MM.SVM.K1 Support Vector Machine

**Opis**

Zaimplementować klasyfikator oparty na modelu maszyny wektorów nośnych (SVM) w wersji liniowej i nieliniowej (kernel: poly/rbf/sigmoid). Porównać wyniki dla strategii one-vs-one i one-vs-all do klasyfikacji zbioru ręcznie pisanych cyfr. Zbiór danych do użycia: MNIST - http://yann.lecun.com/exdb/mnist/. Uzyskane rezultaty porównać z wynikami dla wybranej implementacji algorytmu ML (machine learning) z dostępnych bibliotek np. Scikit-learn, WEKA, MLlib, Tensorflow/Keras etc.

**Wypunktowany opis zadania**

Zadanie

1. Zaimplementować algorytm klasyfikacji **liniowej i nieliniowej** dla SVM.
2. **Porównać wyniki dla strategii one-vs-one** (sprawdzamy czy należy do klasy czy nie należy do klasy – predict one class from another class) i **one-vs-all** (sprawdzanie czy należy do jednej danej klasy czy do pozostałych wielu klas) do klasyfikacji zbioru ręcznie pisanych cyfr.

Porównanie one-vs-one i one-vs-all : <https://www.youtube.com/watch?v=_s3z8dQX3pM>

1. Uzyskane rezultaty **porównać z wynikami dla wybranej implementacji** algorytmu ML (machine learning) z dostępnych **bibliotek** np. Scikit-learn, WEKA, MLlib, Tensorflow/Keras.

Do wyświetlania :

1. Algorytm liniowy :
   1. Ze strategią one-one
   2. Ze strategia one-all
2. Algorytm nieliniowy:
   1. Gaussa :
      1. Ze strategia one-one
      2. Ze strategia one-all
   2. Wielomianowy
      1. Ze strategia one-one
      2. Ze strategia one-all

Zrobić w dokumentacji tabelę z porównanymi wynikami dla jednego algorytmu wraz z biblioteką WEKA.

WEKA – dlatego, że ona została napisana w Javie i porównanie z nią będzie miało logiczny sens.

Pozostałe biblioteki są napisane w Pythonie.

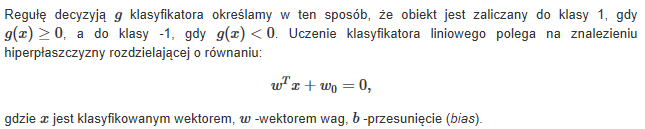
**SVM**

In machine learning, support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

**Liniowy klasyfikator SVM**

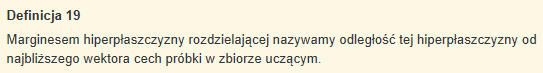
Rozważmy zadanie klasyfikacji dla dwóch klas, oznaczonych etykietami -1 oraz 1, zatem niech dany będzie zbiór uczący postaci:

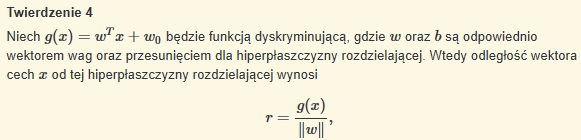


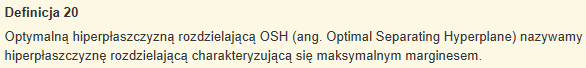


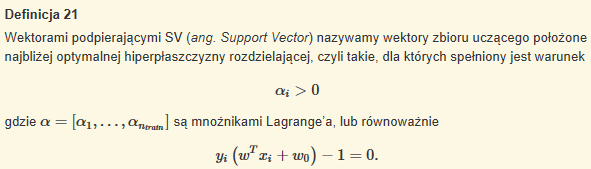
**Przypadek liniowo separowalny**

Rozważanie przypadku liniowo separowalnego gwarantuje istnienie hiperpłaszczyzny poprawnie rozdzielającej wszystkie próbki zbioru uczącego. Wśród nieskończenie wielu rozwiązań za *optymalną* w sensie SVM uważamy hiperpłaszczyznę rozdzielającą maksymalizującą tak zwany *margines*. Pojęcie marginesu i *optymalnej hiperpłaszczyzny rozdzielającej* definiujemy poniżej:

