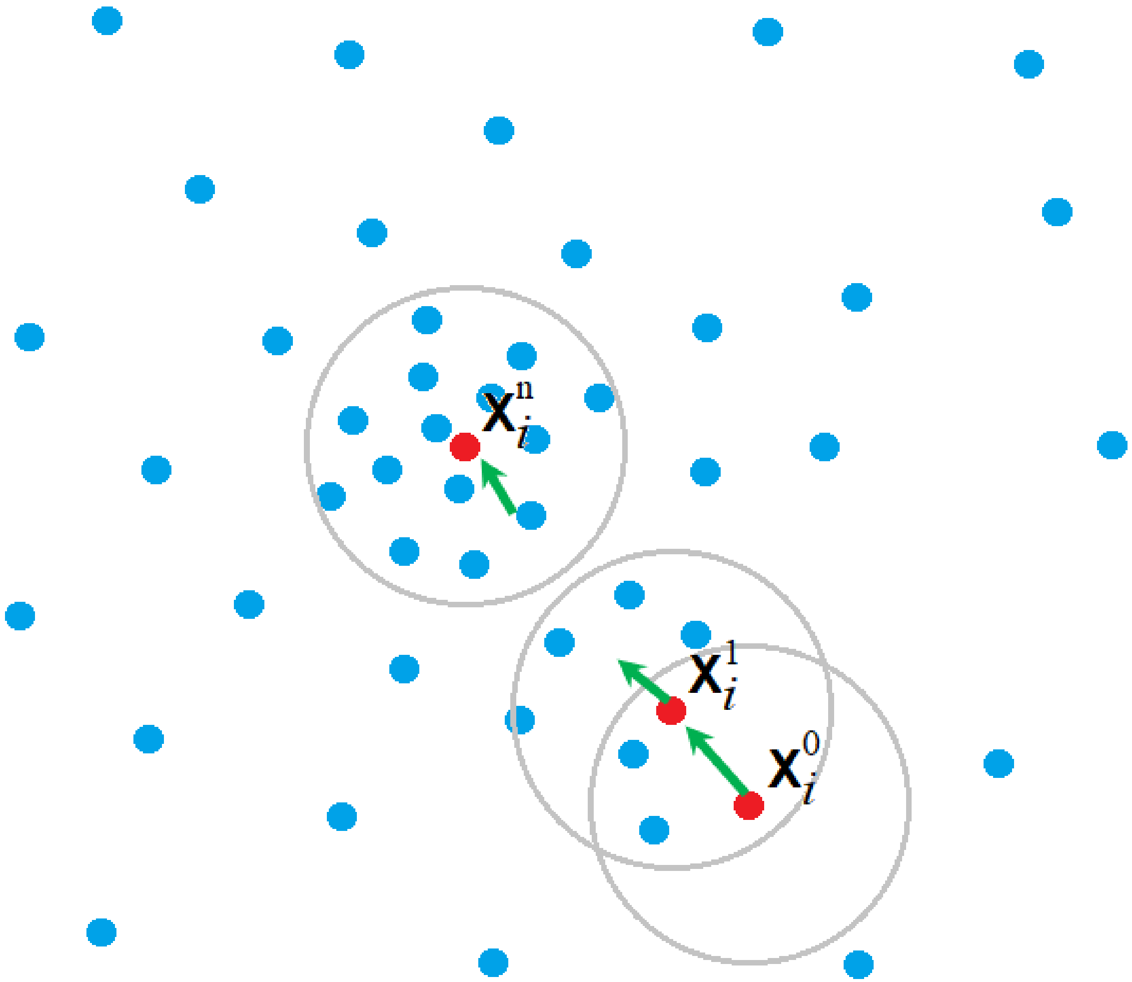
# MeanShift, HMM, MACE, SVM

doc. Ing. Josef Chaloupka, Ph.D.

# MeanShift

- MeanShift – přesun za průměrem
- patří mezi shlukovací (clusterovací) segmentační metody
- Bezparametrické – není potřeba znát počet nebo tvar segmentů
- interativní algoritmus >>> v každé interaci se vypočte vážený průměr okna a do tohoto okna se nové okno posune. Dosažení konvergence (další posun = 0) – konec. Posun jen v úrovni 1px – ukončení při určitém počtu interací nebo po dosažení malého (předem daného) posunu.
- Body v prostoru jsou reprezentovány vekorem $\mathbf{X}_i$ = [x, y, f(y, x)]; x, y souřadnice bodu, f(y, x) – např. jas (3D), RGB + 3 parametry – (5D)
- Algoritmus – 3 parametry: 1) poloměr výpočetního okna, 2) jádro (kernel) K, 3) maximální jasová vzdálenost

