkcja decyzyjna to

$$f^* = \arg\min_{f:X\to A} R(f)$$

Ponieważ w praktyce rzadko znamy rozkłady X, Y to definiujemy też ryzyko empiryczne, które odnosi się bezpośrednio do jakiegoś zbioru danych.

Definicja 2.1.5. Ryzyko empiryczne funkcji decyzyjnej f na danych $D = \left\{\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)\right\}$ definiujemy jako

$$\widehat{R}_m(f) = \frac{1}{m}\sum_{i=1}^{m} \ell\left(f(x^{(i)}), y^{(i)}\right)$$

Z silnego prawa wielkich liczb wiemy, że prawie na pewno mamy zbieżność $\widehat{R}_m \to R_f$

Aby znaleźć dobrą funkcję decyzyjną f mając jakieś dane D będziemy minimalizować ryzyko empiryczne, czyli średnią stratę f na danych.

Ponieważ potencjalnych funkcji decyzyjnych może być dużo i nie wszystkie są dla nas sensowne to wprowadzamy pojęcie przestrzeni hipotez.

Definicja 2.1.6. Przestrzeń hipotez to dowolny zbiór funkcji decyzyjnych.

Dla zadanego zbioru hipotez H będziemy szukać

$$\widehat{f}_H = \arg\min_{f \in H} \frac{1}{m} \sum_{i=1}^{m} \ell\big(f(x^{(i)}), y^{(i)}\big)$$

Rozdział 3

Regresja liniowa

3.1 Definicje

Definicja 3.1.1. Mówimy, że model (przewidujący wyjścia) jest **liniowy** jeśli jego wyjścia można opisać funkcją

$$f_\theta(x) = \theta_0 + \sum_{i=1}^{k} \theta_i \cdot \phi_i(x_i)$$

gdzie funkcje φ_i są dowolne i nazywamy je **funkcjami bazowymi**.

Zauważmy, że model jest liniowy pod względem parametru θ a nie względem x. Używając odpowiednich funkcji bazowych możemy opisywać funkcje f które nie są funkcjami liniowymi.

3.2 Metoda najmniejszych kwadratów

Bierzemy model bez funkcji bazowych tj.

$$h_\theta(x) = \theta_0 + \sum_{i=1}^{k} \theta_i x_i$$

oraz kwadratową funkcję straty

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Tworzymy **macierz planowania** X

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_k^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_k^{(m)} \end{bmatrix}$$

oraz **wektor zmiennej objaśnianej** y

$$y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Możemy teraz elegancko zapisać

$$J(\theta) = \frac{1}{2}(X\theta - y)^T(X\theta - y)$$

Minimalizację robimy albo stosując spadek wzdłuż gradientu albo obliczając analitycznie

$$\theta = (X^T X)^{-1} X^T y$$

3.3 Funkcje bazowe

Możemy obłożyć wejścia jakąś funkcją ϕ – w ten sposób przeniesiemy je do innej przestrzeni w której być może prościej się robi regresję.

Dla modelu

$$h_\theta(x) = \theta^T \phi(x) + \varepsilon$$

gdzie ε jest szumem gaussowskim

tworzymy macierz planowania:

$$\Phi = \begin{bmatrix} \phi_1(x_1^{(1)}) & \dots & \phi_k(x_k^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi_1(x_1^{(m)}) & \dots & \phi_k(x_k^{(m)}) \end{bmatrix}$$

i dalej liczymy tak jak normalnie tj. dostajemy

$$\widehat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T y$$

3.4 Bayesowska regresja liniowa

W normalnej regresji liniowej staraliśmy się pokonać szum i dopasować jedną funkcję. Teraz będziemy chcieli znaleźć rozkład prawdopodobieństwa szukanych funkcji co nam daje dużo więcej informacji.

Mamy:

$$p(\theta) = \mathcal{N}(\theta \mid \mu_0, S_0)$$

oraz

$$p(Y \mid X, \theta) = \prod_{i=1}^{m} \mathcal{N}(y^{(i)}, \theta^T \phi(x^{(i)}), \sigma^2)$$

Rozkład a posteriori zadany jest wzorem:

$$p(\theta \mid Y, X) = \frac{p(Y, X \mid \theta) p(\theta)}{p(X, Y)}$$

Mianownik jest stałą, więc $p(\theta \mid Y, X)$ jest iloczynem wielu rozkładów normalnych zatem

$$p(\theta \mid Y, X) = \mathcal{N}(\theta \mid \mu_m, S_m)$$

3.4.1 Rozkład predykcyjny

Jeszcze lepiej będzie jak zamiast przewidywać sam parametr θ będziemy umieli obliczać rozkład prawdopodobieństwa nowej danej $x^{(m+1)}, y^{(m+1)}$.

Rozdział 4

Regularyzacja

W regresji staramy się znaleźć hipotezę h, która osiąga jak najmniejszy błąd na danych treningowych. Co jeśli dochodzi jednak do przeuczenia? Jednym z powodów może być nadmierna złożoność hipotezy, która jest na tyle „elastyczna"/„złożona" że praktycznie zapamiętuje cały zbiór danych. Będziemy próbowali jakoś odsiać takie hipotezy, które są zbyt złożone.

Zaczynamy w tym celu od wprowadzenia miary złożoności.

Definicja 4.0.1. Miara złożoności dla przestrzeni hipotez H to dowolna funkcja

$$\Omega : H \to [0, +\infty)$$

Miarą złożoności może być:

- liczba rozważanych cech
- stopień dopasowanego wielomianu
- głębokość drzewa
- $\ell_0(\theta) = |\{i : \theta_i \neq 0\}|$
- $\ell_1(\theta) = \sum_{i=1}^{k} |\theta_i|$
- $\ell_2(\theta) = \sum_{i=1}^{k} \theta_i^2$

Mamy dwa rodzaje regularyzacji:

- Regularyzacja Tichonowa – dodajemy do funkcji straty koszt za złożoność hipotezy:

$$l(h) = \ell(h) + \lambda\Omega(h)$$

- Regularyzacja Iwanowa – zawężamy się do tych hipotez, które są mało złożone

$$H' = \{h \in H : \Omega(h) \leq \omega\}$$

Okazuje się, że te dwie definicje są sobie równoważne tj. jeśli mamy optimum h^*_λ dla regularyzacji Tichonowa to jest ono również rozwiązaniem optymalnym dla regularyzacji Iwanowa dla jakiegoś ω.

Analogicznie h^*_ω jest optymalne dla Tichonowa przy pewnym λ.

Jeśli używamy funkcji ℓ_1 to mamy do czynienia z **regresją lasso**, a jeśli używamy ℓ_2 to z **regresją grzbietową**.

Rozdział 5

Klasyfikacja

5.1 Rodzaje klasyfikacji

5.1.1 Klasyfikacja generatywna

W klasyfikacji generatywnej staramy się dla każdej klasy określić jej rozkład. Kiedy dostajemy nowe wejście sprawdzamy z jakim prawdopodobieństwem należy do której z klas i na tej podstawie przydzielamy mu etykietę.

Co ciekawe takie podejście da się stosować w uczeniu bez nadzoru.

Przykładowe klasyfikatory:

- Naiwny Bayes
- Sieci Bayesowskie
- Pola Markowa
- GANy
- gaussowska analiza dyskryminacyjna

5.1.2 Klasyfikacja dyskryminatywna

W klasyfikacji dyskryminatywnej z kolei staramy się rozgraniczyć dane od siebie – nie interesuje nas dokładnie w jaki sposób powstają dane. Decyzję dla nowego wejścia podejmujemy sprawdzając po której stronie granicy się znajduje.

Na ogół to podejście sprawdza się w uczeniu nadzorowanym – skądś musimy mieć dane treningowe na podstawie których tworzymy granice, a danych dobrze jak jest sensownie dużo.

Przykładowe klasyfikatory:

- Regresja logistyczna
- SVM
- Sieci neuronowe
- KNN
- drzewa decyzyjne

5.2 Metody oceny klasyfikatora

Załóżmy że mamy klasyfikator binarny. Jak ocenić czy dobrze sobie radzi?

Dla danego wejścia mamy cztery możliwości jakości odpowiedzi:

- True Positive – poprawnie odpowiedzieliśmy „TAK"
- True Positive – poprawnie odpowiedzieliśmy „NIE"
- False Positive – błędnie odpowiedzieliśmy „TAK"
- False Negative – błędnie odpowiedzieliśmy „NIE"

Mamy trzy miary jakości:

- dokładność – jak wiele danych jest poprawnie zaklasyfikowanych

$$\frac{TP + TN}{TP + TN + FP + FN}$$

- precyzja – jak wiele danych, które uznaliśmy za pozytywne faktycznie jest pozytywnych. Wysoka precyzja to niski poziom fałszywych alarmów.

$$\frac{TP}{TP + FP}$$

- czułość – jak dużo faktycznie pozytywnych danych uznaliśmy za pozytywne. Wysoka czułość to wysoki poziom detekcji pozytywnych przypadków.

$$\frac{TP}{TP + FN}$$

Chcielibyśmy osiągnąć zarówno wysoką precyzję jak i wysoką czułość – np. jeśli wykrywamy obecność wirusa to z jednej strony chcemy przegapiać jak najmniej zarażeń (czułość), ale nie chcemy też wszystkich kierować na kwarantannę (precyzja).

5.2.1 Miara F-Beta

Definicja 5.2.1. Miarę F_β definiujemy jako

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precyzja} \cdot \text{czułość}}{\beta^2 \cdot \text{precyzja} + \text{czułość}}$$

Szczególnym przypadkiem jest miara F_1

$$F_1 = 2 \cdot \frac{\text{precyzja} \cdot \text{czułość}}{\text{precyzja} + \text{czułość}}$$

$\beta > 1$ premiuje czułość, natomiast $\beta < 1$ premiuje precyzję.

5.2.2 Krzywa ROC

Krzywa ROC to wykres, który dla różnych modeli ilustruje zależność czułości czyli poziomu wyników prawdziwie pozytywnych $\frac{TP}{TP+FN}$ od poziomu wyników fałszywie pozytywnych $\frac{FP}{FP+TN}$

Oś $y = x$ odpowiada losowej klasyfikacji, obszar nad nią oznacza dobry klasyfikator, obszar pod nią oznacza zły klasyfikator.

Idealny klasyfikator znajduje się w punkcie (0, 1) – nie ma w ogóle wyników fałszywie pozytywnych a wszystkie prawdziwe dane są poprawnie zaklasyfikowane.

5.3 Funkcje straty w klasyfikacji

Przyjmijmy że mamy dwie klasy tj. $Y = \{-1, 1\}$

Moglibyśmy od razu jako odpowiedź dawać $\widehat{y} \in \{-1, 1\}$ wtedy błąd zadany jest takim wzorem:

$$\ell(f(x), y) = \mathbf{1}[f(x) \neq q]$$

Możemy jednak być nieco sprytniejsi i przewidywać $\widehat{y} \in \mathbb{R}$ (im większe na moduł tym silniejsze przekonanie) i dopiero na podstawie znaku odpowiadać 1 lub -1.

