Dimensionality reduction is a powerful technique used by data scientists to look for hidden structure in data. The method is useful in a number of domains, for example document categorization, protein disorder prediction, and machine learning model debugging[2].

The results of a dimensionality reduction algorithm can be visualized to reveal patterns and clusters of similar or dissimilar data. Even though the data is displayed in only two or three dimensions, structures present in higher dimensions are maintained, at least roughly[7].

The technique is available in many applications, for example Google's let's you view high-dimensional datasets embedded in two or three dimensions under a variety of different projections.

This guide will teach you how to think about these embeddings, and provide a comparison of some of the most popular dimensionality reduction algorithms used today.

Redukcja wymiarów jest potężną techniką wykorzystywaną przez naukowców zajmujących się danymi do poszukiwania ukrytej struktury w danych. Metoda jest przydatna w wielu domenach, na przykład w kategoryzacji dokumentów, przewidywaniu zaburzeń białkowych i debugowaniu modelu uczenia maszynowego.

Wyniki algorytmu redukcji wymiarów można wizualizować w celu ujawnienia wzorców i grup podobnych lub odmiennych danych. Nawet jeśli dane są wyświetlane tylko w dwóch lub trzech wymiarach, struktury obecne w wyższych wymiarach są utrzymywane, przynajmniej z grubsza.

Technika ta jest dostępna w wielu aplikacjach, na przykład Google pozwala przeglądać wysokowydajne zbiory danych osadzone w dwóch lub trzech wymiarach w różnych rzutach.

Ten przewodnik nauczy Cię, jak myśleć o tych osadzeniach i przedstawić porównanie najpopularniejszych obecnie stosowanych algorytmów redukcji wymiarów.

2.

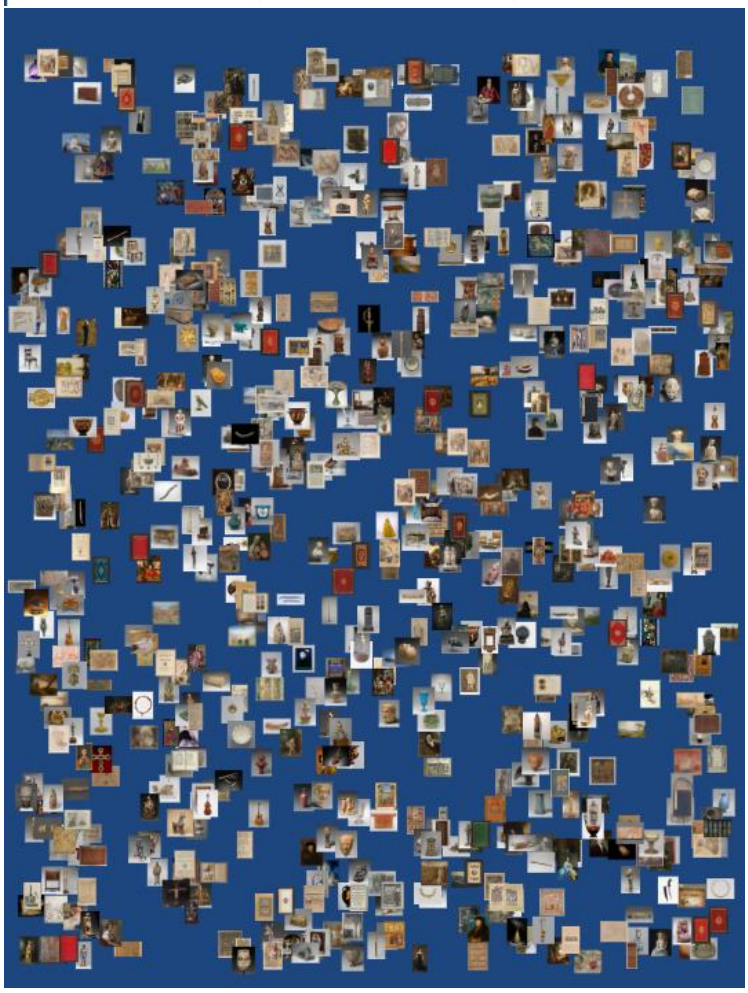
Your browser has just loaded information about roughly 800 artworks from the collection at the Metropolitan Museum of Art. The museum has publicly released a large dataset about their collection[5], just a small fraction are displayed here. They are positioned randomly.
Hover over an artwork to see its details.
Each artwork includes basic metadata, such as its title, artist, date made, medium, and dimensions. Data scientists like to call metadata for each data point (artwork) *features*. Below are some of the features of 10 artworks in the dataset.

