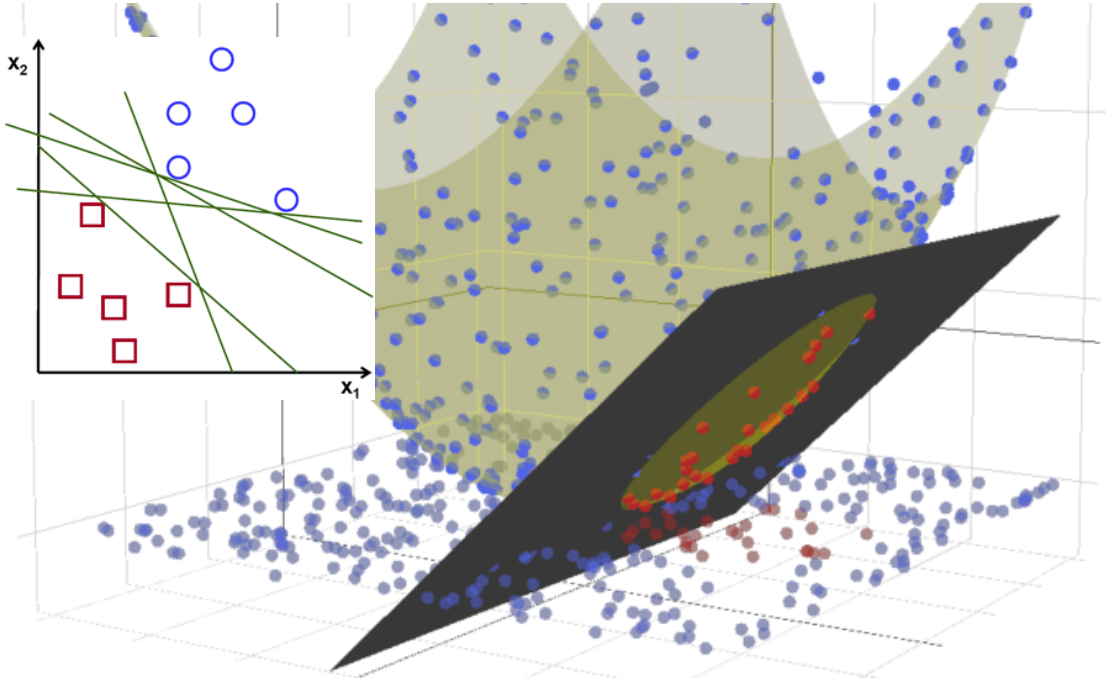
Machine learning

Lecture 4



Marcin Wolter

IFJ PAN

7 kwietnia 2017

- Support Vector Machines (SVM).
- Omówienie zadania zaliczeniowego.

Support Vector Machines

Główne założenie:

należy zbudować model używając **minimalną liczbę wektorów z danych treningowych (Support Vectors).**

Przestrzeń:

może modelować dowolną funkcję.

Funkcjonalnie algorytm podobny do sieci neuronowej, metod jądrowych Parzena itp.

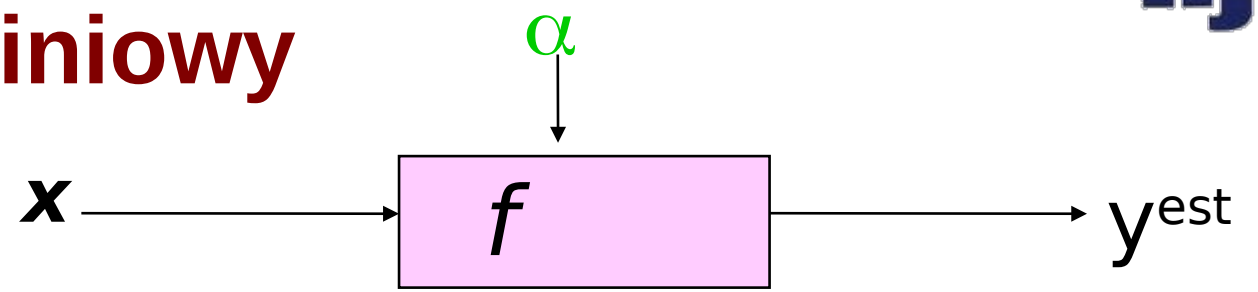
Trochę historii

Wczesne lata sześćdziesiąte – została opracowana metoda “support vectors” w celu konstruowania hiperpłaszczyzn do rozpoznawania obrazu (Vapnik i Lerner 1963, Vapnik i Czervonenkis 1964) – liniowa SVM.

Początek lat 1990-siątych: uogólnienie metody pozwalające na konstruowanie nieliniowych funkcji separujących (Boser 1992, Cortes i Vapnik 1995).

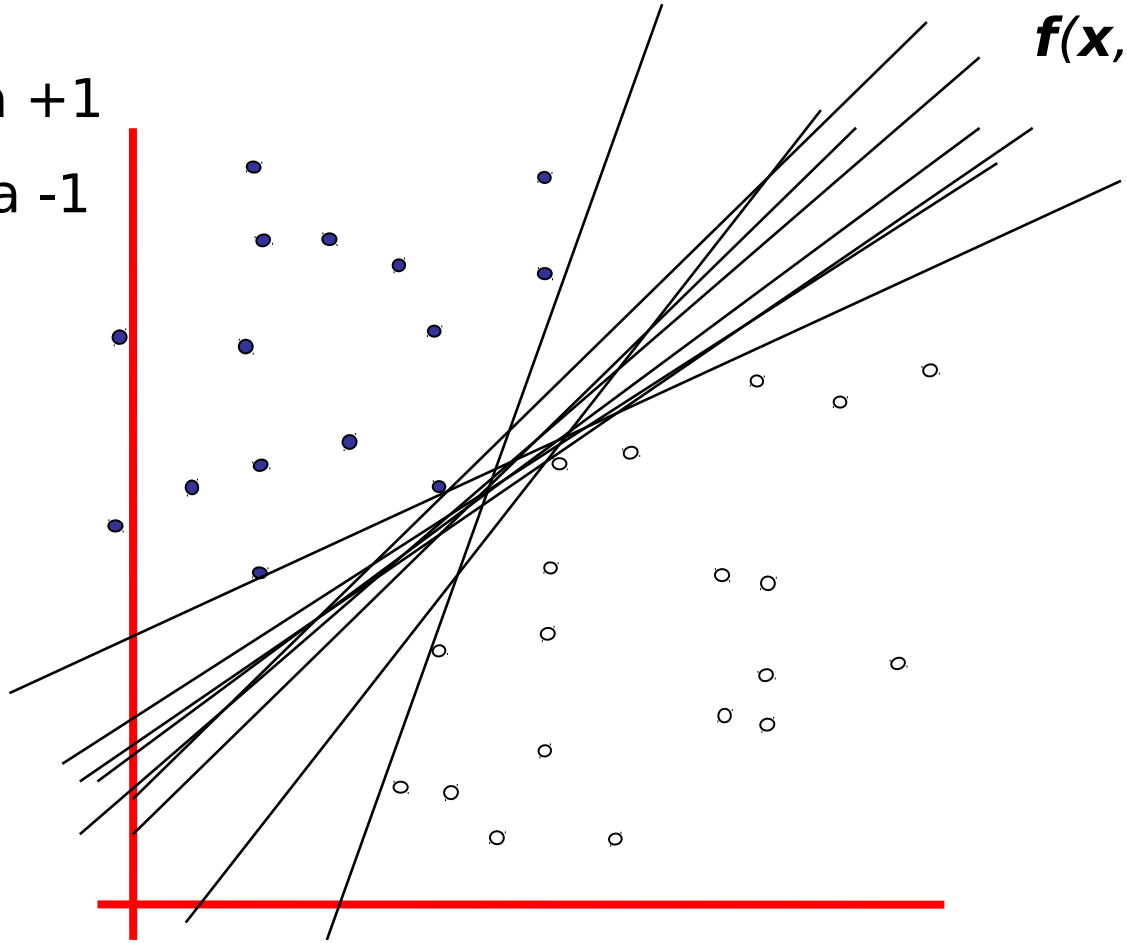
1995: dalsze rozszerzenie pozwalające otrzymać estymację funkcji ciągłej na wyjściu – regresja (Vapnik 1995).

Klasyfikator Liniowy



$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}*\boldsymbol{x} - b)$

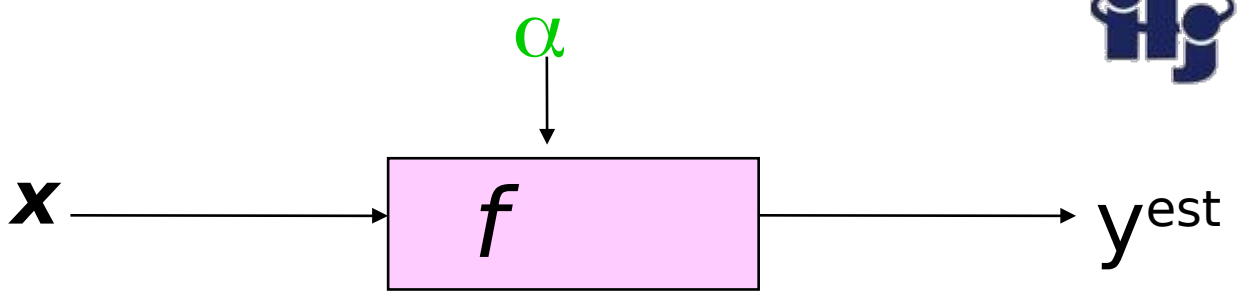
- oznacza +1
- oznacza -1



Każda z tych linii (klasyfikatorów) jest dobra...

...ale która najlepiej oddziela sygnał od tła?

Ta z największym marginesem!