Wtedy szczególności ryzyko empiryczne wyraża się:

$$\widehat{R}_m(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\left[y^{(i)} \neq h(x^{(i)} \leq 0\right]$$

Co jest trochę słabe, bo nie jest to ani ciągłe ani różniczkowalne – słabo się z nich uczy.

Definicja 5.3.1. Marginesem dla przewidywania $\widehat{y} \in \mathbb{R}$ i prawdziwej odpowiedzi y nazywamy $\widehat{y} \cdot y$

Na podstawie marginesu m możemy zdefiniować różne funkcje straty:

- strata zawiasowa
$$\max(0, 1 - m)$$
- strata Savage'a
$$\frac{1}{(1 + e^m)^2}$$
- strata logistyczna
$$\ln(1 + e^{-m})$$
- strata wykładnicza
$$e^{-m}$$
- strata kwadratowa
$$(1 - m)^2$$

5.4 Gaussowska analiza dyskryminacyjna

Mamy dwie klasy $Y = \{0, 1\}$

Zakładamy, że klasę 1 wybieramy z prawdopodobieństwem ϕ oraz że w obrębie każdej klasy dane są generowane rozkładem normalnym, z tą samą macierzą kowariancji tj.

$$p(x \mid y = i) \sim \mathcal{N}(\mu_i, \Sigma)$$

gdzie

$$\mathcal{N}(x \mid \mu_i, \Sigma) = \frac{1}{(2\pi)^{k/2}\sqrt{|\Sigma|}} \exp\left(-\frac{(x-\mu_i)^T\Sigma^{-1}(x-\mu_i)}{2}\right)$$

Mając dame etykiety będziemy chcieli znaleźć parametry $\theta = (\phi, \mu_0, \mu_1, \Sigma)$.

Maksymalizujemy log-wiarygodność

$$\begin{aligned}
\ell(\theta) &= \ln \prod_{i=1}^{m} p(x^{(i)}, y^{(i)} \mid \theta) \\
&= \ln \prod_{i=1}^{m} p(x^{(i)} \mid y^{(i)}, \theta) p(y^{(i)} \mid \theta) \\
&= \sum_{i=1}^{m} \ln p(x^{(i)} \mid y^{(i)}, \theta) + \sum_{i=1}^{m} \ln p(y^{(i)} \mid \theta) \\
&= \sum_{i=1}^{m} \ln \mathcal{N}(x^{(i)} \mid \mu_{y^{(i)}}, \Sigma) + \sum_{i=1}^{m} \ln\left(\phi^{y^{(i)}}(1-\phi)^{1-y^{(i)}}\right)
\end{aligned}$$

Założenie, że oba rozkłady mają tę samą macierz kowariancji jest istotne – dzięki temu dostajemy

$$p(y = 1 \mid x) = \frac{1}{1 + \exp(-\eta^T x)}$$

gdzie η jest stałą wyznaczoną na podstawie parametrów modelu.

Innymi słowy – mamy tutaj regresję logistyczną, jeśli macierze kowariancji byłyby różne to już nie byłoby tak wesoło.

Rozdział 6

Naiwny klasyfikator bayesowski

Rozdział 7

Modele graficzne

7.1 Warunkowa niezależność cech

Rozważmy $x = [x_1, \ldots, x_k]$ oraz daną wyjściową y.

Mamy

$$\begin{aligned} p(x \mid y) &= p(x_1, \ldots, x_k \mid y) \\ &= p(x_1 \mid y) \cdot p(x_2 \mid y, x_1) \ldots p(x_k \mid y, x_1, \ldots x_{k-1}) \end{aligned}$$

Jeśli wszystkie cechy są binarne to aby opisać zachowanie takiego modelu potrzebujemy mieć aż $1 + 2 + 4 + \cdots + 2^{k-1} = 2^k - 1$ parametrów – to dużo.

Jeśli założymy, że wszystkie te cechy są warunkowo niezależne to dostajemy

$$p(x \mid y) = \prod_{i=1}^{k} p(x_i \mid y)$$

I tutaj wystarczy nam k parametrów.

7.1.1 Przykład

Załóżmy że mieszkamy w jakimś obszarze w którym mają miejsce trzęsienia ziemi i włamania. Dostaliśmy informację, że w domu włączył się alarm więc jedziemy do domu sprawdzić co się stało. Podczas drogi powrotnej radiu powiedzieli, że miało miejsce lekkie trzęsienie ziemi.

Zastanawiamy się teraz co faktycznie jest przyczyną alarmu i czy lepiej jest jechać się upewnić czy może wracać do roboty.

Mamy więc cztery binarne **zmienne losowe** (nie zdarzenia):

1. E – miało miejsce trzęsienie ziemi
2. R – radio podało informację o trzęsieniu
3. B – ktoś się do nas włamał
4. A – włączył się alarm

Możemy rozpisać

$$p(E, R, B, A) = p(E) \cdot p(B \mid E) \cdot p(R \mid E, B) \cdot p(A \mid R, E, B)$$

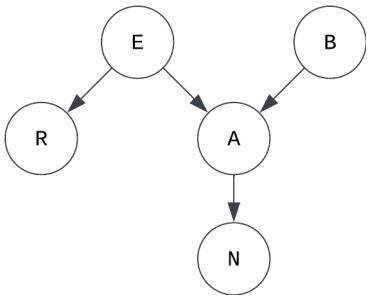
Mamy tutaj 15 parametrów, które musimy znać żeby znać pełen rozkład.

Jak się jednak dobrze zastanowimy to:

- $p(A \mid R, E, B) = p(A \mid E, B)$ – to czy u nas działa alarm jest raczej niezależne od radia jeśli wiemy czy się do nas włamali i czy miało miejsce trzęsienie ziemii.
- $p(R \mid E, B) = p(R \mid E)$ – radio nie jest zainteresowane naszym domem.
- $p(B \mid E) = p(B)$ – zakładamy że trzęsienie ziemi nie ma wpływu na włamania.

Powiedzmy że mamy jeszcze piątą zmienną N, która mówi czy sąsiad do nas zadzwonił poinformować nas o alarmie.

Możemy zobrazować zależności za pomocą diagramu:



Rysunek 7.1: Model graficzny dla przykładu z trzęsieniem ziemi

7.2 Model graficzny

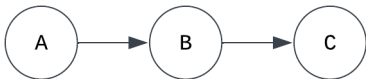
Model graficzny służy do wyrażenia zależności między zmiennymi losowymi za pomocą DAGu.

Bezpośrednie zależności między dwoma zmiennymi modelujemy jako krawędź skierowana od przyczyny do skutku.

Będziemy oznaczać $X \perp Y$ jeśli te dwie zmienne są **niezależne**.

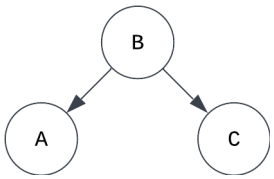
Jest kilka ciekawych struktur, które mogą wystąpić:

- mediator



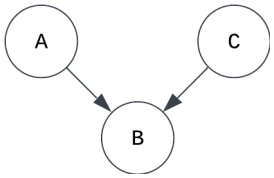
Rysunek 7.2: $A \not\perp C$ ale $A \perp C \mid B$

- confounder



Rysunek 7.3: $A \not\perp C$ ale $A \perp C \mid B$

- collider

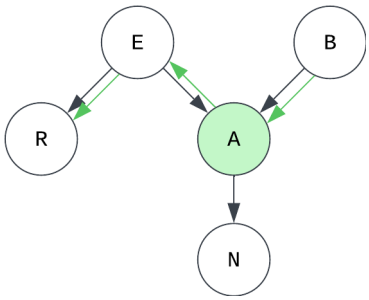


Rysunek 7.4: $A \perp C$ ale $A \not\perp C \mid B$

Aby sprawdzić czy dane dwie zmienne są od siebie zależne szukamy między nimi ścieżki zależności. Mediator i Confounder zamykają ścieżkę, natomiast Collider ją otwiera w momencie gdy warunkujemy się po zmiennej w danym węźle.

W naszym przykładzie z trzęsieniem ziemii mamy zatem $E \perp B$, $A \not\perp R$, $B \perp R$ – intuicyjne ma to sens, bo te rzeczy są od siebie naturalnie niezależnie.

Jeśli jednak wiemy czy alarm dzwoni czy nie to sytuacja się zmienia i mamy $E \not\perp B, B \not\perp R$. Jeśli wiemy że dzwoni alarm to sytuacja że w radiu nie ma informacji o trzęsieniu a do nas się nie włamują jest teraz bardzo mało prawdopodobna.



Rysunek 7.5: Ścieżka zależności $B \not\perp R \mid A$

Rozdział 8

Klasyfikatory liniowe

8.1 Regresja logistyczna

Regresja logistyczna korzysta z funkcji decyzyjnej zadanej wzorem

$$h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Traktujemy wyjście jako rozkład prawdopodobieństwa:

$$p(y = 1 \mid x, \theta) = h_\theta(x)$$

$$p(y = 0 \mid x, \theta) = 1 - h_\theta(x)$$

Możemy też zapisać sprytnie jako

$$p(y \mid x, \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

Chcemy zmaksymalizować log-wiarygodność

$$\begin{aligned}\ell(\theta) &= \log \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}, \theta) \\ &= \sum_{i=1}^{m} \big(y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\big)\end{aligned}$$

8.2 Uogólnione modele liniowe

8.2.1 Wykładnicza rodzina rozkładów

Definicja 8.2.1. Niech $P_\Theta = \{p(y \mid \theta) : \theta \in \Theta\}$ będzie parametryczną rodziną rozkładów.

Mówimy, że P_Θ jest **wykładniczą rodziną rozkładów** jeśli dla każdego $\theta \in \Theta$ istnieje wektor η (parametr naturalny) oraz funkcje $a(\eta), b(y), T(y)$ takie, że:

$$p(y \mid \eta) = b(y) \exp\big(\eta^T T(y) - a(\eta)\big)$$

(nie mylić z rodziną wykładniczych rozkładów, to nie o to chodzi).