Twoja przeglądarka właśnie załadowała informacje o około 800 pracach z kolekcji w Metropolitan Museum of Art. Muzeum publicznie opublikowało duży zbiór danych o ich kolekcji [5], tylko niewielki ułamek jest tutaj wyświetlany. Są rozmieszczone losowo.

Najedź na grafikę, aby zobaczyć jej szczegóły.

Każda grafika zawiera podstawowe metadane, takie jak tytuł, artysta, data, medium i wymiary. Dane naukowcy lubią wywoływać metadane dla każdego punktu danych (kompozycji). Poniżej znajdują się niektóre cechy 10 dzieł w zbiorze danych.

Year	Title	Artist
1680	Spindle-back armch...	
1875	Armchair	Pottier and Stymus ...
1776	Basket	Myer Myers
1650	Basin	Master Potter A
1710	Two-handled Bowl	Cornelius Kierstede
1876	The Bryant Vase	James Horton White...
1765	Bureau table	John Townsend
1880	Cabinet	Daniel Pabst
1866	Cabinet	Alexander Roux
1835	Celery vase	Boston & Sandwich ...



Projecting onto a line

These features can be thought of as vectors existing in a high-dimensional space. Visualizing the vectors would reveal a lot about the distribution of the data, however humans can't see so many dimensions all at once.

Instead the data can be projected onto a lower dimension, one that can be visualized directly. This kind of projection is called an *embedding*.

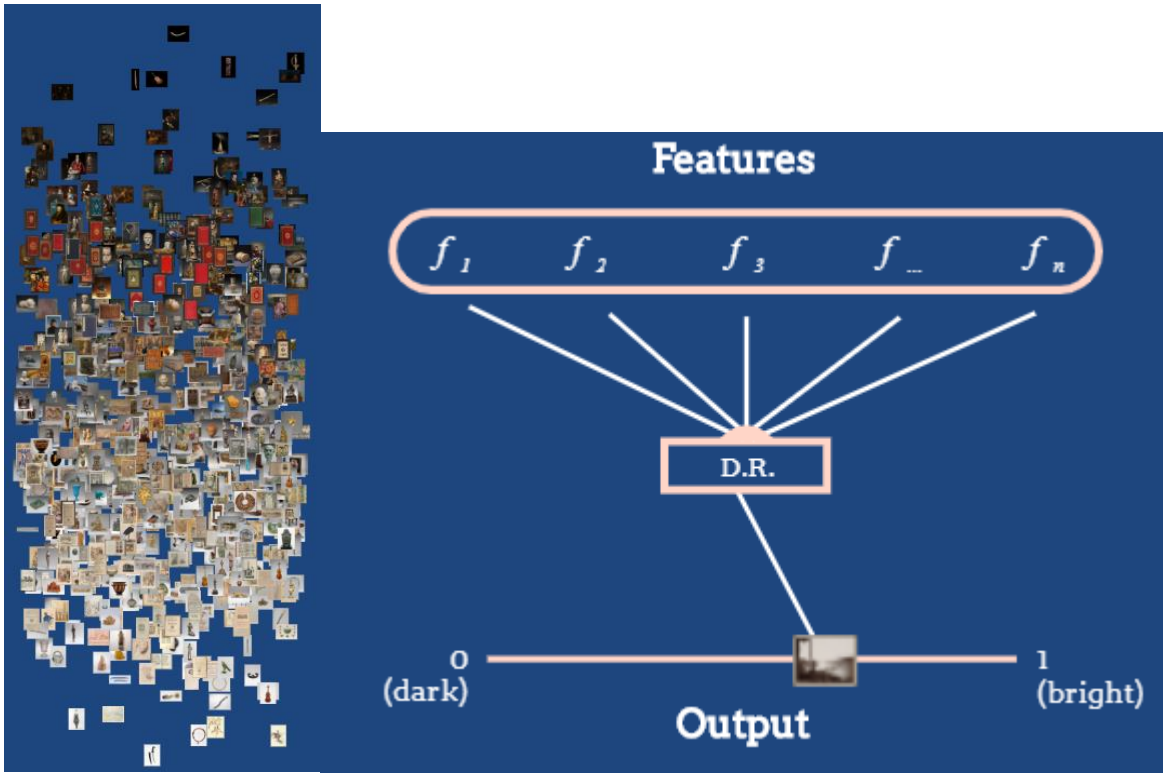
Computing a 1-dimensional embedding requires taking each artwork and computing a single number to describe it. A benefit of reducing to 1D is that the numbers, and the artworks, can be sorted on a line.