# MeanShift



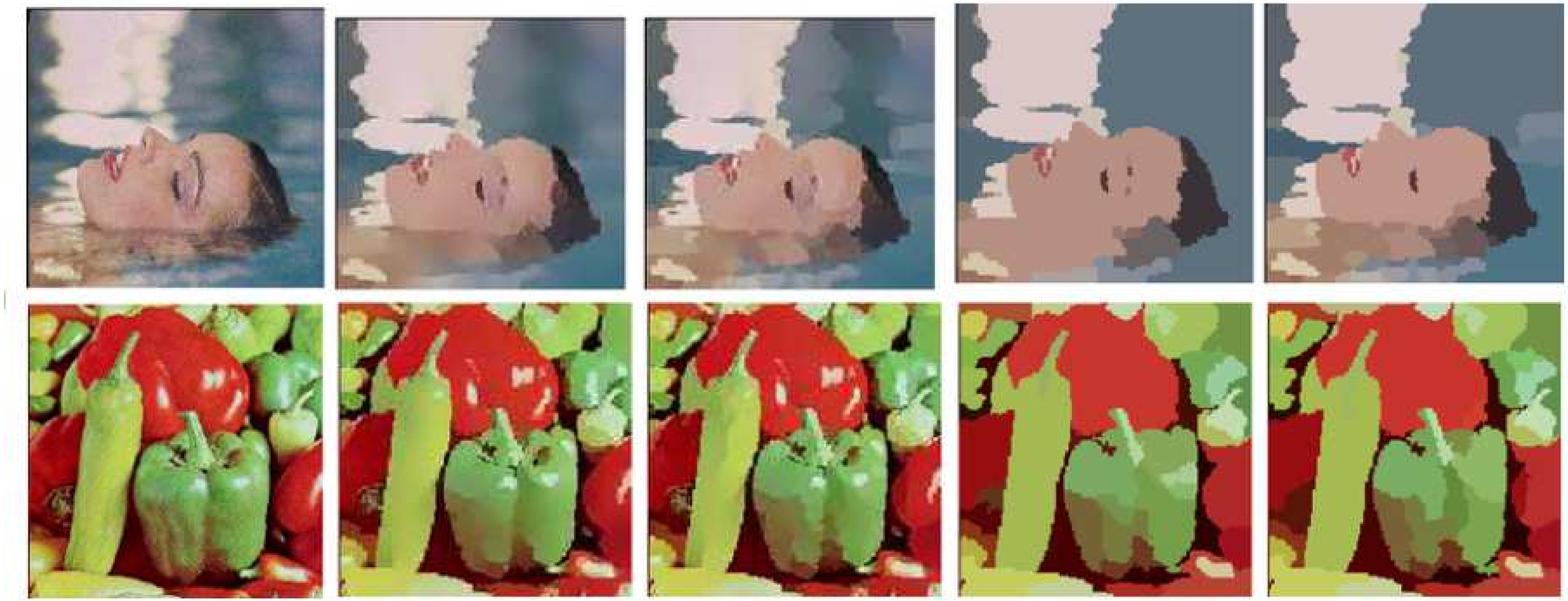

interace

# MeanShift

- MeanShift – sledování objektů

https://docs.opencv.org/3.4/d7/d00/tutorial_meanshift.html

# MeanShift



Backhouse, I. Y. H. Gu and T. Wang, "ML Nonlinear Smoothing for Image Segmentation and its Relationship to the Mean Shift," *2007 IEEE International Conference on Image Processing*, San Antonio, TX, USA, 2007, pp. IV - 337-IV - 340

# MeanShift

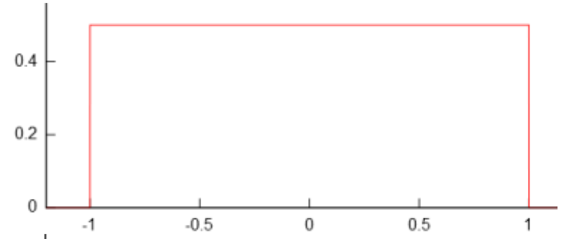
- Body v prostoru jsou reprezentovány vekorem $\mathbf{X}_i$ = [x, y, f(y, x)]; x, y souřadnice bodu, f(y, x) – např. jas (3D) **d** = 3, RGB + 3 parametry – (5D) **d** = 5
- Algoritmus – 3 parametry: 1) poloměr výpočetního okna **h**, 2) jádro (kernel) **K**, 3) maximální jasová vzdálenost
- Obrazové body blízko u sebe se stejnou intenzitou budou v prostoru (3D, 5D) vytvářet shluky
- V každém kroku algoritmu se z vybraného bodu posunujeme ve směru váženého průměru (nového těžiště)
- Postup: 1) z každého bodu pustíme MeanShift a uložíme si lokální max., kam algoritmus dokonvergoval, 2) Shlukneme body, které dokonvergovaly do stejného místa (s jistou tolerancí)
- Odhad hustoty v bodě x: $$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

h – šířka kernelu, $x_{1..n}$ jsou vzorky

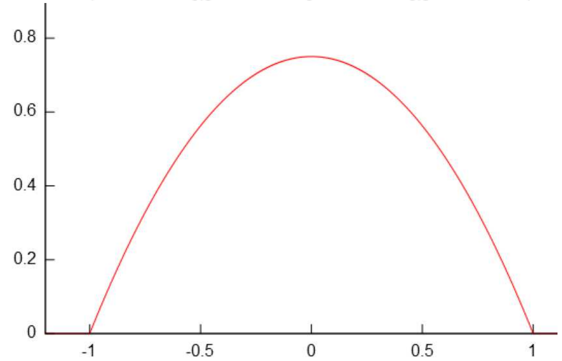
# MeanShift

- K – kernel (jádro) – radiálně symetrická funkce $K(\mathbf{x}) = c_k k(\|\mathbf{x}\|^2)$
- k – profil kernelu
- Představa 2D – kruh, poloměr h, 3D koule, ND – ND koule