8.2.1.1 Rozkład Bernoulliego

Mamy rozkład z jednym parametrem ϕ, $p(y \mid \phi) = \phi^y(1-\phi)^{1-y}$

Rozpisujemy

$$\begin{aligned}\phi^y(1-\phi)^{1-y} &= \exp(y\ln\phi + (1-y)\ln(1-\phi)) \\ &= \exp(y(\ln\phi - \ln(1-\phi)) + \ln(1-\phi)) \\ &= \exp\left(y\ln\frac{\phi}{\ln(1-\phi)} + \ln(1-\phi)\right)\end{aligned}$$

W efekcie czego dostajemy:

- $\eta = \ln\frac{\phi}{1-\phi}$
- $T(y) = y$
- $a(\eta) = -\ln(1-\phi) = \ln(1+e^\eta)$
- $b(y) = 1$

8.2.1.2 Rozkład normalny

8.2.2 Uogólnione modele liniowe

Chcemy przewidywać wartości y na podstawie obserwacji x z jakiejś przestrzeni $X \times Y$

Założenia modelu:

- Dla wejścia x oraz parametru θ zmienna $y|x,\theta$ pochodzi z wykładniczej rodziny rozkładów o parametrze η
- $\eta = \theta^T x$
- $h_\theta(x) = \mathbb{E}[T(y) \mid x, \theta]$
- $g(\eta) = \mathbb{E}[T(y) \mid \eta]$ to **kanoniczna funkcja odpowiedzi**
- $g^{-1}(\eta)$ to **kanoniczna funkcja łącząca**

8.2.2.1 Regresja liniowa

W regresji liniowej z szumem gaussowskim mamy $p(y \mid x) \sim \mathcal{N}(\mu, \sigma^2)$

Niech σ^2 będzie ustalone. Wtedy $\{\mathcal{N}(\mu,\sigma^2) : \mu \in \mathbb{R}\}$ jest rodziną wykładniczą.

Kładziemy $\eta = \mu, T(y) = y, \mathbb{E}[T(y) \mid x, \theta] = \mu$, $h_\theta(x) = \mu = \eta = \theta^T x$

8.2.2.2 Regresja logistyczna

W regresji logistycznej mamy z kolei $p(y \mid x) \sim \text{Bern}(\phi)$, co jest elegancką rodziną wykładniczą.

Mamy $\eta = \ln\frac{\phi}{1-\phi}$ zatem $\phi = \frac{1}{1+\exp(-\eta)}$

Ponadto, ponieważ jest to indykator to mamy $\mathbb{E}[y \mid x, \theta] = \phi$ czyli mamy

$$h_\theta(x) = \phi = \frac{1}{1+\exp(-\theta^T x)}$$

8.2.2.3 Regresja wieloklasowa (softmax)

Załóżmy, że chcemy klasyfikować do więcej niż jednej klasy tj, $Y = \{1, \ldots, K\}$.

Mamy więc

$$p(y = i \mid x) = \phi_i = \prod_{j=1}^{K} \phi_j^{\mathbf{1}[y=j]}$$

przy czym

$$\sum_{i=1}^{K} \phi_i = 1$$

aby dostać wykładniczą rodzinę przekształcamy:

$$\begin{aligned} P(y \mid x) &= \exp\left(\sum_{j=1}^{K} \mathbf{1}[y = j] \ln \phi_j\right) \\ &= \exp\left(\ln \phi_K + \sum_{j=1}^{K} \mathbf{1}[y = j](\ln \phi_j - \ln \phi_K)\right) \\ &= \exp\left(\ln \phi_K + \sum_{j=1}^{K-1} \mathbf{1}[y = j] \ln \frac{\phi_j}{\phi_K}\right) \end{aligned}$$

Sprowadzamy do takiej a nie innej postaci, bo możemy, i ma ona fajne konsekwencje.

Dostajemy

- $\eta_i = \ln \frac{\phi_i}{\phi_K}$
- $a(\eta) = -\ln \phi_K$
- $b(y) = 1$
- $T(i)$ – wektor długości $K - 1$ z jedynką na i-tym miejscu, $T(K) = 0$

Zauważamy, że

$$\frac{\phi_i}{\phi_K} = \exp(\eta_i)$$

a ponadto

$$\frac{1}{\phi_K} = \sum_{i=1}^{K} \frac{\phi_i}{\phi_K} = \sum_{i=1}^{K} \exp(\eta_i)$$

z czego

$$\phi_i = \frac{\exp(\eta_i)}{\sum_{i=1}^{K} \exp(\eta_i)}$$

W takim razie

$$\begin{aligned} h_\theta(x) &= \mathbb{E}[T(y) \mid x, \theta] \\ &= [\phi_1, \ldots, \phi_{K-1}]^T \end{aligned}$$

Czyli

$$\begin{aligned}\theta_i &= \phi_i \\ &= \frac{\exp(\eta_i)}{\sum_{i=1}^{K}\exp(\eta_i)} \\ &= \frac{\exp(\theta_i^T x)}{\sum_{i=1}^{K}\exp(\theta_i^T x)}\end{aligned}$$

8.3 Perceptron

8.4 Maszyny wektorów nośnych

8.4.1 Margines

Chcemy jakoś wyrazić fakt, że nasza hiperpłaszczyzna $w^T x + b$ dobrze separuje dane. Wprowadzamy zatem pojęcie marginesu.

Definicja 8.4.1. Margines dla obserwacji $x^{(i)}, y^{(i)}$ oraz hiperpłaszczyzny $w^T x + b$ definiujemy jako

$$\widehat{\gamma}^{(i)} = y^{(i)} \cdot \left(w^T x^{(i)} + b\right)$$

Natomiast dla zbioru D

$$\widehat{\gamma} = \min_i \widehat{\gamma}^{(i)}$$

Ponieważ wartości w, b mogą być dowolnie duże i definiować tę samą hiperpłaszczyzny to chcielibyśmy aby były one znormalizowane. Definiujemy zatem

Definicja 8.4.2. Margines geometryczny definiujemy jako

$$\gamma^{(i)} = \frac{\widehat{\gamma}^{(i)}}{\|w\|}$$

$$\gamma = \frac{\widehat{\gamma}}{\|w\|}$$

Będziemy chcieli zmaksymalizować margines geometryczny czyli nasz problem jest postaci:

$$\begin{aligned}&\text{zmaksymalizować } \gamma \\ &\quad \text{pod warunkami } y^{(i)} \cdot \left(w^T x^{(i)} + b\right) \geq \gamma \\ &\qquad\qquad \|w\| = 1\end{aligned}$$

co jest równoważne

$$\begin{aligned}&\text{zminimalizować } \|w\|^2 \\ &\text{pod warunkami } y^{(i)} \cdot \left(w^T x^{(i)} + b\right) \geq 1\end{aligned}$$

Punkty $x^{(i)}$ które realizują równość nazywamy **wektorami wspierającymi** (support vectors).

Często dane nie będą idealnie liniowo separowalne i będziemy mieli obserwacje odstające. Dodajemy zatem czynnik regularyzacyjny.

$$\begin{aligned} &\text{zminimalizować } \|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \\ &\text{pod warunkami } y^{(i)}\cdot\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i \\ &\qquad\qquad \xi_i \geq 0 \end{aligned}$$

Co jest w bardzo fajny sposób równoważne minimalizacji zawiasowej funkcji straty z regularyzacją ℓ^2

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\max\left\{0, 1 - y^{(i)}\cdot\left(w^T x^{(i)} + b\right)\right\}$$

8.5 Funkcje jądrowe

W praktyce dane rzadko są liniowe, a SVM umie dopasować tylko hiperpłaszczyznę. Aby sobie z tym poradzić moglibyśmy wprowadzić funkcje bazowe ϕ tak jak miało to miejsce w przypadku regresji liniowej.

Nasz problem jest wtedy postaci

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\left(1 - y^{(i)}\cdot\left(w^T\phi(x^{(i)}) + b\right)\right)$$

co wiemy że jest równoważne maksymalizacji

$$\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j\cdot y^{(i)}y^{(j)}\cdot\phi(x^{(i)})^T\phi(x^{(j)})$$

pod warunkiem

$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$$

gdzie $\alpha_i \in \left[0, \frac{C}{m}\right]$

Zauważamy, że możemy stworzyć funkcję jądrową κ

$$\kappa(x^{(i)}, x^{(j)}) = \phi(x^{(i)})^T\phi(x^{(j)}) = \left\langle\phi(x^{(i)}), \phi(x^{(j)})\right\rangle$$

Przykładową funkcją jądrową, która jest fajna jest jądro gaussowskie:

$$\kappa(x, z) = \exp\left(\frac{-\|x - z\|^2}{2\tau^2}\right)$$

Definicja 8.5.1. Dla jądra κ oraz elementów $x^{(1)}, \ldots, x^{(m)}$ definiujemy **macierz Grama**

$$K = \begin{bmatrix} \kappa(x^{(1)}, x^{(1)}) & \ldots & \kappa(x^{(1)}, x^{(m)}) \\ \vdots & \ddots & \vdots \\ \kappa(x^{(m)}, x^{(1)}) & \ldots & \kappa(x^{(m)}, x^{(m)}) \end{bmatrix}$$

Twierdzenie 8.5.1 (Mercer)**.** κ jest funkcją jądrową wtedy i tylko wtedy, gdy dla dowolnych obserwacji macierz Grama jest dodatnio półokreślona.

Twierdzenie 8.5.2. Następujące warunki są równoważne:

- M jest dodadnio półokreślona
- $M = R^T R$ dla pewnego R
- wartości własne M są nieujemne

8.5.1 Przykłady funkcji jądrowych

- Jądro gaussowskie
$$\kappa(x, z) = \exp\left(\frac{-\|x - z\|^2}{2\tau^2}\right)$$
- Dla skończonego zbioru D i jego podzbiorów A_1, A_2
$$\kappa(A_1, A_2) = 2^{|A_1 \cap A_2|}$$
- Dla rozkładu prawdopodobieństwa p
$$\kappa(x, z) = p(x)p(z)$$
- Dla rodziny rozkładów $\{p_i : i \in \mathbb{N}\}$ oraz rozkładu wag $p(i)$
$$\kappa(x, z) = \sum_{i \in \mathbb{N}} p_i(x)p_i(z)p(i)$$
- Mnożenie przez stałą
$$c\kappa(x, z)$$
- Mnożenie przez funkcję
$$f(x)\kappa(x, z)f(z)$$
- Suma
$$\kappa_1(x, z) + \kappa_2(x, z)$$
- Przeliczalna suma
$$\sum_{i \in \mathbb{N}} \kappa_i(x, z)$$
- Iloczyn
$$\kappa_1(x, z)\kappa_2(x, z)$$
- Zastosowanie wielomianu o nieujemnych współczynnikach
$$P(\kappa(x, z))$$
- Zastosowanie funkcji wykładniczej
$$\exp(\kappa(x, z))$$

- Złożenie z funkcją
$$\kappa(\phi(x), \phi(z))$$
- Mnożenie przez symetryczną dodatnio półokreśloną macierz
$$\kappa(x, z) = x^T A z$$
- Suma po współrzędnych
$$\sum_{i=1}^{k} \kappa_i(x_i, z_i)$$
- Iloczyn po współrzędnych
$$\prod_{i=1}^{k} \kappa_i(x_i, z_i)$$

8.6 Twierdzenie o reprezentacji

8.6.1 Pomocnicze lematy

Lemat 8.6.1. Norma $\|\cdot\| : \mathbb{R}^d \to \mathbb{R}$ pochodzi od iloczynu skalarnego tj. $\|x\|^2 = \langle x, x\rangle$ wtedy i tylko wtedy gdy
$$\forall_{x,y\in\mathbb{R}^d} : 2\|x\|^2 + 2\|y\|^2 = \|x+y\|^2 + \|x-y\|^2$$

Dowód.