Wyświetlanie na linii

Cechy te mogą być uważane za wektory istniejące w przestrzeni o dużych wymiarach. Wizualizacja wektorów ujawniłaby wiele na temat dystrybucji danych, jednak ludzie nie mogą widzieć tak wielu wymiarów jednocześnie.

Zamiast tego dane mogą być rzutowane na niższy wymiar, który można bezpośrednio wizualizować. Ten rodzaj projekcji nazywany jest osadzaniem.

Obliczanie osadzania jednowymiarowego wymaga zabrania każdej grafiki i obliczenia pojedynczej liczby, aby ją opisać. Zaletą zmniejszenia do 1D jest to, że liczby i dzieła sztuki mogą być sortowane w linii.



For the mathematically inclined

Dimensionality reduction can be formulated mathematically in the context of a given dataset. Consider a dataset represented as a matrix X, where X is of size $m \times n$, where m represents the number of rows of X, and n representes the number of columns.

Typically, the rows of the matrix are *data points* and the columns are *features*. Dimensionality reduction will reduce the number of features of each data point, turning X into a new matrix, X', of size $m \times d$, where $d < n$. For visualizations we typically set d to be 1, 2 or 3.

Say $m=n$, that is X is a square matrix. Performing dimensionality reduction on X and setting $d=2$ will change it from a square matrix to a tall, rectangular matrix.

Dla skłonnych matematycznie

Redukcję wymiarów można sformułować matematycznie w kontekście danego zbioru danych. Rozważmy zbiór danych reprezentowany jako macierz XX, gdzie XX jest sizem razy nm × n, gdzie mm oznacza liczbę wierszy XX, ann reprezentuje liczbę kolumn.

Zazwyczaj wiersze macierzy są punktami danych, a kolumny są cechami. Redukcja wymiarów zmniejszy liczbę cech każdego punktu danych, zamieniając XX w nową macierz, X'X ', o rozmiarze m razy dm × d, gdzie d <nd <n. W przypadku wizualizacji zazwyczaj ustawiamy dd na 1, 2 lub 3.

Powiedz m = nm = n, czyli XX jest kwadratową macierzą. Wykonanie redukcji wymiarowości na XX i ustawienie d = 2d = 2 zmieni ją z kwadratowej macierzy na wysoką prostokątną matrycę.

$$X = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \Longrightarrow \begin{bmatrix} x' & x' \\ x' & x' \\ x' & x' \end{bmatrix} = X'$$

Each data point only has two features now, i.e., each data point has been reduced from a 3 dimensional vector to a 2 dimensional vector.

Każdy punkt danych ma teraz tylko dwie cechy, tj. Każdy punkt danych został zredukowany z wektora trójwymiarowego do wektora dwuwymiarowego.

5.

Embedding data in two dimensions

The same brightness feature can be used to position the artworks in 2D space instead of 1D. The pieces have more room to spread out.

On the right you see a simple 2-dimensional embedding based on image brightness, but this isn't the only way to position the artworks. In fact, there are many, and some projections are more useful than others.