$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}\bullet\boldsymbol{x} - b)$

"Maximum margin linear classifier"

Najprostszy rodzaj SVM, zwany LSVM (liniowa SVM)

• oznacza $y_i=+1$

○ oznacza $y_i=-1$

Support Vectors

-punkty ograniczające margines, czyli te na których on się wspiera.

$$\forall_i \quad y_i(\vec{w}\cdot\vec{x}_i+b)-1\geqslant 0$$

$\vec{w}\cdot\vec{x}+b=0$ *równanie prostej*

$$margin=\frac{2}{|\vec{w}|}$$

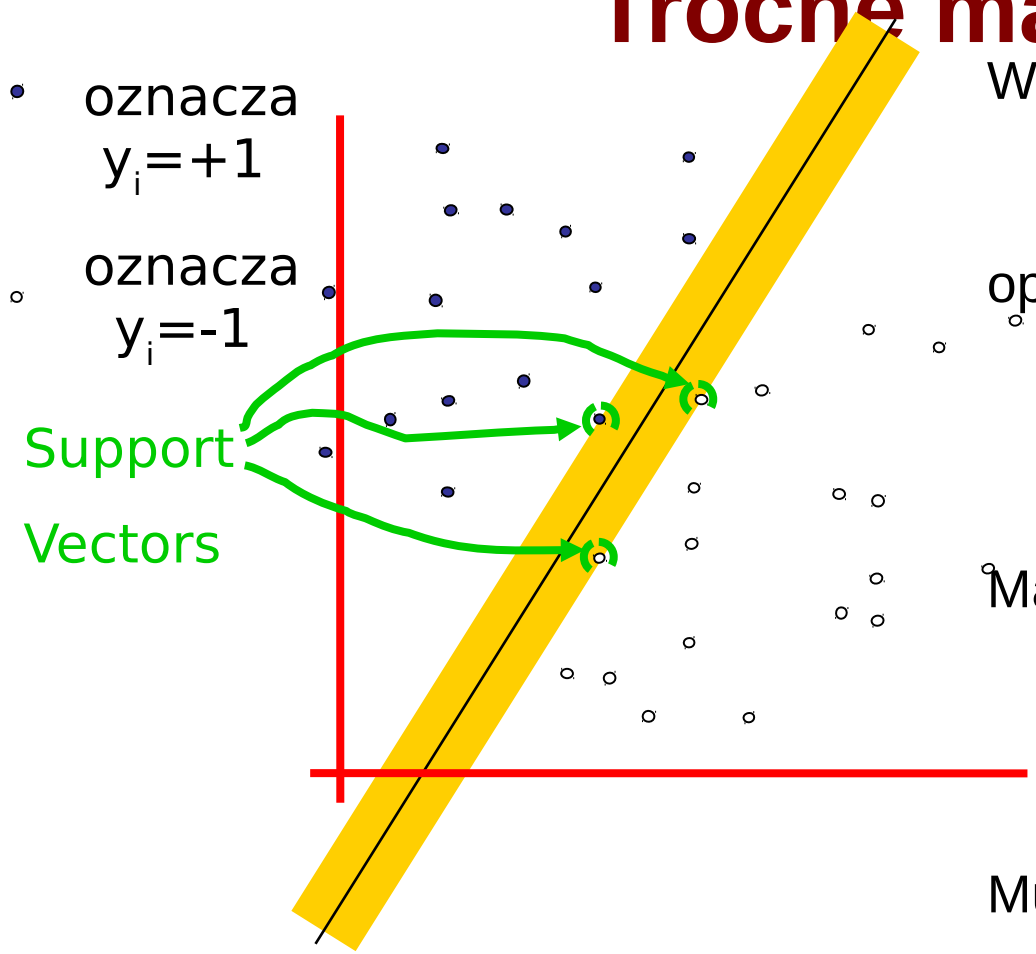
Intuicyjnie najlepiej.

Nieczuły na błędy w położeniu klasyfikatora.

Separacja zależy tylko od wektorów wspierających.

Działa w praktyce!

Troche matematyki



Warunek jaki musi spełniać prosta:

$$\forall_i \quad y_i(\vec{w}\cdot\vec{x}_i+b)-1\geqslant 0$$

opisana wzorem:

$$\vec{w}\cdot\vec{x}+b=0$$

Margines określony jest wzorem:

$$margin=\frac{2}{|\vec{w}|}$$

Musimy maksymalizować margines, czyli minimalizować $|\mathbf{w}|^2$.

Klasyfikator zależy tylko od części danych – **wektorów wspierających**.
Używamy tylko podzbioru wszystkich danych aby zoptymalizować separację.

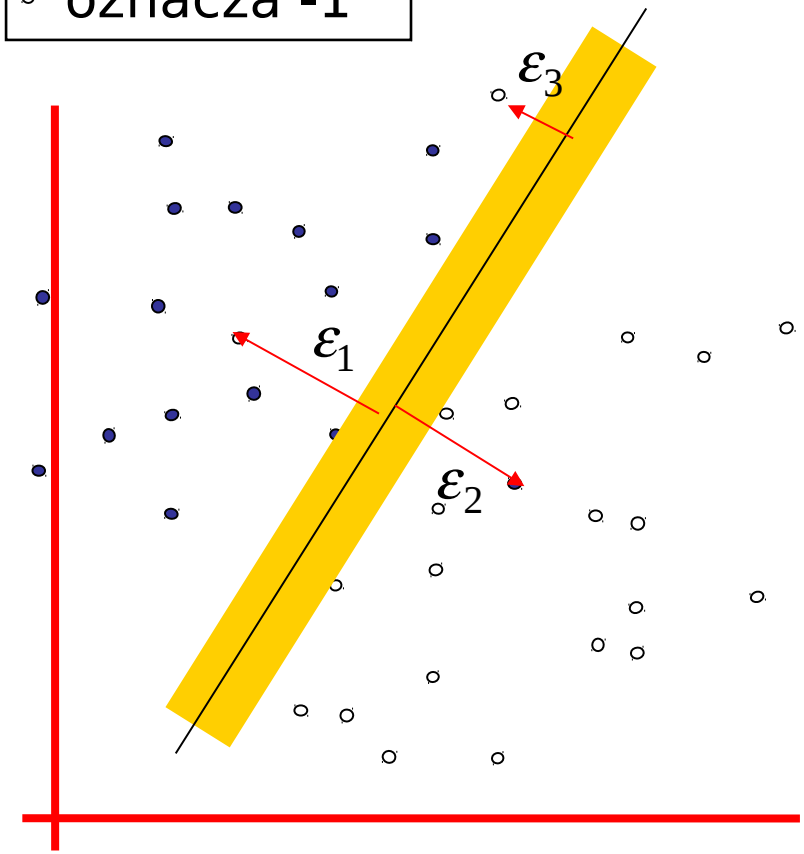
Co zrobić jeśli dane nie są separowalne?

Dodajemy dodatkowy człon do naszych równań (“slack variable”):

$\xi_i = 0$ x_i *poprawnie sklasyfikowane*

$\xi_i =$ *odległość* x_i *sklasyfikowane niepoprawnie*

- oznacza +1
- oznacza -1



I uzyskujemy:

$$\forall_i \quad y_i(\vec{w}\cdot\vec{x}_i + b) - 1 + C\,\xi_i \geqslant 0$$

Dla klasyfikatora liniowego (prostej)

$$\vec{w}\cdot\vec{x} + b = 0$$

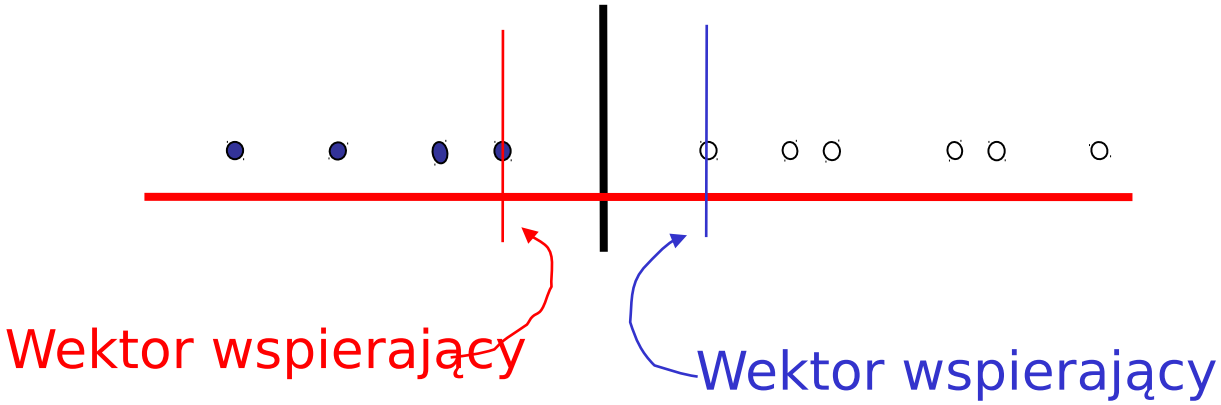
$$\textit{minimalizuj}: \frac{1}{2}|\vec{w}|^2 + C\sum_i \xi_i$$

gdzie C jest arbitralnym parametrem.

Funkcja straty jest liniowa, nie kwadratowa!

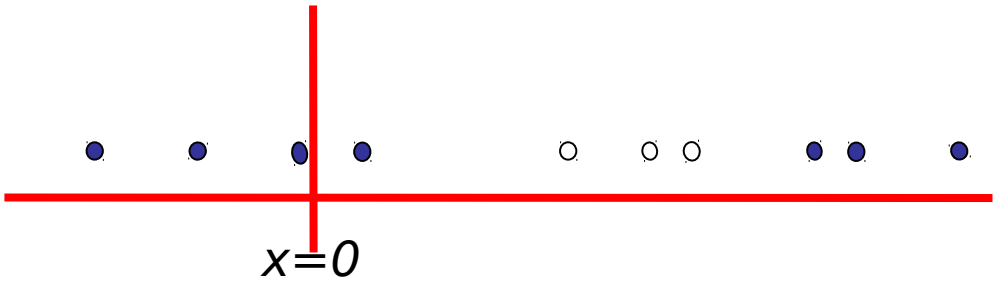
Przykład jednowymiarowy

Super łatwe!!!



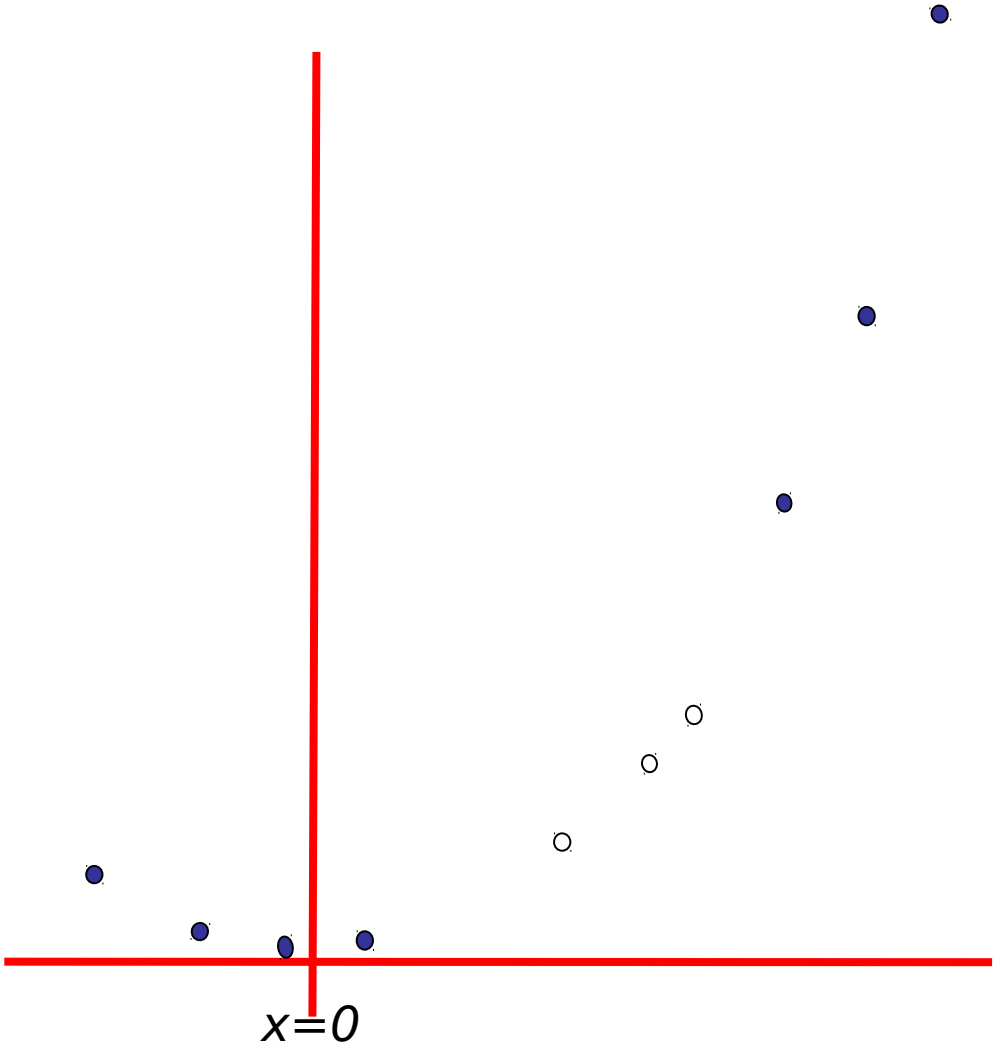
Trudniejszy przykład jednowymiarowy

Te dane nie są liniowo separowalne!