$\Longrightarrow$

$$\|x+y\|^2 + \|x-y\|^2 = \langle x+y, x+y\rangle + \langle x-y, x-y\rangle$$

$\Longleftarrow$

Definiujemy
$$\langle x, y\rangle = \frac{1}{4}\big(\|x+y\|^2 - \|x-y\|^2\big)$$

1. $\langle x\rangle = \|x\|^2$
2. $\langle x, y\rangle = \langle y, x\rangle$
3. $\langle x+y, z\rangle = \langle x, z\rangle + \langle y, z\rangle$
4. $\langle ax, y\rangle = a\langle x, y\rangle$

□

Lemat 8.6.2. Niech M będzie domkniętą podprzestrzenią $\mathbb{R}^d$. Wtedy dla dowolnego $x \in \mathbb{R}^d$ istnieje dokładnie jedno $P_M(x) = m_0 \in M$ takie że
$$\forall_{m\in M} \|x - m_0\| \leq \|x - m\|$$
gdzie $P_M(x)$ jest rzutem ortogonalnym tj.

- $P_M(ax + by) = aP_M(x) + bP_M(y)$

- $P_M = P_M \circ P_M$
- $\langle P_M(x), y\rangle = \langle x, P_M(y)\rangle$
- $\|P_M(x)\| \leq \|x\|$

Dowód.

Bierzemy

$$d = \inf_{m\in M} \|x - m\|$$

Istnieje ciąg

$$(z_n)_{n\in\mathbb{N}} \subseteq M : \|x - z_n\|^2 \leq d^2 + \frac{1}{n}$$

Dla $n_1, n_2 \in \mathbb{N}$, $\frac{z_{n_1}+z_{n_2}}{2} \in M$

$$\begin{aligned}\|z_{n_1} - z_{n_2}\|^2 &= \|z_{n_1} - x + x - z_{n_2}\|^2 \\ &= 2\|z_{n_1} - x\|^2 + 2\|z_{n_2} - x\|^2 - \|z_{n_1} - x - x + z_{n_2}\|^2 \\ &\leq 4d^2 + \frac{2}{n_1} + \frac{2}{n_2} - 4\|\ldots\|\end{aligned}$$

W takim razie $\lim_{n\to\infty} z_n = m_0$ Czyli d jest realizowane przez m_0.

Jeszcze trzeba pokazać jedyność: Niech istnieje m_1 takie że

$$\forall_{m\in M} \|x - m_0\| = \|x - m_1\| \leq \|x - m\|$$

Policzmy

$$\begin{aligned}\|m_0 - m_1\| &= \|m_0 - x + x - m_1\|^2 \\ &= 2\|m_0 - x\|^2 + 2\|m_1 - x\| - \|m_0 - x + m_1 - x\| \\ &\leq 2d^2 + 2d^2 - 4d^2 = 0\end{aligned}$$

Czyli $m_0 = m_1$

□

8.6.2 Właściwe twierdzenie

Twierdzenie 8.6.1 (O reprezentacji)**.** Niech $x^{(1)}, \ldots, x^{(m)} \in X$, $\psi : X \to \mathbb{R}^k$ $L : \mathbb{R}^k \to \mathbb{R}$ oraz niech $R : \mathbb{R}_{\geq 0} \to 0$ będzie niemalejąca. Definiujemy funkcję straty

$$J(w) = R(\|w\|) + L(\langle w, \psi(x^{(1)})\rangle, \ldots \langle w, \psi(x^{(1)})\rangle)$$

Jeśli $J(w)$ osiąga minimum to istnieje

$$\arg\min J(w) = w^* = \sum_{i=1}^{m} \alpha_i \psi(x^{(i)})$$

A ponadto, jeśli R jest silnie rosnąca to wszystkie argumenty są tej postaci.

Dowód. Niech $M = \operatorname{span}\{\psi(x^{(1)}), \dots \psi(x^{(m)})\}$

Niech w minimalizuje $J(w)$.

$w^* = P_M(w)$

$w^\perp = w - w^*$

Dla dowolnego $z \in M$ mamy

$$\begin{aligned}\langle w^\perp, z\rangle &= \langle w - w^*, z\rangle \\ &= \langle w - w^*, P_M(z)\rangle \\ &= \langle w, P_M(z)\rangle - \langle w^*, P_M(z)\rangle \\ &= \langle P_M(w), z\rangle - \langle P_M(w^*), z\rangle \\ &= 0\end{aligned}$$

W takim razie

$$\langle w^*, \psi(x^{(i)})\rangle = \langle w, \psi(x^i)\rangle$$

Zatem wartość funkcji L się nie zmienia. Mamy ponadto $\|w^*\| \leq \|w\|$ a R jest niemalejąca, zatem J nie może być większe. □

Rozdział 9

Złożoność hipotez

9.1 PAC-nauczalność

Będziemy zajmowali się klasyfikacją binarną, tj. wejścia X będą dowolne, ale wyjścia $Y = \{0, 1\}$

Definicja 9.1.1. Pojęcie to dowolna funkcja $c : X \to Y$ lub równoważnie dowolny podzbiór $c \subseteq X$ zawierający pozytywne wejścia.

Definicja 9.1.2. Klasa pojęć to dowolny podzbiór $C \subseteq Y^X$

Będziemy się zajmowali różnymi zbiorami hipotez $H \subseteq Y^X$ – nie muszą być tym samym co C

Zakładamy, że próbka $S = \left\{x^{(i)}, \ldots, x^{(m)}\right\} \subseteq X$ została wybrana niezależnie z jakiegoś rozkładu D, a etykiety są zadane jakimś (ustalonym, ale nieznanym) pojęciem $y^{(i)} = c(x^{(i)})$

Naszym celem jest znaleźć algorytm, który dla danych wejściowych stworzonych w ten sposób minimalizuje ryzyko

$$R(h) = \mathbb{E}[\mathbf{1}[h(x) \neq c(x)]]$$

Definicja 9.1.3. Mówimy, że klasa pojęć C jest **PAC-nauczalna** jeśli istnieje taki algorytm oraz wielomianowa funkcja $Q : \mathbb{R}^2 \to \mathbb{R}$ taka, że dla dowolnego rozkładu D na danych wejściowych oraz dowolnego pojęcia c a ponadto dowolnych $\varepsilon, \delta > 0$ jeśli mamy próbkę S o rozmiarze $|S| \geq Q(\frac{1}{\varepsilon}, \frac{1}{\delta})$ to nasz algorytm konstruuje hipotezę h_S dla której:

$$p(R(h_S) \leq \varepsilon) \geq 1 - \delta$$

Innymi słowy – jeśli wiemy jakie rodzaje pojęć dostajemy to mając wystarczająco dużo danych z dowolnie dużym prawdopodobieństwem mamy dowolnie dużą skuteczność.

9.2 Złożoność próbkowa

Definicja 9.2.1. Mówimy, że hipoteza h_S utworzona na próbce S jest **spójna** jeśli błąd empiryczny jest zerowy czyli $\forall_{x \in S} : h_S(x) = c(x)$.

Twierdzenie 9.2.1. Niech $H \subseteq Y^X$ będzie skończonym zbiorem hipotez a $c \in H$ będzie pojęciem. Jeśli dla dowolnej próbki S zwraamy spójną hipotezę h_S to dla dowolnych $\varepsilon, \delta > 0$ o ile

$$m \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right)$$

to

$$p(R(h_S) \leq \varepsilon) \geq 1 - \delta$$

Dowód. Pokażemy że możemy dobrać takie m dla którego

$$p(R(h_S) > \varepsilon) \leq \delta$$

Skoro $R(h_S) > \varepsilon$ to z definicji ryzyka $p(h_S(x) \neq c(x)) > \varepsilon$.

Ale skoro h_S jest spójna to prawdopodobieństwo wylosowania spójnego z nią S wynosi co najwyżej $(1-\varepsilon)^m$.

Pesymistycznie mamy $|H| - 1$ hipotez z których każda może być spójna, ale niepoprawna – algorytm może wybrać dowolną z nich.

Prawdopodobieństwo, że S jest spójny z dowolną z nich ograniczamy od góry poprzez union bound:

$$p(R(h_S) > \varepsilon) \leq (|H| - 1)(1-\varepsilon)^m \leq |H| \exp(\varepsilon m)$$

Przekształcając dostajemy tezę:

$$\begin{aligned}
|H| \exp(\varepsilon m) &\leq \delta \\
\exp(\varepsilon m) &\leq \frac{\delta}{|H|} \\
\varepsilon m &\leq \ln \delta - \ln |H| \\
m &\leq \frac{1}{\varepsilon}(\ln \delta - \ln |H|) \\
m &\geq \frac{1}{\varepsilon}(\ln |H| - \ln \delta) \\
m &\geq \frac{1}{\varepsilon}\left(\ln |H| + \ln \frac{1}{\delta}\right)
\end{aligned}$$

□

9.3 Wymiar Wapnika-Czerwonienkisa

Definicja 9.3.1. Dla przestrzeni hipotez H definiujemy **funkcję wzrostu** $\Pi_H : \mathbb{N} \to \mathbb{N}$ jako:

$$\Pi_H(m) = \max_{x^{(1)},\ldots,x^{(m)} \in X} \left| \left\{ (h(x^{(1)}), \ldots, h(x^{(m)})) : h \in H \right\} \right|$$

Innymi słowy – dla każdego m-elementowego podzbioru X zliczamy liczbę etykietowań, które umiemy uzyskać i wybieramy największą wartość.

Definicja 9.3.2. Mówimy, że H rozbija m-elementowy $S \subseteq X$ jeśli $\Pi_H(m) = 2^m$ tj. realizuje wszystkie możliwe jego etykietowania.

Definicja 9.3.3. Wymiar Wapnika-Czerwonienkisa przestrzeni hipotez H to moc największego zbioru, który jest rozbity przez H

$$VC(H) = \max\{m : \Pi_H(m) = 2^m\}$$

Lemat 9.3.1 (Sauer)**.** Niech $VC(H) = d$. Wtedy dla dowolnego $m \in \mathbb{N}$ zachodzi

$$\Pi_H(m) \leq \sum_{i=0}^{d} \binom{m}{i}$$

Dowód. □

Rozdział 10

KNN

10.1 KNN

Co tu dużo mówić – mamy jakieś dane, jak chcemy przewidywać wyjście dla nowych wartości to znajdujemy k najbliższych sąsiadów według jakiejś metryki i wybieramy najczęstszą odpowiedź.