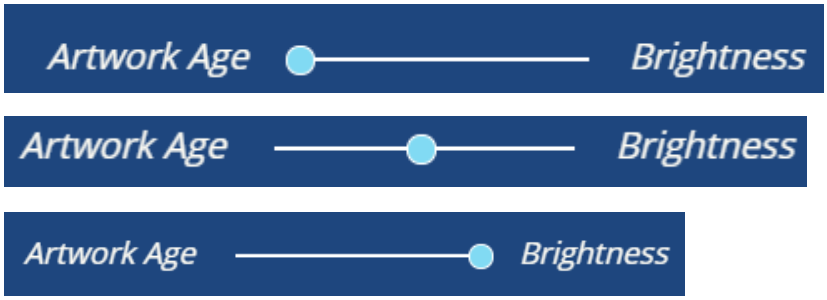
*Use the slider to vary the influence that the brightness and artwork age have in determining the embedding positions.*As you move the slider from brightness to artwork age, the embedding changes from highlighting bright and dark images, and starts to cluster recent modern-day images in the bottom left corner whereas older artworks are moved farther away (*hover over images to see their date*).

Osadzanie danych w dwóch wymiarach

Ta sama funkcja jasności może być używana do pozycjonowania dzieł sztuki w przestrzeni 2D zamiast 1D. Kawałki mają więcej miejsca do rozłożenia.

Po prawej stronie widać proste dwuwymiarowe osadzanie oparte na jasności obrazu, ale nie jest to jedyny sposób na umieszczenie dzieł sztuki. W rzeczywistości jest ich wiele, a niektóre projekcje są bardziej przydatne niż inne.

Użyj suwaka, aby zmienić wpływ jasności i wieku kompozycji na określenie pozycji osadzania. Przesuwając suwak z jasności do wieku grafiki, zmiany osadzania z podświetlania jasnych i ciemnych obrazów i zaczynają grupować najnowsze obrazy współczesne w lewym dolnym rogu, podczas gdy starsze dzieła są przenoszone dalej (najedź kursorem na zdjęcia, aby zobaczyć ich datę).



The embedding you see here is actually a linear 1D embedding, whose resulting scalar is then mapped on a space-filling Hilbert curve[1] to give the illusion of a 2D embedding, since space-filling curves preserve locality fairly well[6].

Each artwork's 1D reduced projection is computed by a linear combination of the three features above.

Let a be a given artwork, and let each slider's value be a weight w_i. We will compute a', the scalar projection of a R.

Osadzenie, które tutaj widzisz, jest w rzeczywistości liniowym osadzeniem 1D, którego wynikowy skalar jest następnie mapowany na wypełniającej przestrzeń krzywej Hilberta [1], aby dać złudzenie osadzania 2D, ponieważ krzywe wypełniające przestrzeń dość dobrze zachowują lokację [6].

Obniżona projekcja każdej grafiki 1D jest obliczana przez liniową kombinację trzech powyższych funkcji.

Pozwól, aby była daną kompozycją, i niech wartość każdego suwaka będzie wagą wi. Obliczymy ', skalarną projekcję R.

$$a' = (a_{\text{brightness}} \times w_{\text{brightness}}) + (a_{\text{age}} \times w_{\text{age}})$$

Ostateczna pozycja każdego dzieła jest losowo pomijana, aby zapobiec nadmiernemu nakładaniu się.

Each artwork's final position is randomly jittered to prevent excessive overlap.

6.

Real-world algorithms

The previous section showed an example of a user-driven embedding, where the exact influence of each feature is known. However, you may have noticed that it's hard to find meaningful combinations of feature weights.

State-of-the-art algorithms can find an optimal combination of features so that distances in the high dimensional space are preserved in the embedding. Use the tool below to project the artworks using three commonly used algorithms.