???

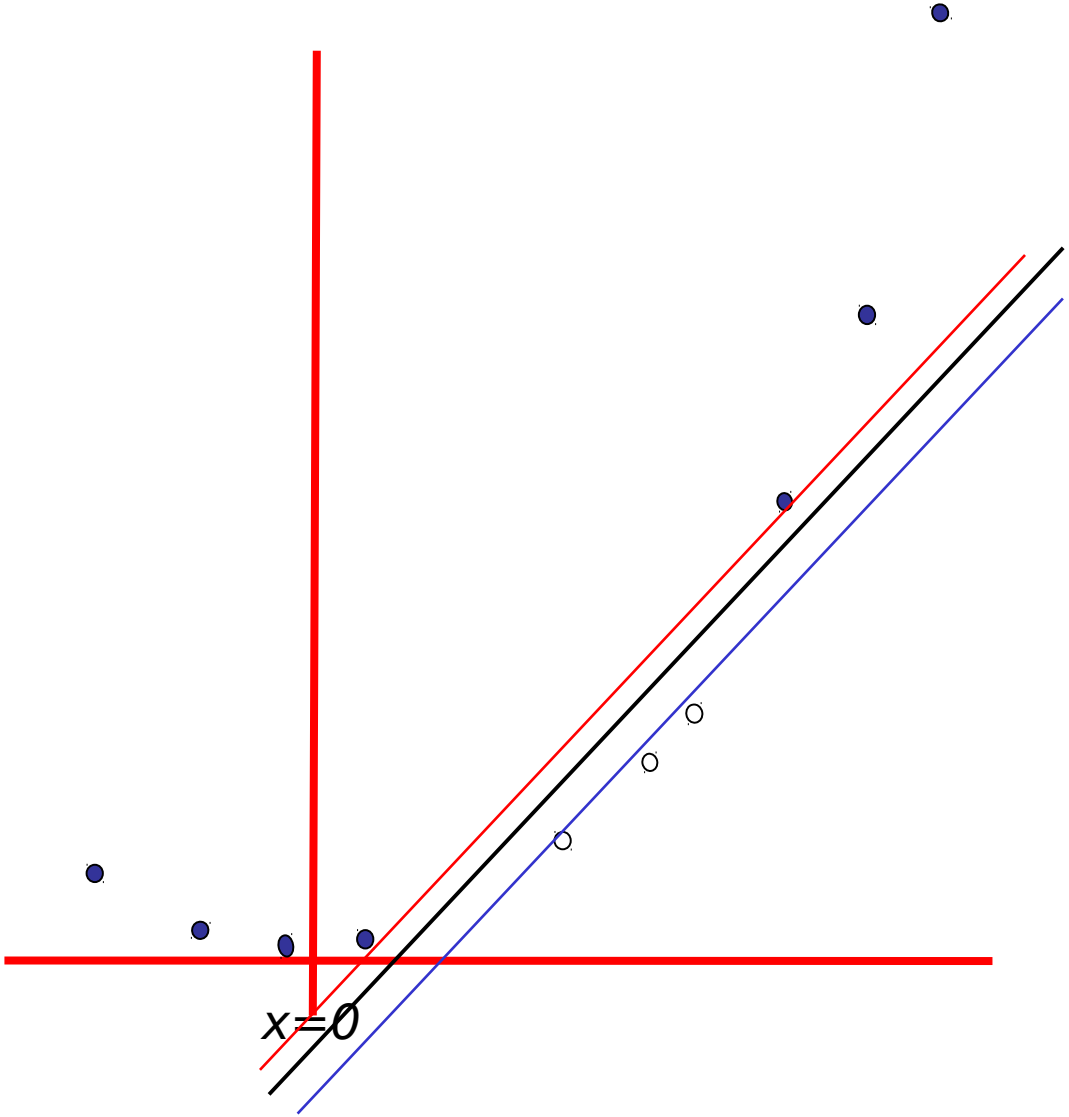
A może przejść do przestrzeni 2-wymiarowej?



$x \rightarrow (x, x^2)$

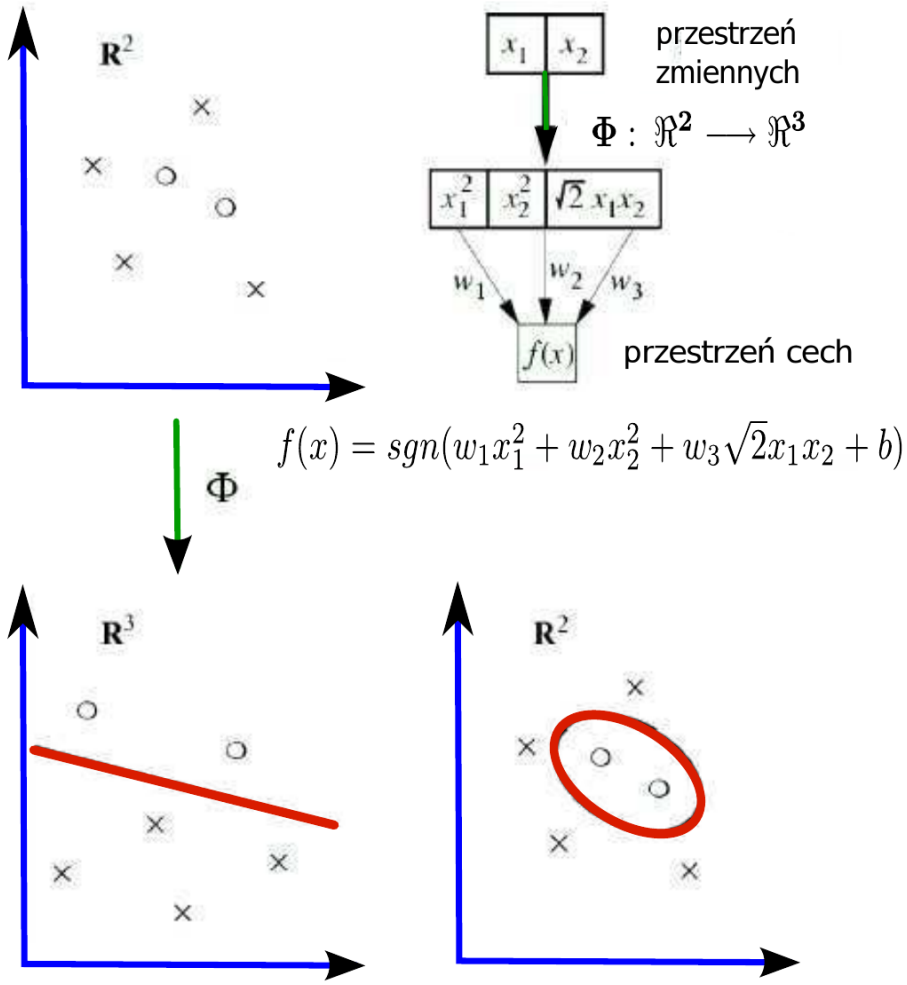
W tej przestrzeni punkty są liniowo separowalne!

W przestrzeni 2-wymiarowej



Teraz mamy separację

Jeśli dane nie są liniowe



$$f(x) = sgn(w_1 x_1^2 + w_2 x_2^2 + w_3 \sqrt{2} x_1 x_2 + b)$$

Należy dokonać transformacji do przestrzeni o większej liczbie wymiarów, gdzie dane będą liniowo separowalne.

W naszym przykładzie dane separowalne przez krzywą eliptyczną w R^2 będą separowalne liniowo w R^3.

Potrzebna jest uniwersalna metoda transformacji do przestrzeni o większej liczbie wymiarów.

Mnożniki Lagrange'a *(przypomnienie)*

Mnożniki Lagrange’a – metoda obliczania ekstremum warunkowego funkcji różniczkowalnej wykorzystywana w teorii optymalizacji. Dla przypadku dwuwymiarowego problem optymalizacji polega na maksymalizacji L(x, y) przy spełnieniu warunku G(x, y)= 0. W tym celu wprowadzamy nową zmienną α zwaną mnożnikiem Lagrange’a oraz budujemy funkcję pomocniczą:

$F(x, y, \alpha) = L(x, y) + \alpha G(x, y)$.

Wszystkie punkty, które mogą być ekstremami warunkowymi są rozwiązaniami układu równań:

$$\begin{cases} \frac{\partial F(x,y,\alpha)}{\partial x} & = 0 \\ \frac{\partial F(x,y,\alpha)}{\partial y} & = 0 \\ G(x,y) & = 0 \end{cases} .$$

Problem *n* zmiennych z *k* więzami jest redukowany do problemu *n+k* zmiennych bez więzów.