Wybór k:

- Za małe k jest podatne na szum/obserwacje odstające
- Za duże k oznacza duży koszt obliczeniowy
- Zazwyczaj chcemy nieparzyste k
- Przyjmuje się, że $k \leq \sqrt{m}$

Optymalizacje:

- Wybranie mniejszej liczby cech
- Użycie kd-drzew aby szybko znajdować bliskie punkty
- Nie liczyć odległości dokładnie tylko ją szacować
- Eliminacja redundancji

10.1.1 Wady i zalety

Zalety:

- brak założeń o danych
- prostota
- wysoka dokładność
- dobre dla regresji i klasyfikacji

Wady:

- drogie obliczenia
- potrzebujemy trzymać zbiór treningowy
- mało istotne cechy są brane pod uwagę

Rozdział 11

Drzewa decyzyjne

11.1 Drzewa decyzyjne

Niech $x = (x_1, \ldots, x_k) \in X$ będzie elementem z przestrzeni wejść.

Będziemy konstruować drzewa binarne w których węzły są jednej z dwóch postaci:

- $x_i \leq t$, $t \in \mathbb{R}$ gdy i-ta cecha jest ciągła
- $x_i \in A_0$ gdzie A_0 jest jakimś nietrywialnym podzbiorem wartości jakie przyjmuje ta cecha.

W ten sposób otrzymujemy podział X na parami rozłączne zbiory $R_1, \ldots, R_M$

11.1.1 Regresja

Każdemu R_j przypisujemy jakąś wartość, którą przewidujemy dla wszystkich jego elementów. Innymi słowy funkcja decyzyjna jest postaci

$$h(x) = \sum_{j=1}^{M} \widehat{c}_j \cdot \mathbf{1}[x \in R_j]$$

Jeśli $l(y, \widehat{y}) = (y - \widehat{y})^2$ to $\widehat{c}_j$ jest średnią arytmetyczną wartości w danym obszarze.

$$\widehat{c}_j = \frac{1}{|R_j|} \sum_{i: x^{(i)} \in R_j} y^{(i)}$$

11.1.2 Złożoność drzewa

Oczywiście minimalny błąd treningowy uzyskujemy przy najmniejszym rozdrobnieniu, gdy każdy obszar zawiera dokładnie jeden punkt – zazwyczaj nie chcemy takiej sytuacji.

Dla uproszczenia załóżmy, że $x \in R^k$

Dla każdej cechy j oraz punktu s definiujemy dwa zbiory uzyskane w wyniku podziału:

- $R_1(j, s) = \{x : x_j \leq s\}$
- $R_2(j, s) = \{x : x_j > s\}$

oraz odpowiadające im stałe

- $\widehat{c}_1 = \sum_{i:x^{(i)} \in R_1} \frac{y^{(i)}}{|R_1|}$
- $\widehat{c}_2 = \sum_{i:x^{(i)} \in R_2} \frac{y^{(i)}}{|R_2|}$

Chcemy teraz zminimalizować błąd:

$$J(j,s) = \sum_{i:x^{(i)} \in R_1(j,s)} \left(y^{(i)} - \widehat{c}_1(j,s)\right)^2 + \sum_{i:x^{(i)} \in R_2(j,s)} \left(y^{(i)} - \widehat{c}_2(j,s)\right)^2$$

W tym celu dla każdej cechy j osobno sortujemy obserwacje rosnąco po x_j uzyskując ciąg $x^{(j_1)}, \ldots, x^{(j_m)}$.

Możliwych punktów podziału jest teraz $m-1$ – są to środki pomiędzy kolejnymi punktami. Dla każdej cechy wybieramy ten punkt, który minimalizuje błąd, a następnie wybieramy tę cechę dla której ten błąd jest najmniejszy. Dzielimy drzewo i kontynuujemy rekurencyjnie.

11.1.3 Pruning

Kiedyś musimy zakończyć budowę drzewa aby nie doszło do przeuczenia.

Idea pruningu polega na tym, że celowo konstruujemy duże (przeuczone) drzewo, z którego będziemy stopniowo kasować węzły aby otrzymać coś sensownego.

Definicja 11.1.1. Niech T_0 będzie drzewem decyzyjnym. Mówimy, że $T \subset T_0$ jest jego **poddrzewem** jeśli możemy je otrzymać poprzez usunięcie niektórych węzłów z T_0.

Definicja 11.1.2. Złożoność drzewa T, oznaczaną przez $|T|$ będziemy oznaczać liczbę liści tego drzewa.

Dla ustalonego α definiujemy funkcję kosztu drzewa T

$$J_\alpha(T) = \widehat{R}(T) + \alpha|T|$$

Intuicyjnie odpowiada to znalezieniu balansu pomiędzy ryzykiem empirycznym a rozmiarem drzewa – w naturalny sposób większe drzewa mają mniejsze ryzyko empiryczne kosztem możliwego przeuczenia ze względu na rozmiar.

Parametr α dobieramy eksperymentalnie za pomocą danych walidacyjnych.

Optymalnego drzewa szukamy zachłannie w kolejnych iteracjach – T_1 powstaje przez usunięcie jednego z liści T_0 minimalizując przy tym $\widehat{R}(T_1) - \widehat{R}(T_0)$.

T_2 konstruujemy kasując minimalny liść z T_1 i tak dalej, aż do wyczerpania zapasów (tj. aż zostanie nam jeden węzeł).

Dostajemy w ten sposób $|T_0|$ drzew, wśród których znajduje się $\arg\min_{T \subseteq T_0} J_\alpha(T)$ dla dowolnego $\alpha \geq 0$

11.1.4 Klasyfikacja

Załóżmy że mamy K klas które jakoś chcemy przypisać danym.

Mając drzewo decyzyjne dzielące X na obszary $R_1, \ldots, R_M$ obliczamy prawdopodobieństwa

$$\widehat{p}_{n,k} = \frac{1}{|R_n|} \sum_{i:x^{(i)} \in R_n} \mathbf{1}\left[y^{(i)} = k\right]$$

Naturalnym wyborem funkcji decyzyjnej jest przypisanie każdemu liściowi najczęstszej klasy.

Aby wartości $\widehat{p}_{n,k}$ miały jakiś sens wprowadzamy *miarę nieczystości*. Intuicyjnie liść jest czysty jeśli jakaś klasa wyraźnie w nim dominuje.

Przykładowe funkcje nieczystości to:

- błąd klasyfikacji
$$1 - \widehat{p}_{n,h(n)}$$
- współczynnik Giniego
$$\sum_{k=1}^{K} \widehat{p}_{n,k}(1 - \widehat{p}_{n,k})$$
- entropia
$$-\sum_{k=1}^{K} \widehat{p}_{n,k} \ln \widehat{p}_{n,k}$$

Tak jak w przypadku regresji wybieraliśmy podział minimalizujący ryzyko empiryczne, tak tutaj będziemy chcieli minimalizować ważoną nieczystość – dla miary nieczystości Q oraz potencjalnego podziału na R_1, R_2 minimalizujemy

$$|R_1| \cdot Q(R_1) + |R_2| \cdot Q(R_2)$$

11.1.5 Niebinarne cechy

Jeśli jakaś cecha x_j może przyjąć q możliwych, nieporównywalnych wartości to potencjalnie mamy $2^{q-1} - 1$ podziałów na niepuste podzbiory – to dość dużo.

Zamiast tego sortujemy cechy $A_1, \ldots, A_q$ rosnąco po $p(y = 1 \mid x_j = A_i)$.

Mamy dzięki temu $q - 1$ możliwych podziałów, co jest już dużo lepsze.

11.1.6 Wady i zalety

Zalety:

- łatwo zrozumieć co robią
- szybko klasyfikują jak już je skonstruujemy
- nie wymagają przetwarzania danych

Wady:

- nie zawsze są optymalne
- wrażliwe na zmiany
- długi czas uczenia
- dla danych ciągłych podziały są bardzo sztywne

11.2 Lasy losowe

11.2.1 Bootstrapping

Definicja 11.2.1. Niech $S = \{x^{(1)}, \ldots, x^{(m)}\} \subseteq X$

Próbą bootstrapową z próbki S nazywamy dowolny m-elementowy multizbiór elementów wybranych S losowanych jednostajnie ze zwracaniem.

Można wyliczyć, że w takiej próbce znajdzie się ok. 63% obserwacji z S

Okazuje się, że jeśli szacujemy jakiś parametr ϕ zgodnie z którym zostały wybrane dane do S to histogram na próbkach boostrapowych jest podobny do niezależnie utworzonych próbek.

Dzięki bootstrapowaniu możemy wygenerować dużo próbek co jest pomocne gdy nie mamy dostępu do całej populacji.

Ponadto mamy ciekawą własność – niech $Z_1, \ldots, Z_n$ będą parami niezależnymi zmiennymi z tego samego rozkładu o średniej μ i wariancji σ^2.

Mamy wtedy

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} Z_i\right] = \mu$$

oraz

$$\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} Z_i\right) = \frac{\sigma^2}{n}$$

dostaliśmy mniejszą wariancję.

Jedyny problem jest taki, że próbki są niezależne pod warunkiem oryginalnej próbki S, ale są zależne względem całej populacji.

11.2.2 Bagging - bootstrap aggregate

Niech $S_1, \ldots, S_B$ będą niezależnymi m-elementowymi zbiorami treningowymi. Dostajemy B hipotez $\widehat{h}_1(x), \ldots, \widehat{h}_B(x)$ na podstawie których tworzymy nową hipotezę.

Dla regresji:

$$\widehat{h}(x) = \frac{1}{B}\sum_{i=1}^{B} \widehat{h}_i(x)$$

podobnie dla klasyfikacji

$$\widehat{h}(x) = \operatorname*{arg\,max}_{k\in\{1,\ldots,K\}} \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\left[\widehat{h}_i(x) = k\right]$$

W praktyce mimo zależności uśrednianie hipotez daje sensowne wyniki.

11.2.3 Błąd out-of-bag

Jak zauważyliśmy, każda próbka zawiera ok. 63% obserwacji, pozostałe potraktujemy jako dane walidacyjne.

Dla $x^{(i)} \in S$ definiujemy zbiór próbek do których nie trafił:

$$S^{(i)} = \left\{b : x^{(i)} \notin S_b\right\}$$

Tworzymy hipotezę out-of-bag:

$$\widehat{h}_{OOB}\left(x^{(i)}\right) = \frac{1}{|S^{(i)}|} \sum_{b \in S^{(i)}} \widehat{h}_b\left(x^{(i)}\right)$$

Błąd dla tej hipotezy nazywamy błędem out-of-bag.

