

Kompaktowe reprezentacje binarne i modele generatywne

Jakub Zadrozny

Promotor: dr Rafał Nowak

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

6 września 2019

Chmura punktów – N trójek współrzędnych (x, y, z) , gdzie N – rozdzielczość chmury. Dalej zakładamy $N = 2048$.

Chmura punktów – N trójek współrzędnych (x, y, z) , gdzie N – rozdzielczość chmury. Dalej zakładamy $N = 2048$.

Zadanie

Chcemy efektywnie przechowywać i przetwarzać ogromne zbiory chmur punktów.

Chmura punktów – N trójek współrzędnych (x, y, z) , gdzie N – rozdzielczość chmury. Dalej zakładamy $N = 2048$.

Zadanie

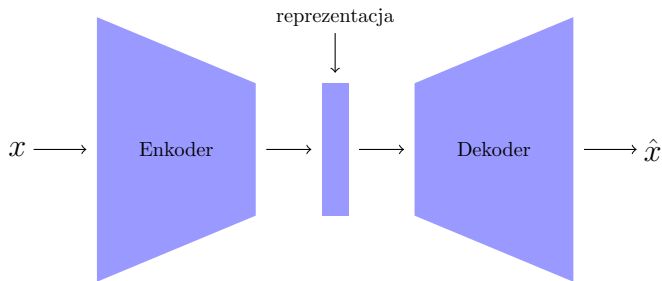
Chcemy efektywnie przechowywać i przetwarzać ogromne zbiory chmur punktów.

Rozwiązanie naiwne wymaga aż $2048 \times 3 \times 4B \approx 25KB$ pamięci na każdą chmurę!

Reprezentacją danych nazywamy przyporządkowanie danym pewnego wektora $z \in \mathbb{R}^d$, na podstawie którego potrafimy je odtworzyć (bez dodatkowych informacji).

Reprezentacją danych nazywamy przyporządkowanie danym pewnego wektora $z \in \mathbb{R}^d$, na podstawie którego potrafimy je odtworzyć (bez dodatkowych informacji).

Do znalezienia reprezentacji posiadanych (i potencjalnie nowych) danych możemy posłużyć się **autoenkoderem**:



Używając autoenkodera z odpowiednio dobraną architekturą i funkcją kosztu, potrafimy znaleźć satysfakcjonującą reprezentację chmur punktów w 128-wymiarowej przestrzeni.

Używając autoenkodera z odpowiednio dobraną architekturą i funkcją kosztu, potrafimy znaleźć satysfakcjonującą reprezentację chmur punktów w 128-wymiarowej przestrzeni.

Uwaga

128-wymiarowa reprezentacja zajmuje $128 \times 4B \approx 0.5KB$ pamięci, co daje ok. 50-krotną skalę kompresji danych.

Przechowywanie i przetwarzanie danych w takiej reprezentacji pozwala rozwiązać podstawowy problem narzutu pamięciowego i obliczeniowego.

Chcielibyśmy jednak opracować model, który pozwoliłby na:

Chcielibyśmy jednak opracować model, który pozwoliłby na:

- generowanie nowych, nieistniejących do tej pory danych,

Chcielibyśmy jednak opracować model, który pozwoliłby na:

- generowanie nowych, nieistniejących do tej pory danych,
- w tym generowanie danych podobnych do pewnej zadanej chmury,

Chcielibyśmy jednak opracować model, który pozwoliłby na:

- generowanie nowych, nieistniejących do tej pory danych,
- w tym generowanie danych podobnych do pewnej zadanej chmury,
- gładkie interpolowanie pomiędzy obiektami (stwierdzenie co jest *pomiędzy* obiektem A i B),

Chcielibyśmy jednak opracować model, który pozwoliłby na:

- generowanie nowych, nieistniejących do tej pory danych,
- w tym generowanie danych podobnych do pewnej zadanej chmury,
- gładkie interpolowanie pomiędzy obiektami (stwierdzenie co jest *pomiędzy* obiektem A i B),
- tworzenie obiektów na bazie posiadanych chmur (np. sklejenie połowy obiektu A i połowy B).

Nowe założenie

Dane są obserwacjami zmiennej losowej x , której rozkład jest postaci

$$f_{\theta}(z, x) = f(z)f_{\theta}(x|z),$$

gdzie z jest dodatkową zmienną ukrytą, a θ parametrami rozkładu.

Nowe założenie

Dane są obserwacjami zmiennej losowej x , której rozkład jest postaci

$$f_{\theta}(z, x) = f(z)f_{\theta}(x|z),$$

gdzie z jest dodatkową zmienną ukrytą, a θ parametrami rozkładu.