Trochę matematyki

Wprowadzamy Lagrangian i mnożniki Lagrange'a:

$$L(\vec{w},b,\vec{\alpha}) = 1/2|w|^2 - \sum_i \alpha_i \underbrace{(y_i[\langle \vec{w},\vec{x}\rangle + b] - 1)}$$

$$\alpha_i \geq 0$$

$$y_i[\langle \vec{w},\vec{x}\rangle + b] - 1 \geq 0 \quad \textit{nasze więzy}$$

Lagrangian musi być minimalizowany ze względu na **w** i b oraz maksymalizowany ze względu na α_i.

$$y_i[\langle \vec{w},\vec{x}\rangle + b] - 1 > 0 \;\rightarrow\; \alpha_i = 0 \quad (\textit{bez znaczenia})$$

$$y_i[\langle \vec{w},\vec{x}\rangle + b] - 1 = 0 \qquad \textit{support vectors}$$

w punkcie ekstremum:

$$\frac{\partial L}{\partial b} = 0 \;,\; \frac{\partial L}{\partial \vec{w}} = 0$$

$$\rightarrow \sum_i \alpha_i y_i = 0 \quad \vec{w} = \sum_i \alpha_i y_i \vec{x}_i$$

podstawiamy do L i mamy maksymalizację funkcji W(α) w przestrzeni α_i

$$W(\vec{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \vec{x}_i, \vec{x}_j \rangle$$

$$\alpha_i \geq 0 \;,\; \sum_i y_i \alpha_i = 0$$

$$f(\vec{x}) = sgn(\langle \vec{w},\vec{x}\rangle + b) = sgn\left(\sum_i \alpha_i y_i \langle \vec{x}_i,\vec{x}\rangle + b\right)$$

W tej przestrzeni W jest funkcją iloczynów $x_i * x_j$

f – funkcja dyskryminująca

Trik – zastosowanie jądra (kernel)

Funkja Φ przeprowadza przestrzeń wektorów wejściowych do nowej przestrzeni w której dane są separowalne przez hiperpłaszczyznę:

$$\Phi(x): input\, \Re^n \rightarrow \Re^N \quad (N \geq n)$$

Wtedy

$$\langle x_i, x_j \rangle \rightarrow \langle \Phi(x_i), \Phi(x_j) \rangle = K(x_i, x_j)$$

I możemy używać tych samych równań.

Nie musimy znać funkcji Φ, wystarczy znać jądro (kernel) i można pracować w nowej przestrzeni.

$$W(\vec{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j)$$

$$\alpha_i \geq 0 \quad , \quad \sum_i y_i \alpha_i = 0$$

$$f(\vec{x}) = sgn\left(\sum_i \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \right)$$

Typowo stosowane jądra

- Polynomial

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle + c\right)^d$$

- Sigmoid

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\kappa \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \theta\right)$$

- Gaussian

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

Jądro musi spełniać warunek:

$$K(x_i, x_j) = K(x_j, x_i)$$

Lagrangian

Lagrangian w przypadku danych nieseparowalnych:

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$\text{subject to}: \quad y_i(w \cdot z_i - b) \geq 1 - \xi_i \;\; \forall\, i; \;\; \xi_i \geq 0 \;\forall\, i$$

This problem is referred to as the *primal* problem. The Lagrangian for this problem is:

$$L = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i + \sum_i \alpha_i[1 - \xi_i - y_i(w \cdot z_i - b)] - \sum_i \pi_i \xi_i$$

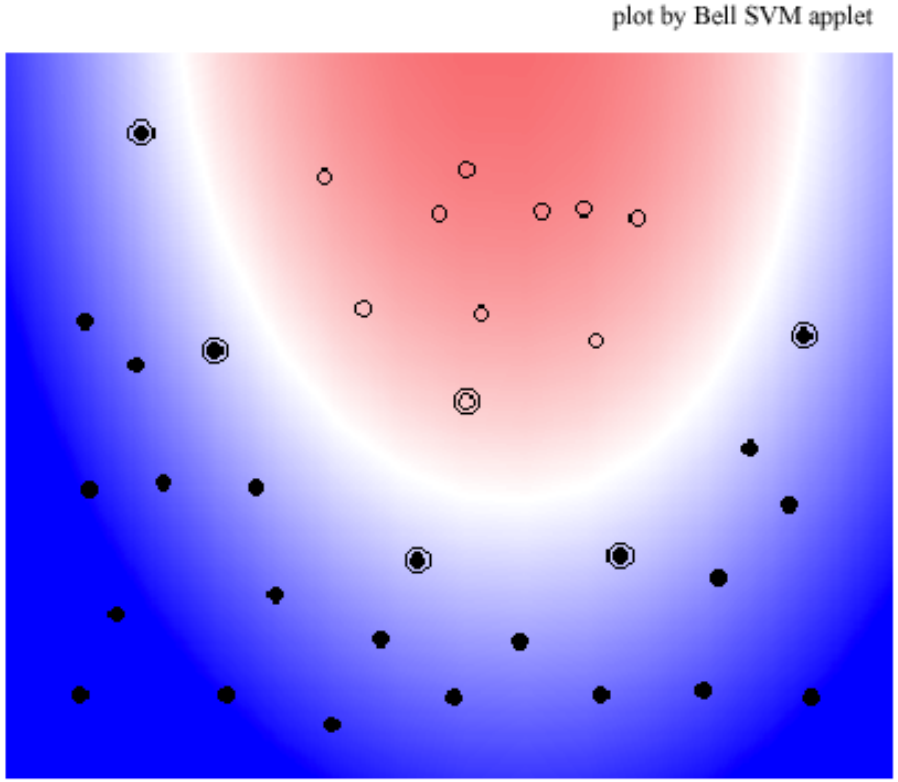
M. Krzyśko, *Systemy uczące się: rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości*. WNT, 2008. http://books.google.com/books?id=wx6DPgAACAAJ

C. J. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery **2** (1998) 121–167. http://dx.doi.org/10.1023/A:1009715923555

Nieliniowe jądro (I)

Przykład: SVM z jądrem wielomianowym 2-go stopnia

Kernel: $K(\vec{x}_i,\vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^2$

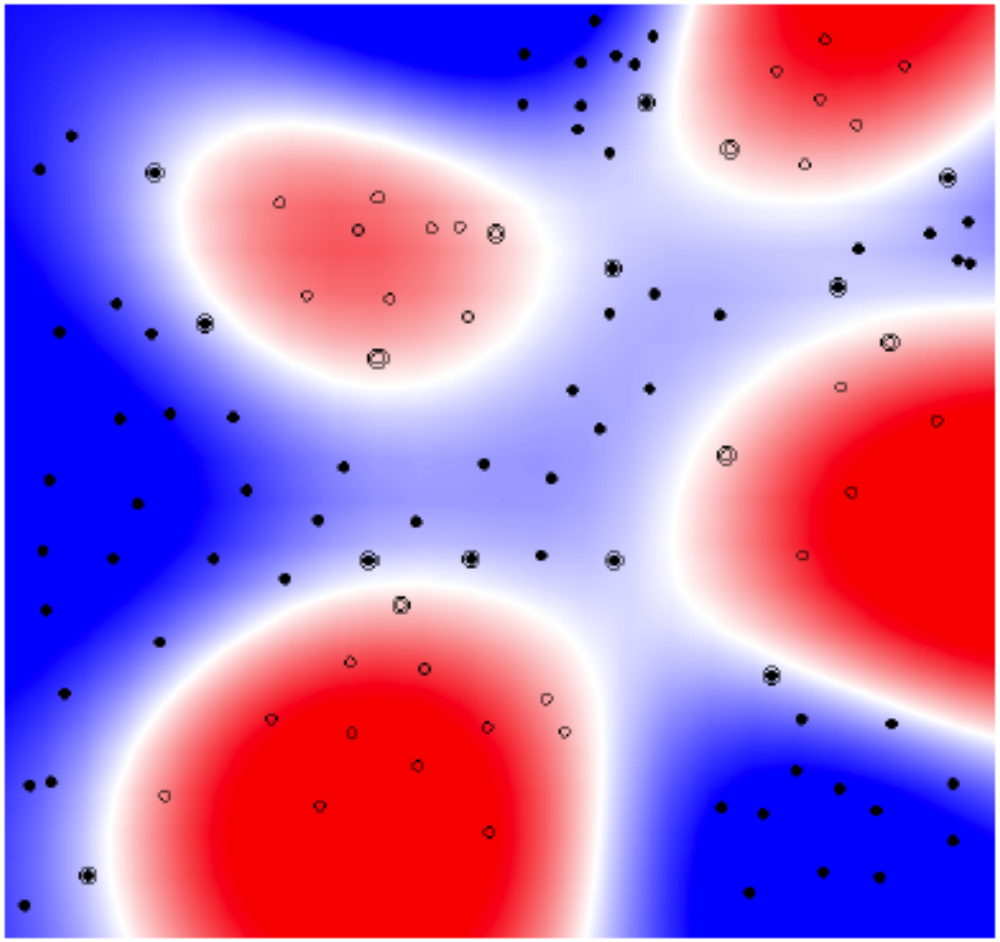


Nieliniowe jądro (II)

Jądro: funkcja Gaussa

Kernel: $K(\vec{x}_i,\vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2/\sigma^2)$

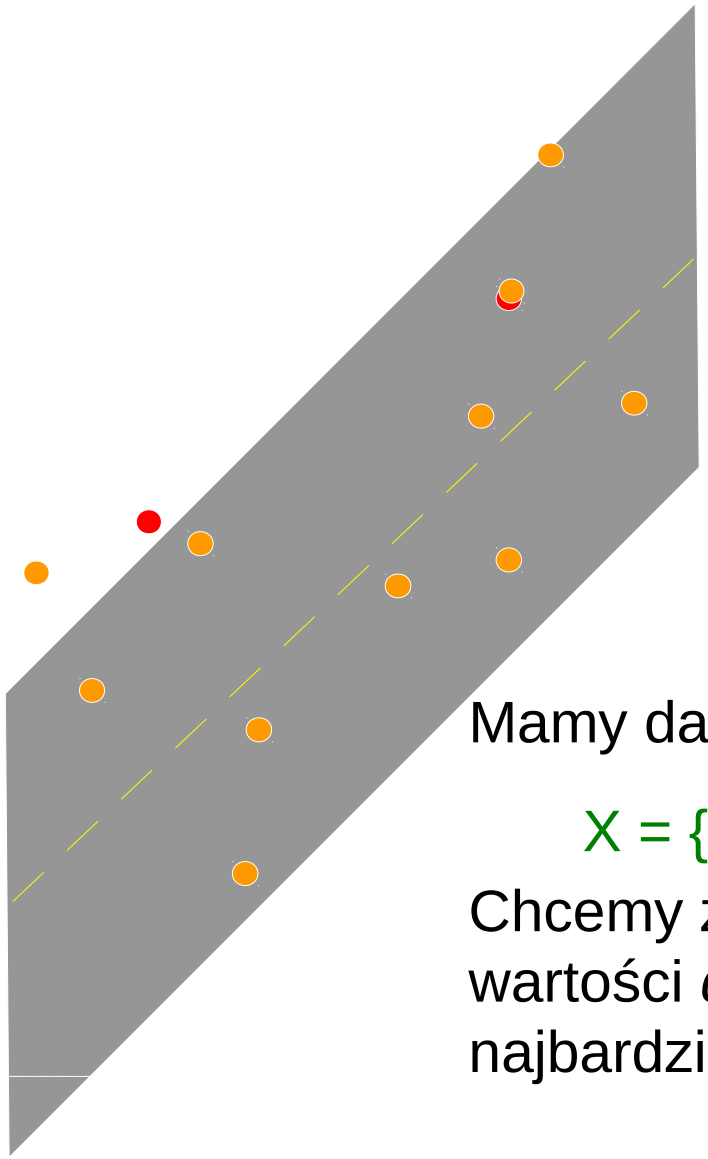
plot by Bell SVM applet

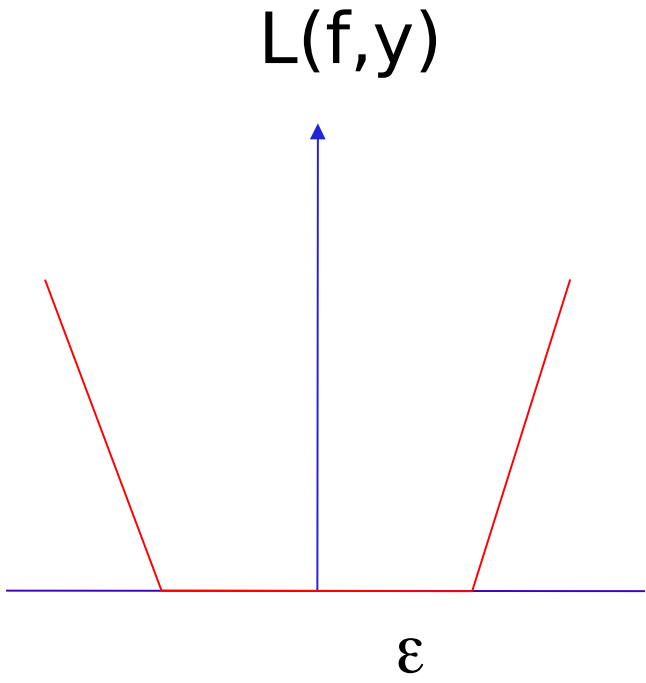


Applet pokazujący działanie SVM

http://cs.stanford.edu/people/karpathy/svmjs/demo/

Regresja – “ε insensitive loss”