11.2.4 Las losowy

Las losowy powstaje przez utworzenie wielu drzew decyzyjnych, każde na innej próbce bootstrapowej. Podczas konstrukcji drzewa w każdym węźle losujemy podzbiór cech do którego rozważań się zawężamy.

11.2.5 Wady i zalety

Zalety:

- mają mniejszą wariancję niż drzewa decyzyjne – mniejsze przeuczenie
- radzą sobie z brakującymi danymi
- dobre dla dużych wielowymiarywych danych
- mniej wrażliwe na zmiany w obserwacjach

Wady:

- mniej intuicyjne
- nie zawsze dobre dla regresji
- jeszcze dłużej się liczy niż drzewa decyzyjne
- ryzyko przeuczenia gdy cechy są silnie skorelowane

Rozdział 12

Boosting

12.1 Słabe klasyfikatory

Definicja 12.1.1. Algorytm uczący nazwiemy **słabym** jeśli istnieje $\gamma > 0$ taka że dla dowolnego rozkładu D na danych wejściowych, dowolnej $\delta > 0$ oraz próbki $S \subseteq X$ o odpowiednim rozmiarze m algorytm zwraca hipotezę h_S dla której zachodzi

$$p(R(h_S) \leq \frac{1}{2} - \gamma) \geq 1 - \delta$$

Innymi słowy, z dużym prawdopodobieństwem dostajemy coś co klasyfikuje choć trochę lepiej niż losowo, w szczególności dobre klasyfikatory są również słabe w myśl tej definicji, a sama „słabość" odnosi się raczej do naszych wymagań co do klasyfikatora.

Na podstawie zachowania słabych klasyfikatorów będziemy przydzielać obserwacjom i klasyfikatorom jakieś wagi i na ich podstawie stworzymy końcową hipotezę.

Jeśli mamy T słabych klasyfikatorów $h_1, \ldots, h_T$ to stworzymy jeden silny klasyfikator

$$h(x) = \operatorname{sgn}\left(\sum_{i=1}^{T} \alpha_i \cdot h_t(x)\right)$$

.

Naszym celem jest odpowiednie dobranie wag α_i. Ponadto każda obserwacja $(x^{(i)}, y^{(i)}$ dostanie jakąś wagę $w_t^{(i)}$ – im trudniej ją klasyfikować tym większa ta waga.

12.2 AdaBoost

Zaczynamy od przydzielenia wszystkim obserwacjom takiej samej wagi $w_1^{(i)} = \frac{1}{m}$

W t-tej iteracji wybieramy ten klasyfikator, który popełnia najmniejszy ważony błąd $\epsilon_t \leq \frac{1}{2}$

Oznaczamy ten klasyfikator przez h_t i nadajemy mu wagę

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t} \in [0, \infty)$$

Następnie modyfikujemy wagi obserwacjom:

$$w_{t+1}^{(i)} = \begin{cases} w_t^{(i)} \cdot \exp(\alpha_t) & \text{gdy } h_t(x^{(i)}) \neq y^{(i)} \\ w_t^{(i)} \cdot \exp(-\alpha_t) & \text{gdy } h_t(x^{(i)}) = y^{(i)} \end{cases}$$

W ten sposób wagi dla dobrze trafionych obserwacji maleją, a nietrafionych rosną.

Po każdej iteracji normalizujemy wagi tak aby ich suma wynosiła 1.

Zatrzymujemy jeśli zajdzie jeden z przypadków:

- Wykonamy T iteracji
- Znajdziemy h_t z zerowym błędem
- Wszystkie klasyfikatory mają ważony błąd $\frac{1}{2}$ – wtedy $\alpha_t = 0$ i już nic się nie zmieni.

Zachodzi twierdzenie:

Twierdzenie 12.2.1. Po T iteracjach ryzyko empiryczne klasyfikatora zwróconego przez AdaBoost jest ograniczony

$$\widehat{R}(h) \leq \exp\left(-2\sum_{t=1}^{T}\left(\frac{1}{2} - \epsilon_t\right)^2\right)$$

ponadto jeśli $\forall_{t\in\{1,\ldots,T\}} : \gamma \leq \frac{1}{2} - \epsilon_t$ to

$$\widehat{R}(h) \leq \exp\bigl(-2\gamma^2 T\bigr)$$

12.2.1 Wady i zalety

Zalety

- prosty
- tylko jeden hiperparametr – T
- nie przeucza się
- działa dla różnych słabych klasyfikatorów

Wady:

- podatny na obserwacje odstające
- czasem nie działa jeśli źle dobierzemy klasyfikatory

Rozdział 13

Klasteryzacja

13.1 Grupowanie hierarchiczne

Idea jest bardzo prosta – mamy m punktów, chcemy je pogrupować. No to grupujemy je poprzez scalanie dwóch grupek w kolejnych iteracjach. To które dwie grupki będziemy łączyć to zależy od metody jaką wybierzemy.

Zaczynamy od wyznaczenia odległości między każdymi dwoma punktami $d(a, b)$ – może to być odległość w dowolnej metryce w zależności od problemu.

Następnie wybieramy klastry które są najpodobniejsze do siebie. Możliwe kryteria:

- łączenie pojedyncze
$$D(A, B) = \min\{d(a, b) : a \in A, b \in B\}$$
Radzi sobie z niekulistymi kształtami ale jest wrażliwe na obserwacje odstające
- łączenie pełne
$$D(A, B) = \max\{d(a, b) : a \in A, b \in B\}$$
Jest mniej czułe na obserwacje odstające ale rozbija duże klasy.
- łączenie średnie
$$D(A, B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$$
Ma tendencje kuliste i jest mniej podatne na obserwacje odstające.
- łączenie centroidalne
$$D(A, B) = d(c_A, c_B)$$
Dobrze się sprawdza ale jest kosztowne ze względu na liczenie centroidów
- łączenie Warda
$$D(A, B) = \sum_{a \in A \cup B} d(a, c_{a \cup B})^2$$
Działa trochę jak średnie.

gdzie centroid to punkt minimalizujący sumę kwadratów odległości.

Aby nieco usprawnić liczenie odległości istnieje algorytm Lance'a-Williamsa – zaczynamy od zainicjowania macierzy D odległościami między punktami, a następnie gdy łączymy dwa klastry to kasujemy odpowiadające im wiersze/kolumny i dodajemy nowy wiersz i nową kolumnę w których wartości obliczamy na podstawie właśnie usuniętych danych.

13.1.1 Wady i zalety

Zalety:

- Nie wymaga znajomości liczby klastrów
- Algorytm nie jest randomizowany – na tych samych danych działa tak samo
- Intuicyjny

Wady:

- Wolny
- Nie działa gdy brakują wartości
- Nie działa dobrze jeśli cechy są różnych typów

13.2 k-means

13.2.1 k-means

Tutaj ustalamy a priori liczbę klastrów k i wykonujemy algorytm:

1. Wylosuj k punktów $\mu_1, \ldots, \mu_k$ – będą to „centroidy" naszych klastrów
2. Aż do stabilizacji powtarzaj:
   (a) Przydziel każdy punkt x do klastra najbliższego μ_i
   (b) Policz prawdziwe centroidy tak uzyskanych klastrów

13.2.2 k-means++

Jeśli mamy pecha i wybierzemy słabe centroidy to k-means daje złe wyniki. Aby sobie nieco z tym poradzić chcemy wybierać centroidy w trochę bardziej zbalansowany sposób – tak aby były one mniej więcej równo rozłożone po przestrzeni.

Oczekiwany błąd jest co najwyżej $8(\lg k + 2)$ razy gorszy niż minimalny jaki możemy uzyskać.

13.2.3 Wady i zalety

Zalety:

- Proste
- dobre dla dużych i kulistych danych
- działa w czasie liniowym
- klastry są zwarte

Wady:

- wymaga wielu przebiegów ze względu na randomizację
- wymaga skalowania danych
- działa tylko na danych numerycznych
- nie radzi sobie z niekulistymi kształtami

13.3 Klasteryzacja spektralna

Na podstawie danych konstruujemy graf podobieństwa, mamy kilka sposobów:

- Łączymy punkty które są wystarczająco blisko siebie, bez wag
- Łączymy k-najbliższych sąsiadów, bez wag
- Tworzymy graf pełny gdzie wagi na krawędziach oznaczają jak blisko siebie są punkty np. jądrem gaussowskim lub odwrotnością odległości

Definiujemy dwie macierze:

$$A_{i,j} = \begin{cases} w_{i,j} & \text{gdy } \{i,j\} \in E \\ 0 & \text{wpp} \end{cases}$$

oraz macierz diagonalną

$$D_{i,i} = \sum_{j:\{i,j\}\in E} w_{i,j}$$

W ten sposób otrzymujemy **laplasjan** $L = D - A$ Liczymy jego wartości własne $\lambda_1 \leq \cdots \leq \lambda_m$.

Możemy też policzyć znormalizowany laplasjan:

$$L_N = D^{-0.5} L D^{-0.5}$$

i wyznaczyć jego k najmniejszych wartości własnych $\lambda_1, \ldots, \lambda_k$. Bierzemy wektory własne $v_1, \ldots, v_k$ które dajemy jako kolumny macierzy U o wymiarach $m \times k$.

Mamy teraz m wierszy na których możemy odpalić znowu klasteryzację i dostać coś sensownego.

13.3.1 Wady i zalety

Zalety:

- Klastry mogą mieć ciekawsze kształty
- Nie trzeba przechowywać całości danych
- Stoi za tym jakaś matematyka

Wady:

- Wymaga wyboru k
- Kosztowne obliczeniowo

13.4 Gaussowskie modele mieszane

W GMM zakładamy, że mamy K klastrów, które mogą się na siebie nakładać. W obrębie każdego klastra punkty są rozłożone zgodnie z pewnym rozkładem normalnym. Aby wylosować punkt najpierw losujemy klaster, a następnie losujemy z niego punkt co zapisujemy jako

$$p(x) = \sum_{i=1}^{K} \pi_i \mathcal{N}(x \mid \mu_i, \Sigma_i)$$

gdzie

$$\sum_{i=1}^{K} \pi_i = 1$$

13.4.1 MLE

Mamy próbkę $D = \left\{x^{(1)}, \ldots, x^{(m)}\right\}$ wylosowaną z pewnego GMM z parametrami

$$\begin{aligned}\pi &= (\pi_1, \ldots, \pi_k)\\ \mu &= (\mu_1, \ldots, \mu_k)\\ \Sigma &= (\Sigma_1, \ldots, \Sigma_k)\end{aligned}$$

Mamy funkcję wiarygodności zadaną wzorem

$$\begin{aligned}L(\pi, \mu, \Sigma) &= p(D \mid \pi, \mu, \Sigma)\\ &= \prod_{i=1}^{m} p(x^{(i)} \mid \pi, \mu, \Sigma)\\ &= \prod_{i=1}^{m} \sum_{j=1}^{K} \pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)\end{aligned}$$

Log-wiarygodność zadana jest wzorem

$$\begin{aligned}\ell(\pi, \mu, \Sigma) &= \ln L(\pi, \mu, \Sigma)\\ &= \sum_{i=1}^{m} \ln\left(\sum_{j=1}^{K} \pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)\right)\end{aligned}$$

No i tutaj napotykamy na taki problem, że logarytm średnio łączy się z sumą. Co gorsza, rozważana funkcja nie jest nawet wypukła i ma wiele maksimów.