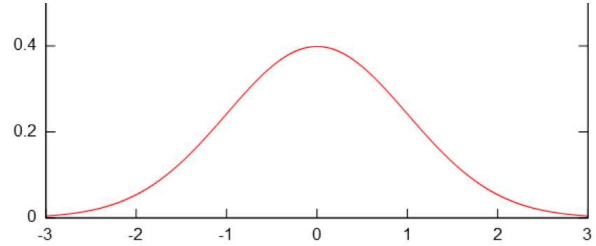
$$K(x - x_i) = \begin{cases} 1 & \text{if } \|x - x_i\| \leq \mathrm{h} \\ 0 & \text{if } \|x - x_i\| > \mathrm{h} \end{cases}$$




- Uniformní kernel K(x) = 0,5
pro {|x| ≤ 1}, jinak 0




- Epanechnikův kernel K(x) = (3 / 4)* (1 – $x^2$)
pro {|x| ≤ 1}, jinak 0




- Gaussův kernel K(x) = $\frac{1}{\sqrt{2\pi}} e^{-0,5\, x^2}$

# MeanShift

- Hustota pravděpodobnosti není tak důležitá, důležitý je její gradient:

$$\nabla f(\mathbf{x}) = \frac{2c_k}{nh^d} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k' \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)$$

$$= \frac{2c_k}{nh^d} \left[ \sum_{i=1}^{n} g \left( \left\| \frac{\mathbf{x} - \mathbf{x_i}}{h} \right\|^2 \right) \right] \left[ \frac{\sum_{i=1}^{n} \mathbf{x}_i g \left( \left\| \frac{\mathbf{x} - \mathbf{x_i}}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left( \left\| \frac{\mathbf{x} - \mathbf{x_i}}{h} \right\|^2 \right)} - \mathbf{x} \right] \qquad g(x) = -k'(x)$$

- Postupujeme z bodu x ve směru gradientu dokud změna posunu není menší než námi zvolený práh, $x_{j+1}$ - $x_j$ **meanshift krok**

$$\mathbf{x}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i g \left( \left\| \frac{\mathbf{x}_i - \mathbf{x}_j}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left( \left\| \frac{\mathbf{x}_i - \mathbf{x}_j}{h} \right\|^2 \right)}$$

```
while(1)
  xj+1 = …
  i = i +1
  if || xj+1 - xj || < T break
```

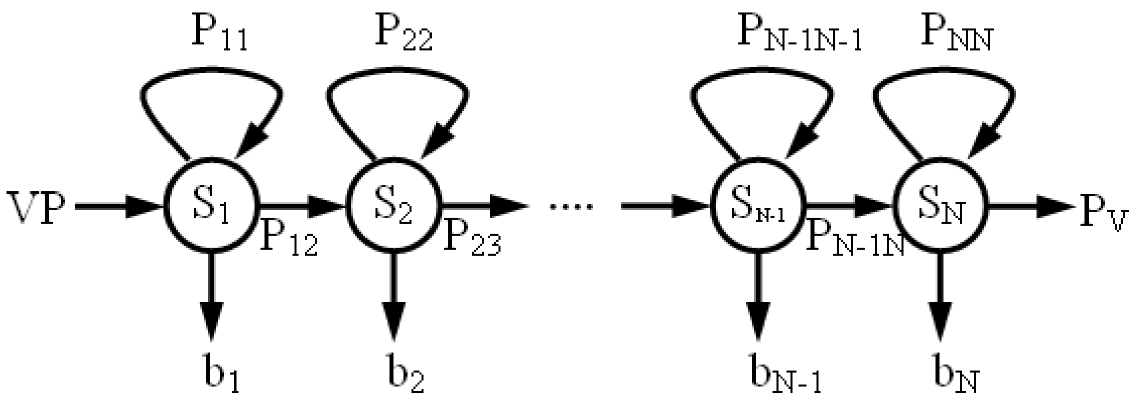
- Segmentace – shlukování bodů: pro každý bod máme spočítané lokální maximum

(atraktor) z = [x, y, f(y, x)], sousední body A a B sloučíme pokud abs($z_A$-$z_B$) ≤ **t**

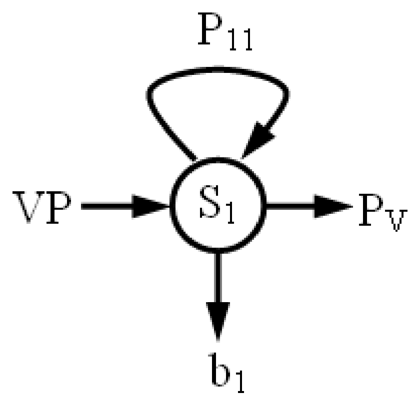
# MeanShift

- **Nastavení algoritmu**
- K – kernel nejmenší vliv
- Minimální jasová vzdálenost – nutnost nastavit
- Poloměr výpočetního okna – velký vliv na výpočet (čas), špatně zvolen – špatná segmentace
- Zastavit včas algoritmus – nespoléhat, že při konvergenci posun = 0, zastavit dřív při malém posunu, nastavit maximální počet interakcí
- Stejnobarevné plochy tvořené více atraktory, sloučení atraktorů dle minimální jasové vzdálenosti
- Modifikace Meanshift – 1) zlepšení segmentační schopnosti – volba vyhlazovacích parametrů, 2) zrychlení výpočtu – zjednodušení vyhledávání sousedních bodů, omezení počtu interací…