Mamy dane wejściowe:

$$X = \{(\underline{x}_1, d_1), \ldots, (\underline{x}_N, d_N)\}$$

Chcemy znaleźć funkcję f(x) która ma odchylenie od wartości *d* nie większe niż ε oraz będącą jak najbardziej “gładką”.

Regresja

Zadanie: przeprowadzić uogólnienie algorytmu klasyfikacji tak, aby uzyskać regresję.

Zachować wszystkie właściwośći SVM

Definiujemy funkcję:

$$|y - f(\mathbf{x})|_\varepsilon := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}$$

Minimalizacja:

$$\frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|_\varepsilon$$

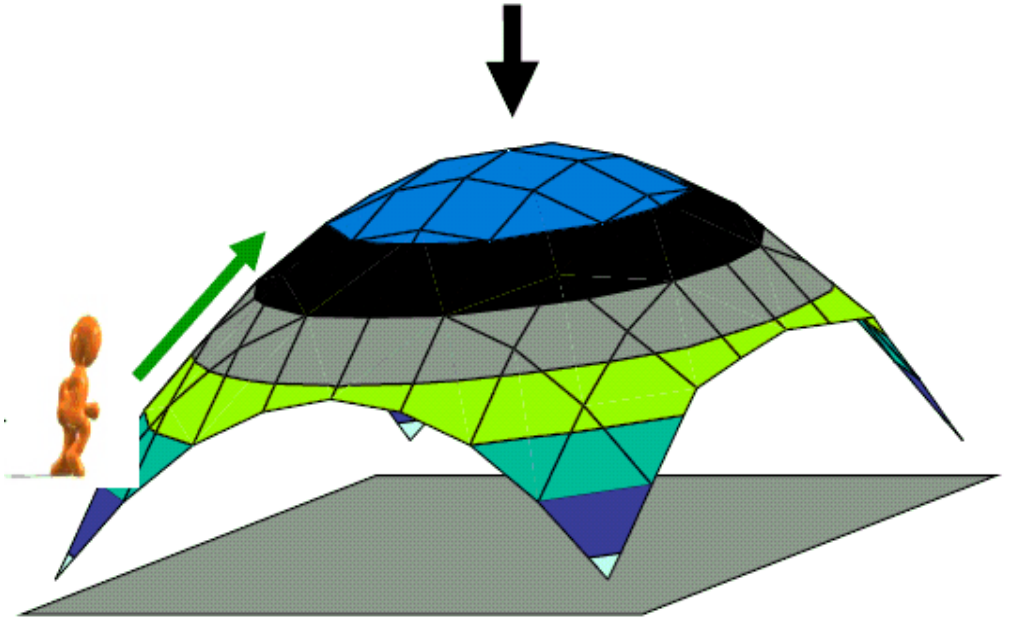
I powtarzamy procedurę przejścia do wyższych wymiarów, stosujemy funkcję jądra itd...

SVM w pakiecie*T*MVA

Algorytm SVM został zaimplementowany do pakietu TMVA – Andrzej Zemła, praca magisterska.

Dostępne funkcje jądra:

- Liniowa – zapewnia tylko separację liniową
- RBF (funkcja Gaussa) – dowolny kształt powierzchni separującej
- Wielomianowa



Rozwiązanie problemu kwadratowego - SMO (Sequential Minimal Optimization - J. Platt, Microsoft) z modyfikacjami Keerthi'ego:

- rozbicie problemu minimalizacji na wiele małych kroków
- znalezienie optymalnej pary wektorów, redukcja do problemu dwóch zmiennych i rozwiązanie analityczne
- iteracyjne powtarzanie procedury aż do znalezienia minimum.

Porównanie SMO z innymi algorytmami

Z prezentacji J. Platt

Experiment	SMO Time (sec)	SVM^{light} Time (sec)	Chunking Time (sec)	SMO Scaling Exponent	SVM^{light} Scaling Exponent	Chunking Scaling Exponent
AdultLin	**13.7**	217.9	20711.3	**1.8**	2.1	3.1
AdultLinD	**21.9**	n/a	21141.1	**1.0**	n/a	3.0
WebLin	**339.9**	3980.8	17164.7	**1.6**	2.2	2.5
WebLinD	**4589.1**	n/a	17332.8	**1.5**	n/a	2.5
AdultGaussK	442.4	**284.7**	11910.6	2.0	2.0	2.9
AdultGauss	**523.3**	737.5	n/a	2.0	2.0	n/a
AdultGaussKD	**1433.0**	n/a	14740.4	**2.5**	n/a	2.8
AdultGaussD	1810.2	n/a	n/a	2.0	n/a	n/a
WebGaussK	**2477.9**	2949.5	23877.6	**1.6**	2.0	2.0
WebGauss	**2538.0**	6923.5	n/a	**1.6**	1.8	n/a
WebGaussKD	**23365.3**	n/a	50371.9	2.6	n/a	**2.0**
WebGaussD	24758.0	n/a	n/a	1.6	n/a	n/a
MNIST	**19387.9**	38452.3	33109.0	n/a	n/a	n/a

Table 2: Timings of algorithms on various data sets.

Przykład (pakiet TMVA): Identyfikacja leptonów τ w eksperymencie ATLAS

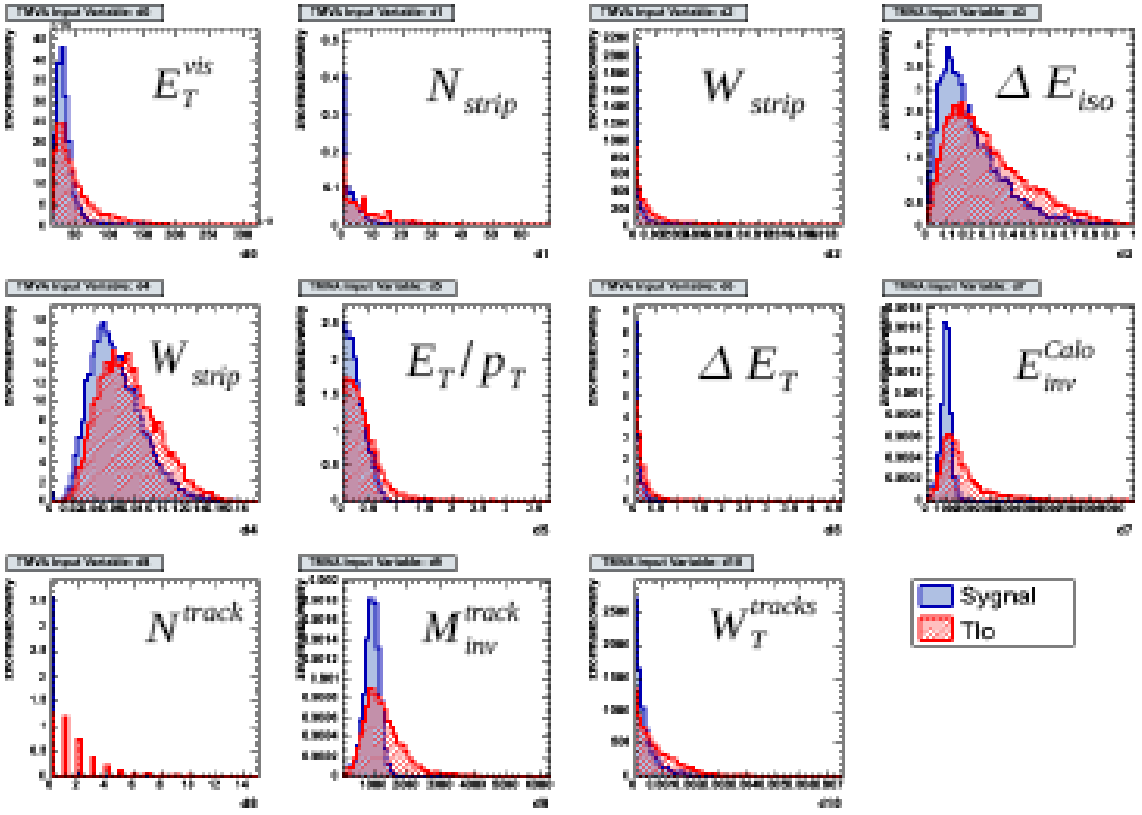
Eksperyment ATLAS – jeden z eksperymentów (CMS, ALICE, LHCb) budowanych obecnie na akceleratorze LHC w laboratorium CERN (start 2008)

Dane do analizy: hadronowe rozpady τ zrekonstruowane za pomocą algorytmu tau1p3p (z danych Monte-Carlo)

- 1-prong (1 ślad hadronowy)
- 2-prong (2 ślady hadronowe)
- 3-prong (3 ślady hadronowe)

Identyfikacja – zrekonstruowane zmienne:

- rozpady 1p – 9 zmiennych
- rozpady 2p i 3p – 11 zmiennych

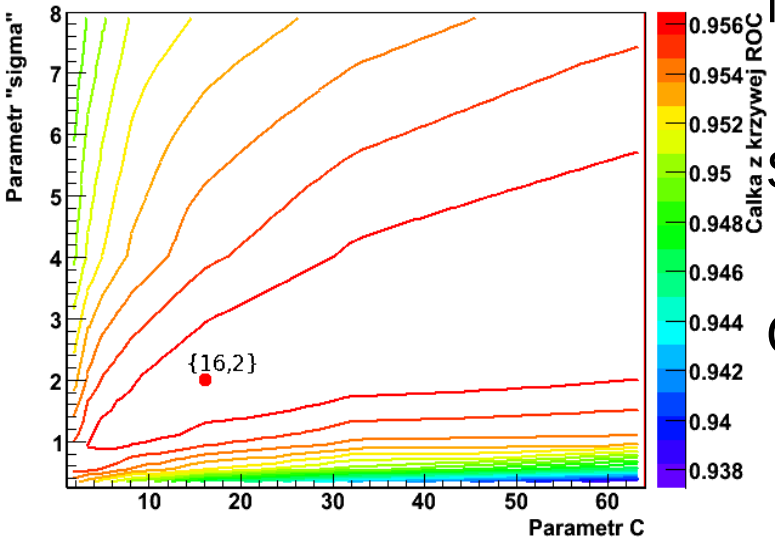


Dane 3-prong

Żadna zmienna samodzielnie nie pozwala na dobre rozróżnienie sygnału od tła.