Możemy jednak zauważyć ciekawą rzecz – nasz problem składa się z dwóch części:

- przydzielenia etykiet klastrów do punktów
- oszacowania rozkładów klastrów

Jeśli znamy którąś z tych dwóch rzeczy to jesteśmy w stanie bez większego problemu wyznaczyć drugą.

Jeśli znamy $\theta = (\pi, \mu, \Sigma)$ to umiemy oszacować z jakiego klastra pochodzi dany punkt.

$$
\begin{aligned}
p(z = c \mid x, \theta) &= \frac{p(x \mid z = c, \theta) p(z = c \mid \theta)}{p(x \mid \theta)} \\
&= \frac{\pi_c \mathcal{N}(x \mid \mu_c, \Sigma_c)}{\sum_{i=1}^{K} \pi_i \mathcal{N}(x \mid \mu_i, \Sigma_i)}
\end{aligned}
$$

Jeśli znamy etykiety $z^{(1)}, \ldots, z^{(m)}$ to nasza log-wiarygodność upraszcza się do

$$
\begin{aligned}
\ell(\pi, \mu, \Sigma) &= \sum_{i=1}^{m} \ln \left(\sum_{j=1}^{K} \mathbf{1}\left[z^{(i)} = j\right] \pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j) \right) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{k} \mathbf{1}\left[z^{(i)} = j\right] \ln\left(\pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)\right) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{k} \mathbf{1}\left[z^{(i)} = j\right] \ln \pi_j + \sum_{i=1}^{m} \sum_{j=1}^{k} \mathbf{1}\left[z^{(i)} = j\right] \ln \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)
\end{aligned}
$$

Dzięki znajomości etykiet udało nam się rozdzielić wybór rozkładu od jego parametrów co pozwala nam na proste policzenie pochodnych.

Załóżmy, że mamy informacje do których klastrów przynależą nasze punkty tj. znamy

$$
\gamma_c^{(i)} = p(z^{(i)} = c \mid x^{(i)}, \theta)
$$

Wtedy maksymalizujemy **oczekiwaną log-wiarygodność** (a przynajmniej tak by wynikało ze slajdów, bo inaczej ciężko sensownie uwzględnić gammy) tj.

$$
\begin{aligned}
\mathbb{E}[\ell(\pi, \mu, \Sigma)] &= \mathbb{E}\left[\sum_{i=1}^{m} \sum_{j=1}^{k} \mathbf{1}\left[z^{(i)} = j\right] \ln\left(\pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)\right) \right] \\
&= \sum_{i=1}^{m} \sum_{j=1}^{k} \mathbb{E}\left[\mathbf{1}\left[z^{(i)} = j\right]\right] \ln\left(\pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)\right) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{k} \gamma_j^{(i)} \ln\left(\pi_j \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)\right) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{k} \gamma_j^{(i)} \ln \pi_j + \sum_{i=1}^{m} \sum_{j=1}^{k} \gamma_j^{(i)} \ln \mathcal{N}(x^{(i)} \mid \mu_j, \Sigma_j)
\end{aligned}
$$

Pochodne obliczamy identycznie jak w poprzednim przypadku – w wyniku po prostu zastępujemy każde $\mathbf{1}\left[z^{(i)} = c\right]$ przez $\gamma_c^{(i)}$

13.4.2 Maksymalizacja wartości oczekiwanej

Bazując na powyższych spostrzeżeniach możemy szacować parametry π, μ, Σ iteracyjnie wykonując na zmianę dwa kroki – raz obliczamy etykiety, a raz obliczamy rozkłady klastrów.

1. Wylosuj π, μ, Σ
2. Wykonuj aż do warunku stopu (np. bardzo małej zmiany)

(a) **E** (Expectation)

Dla każdego $x^{(i)}$ wylicz wszystkie $\gamma_c^{(i)}$

(b) **M** (Maximization)

Uaktualnij π, μ, Σ tak, aby maksymalizowały oczekiwaną log-wiarygodność.

13.5 Algorytm maksymalizacji wartości oczekiwanej

13.5.1 Dywergencja Kullbacka-Leiblera

Chcielibyśmy mieć jakiś sposób na obliczanie podobieństwa dwóch rozkładów, będzie nam to za chwilę przydatne w konstrukcji algorytmu.

Definicja 13.5.1. Dywergencja Kullbacka-Leiblera dla rozkładów q, p zadana jest wzorem dla rozkładów ciągłych

$$\mathrm{KL}(q \parallel p) = \int q(x) \ln \frac{q(x)}{p(x)} \, \mathrm{d}x$$

oraz

$$\mathrm{KL}(q \parallel p) = \sum_x q(x) \ln \frac{q(x)}{p(x)}$$

Podstawowe jej własności:

- $\mathrm{KL}(p \parallel p) = 0$
- $\mathrm{KL}(p \parallel q) \geq 0$
- $\mathrm{KL}(p \parallel q)$ nie musi być równe $\mathrm{KL}(q \parallel p)$

13.5.2 Sformułowanie problemu

Podobnie jak w GMM zakładamy, że dane pochodzą z mieszanego modelu o K rozkładach o parametrach θ. Innymi słowy

$$p(x^{(i)} \mid \theta) = \sum_{c=1}^{K} p(z^{(i)} = c \mid \theta) \cdot p(x^{(i)} \mid z^{(i)} = c, \theta)$$

Będziemy starali się znaleźć taki parametr modelu θ który maksymalizuje log wiarygodność.

$$\theta^* = \arg\max_\theta \sum_{i=1}^{m} \ln \sum_{c=1}^{K} p(x^{(i)}, z^{(i)} = c \mid \theta)$$

Mamy tutaj ten sam problem co wcześniej – nie wejdziemy z logarytmem pod sumę, więc musimy kombinować inaczej.

13.5.3 Ograniczenie dolne

Niech q będzie dowolnym rozkładem na K wartościach. Możemy szacować log-wiarygodność od dołu za pomocą funkcji $\mathcal{L}(\theta, q)$

$$\begin{aligned}
\ln \ell(\theta) &= \sum_{i=1}^{m} \ln \sum_{c=1}^{K} p(x^{(i)}, z^{(i)} = c \mid \theta) \\
&= \sum_{i=1}^{m} \ln \sum_{c=1}^{K} q(z^{(i)} = c) \frac{p(x^{(i)}, z^{(i)} = c \mid \theta)}{q(z^{(i)} = c)} \\
&\geq \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln \frac{p(x^{(i)}, z^{(i)} = c \mid \theta)}{q(z^{(i)} = c)} \\
&= \mathcal{L}(\theta, q)
\end{aligned}$$

Nierówność mamy dzięki nierówności Jensena, wklęsłości logarytmu oraz faktu, że q jest rozkładem prawdopodobieństwa tj. $\sum_{c=1}^{K} q(z = c) = 1$

Skoro $\mathcal{L}(\theta, q)$ jest ograniczeniem dolnym to będziemy chcieli znaleźć największe takie, a więc szukamy q, θ maksymalizujące tę wartość.

Bardziej formalnie – jeśli po k krokach mamy parametr θ_k to obliczamy (krok **E**)

$$q_{k+1} = \arg\max_{q} \mathcal{L}(\theta_k, q)$$

a następnie (krok **M**)

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}(\theta, q_{k+1})$$

13.5.4 Krok E

Szukamy największego ograniczenia dolnego czyli w praktyce minimalizujemy różnicę

$$\begin{aligned}
\ln \ell(\theta) - \mathcal{L}(\theta, q) &= \sum_{i=1}^{m} \ln p(x^{(i)}, \theta) - \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln \frac{p(x^{(i)}, z^{(i)} = c \mid \theta)}{q(z^{(i)} = c)} \\
&= \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln p(x^{(i)}, \theta) - \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln \frac{p(x^{(i)}, z^{(i)} = c \mid \theta)}{q(z^{(i)} = c)} \\
&= \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \left(\ln p(x^{(i)}, \theta) - \ln \frac{p(x^{(i)}, z^{(i)} = c \mid \theta)}{q(z^{(i)} = c)} \right) \\
&= \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln \frac{p(x^{(i)}, \theta) \cdot q(z^{(i)} = c)}{p(x^{(i)}, z^{(i)} = c \mid \theta)} \\
&= \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln \frac{\cdot q(z^{(i)} = c)}{p(x^{(i)}, z^{(i)} = c \mid x^{(i)}, \theta)} \\
&= \mathrm{KL}\big(q(z^{(i)}) \parallel p(z^{(i)} \mid x^{(i)}, \theta)\big)
\end{aligned}$$

A wiemy że jest ona równa zero tylko gdy $q(z^{(i)}) = p(z^{(i)} \mid x^{(i)}, \theta)$.

13.5.5 Krok M

Teraz chcemy znaleźć θ maksymalizujące $\mathcal{L}(\theta, q)$.

Mamy

$$\mathcal{L}(\theta, q) = \sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln \frac{p(x^{(i)}, z^{(i)} = c \mid \theta)}{q(z^{(i)} = c)}$$

Zauważamy jednak, że $q(z^{(i)} = c)$ w mianowniku jest stałe, zatem maksymalizujemy

$$\sum_{i=1}^{m} \sum_{c=1}^{K} q(z^{(i)} = c) \ln p(x^{(i)}, z^{(i)} = c \mid \theta) = \mathbb{E}[\ln p(X, Z \mid \theta)]$$

gdzie wartość oczekiwana przechodzi po rozkładzie wyznaczonym przez q tak jak miało to miejsce w przypadku GMM.

13.5.6 Zbieżność

W każdym kroku tylko maksymalizujemy, więc log-wiarygodność nie maleje

$$\ln p(X \mid \theta_k) = \mathcal{L}(\theta_k, q_{k+1}) \leq \mathcal{L}(\theta_{k+1}, q_{k+1}) \leq \ln p(X \mid \theta_{k+1})$$

Rozdział 14

Redukcja wymiarów

14.1 Rozkład według wartości osobliwych (SVD)

Niech A będzie dowolną macierzą o wymiarach $m \times n$ i niech $m \geq n$.