# HMM - Hidden Markov Model




**HMM**

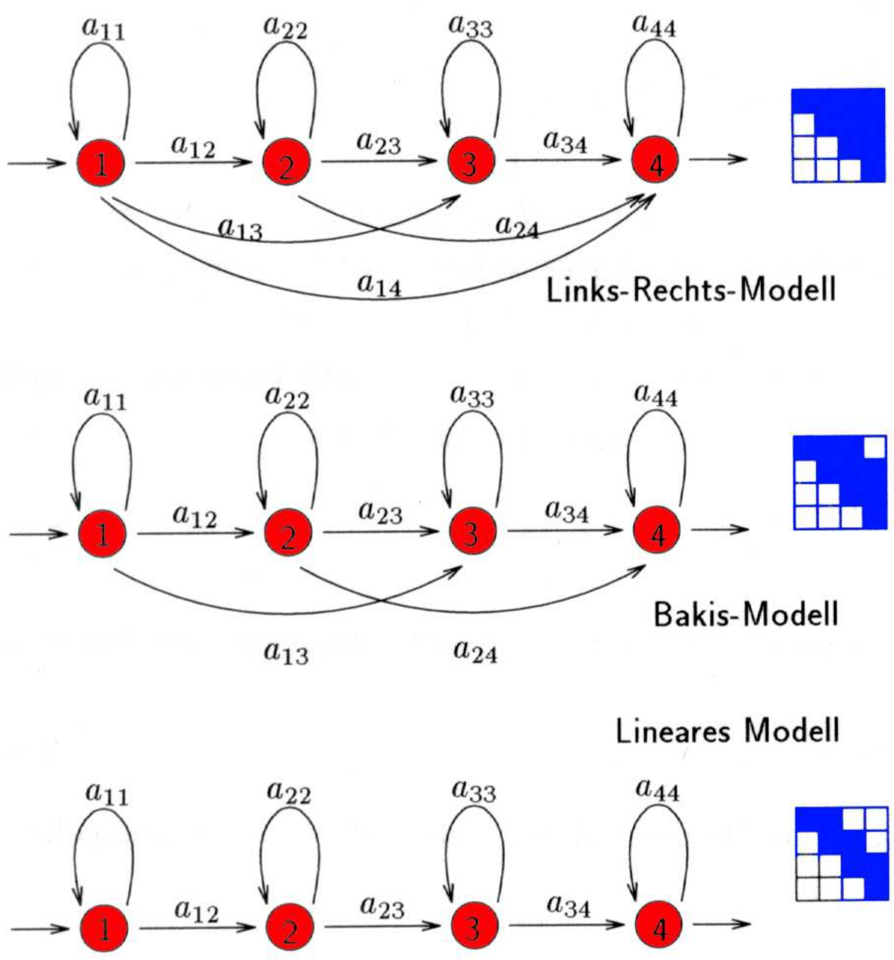


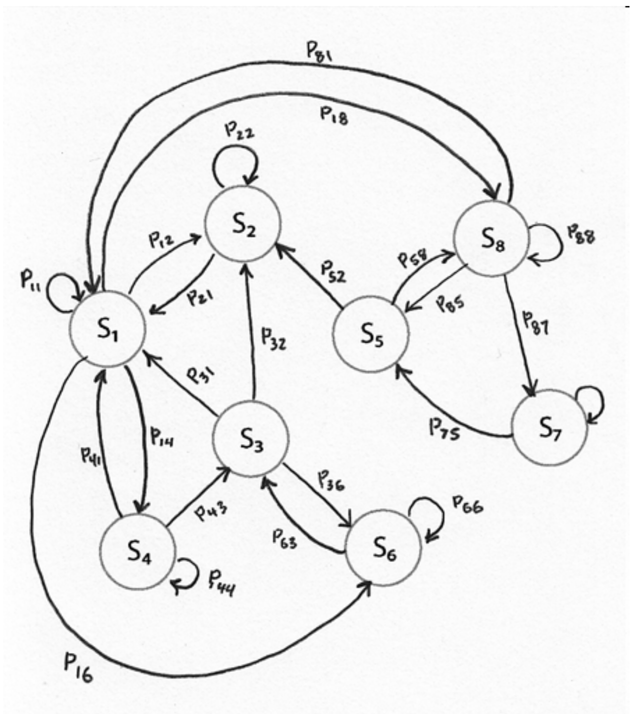

**GMM**

$$b_s(\vec{x}) = \sum_{m=1}^{M} c_{sm} \frac{1}{\sqrt{(2\pi)^P \det \Sigma_{sm}}} \cdot \exp\left[-0.5\left(\vec{x} - \vec{\bar{x}}_{sm}\right)^T \Sigma_{sm}^{-1} \left(\vec{x} - \vec{\bar{x}}_{sm}\right)\right]$$