Optymalizacja punktu pracy

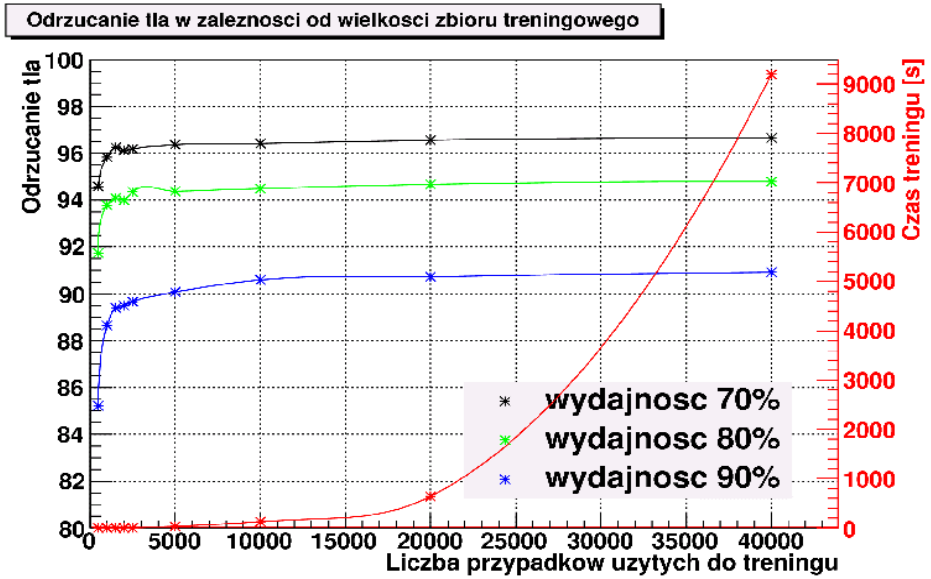
dane 3-prong



Nauka: trening na części danych, sprawdzenie na pozostałych danych (kontrola przeuczenia).

SVM z gaussowską funkcja jądrową – optymalizacja σ funkcji gaussa oraz parametru C.

Optymalizacja – wielokrotny trening, wybór „najlepszej" pary {σ,C} (największa całka z krzywej ROC).



Zależność odrzucania tła oraz czasu treningu od liczby wektorów treningowych.

Dla małych zbiorów treningowych prawie optymalna klasyfikacja (skraca czas treningu, często nie mamy dużych zbiorów danych treningowych).

Separacja sygnału i tła

dane 3-prong

Wyniki dla funkcji jądrowych: liniowej, gausa i wielomianowej 9 stopnia.

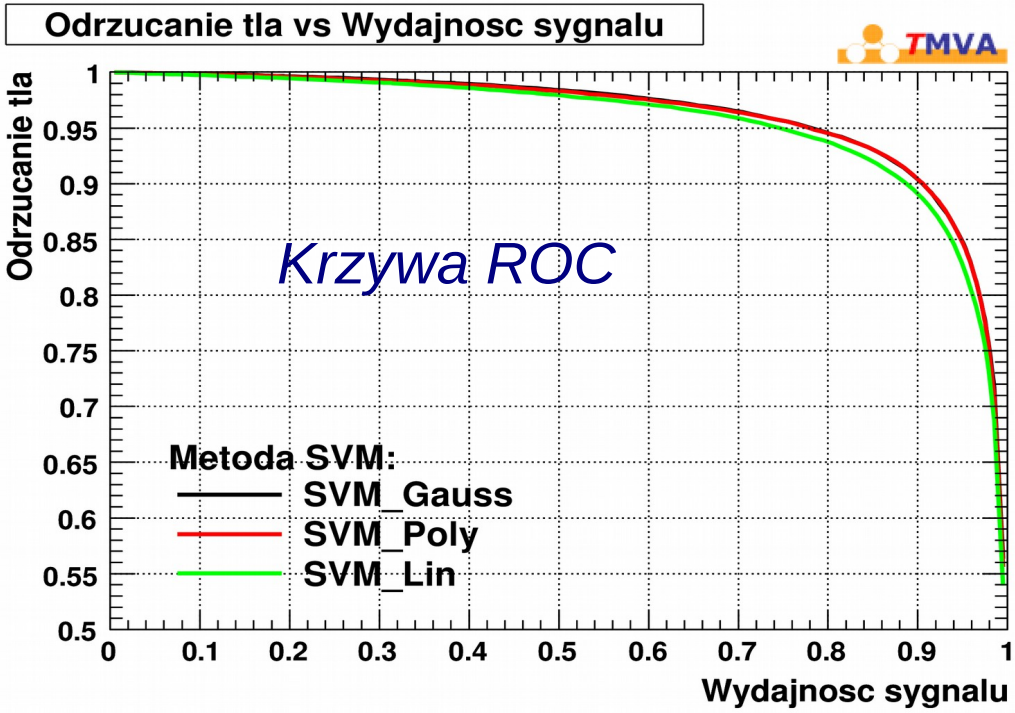
Trening na zbiorze zawierającym po 5000 przypadków sygnału i tła.

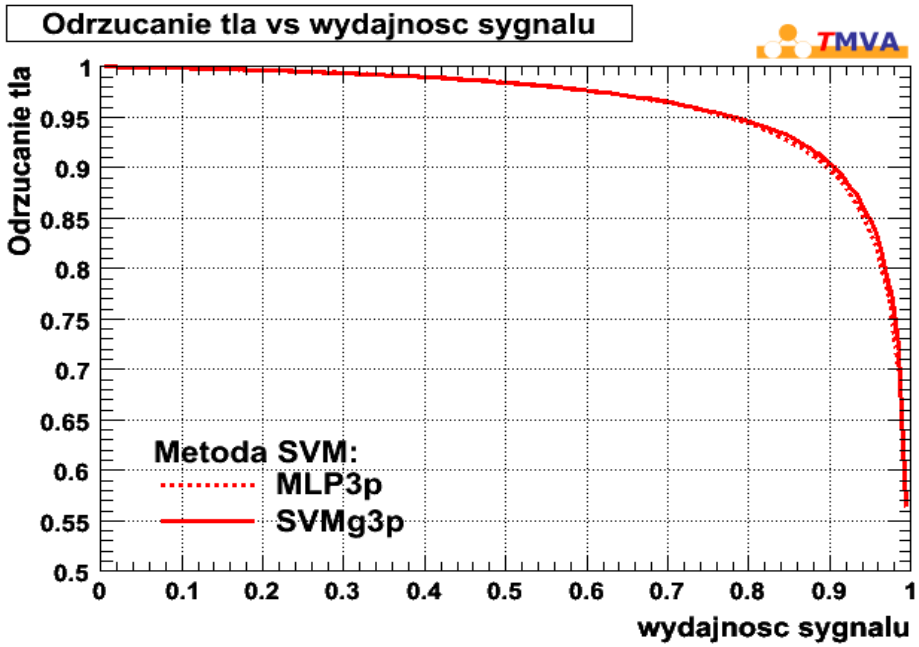
Testowanie: ~62000 p. sygnału
~264000 p. tła

Porównanie SVM z wielowarstwową siecią neuronową

Sieć MLP z pakietu TMVA

2 ukryte warstwy mające po 20 nodów





Sieć MLP oraz SVM z funkcją jądrową gausa dają zbliżone rezultaty

SVM <-> multilayer perceptron (sieć neuronowa) Fundamentalne różnice

MLP – stopień skomplikowania kontrolowany poprzez liczbę ukrytych nodów

SVM – stopień skomplikowania kontrolowany niezależnie od liczby wymiarów

SVM – mapowanie powoduje, że płaszczyzna separująca jest konstruowana w przestrzeni wielo (często nieskończenie) wymiarowej.

Problem z wielka liczbą wymiarów jest rozwiązywany poprzez użycie funkcji jądra (kernel).

Jianfeng Feng, Sussex University

Mocne strony SVM

Stopień skomplikowania/pojemność jest niezależny od liczby wymiarów.

Dobra podbudowa statystyczna.

Znajdowanie minimum. Minimalizujemy funkcję kwadratową co gwarantuje zawsze znalezienie minimum. Algorytm jest wydajny i SVM generuje prawie optymalny klasyfikator. Nie jest tez czuły na przetrenowanie.

Dobre uogólnianie dzięki wielowymiarowej “feature space”.

Najważniejsze: poprzez użycie odpowiedniej funkcji jądra SVM automatycznie dobiera wszystkie parametry sieci (liczbę ukrytych nodów, wagi).

Jianfeng Feng, Sussex University

Słabe strony SVM

Powolny trening – minimalizacja funkcji, szczególnie dokuczliwy przy dużej ilości danych użytych do treningu.

Rozwiązania też są skomplikowane (normalnie >60% wektorów użytych do nauki staje się wektorami wspierającymi), szczególnie dla dużych ilości danych.

> *Przykład (Haykin): poprawa o 1.5% ponad wynik osiągnięty przez MLP. Ale MLP używał 2 ukrytych nodów, SVM 285.*

Trudno dodać własną wiedzę (prior knowledge)

Jianfeng Feng, Sussex University

Zadanie dla ambitnych

https://www.kaggle.com/c/higgs-boson

Open Higgs challenge!!!!!!!!!!!!!!

Machine Learning and HEP

- 90'ies - Neural Nets used by LEP experiments
- BDT (Adaboost) invented in 97
- Machine Learning used extensively at D0/CDF (mostly BDT, also Neural Nets) in the 00'ies
- Last years – mostly BDT built in TMVA ROOT package (popular among physicists). Neural Nets and other techniques treated as obsolete.

- **Not much work within LHC experiments on studying possible better MVA techniques.**
- **Enormous development of Machine Learning in the outside world in the last 10 years ("Big Data", "Data Science", even "Artificial Intelligence" is back).**

- **We have to catch up and learn from computer scientists:**

Make an open Higgs challenge!

- **Task: identify H->tau tau signal out of background in the simulated data.**

How did it work ?