Szukamy:

- wektorów ortonormalnych
$$v_1, \ldots, v_n \in \mathbb{R}^n$$
- wektorów ortonormalnych
$$u_1, \ldots, u_n \in \mathbb{R}^m$$
- wartości osobliwych
$$\sigma_1, \ldots, \sigma_n \in \mathbb{R}_{\geq 0}$$

dla których

$$Av_j = \sigma_j u_j$$

W postaci macierzowej mamy

$$A \begin{bmatrix} v_1 \ldots v_n \end{bmatrix} = \begin{bmatrix} u_1 \ldots u_n \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_n \end{bmatrix}$$

co zapisujemy dla uproszczenia:

$$AV = \widehat{U}\widehat{\Sigma}$$

Ponieważ V składa się z wektorów ortonormalnych to mamy

$$VV^T = V^T V = I$$

zatem

$$A = \widehat{U}\widehat{\Sigma}V^T$$

Możemy rozszerzyć $\widehat{U}$, które ma wymiar $m \times n$ do macierzy unitarnej U, która już jest kwadratowa i ma wymiary $m \times m$.

Możemy też uzupełnić $\widehat{\Sigma}$ zerami do macierzy o wymiarach $m \times n$

Rozkład $A = U\Sigma V^T$ nazywamy **rozkładem na wartości osobliwe**

Zachodzi twierdzenie

Twierdzenie 14.1.1. Każda macierz ma rozkład na wartości osobliwe, przyczym $\sigma_1, \ldots, \sigma_n$ są jednoznaczne, a ponadto jeśli są parami różne to $u_1, \ldots, u_n$ oraz $v_1, \ldots, v_n$ są jednoznaczne z dokładnością do znaku.

14.1.1 Wyznaczanie rozkładu

$$A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T \Sigma V^T$$

mamy zatem

$$A^T AV = V\Sigma^T \Sigma$$

W szczególności mamy

$$A^T Av_i = \sigma_i^2 v_i$$

czyli kolumny macierzy V są wektorami własnymi $A^T A$.

W podobny sposób kolumny U są wektorami własnymi AA^T

14.2 Analiza składowych głównych (PCA)

Mamy dane w jakiejś przestrzeni $\mathbb{R}^d$. Szukamy hiperpłaszczyzny takiej, że jak zrzutujemy na nią punkty to otrzymamy wysoką wariancję. Równoważnie chcemy aby suma kwadratów odległości do tejże hiperpłaszczyzny była jak najmniejsza.

Szukamy zatem rzutowania $P : R^d \to R^l$, takiego że $P^2 = P$

Ponieważ będziemy szukać odwzorowań liniowych to zakładamy, że nasze dane są wyśrodkowane tj. $\sum_i x^{(i)} = 0$

14.2.1 Przestrzeń jednowymiarowa

W przypadku jednowymiarowym ($l = 1$) nasze rzutowanie jest postaci $\pi(x) = u^T x$ gdzie $u \in \mathbb{R}^d$

Chcemy zmaksymalizować

$$\operatorname{Var}\left(u^T X\right)$$

przy czym zakładamy, że $\|u\| = 1$.

Mamy ponadto

$$\operatorname{Var}\left(u^T X\right) = u^T \operatorname{Cov}(X) u = \frac{1}{m} u^T X^T X u$$

Dla skrócenia zapisu przyjmujemy $C = X^T X$ i mówimy, że szukamy

$$\underset{u \in \mathbb{R}^d, \|u\|=1}{\arg\max}\ u^T C u$$

Aby pozbyć się warunku $\|u\| = 1$ przerabiamy ten problem na znane i lubiane mnożniki Lagrange'a

$$L(u, \lambda) = u^T C u - \lambda(u^T u - 1)$$

dostajemy

$$\frac{\partial L}{\partial u} = 2Cu - 2\lambda u$$

Z czego mamy

$$Cu = \lambda u$$

W takim razie u jest wektorem własnym, a ponadto:

$$u^T Cu = u^T \lambda u = \lambda$$

Szukamy zatem największej wartości własnej

14.2.2 Przestrzeń wielowymiarowa

Jeśli $l > 1$ to po prostu szukamy przestrzeni rozpiętej na l wektorach i analogicznie jak w przypadku $l = 1$ znajdujemy l największych wartości własnych macierzy $X^T X$

14.2.3 Zastosowanie SVD

Dzięki SVD mamy następujący algorytm na rzutowanie:

1. Rozkładamy $X = U\Sigma V^T$
2. Wybieramy l największych wartości i wektorów własnych dostając macierz

$$V_r = \begin{bmatrix} v_1 \dots v_l \end{bmatrix}$$

3. Definiujemy rzutowanie

$$\pi(x) = V_r^T x$$

14.2.4 Wybór liczby składowych

Pozostaje pytanie – jakie l wybieramy ? Ano takie, dla którego suma wybranych składowych jest wystarczająco duża np.

$$\frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 0.95$$

14.3 Jądrowa wersja PCA

Liniowe odwzorowania mogą być czasem niewystarczające, będziemy się zatem starali otrzymać nieliniowe główne składowe.

14.3.1 Zależność między macierzą Grama a macierzą kowariancji

Lemat 14.3.1. Niech $x^{(1)}, \dots x^{(m)} \in \mathbb{R}^d$ będą wyśrodkowanymi danymi. Niech $K = XX^T$ będzie macierzą Grama dla tych danych oraz $\lambda \in \mathbb{R}$ oraz $a \in \mathbb{R}^m$ będą takie, że $Ka = \lambda a$

Niech

$$v = X^T a$$

Wtedy:

1. Jeśli $v \neq 0$ to v jest wektorem własnym macierzy $C = X^T X$ dla wartości własnej λ

2. Jeśli $\|a\| = 1$ to $\|v\| = \sqrt{\lambda}$

Dowód.

1.

$$\begin{aligned} Ka &= \lambda a \\ XX^T a &= \lambda a \\ X^T\big(XX^T a\big) &= X^T(\lambda a) \\ \big(X^T X\big)\big(X^T a\big) &= \lambda X^T a \\ Cv &= \lambda v \end{aligned}$$

2.

$$\begin{aligned} \|v\|^2 &= \left\| \sum_{i=1}^{m} a_i x^{(i)} \right\|^2 \\ &= \left\langle \sum_{i=1}^{m} a_i x^{(i)}, \sum_{i=1}^{m} a_i x^{(i)} \right\rangle \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j \big\langle x^{(i)}, x^{(j)} \big\rangle \\ &= a^T K a \\ &= a^T \lambda a \\ &= \lambda \end{aligned}$$

□

Mamy podobny lemat w drugą stronę

Lemat 14.3.2. Niech $Cv = \lambda v$, definiujemy a

$$a = \frac{1}{\lambda} X v$$

wtedy a jest wektorem własnym macierzy Grama dla wartości λ

Dowód.

Zaczniemy od pokazania, że niezerowe wektory własne macierzy C są liniowymi kombinacjami $x^{(i)}$.

Mamy

$$\begin{aligned} v &= \frac{1}{\lambda} Cv \\ &= \frac{1}{\lambda} \sum_{i=1}^{m} x^{(i)} x^{(i)^T} v \\ &= \frac{1}{\lambda} \sum_{i=1}^{m} x^{(i)} \langle x^{(i)}, v \rangle \\ &= \sum_{i=1}^{m} x^{(i)} \frac{1}{\lambda} \langle x^{(i)}, v \rangle \\ &= \sum_{i=1}^{m} x^{(i)} a_i \end{aligned}$$

Teraz pokażemy, jak wyrazić wektory własne macierzy C mając wektory własne macierzy K

$$\begin{aligned} Cv &= \lambda v \\ \left(\sum_{i=1}^{m} x^{(i)} x^{(i)^T} \right) \left(\sum_{i=1}^{m} x^{(i)} a_i \right) &= \lambda \sum_{i=1}^{m} a_i x^{(i)} \\ \sum_{i=1}^{m} \sum_{j=1}^{m} x^{(j)} x^{(j)^T} x^{(i)} a_i &= \lambda \sum_{i=1}^{m} a_i x^{(i)} \\ x^{(n)^T} \left(\sum_{i=1}^{m} \sum_{j=1}^{m} a_i x^{(j)} \langle x^{(j)}, x^{(i)} \rangle \right) &= \lambda \sum_{i=1}^{m} a_i x^{(n)^T} x^{(i)} \\ \sum_{i=1}^{m} \sum_{j=1}^{m} a_i \langle x^{(n)}, x^{(j)} \rangle \langle x^{(j)}, x^{(i)} \rangle &= \lambda \sum_{i=1}^{m} a_i x^{(n)^T} x^{(i)} \\ (K^2 a)_n &= \lambda (Ka)_n \\ Ka &= \lambda a \end{aligned}$$

□

14.3.2 Funkcje jądrowe

Możemy teraz skorzystać z powyższych dwóch lematów i zaproponować poniższy algorytm:

1. Wyznacz macierz Grama K i jej wektor własny a
2. Znormalizuj a aby $\|a\| = 1$
3. Oblicz $v = \frac{1}{\sqrt{\lambda}} \sum_{i=1}^{m} a_i x^{(i)}$

Problem jest taki, że my korzystamy z $x^{(i)}$ przy obliczaniu v – chcemy się tego pozbyć. Okazuje się, że nie musimy obliczać wektorów własnych per se, wystarczy nam znać projekcje zbioru danych na przestrzeń na nich rozpiętą

W przypadku jednowymiarowym mamy

$$v = \sum_{i=1}^{m} a_i x^{(i)}$$

zatem

$$\pi_v(x^{(i)}) = v^T x^{(i} = \sum_{j=1}^{m} a_j \langle x^{(j)}, x^i \rangle = \sum_{j=1}^{m} a_i \kappa(x^{(j)}, x^{(i)})$$

No i super – możemy korzystać z dowolnych funkcji jądrowych.

Przydałoby się jeszcze przeskalować i uśrednić dane w przestrzeni cech, ale robimy to dość prosto. Niech M będzie macierzą wypełnioną wartością $\frac{1}{m}$ a K macierzą Grama po zastosowaniu funkcji jądrowych.

Obliczamy

$$K \leftarrow K - MK - KM + MKM$$

po rozpisaniu tego mnożenia okaże się że istotnie dostaniemy K takie, jak gdyby wektory $\phi(x^{(i)})$ były uśrednione.

Algorytm jest bardzo podobny jak bez funkcji jądrowych tj.:

1. Obliczamy macierz Grama K
2. Środkujemy dane w przestrzeni cech
3. Wyznaczamy rozkład SVD $K = U\Sigma V^T$
4. Definiujemy V_r
$$V_r = \left[\frac{1}{\sqrt{\lambda_1}} v_1 \dots \frac{1}{\sqrt{\lambda_l}} v_l\right]$$
5. Rzutujemy dane z macierzy Grama na przestrzeń rozpiętą na wektorach z V_r
$$z^{(i)} = V_r^T K_i$$