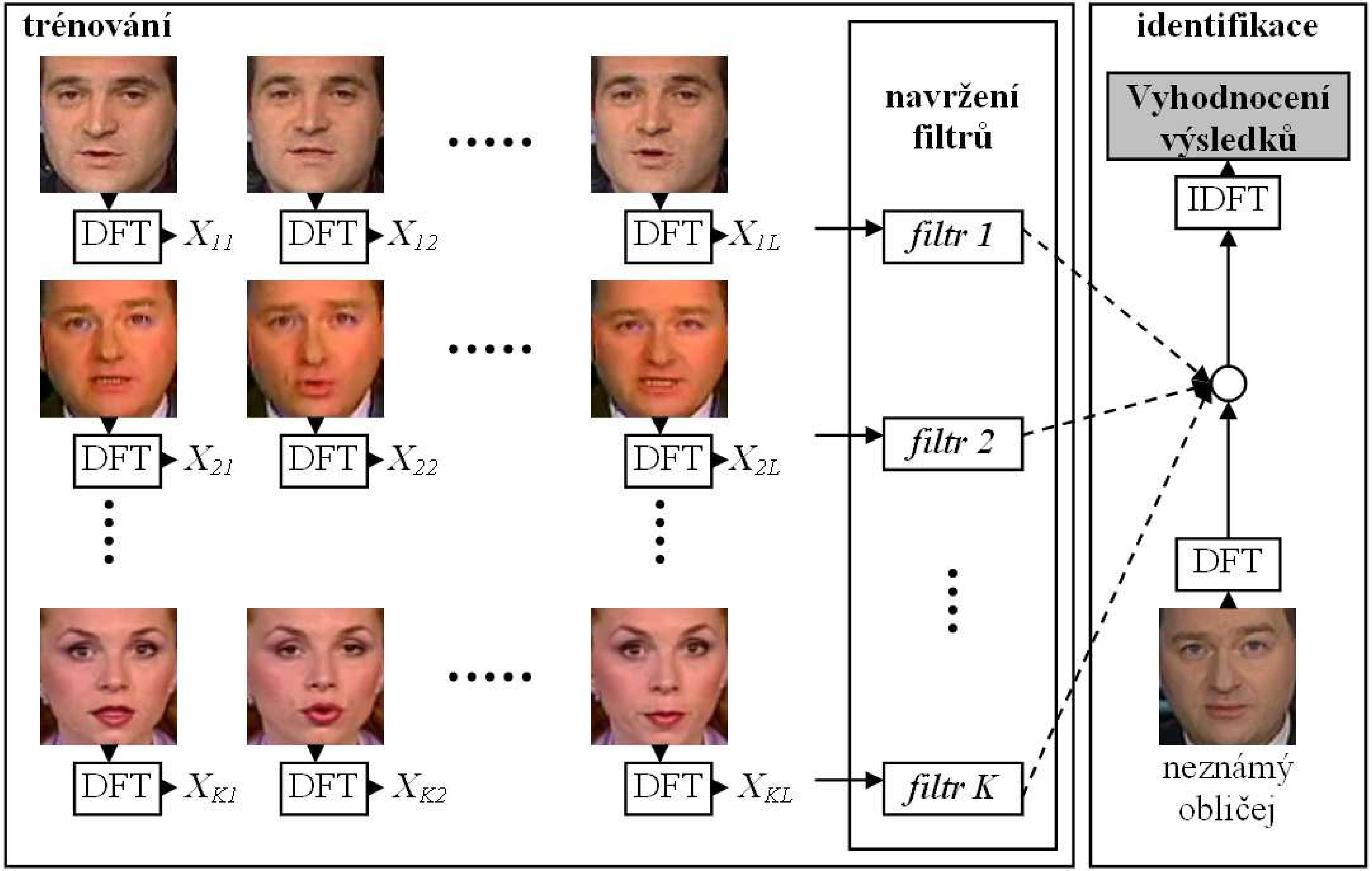
$$b_s(\vec{x}) = \prod_{t=1}^{T} \left( \sum_{m=1}^{M} c_{stm} \frac{1}{\sqrt{(2\pi)^P \det \Sigma_{stm}}} \cdot \exp\left[-0.5\left(\vec{x} - \vec{\bar{x}}_{stm}\right)^T \Sigma_{stm}^{-1} \left(\vec{x} - \vec{\bar{x}}_{stm}\right)\right] \right)^{\gamma_t}$$

# HMM - Hidden Markov Model



# MACE - Minimum Average Correlation Energy



# MACE - Minimum Average Correlation Energy

**Návrh filtru MACE** $\boldsymbol{H} = \boldsymbol{D}^{-1}\boldsymbol{X}\left(\boldsymbol{X}^{+}\boldsymbol{D}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{u}$