- People register to Kaggle web site hosted https://www.kaggle.com/c/higgs-boson . (additional info on https://higgsml.lal.in2p3.fr).
- …download training dataset (with label) with 250k events
- …train their own algorithm to optimize the significance (à la s/sqrt(b))
- …download test dataset (without labels) with 550k events
- …upload their own classification
- The site automatically calculates significance. Public (100k events) and private (450k events) leader boards update instantly. (Only the public is visible)
- 1785 teams (1942 people) have participated
- most popular challenge on the Kaggle platform (until a few weeks ago)
- 35772 solutions uploaded

Funded by: Paris Saclay Center for Data Science, Google, INRIA

Final leaderboard

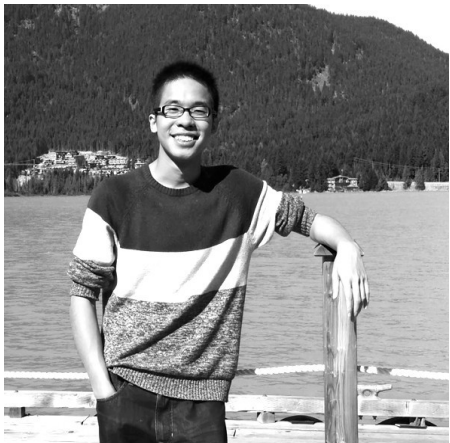
#	Δrank	Team Name ‡ model uploaded * in the money		Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑1	Gábor Melis ‡ *	7000$	3.80581	110	Sun, 14 Sep 2014 09:10:04 (-0h)
2	↑1	Tim Salimans ‡ *	4000$	3.78913	57	Mon, 15 Sep 2014 23:49:02 (-40.6d)
3	↑1	nhlx5haze ‡ *	2000$	3.78682	254	Mon, 15 Sep 2014 16:50:01 (-76.3d)
4	↑38	ChoKo Team		3.77526	216	Mon, 15 Sep 2014 15:21:36 (-42.1h)
5	↑35	cheng chen		3.77384	21	Mon, 15 Sep 2014 23:29:29 (-0h)
6	↑16	quantify		3.77086	8	Mon, 15 Sep 2014 16:12:48 (-7.3h)
7	↑1	Stanislav Semenov & Co (HSE Yandex)		3.76211	68	Mon, 15 Sep 2014 20:19:03
8	↓7	Luboš Motl's team	Best physicist	3.76050	589	Mon, 15 Sep 2014 08:38:49 (-1.6h)
9	↑8	Roberto-UCIIIM		3.75864	292	Mon, 15 Sep 2014 23:44:42 (-44d)
10	↑2	Davut & Josef		3.75838	161	Mon, 15 Sep 2014 23:24:32 (-4.5d)
45	↑5	crowwork ‡	HEP meets ML award XGBoost authors Free trip to CERN	3.71885	94	Mon, 15 Sep 2014 23:45:00 (-5.1d)
782	↓149	Eckhard		3.49945	29	Mon, 15 Sep 2014 07:26:13 (-46.1h)
991	↑4	Rem.		3.20423	2	Mon, 16 Jun 2014 21:53:43 (-30.4h)

The winners

- See http://atlas.ch/news/2014/machine-learning-wins-the-higgs-challenge.html
- 1 : **Gabor Melis** (Hungary) software developer and consultant : wins 7000$.
- 2 : **Tim Salimans** (Neitherland) data science consultant: wins 4000$
- 3 : **Pierre Courtiol** (nhlx5haze) (France) ? : wins 2000$
- HEP meets ML award: (team crowwork), **Tianqi Chen** (U of Washington PhD student in Data Science) and **Tong He** (graduate student Data Science SFU). Provided **XGBoost public software** used by many participants.

https://github.com/dmlc/xgboost

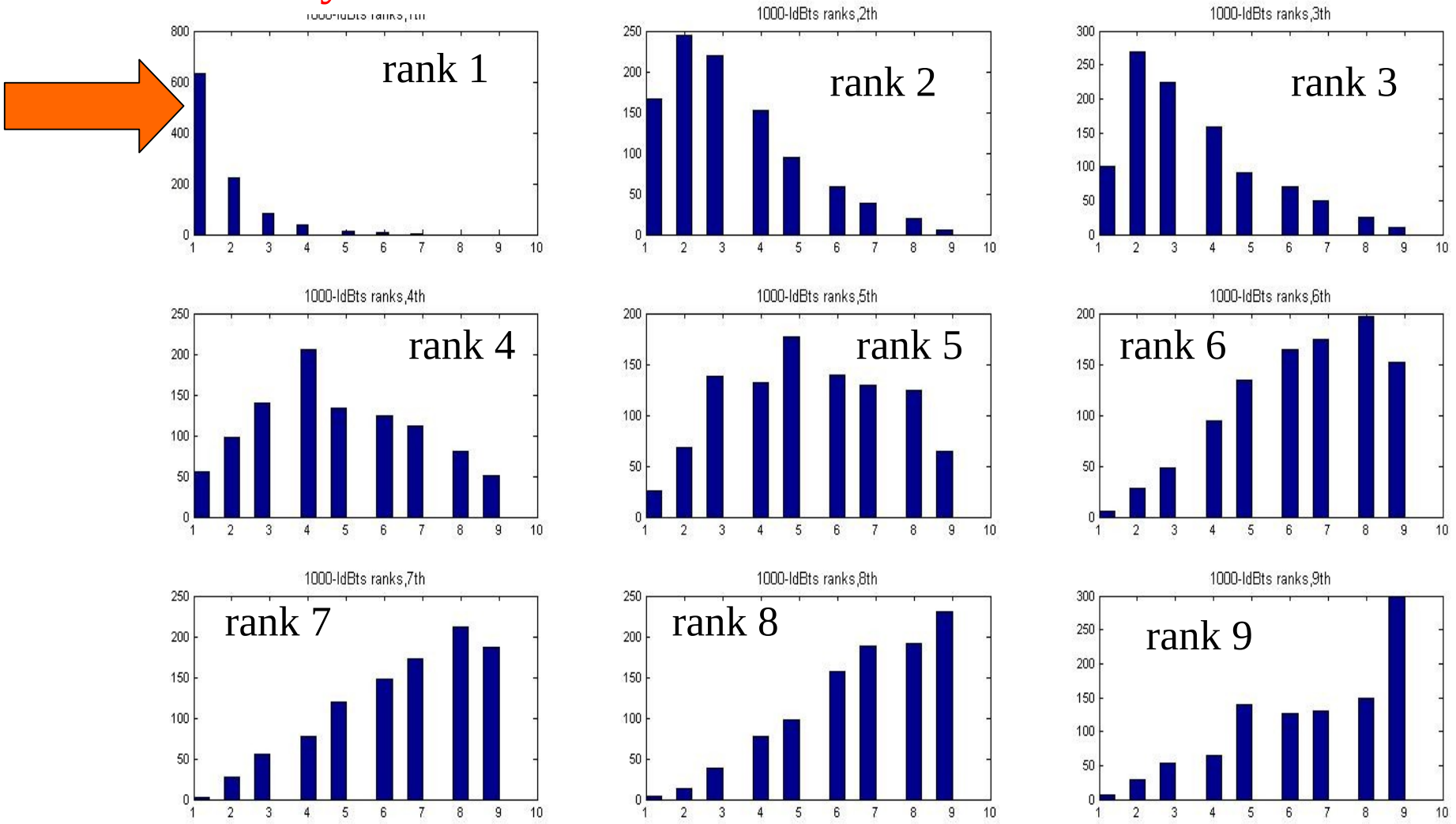
?



Rank distribution after bootstrap

Distribution of rank of participant of rank i after 1000 bootstraps of the test sample.

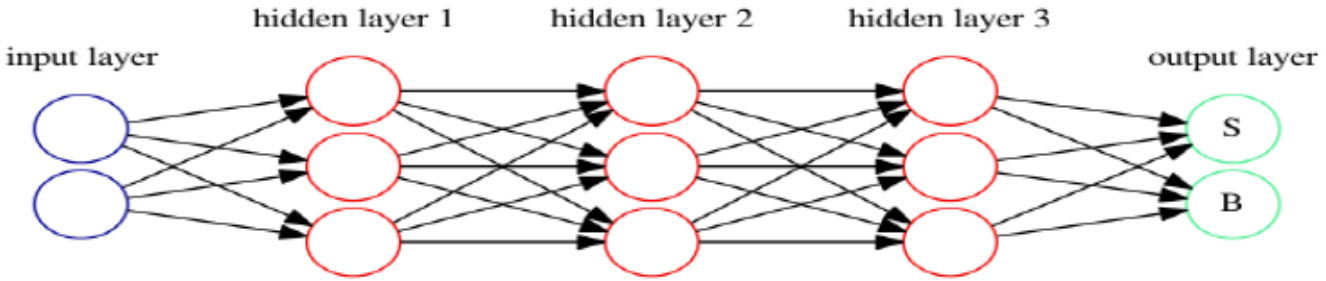
! Gabor clearly better




Who are the
XGBoost)

Deep neural network

- Hierarchical feature extraction – first build abstract objects, than find dependencies between them.
- Deep neural network (DNN)- an artificial neural network with multiple hidden layers of units between the input and output layers.
- Extra layers - composition of features from lower layers, potential of modeling complex data with fewer units than a similarly performing shallow network.



- inputs: normalized features (~30), some log transformed
- 3 hidden layers of 600 neurons each
- output layer: 2 softmax units (one for signal, one for background)
- activation function: “max channel” in groups of 3
- trained to minimize cross entropy
- regularization: dropout on hidden layers, $L_1 + L_2$ penalty and a mild sparsity constraint input weights