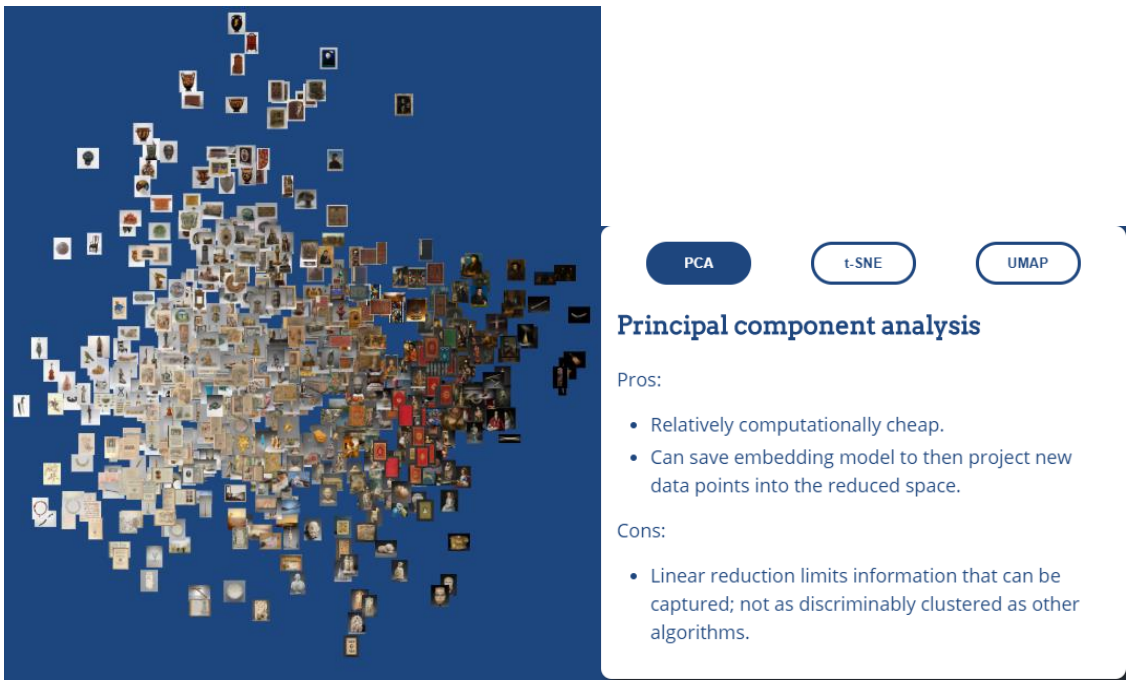
In this example the reduction is performed on the pixels of each image: each image is flattened into a single vector, where each pixel represents one feature. The vectors records are then reduced to two dimensions.

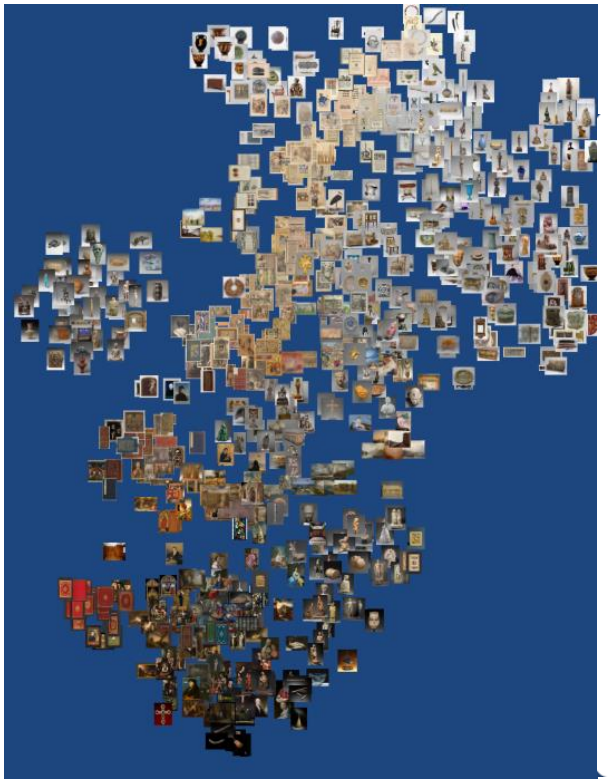
Algorytmy ze świata rzeczywistego

W poprzedniej części przedstawiono przykład osadzania sterowanego przez użytkownika, w którym znany jest dokładny wpływ każdej funkcji. Być może zauważyłeś, że trudno jest znaleźć sensowne kombinacje wag obiektów.

Najnowocześniejsze algorytmy mogą znaleźć optymalną kombinację funkcji, dzięki czemu odległości w przestrzeni o dużych wymiarach zostaną zachowane w osadzaniu. Użyj poniższego narzędzia do projekcji dzieł przy użyciu trzech powszechnie stosowanych algorytmów.

W tym przykładzie redukcja jest przeprowadzana na pikselach każdego obrazu: każdy obraz jest spłaszczony w pojedynczy wektor, gdzie każdy piksel reprezentuje jedną cechę. Rekordy wektorów są następnie redukowane do dwóch wymiarów.