X – matice z vektorů: DFT počítaná z trénovacích obrazů, lexikograficky řazené

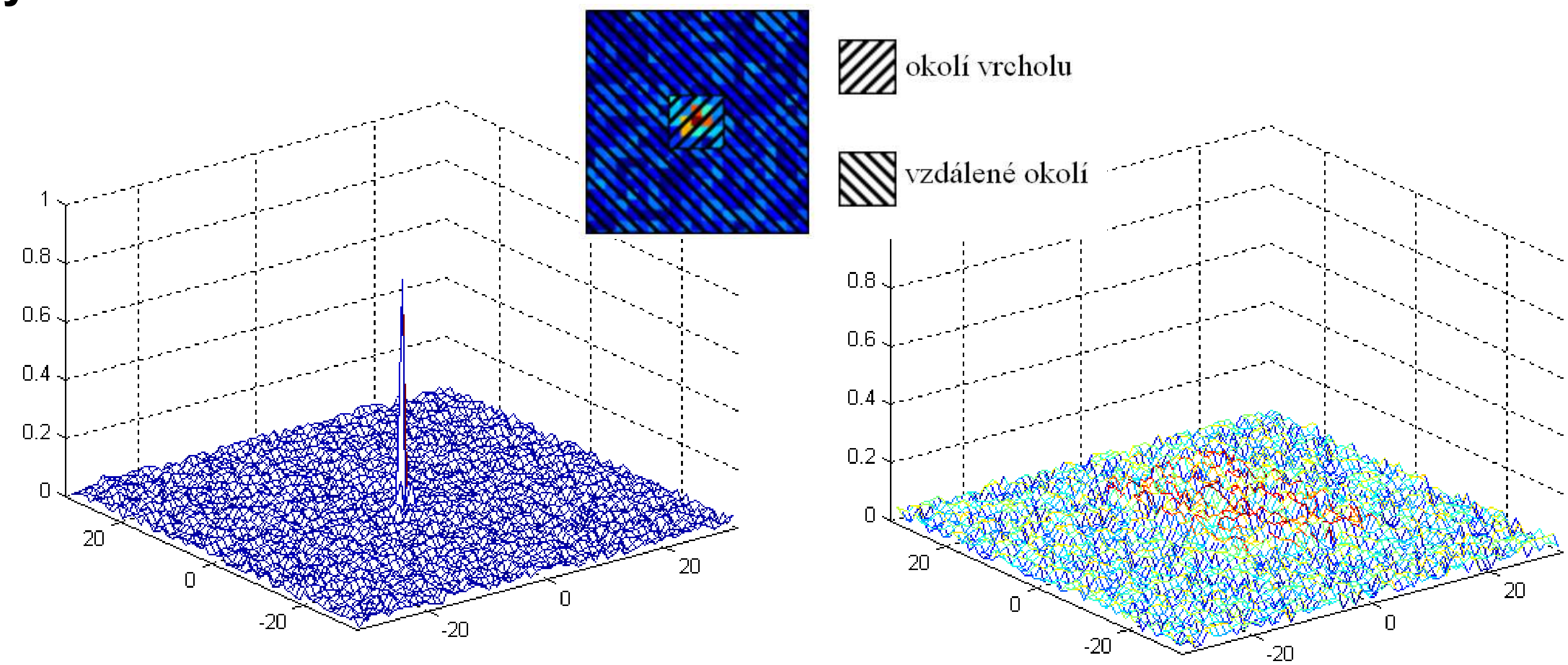
$X^+$– matice transponovaná a komplexně sdružená k X

u – sloupcový vektor, hodnoty obvykle 1

D – diagonální matice: hodnoty spočítané jako průměr výkonového spektra z řádků matice X

$$d_{jj} = \frac{1}{L}\sum_{i=1}^{L}\left|\boldsymbol{x}_i\,(j)\right|^2 \qquad j = 1,\ldots,N$$