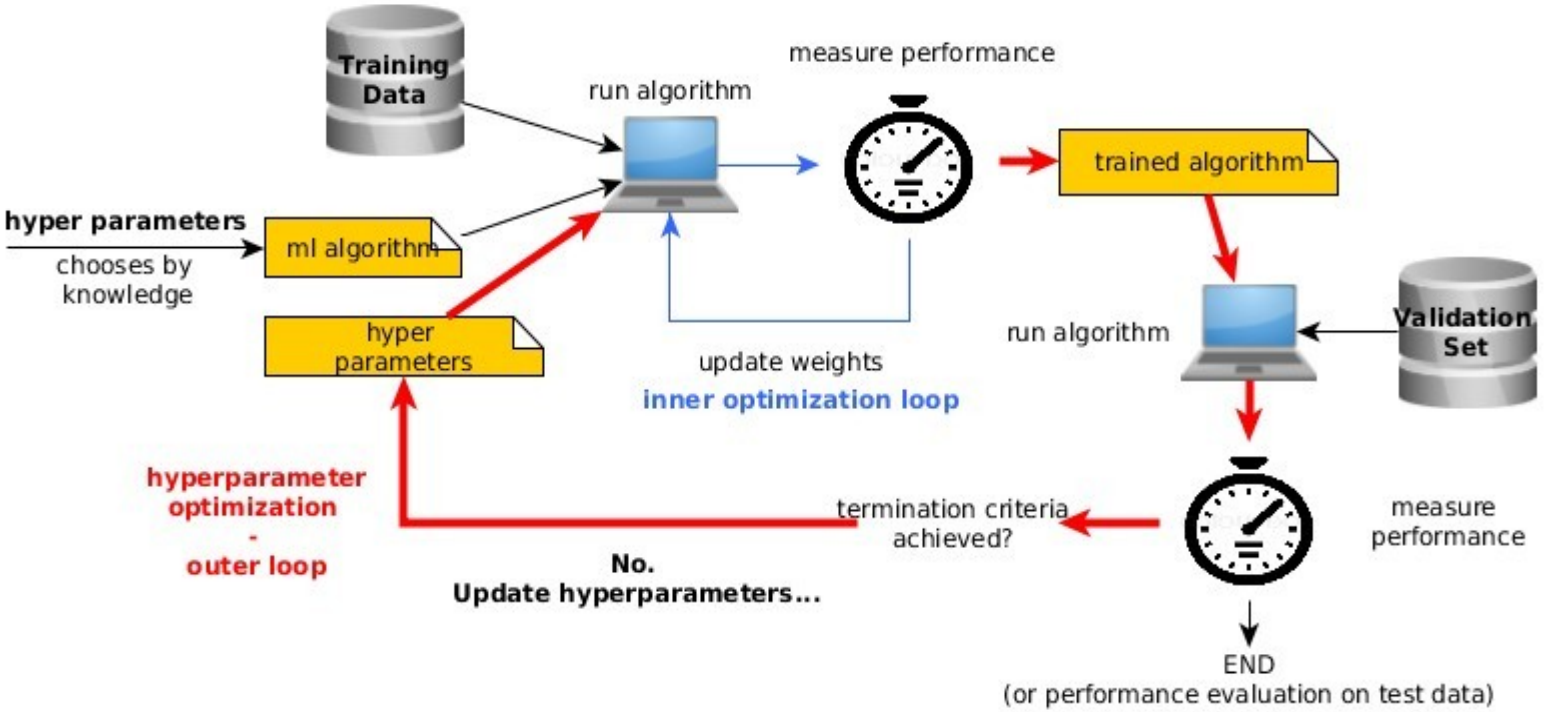
Challenge winning
Gabor's deep neural network
(from Gabor's presentation)

- CV bagged NNs: 3.83
- CV bagged xgboost: 3.79

Remark:
Few years ago some experts claimed neural networks are an obsolete tool :)

Automatic optimization of hyperparameters

- Manual optimization of NN (or any other method) is time consuming.
- Fortunately the Bayesian optimization methods can rival and surpass human domain experts in finding good hyperparameter settings.
- SMAC, SPEARMINT, TPE (and others) are doing that with great success: http://www.cs.ubc.ca/~hutter/papers/13-BayesOpt_EmpiricalFoundation.pdf



Analiza podczas praktyk studenckich

- Próbowaliśmy powtórzyć HiggsChallenge podczas praktyk studenckich.
- Udało się za pomocą TMVA (konwersja danych do formatu root) oraz pakietu XGBoost
- Optymalizacja parametrów XGBoost za pomocą programu hyperopt

A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss (2009)
TMVA 4 Package Documentation
https://tmva.sf.net

Tianqi Chen, Tong He, Bing Xu and Michael Benesty (2014)
XGBoost Package Documentation
https://github.com/dmlc/xgboost

James Bergstra, Dan Yamins, and David D. Cox (2013)
Hyperopt Package Documentation
https://github.com/hyperopt

Rozwiązania Kaggle-Higgs vs Hyperopt

Porównianie wyników uzyskanych przez nas automatycznie z wynikami z najlepszymi znalezionymi parametrami dla XGBoost.

Kto	9. K-H	M. Wolter	Nasze obliczenia
Maks. głębokość	9	10	9
Wsp. uczenia	0.01	0.089	0.059
Liczba drzew	3000	150/250/500	300
Liczba testów	-	300	100
Sub_sample	0.9	1	0.9
Maks. ROC	0.987	0.933/0.934/0.933	0.934

Sub_sample - jaka cześć danych brana jest do procesu uczenia - wprowadza pewną losowość i zapobiega przeuczaniu

Jak widać wyniki przez nas osiągnięte są znacznie słabsze. Prowadziliśmy poszukiwania w innym regionie parametrów.

Zadanie drugie
ATLAS Z→tau tau selection

- Dane:
  - mc12/Ztautau.root - sygnał
  - Powheg_ttbar.root - tło
  - Wenu.root - tło
  - Wmunu.root - tło
  - Wtaunu.root - tło
  - Zee.root - tło
  - Zmumu.root - tło

- Zmienne:

  preselekcja:

  if(!(evtsel_is_dilepVeto > 0 && evtsel_is_tau > 0 && fabs(evtsel_tau_eta) < 2.47 && evtsel_is_conf_lep_veto == 1 && evtsel_tau_numTrack == 1 && evtsel_lep_pt > 26 && fabs(evtsel_lep_eta) < 2.4 && evtsel_transverseMass < 70)) continue;

ATLAS Z→tau tau selection

- Zmienne użyte do treningu:
  - *evtsel_tau_et*
  - *evtsel_dPhiSum*
  - *evtsel_tau_pi0_n*
  - *evtsel_transverseMass*
  - *sum_cos_dphi*

- Spectator
  - *vis_mass*

- Program:
  - TMVAClassificationMW.C i TMVAClassificationMW.h
    Wykonuje podstawowy trening.

ATLAS Z→tau tau selection

- Zainstalować pakiet root i TMVA
- Ściągnąć dane i przykładowy program:
  –
- Uruchomić przykładowy program:

  root -l

  .L TMVAClassificationMW.C++

  TMVAClassificationMW t

  t.Loop()
- Zmodyfikować go:
  - **Spróbować zoptymalizować parametry wybranej metody**
  - **Spróbować usunąć jakieś zmienne a może dodać?**
  - **Spróbować użyć indywidualnych zmiennych wchodzących w skład np. *sum_cos_dphi***
  - **Użyć wszystkich rodzajów tła – użyć wag *WeightLumi***
- Zaaplikować wyuczony klasyfikator do danych (*data12/Muons.PhysCont.grp14.root*), można się wzorować na przykładzie TMVAClassificationApplication dostępnym na stronie TMVA.

ATLAS Z→tau tau selection

Wykonać tego typu rysunek np. dla masy widzialnej

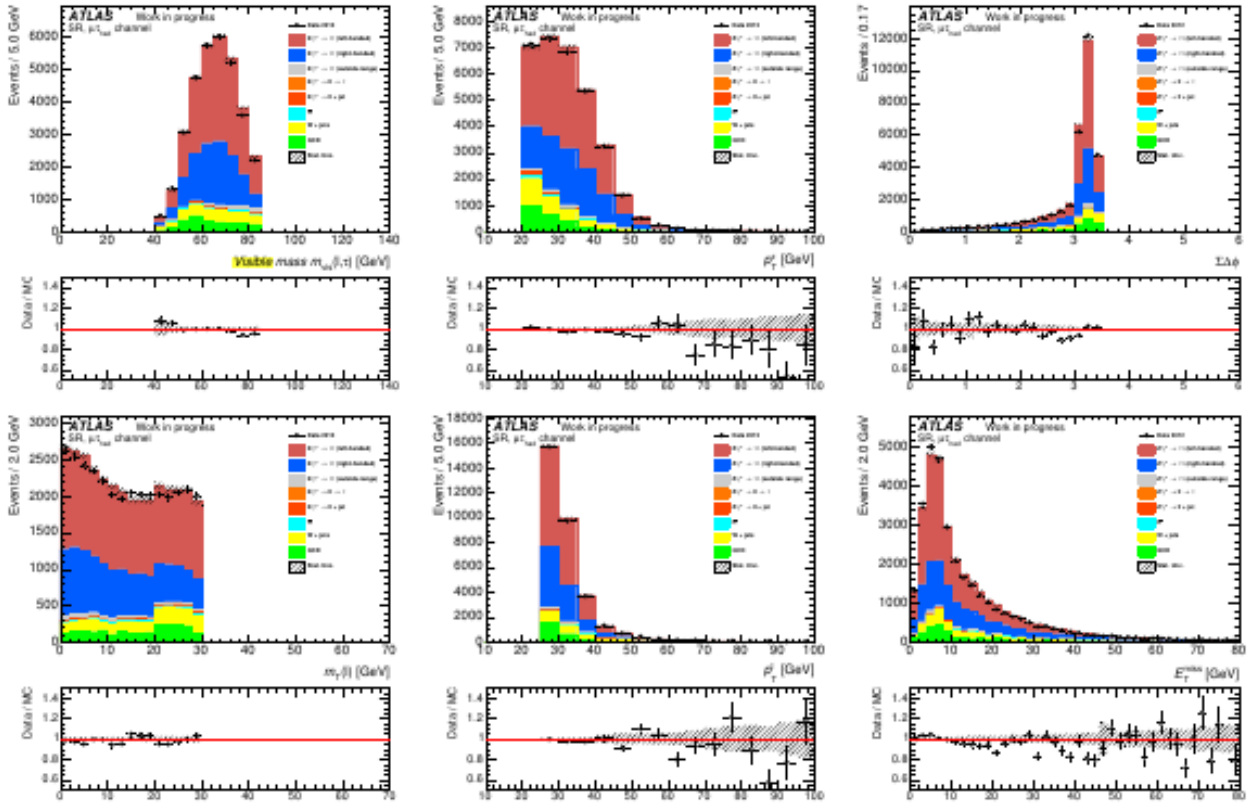


Figure 41: Distributions of variables observed in $Z \to \tau\tau$ (μ-had channel). From top-left: visible mass of τ-lepton system, τ transverse momentum, sum of polar angles between τ and missing-E_T and between lepton and missing-E_T, transverse mass of the lepton-missing-E_T system, lepton transverse momentum and missing-E_T.

ATLAS Z→tau tau selection

Event Selection and Background Estimate

Region / Cut	Signal Region	Same Sign	W Control Region	QCD Control Region
Single lepton trigger + offline lepton pT	evtsel_is_mu / evtsel_is_el			
Isolated Lepton	evtsel_is_isoLep			!evtsel_is_isoLep
Medium Tau ID	evtsel_is_tau			
Veto dileptons	evtsel_is_dilepVeto			
Muon Veto + medium Electron Veto	evtsel_is_conf_lep_veto_medium			
Single Prong tau	evtsel_tau_numTrack == 1			
Transverse Mass	evtsel_transverseMass < 30		evtsel_transverseMass > 70	
Sum Delta Phi	evtsel_dPhiSum < 3.5		evtsel_dPhiSum > 3.5	
Opposite Sign	evtsel_is_oppositeSign	!evtsel_is_oppositeSign	evtsel_is_oppositeSign / !evtsel_is_oppositeSign	evtsel_is_oppositeSign / !evtsel_is_oppositeSign

Cięcia zastosowane w analizie polaryzacji tau pochodzących z rozpadu *Z → taut tau*
Czy używając uczenia maszynowego udało nam się poprawić wynik?