Dodatkowo zakładamy, że

$$\begin{aligned} z &\sim \mathcal{N}(0, I_d), \\ x|z &\sim \mathcal{N}(\mu_x(z; \theta), \mu_{\sigma}(z; \theta)I_k), \end{aligned}$$

gdzie μ_x , μ_{σ} są skomplikowanymi obliczeniami parametryzowanymi przez θ .

Zastosowanie do znalezienia reprezentacji

Reprezentacja: $f_{\theta}(z|x)$.

Rekonstrukcja: $\mu_x(z; \theta)$.

Zastosowanie do znalezienia reprezentacji

Reprezentacja: $f_{\theta}(z|x)$.

Rekonstrukcja: $\mu_x(z; \theta)$.

Problem

Nie znamy prawdziwych parametrów θ , nie możemy nawet efektywnie wyliczyć ich MLE. Ponadto nie potrafimy wyznaczyć rozkładu $f_{\theta}(z|x)$.

Zastosowanie do znalezienia reprezentacji

Reprezentacja: $f_{\theta}(z|x)$.

Rekonstrukcja: $\mu_x(z; \theta)$.

Problem

Nie znamy prawdziwych parametrów θ , nie możemy nawet efektywnie wyliczyć ich MLE. Ponadto nie potrafimy wyznaczyć rozkładu $f_{\theta}(z|x)$.

Rozwiązanie – autoenkoder wariacyjny

Wprowadzamy sparametryzowane przybliżenie rozkładu $f_{\theta}(z|x)$ i optymalizujemy dolne oszacowanie na prawdopodobieństwo wygenerowania obserwowanych danych (**ELBO**).

- 1 Enkoder wyznacza parametry rozkładu normalnego $g_\phi(z|x)$ przybliżającego $f_\theta(z|x)$.

Autoenkoder wariacyjny

- 1 Enkoder wyznacza parametry rozkładu normalnego $g_\phi(z|x)$ przybliżającego $f_\theta(z|x)$.
- 2 Z rozkładu $g_\phi(z|x)$ losuje się reprezentację z .

Autoenkoder wariacyjny

- 1 Enkoder wyznacza parametry rozkładu normalnego $g_\phi(z|x)$ przybliżającego $f_\theta(z|x)$.
- 2 Z rozkładu $g_\phi(z|x)$ losuje się reprezentację z .
- 3 Na podstawie wylosowanej z , dekodek wylicza $\mu_x(z; \theta)$, które przyjmowane jest za rekonstrukcję.

Autoenkoder wariacyjny

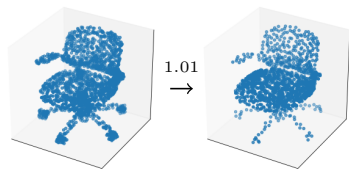
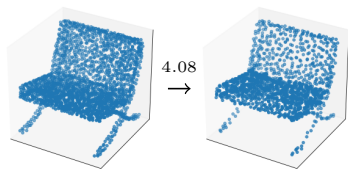
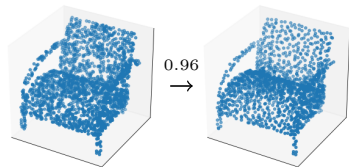
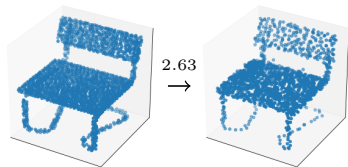
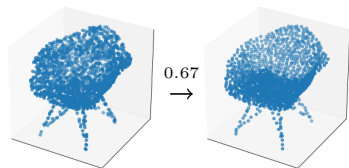
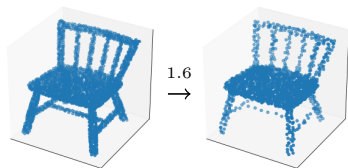
- 1 Enkoder wyznacza parametry rozkładu normalnego $g_\phi(z|x)$ przybliżającego $f_\theta(z|x)$.
- 2 Z rozkładu $g_\phi(z|x)$ losuje się reprezentację z .
- 3 Na podstawie wylosowanej z , dekodek wylicza $\mu_x(z; \theta)$, które przyjmowane jest za rekonstrukcję.

Enkoder: sieć neuronowa parametryzowana przez ϕ .