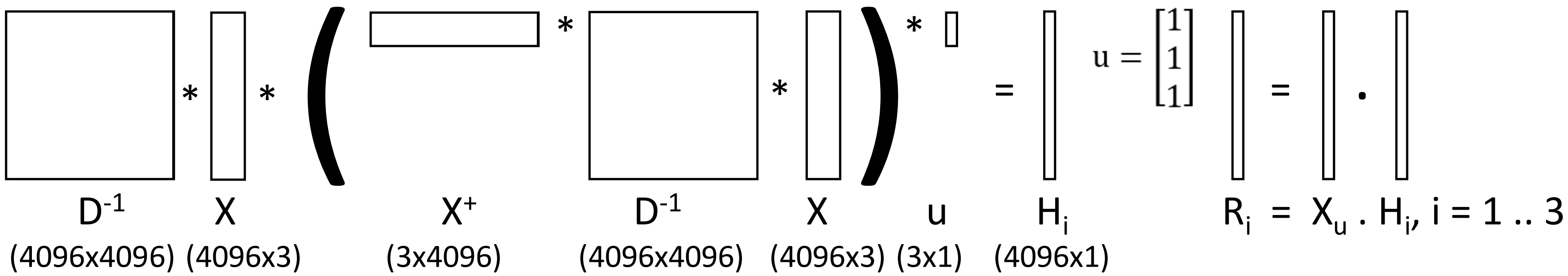
**Vyhodnocení**

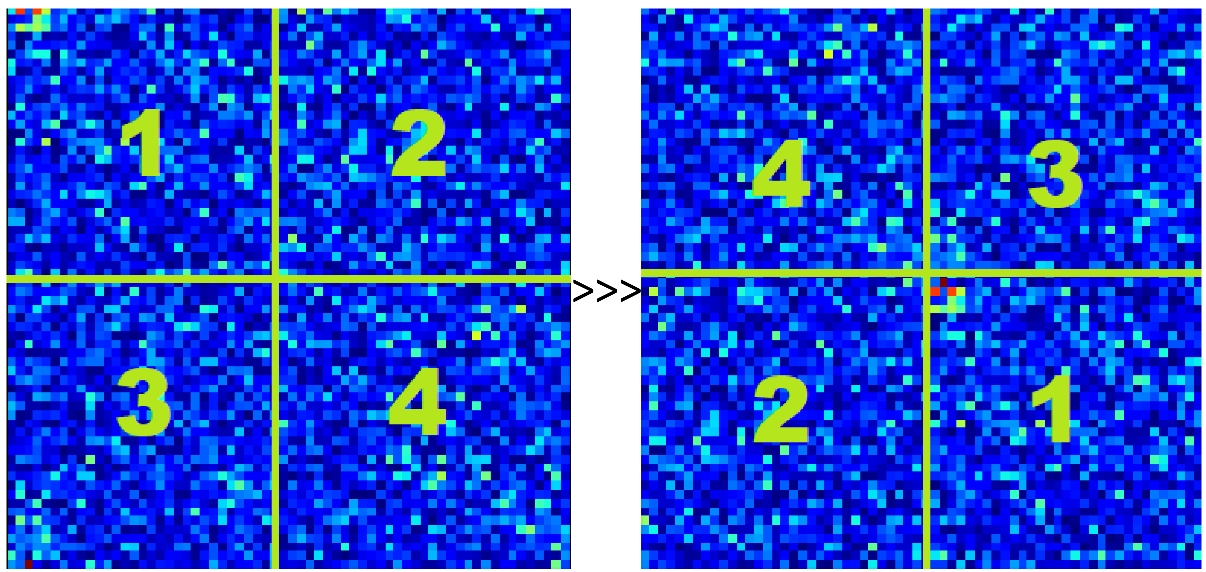


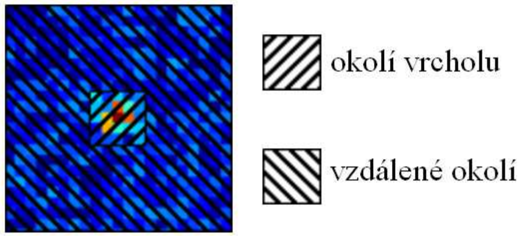
# MACE - Minimum Average Correlation Energy