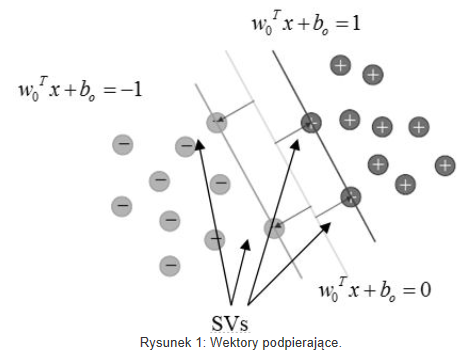


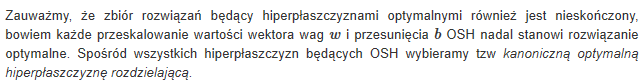


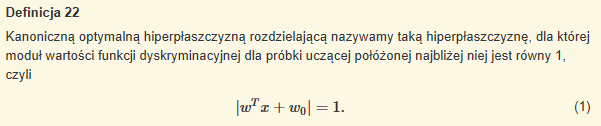


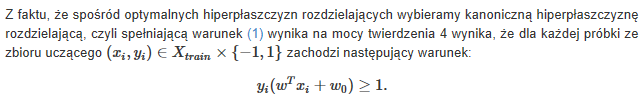


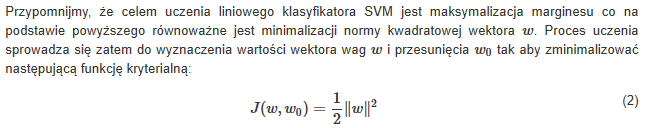
Położenie optymalnej hiperpłaszczyzny rozdzielającej, oraz umiejscowienie wektorów podpierających ilustruje rysunek 1









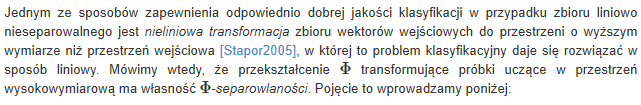


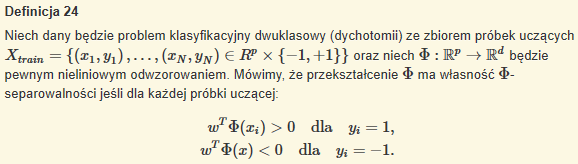




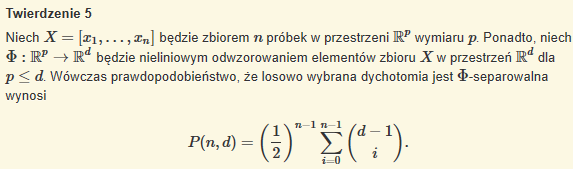
**Nieliniowy klasyfikator SVM**

Zaletą liniowych metod klasyfikacji jest prostota uzyskiwanego modelu. Niestety w przypadku rzeczywistych zbiorów danych rzadko kiedy mamy do czynienia z liniową separowalnością próbek w przestrzeni cech. W takich przypadkach stosowanie liniowych metod klasyfikacji nie daje satysfakcjonujących wyników z racji tego, że powierzchnie rozdzielające klasy w przypadku większości rzeczywistych zbiorów danych mają charakter nieliniowy. Na znalezienie takich powierzchni rozdzielających pozwala klasyfikator nieliniowy.





Uzasadnieniem dla tego podejścia jest poniższe twierdzenie Covera (por. [Ripley1996]).



Dokładny opis : <http://books.icse.us.edu.pl/runestone/static/ai/MaszynyWektorowPodpierajacych/NieliniowyKlasyfikatorSVM.html>