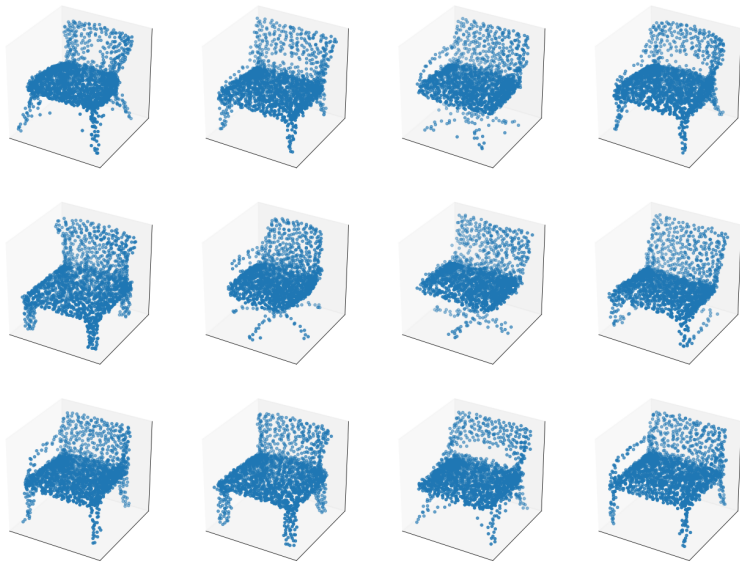
Dekoder: sieć neuronowa parametryzowana przez θ .

Parametry ϕ oraz θ optymalizuje się wspólnie, wariantem metody SGD z funkcją kosztu będącą zmodyfikowanym ELBO (dostosowuje się je do różniczkowania po parametrach ϕ oraz do pracy z chmurami punktów).

Rekonstrukcje



Syntetyczne dane



Model z mieszanką gaussowską

Nowe założenie

Posiadane dane zostały wygenerowane przez mieszankę rozkładów o C komponentach z jednakowymi wagami.

Nowe założenie

Posiadane dane zostały wygenerowane przez mieszankę rozkładów o C komponentach z jednakowymi wagami.

Wprowadzamy dodatkową zmienną dyskretną y i zakładamy, że

$$f_{\theta}(x, y, z) = f(y)f(z|y)f_{\theta}(x|z)$$

oraz

$$y \sim \text{Cat}(1/C),$$

$$z|y \sim \mathcal{N}(\mu_y, I_d),$$

$$x|z \sim \mathcal{N}(\mu_x(z; \theta), \mu_{\sigma}(z; \theta)I_k),$$

gdzie $\mu_y \in \mathbb{R}^d$ są dowolnymi średnimi komponentów mieszanki.

Przybliżamy $f_{\theta}(y, z|x)$ przez pewien rozkład $g_{\phi}(y, z|x)$ przy założeniu, że

$$g_{\phi}(y, z|x) = g_{\phi}(y|x)g_{\phi}(z|y, x),$$

gdzie $g_{\phi}(y|x)$ jest rozkładem dyskretnym, a $g_{\phi}(z|y, x)$ rozkładem normalnym.

Przybliżamy $f_{\theta}(y, z|x)$ przez pewien rozkład $g_{\phi}(y, z|x)$ przy założeniu, że

$$g_{\phi}(y, z|x) = g_{\phi}(y|x)g_{\phi}(z|y, x),$$

gdzie $g_{\phi}(y|x)$ jest rozkładem dyskretnym, a $g_{\phi}(z|y, x)$ rozkładem normalnym.

Wtedy rozkład brzegowy $g_{\phi}(z|x)$ jest mieszanką gaussowską z wagami $g_{\phi}(y|x)$.

Klasteryzacja

Przybliżamy $f_{\theta}(y, z|x)$ przez pewien rozkład $g_{\phi}(y, z|x)$ przy założeniu, że

$$g_{\phi}(y, z|x) = g_{\phi}(y|x)g_{\phi}(z|y, x),$$

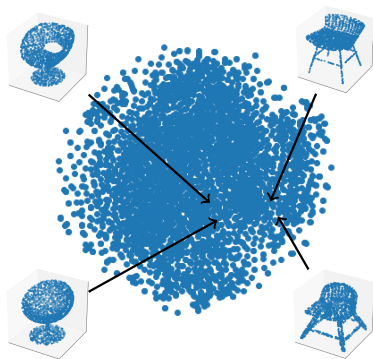
gdzie $g_{\phi}(y|x)$ jest rozkładem dyskretnym, a $g_{\phi}(z|y, x)$ rozkładem normalnym.

Wtedy rozkład brzegowy $g_{\phi}(z|x)$ jest mieszanką gaussowską z wagami $g_{\phi}(y|x)$.

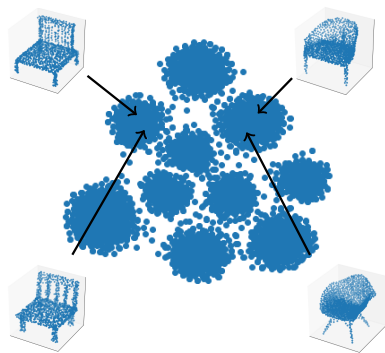
Uwaga

Rozkład $g_{\phi}(y|x)$ możemy interpretować jako prawdopodobieństwa przynależności do każdego z klastrów. Przyporządkowując próbki do klastrów z największym prawdopodobieństwem otrzymujemy naturalny podział danych na podkategorie.