PCA | t-SNE | UMAP

t-Distributed stochastic neighbor embedding

Pros:

- Produces highly clustered, visually striking embeddings.
- Non-linear reduction, captures local structure well.

Cons:

- Global structure may be lost in favor of preserving local distances.
- More computationally expensive.
- Requires setting hyperparameters that influence quality of the embedding.
- Non-deterministic algorithm.

PCA | t-SNE | UMAP

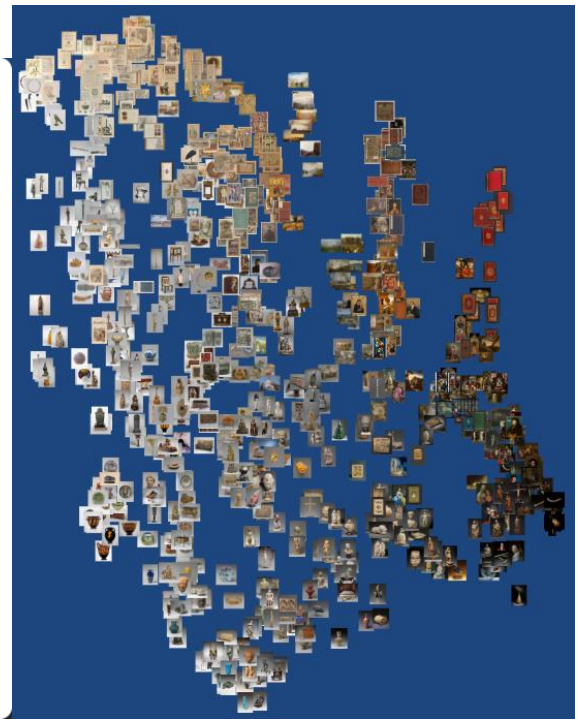
Uniform manifold approximation and projection

Pros:

- Non-linear reduction that is computationally faster than t-SNE.
- User defined parameter for preserving local or global structure.
- Solid theoretical foundations in manifold learning.

Cons:

- New, less prevalent algorithm.
- Requires setting hyperparameters that influence quality of the embedding.
- Non-deterministic algorithm.



There are many algorithms that compute a dimensionality reduction of a dataset. Simpler algorithms such as principal component analysis (PCA) maximize the variance in the data to produce the best possible embedding. More complicated algorithms, such as t-distributed stochastic neighbor embedding (t-SNE)[2], iteratively produce highly clustered embeddings. Unfortunately, whereas before the influence of each feature was explicitly

known, one must relinquish control to the algorithm to determine the best embedding— that means that it is not clear what features of the data are used to compute the embedding. This can be problematic for misinterpreting what an embedding is showing[11].

Dimensionality reduction, and more broadly the field of unsupervised learning, is an active area of research where researchers are developing new techniques to create better embeddings. A new technique, uniform manifold approximation and projection (UMAP)[4], is a non-linear reduction that aims to create visually striking embeddings fast, scaling to larger datasets.

Istnieje wiele algorytmów obliczających redukcję wymiarów zestawu danych. Prostsze algorytmy, takie jak analiza głównych składowych (PCA), maksymalizują wariancję danych w celu uzyskania najlepszego możliwego osadzenia. Bardziej skomplikowane algorytmy, takie jak t-rozproszone stochastyczne osadzanie sąsiadów (t-SNE), wytwarzają iteracyjnie wysoce skupione osadzenia. Niestety, chociaż zanim wpływ każdej funkcji został wyraźnie określony, należy zrezygnować z kontroli nad algorytmem, aby określić najlepsze osadzenie - co oznacza, że nie jest jasne, jakie cechy danych są używane do obliczenia osadzenia. Może to być problematyczne dla błędnej interpretacji tego, co pokazuje osadzanie.

Redukcja wymiarów, a szerzej dziedzina uczenia bez nadzoru, jest aktywnym obszarem badań, w którym naukowcy opracowują nowe techniki tworzenia lepszych osadzeń. Nowa technika, jednolita aproksymacja i projekcja rozmaitości (UMAP), to nieliniowa redukcja, która ma na celu stworzenie wizualnie uderzających osadzeń szybko, skalowanie do większych zbiorów danych.