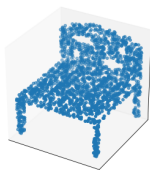
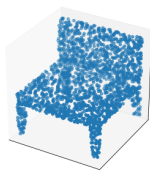
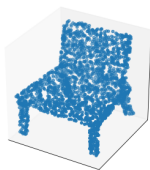
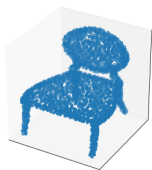
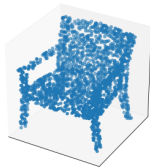
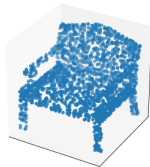
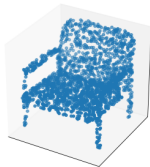
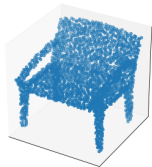
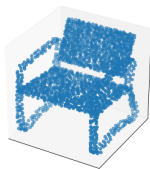
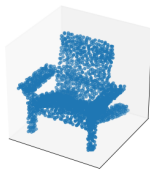
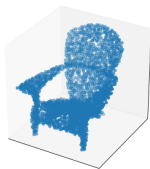
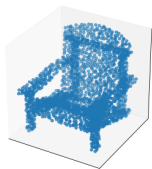
Przestrzeń reprezentacji



(a) VAE- \mathcal{N}



(b) VAE- \mathcal{GM}



Model z rozkładem Beta

Zakładamy, że

$$z = \begin{bmatrix} z_1 \\ \vdots \\ z_d \end{bmatrix}, \quad z_i \sim \text{Beta}(0.01, 0.01),$$

a zmienne z_i są niezależne.

Model z rozkładem Beta

Zakładamy, że

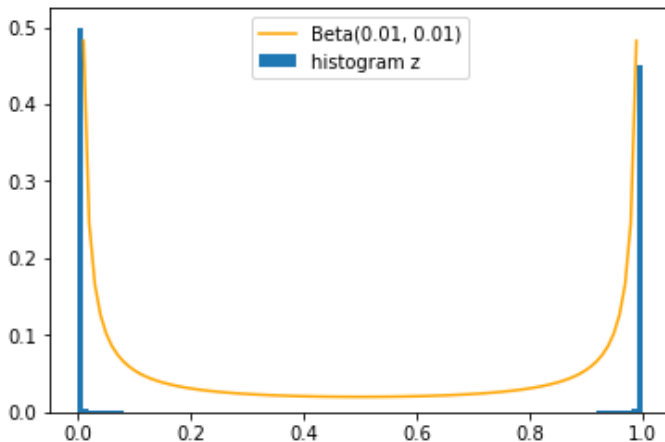
$$z = \begin{bmatrix} z_1 \\ \vdots \\ z_d \end{bmatrix}, \quad z_i \sim \text{Beta}(0.01, 0.01),$$

a zmienne z_i są niezależne.

Podobnie, niech $g_\phi(z|x)$ będzie iloczynem gęstości rozkładów $\text{Beta}(\alpha_i, \beta_i)$, gdzie

$$\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \rho_\alpha(x; \phi), \quad \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} = \rho_\beta(x; \phi).$$

Histogram wartości zmiennej ukrytej



Reprezentacją binarną nazywamy reprezentację, w której każdy wymiar zmiennej ukrytej przyjmuje wartość 0 lub 1.

Reprezentacją binarną nazywamy reprezentację, w której każdy wymiar zmiennej ukrytej przyjmuje wartość 0 lub 1.

Obserwacja

Możemy uzyskać reprezentację binarną dla chmur punktów 3D zaokrąglając reprezentację wylosowaną z rozkładu Beta do jedności (0 lub 1).

Reprezentacją binarną nazywamy reprezentację, w której każdy wymiar zmiennej ukrytej przyjmuje wartość 0 lub 1.

Obserwacja

Możemy uzyskać reprezentację binarną dla chmur punktów 3D zaokrąglając reprezentację wylosowaną z rozkładu Beta do jedności (0 lub 1).

W ten sposób uzyskujemy zaledwie 128-bitową reprezentację, gorszą od 128-wymiarowej ciągłej o niecałe 18%.
Taka reprezentacja zajmuje tylko $4B$, $32x$ mniej niż 128-wymiarowa reprezentacja ciągła i aż ok. $1500x$ mniej niż naiwna reprezentacja danych!

Rekonstrukcje binarne