**příklad:** 3 třídy (různí lidé), 3 snímky (64x64) od každého -> 3 filtry

Návrh filtru MACE $H = D^{-1}X\left(X^{+}D^{-1}X\right)^{-1}u$



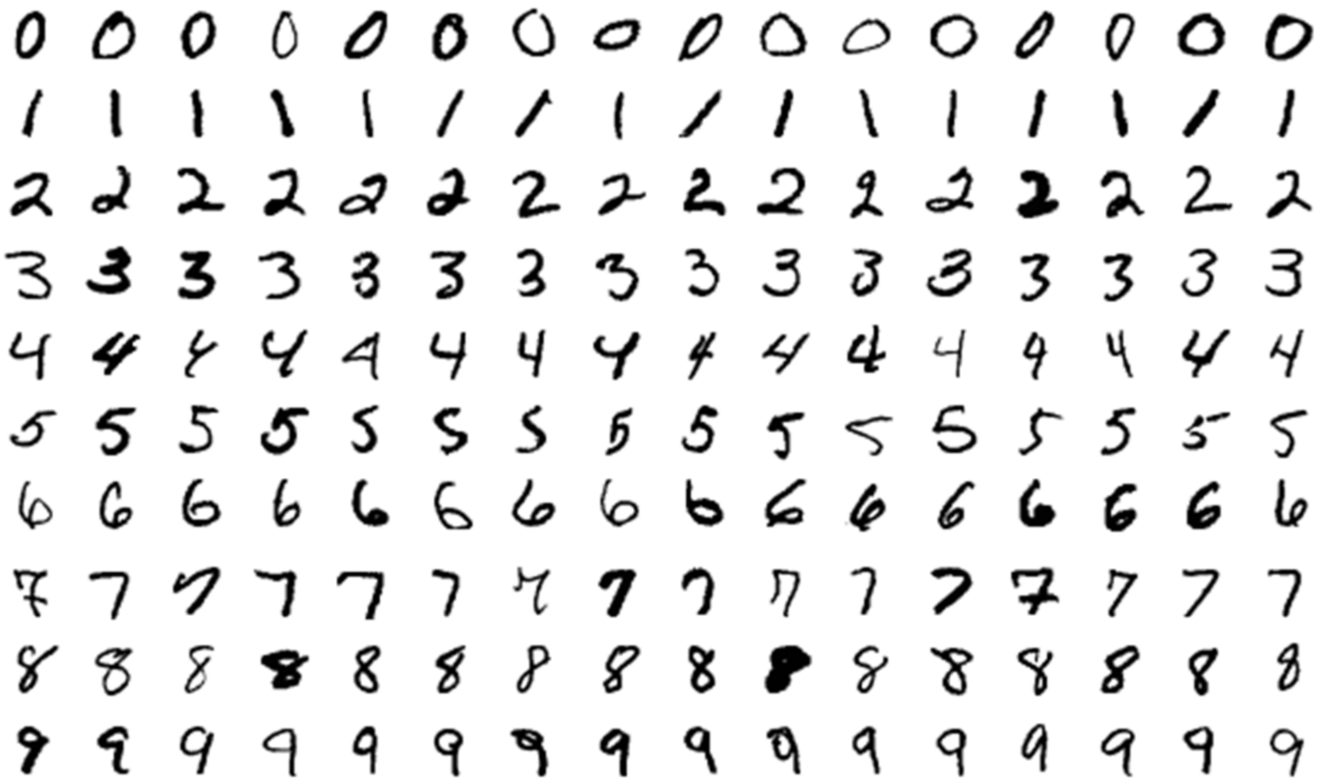




okolí (O) 20x20, vrchol (V) 10x10

$$RES_i = \frac{\max(V) - \text{mean}(O\ (\text{bez } V))}{\text{std}(O\ (\text{bez } V))}$$

# MNIST dataset

- Modified National Institute of Standards and Technology database (1994)
- 60 000 obrázků pro trénink, 10 000 pro testování



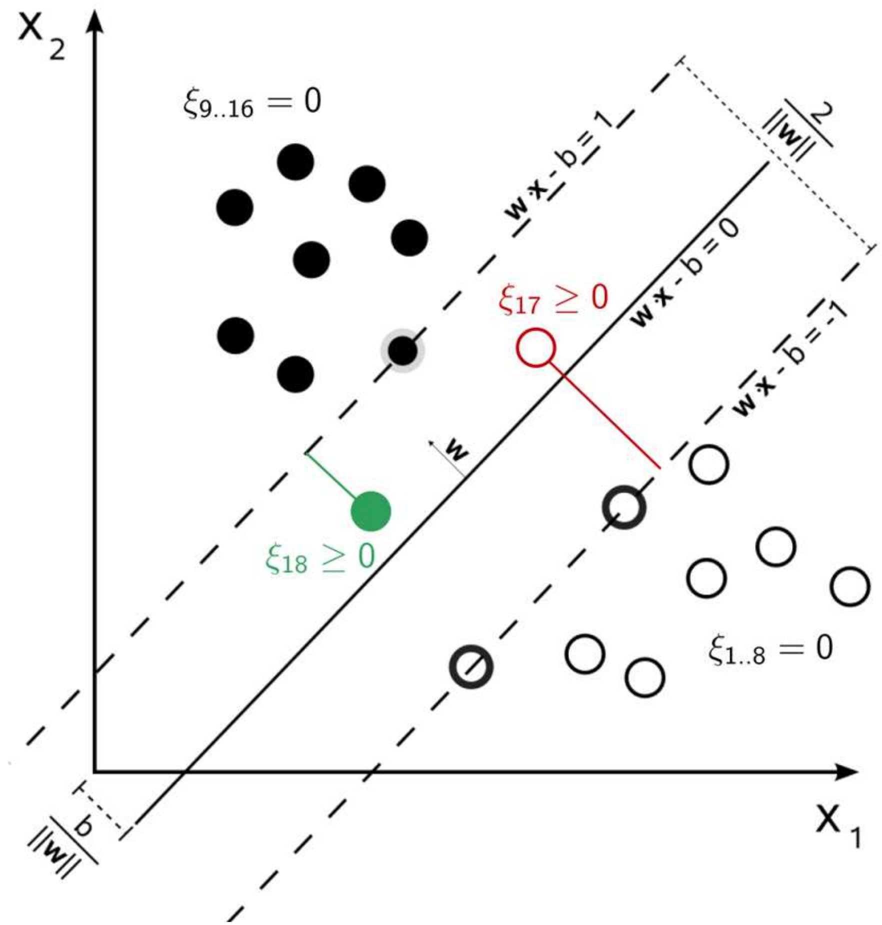
# MNIST dataset

| Type | Classifier | Distortion | Preprocessing | Error rate (%) |
|---|---|---|---|---|
| Linear classifier | Pairwise linear classifier | None | Deskewing | 7.6[10] |
| Decision stream with Extremely randomized trees | Single model (depth > 400 levels) | None | None | 2.7[28] |
| K-Nearest Neighbors | K-NN with rigid transformations | None | None | 0.96[29] |
| K-Nearest Neighbors | K-NN with non-linear deformation (P2DHMDM) | None | Shiftable edges | 0.52[30] |
| Boosted Stumps | Product of stumps on Haar features | None | Haar features | 0.87[31] |
| Non-linear classifier | 40 PCA + quadratic classifier | None | None | 3.3[10] |
| Random Forest | Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC)[32] | None | Simple statistical pixel importance | 2.8[33] |
| Support-vector machine (SVM) | Virtual SVM, deg-9 poly, 2-pixel jittered **2002** | None | Deskewing | 0.56[34] |
| Neural network | 2-layer 784-800-10 | None | None | 1.6[35] |
| Neural network | 2-layer 784-800-10 **2003** | Elastic distortions | None | 0.7[35] |
| Deep neural network (DNN) | 6-layer 784-2500-2000-1500-1000-500-10 **2010** | Elastic distortions | None | 0.35[36] |
| Convolutional neural network (CNN) | 6-layer 784-40-80-500-1000-2000-10 | None | Expansion of the training data | 0.31[37] |
| Convolutional neural network | 6-layer 784-50-100-500-1000-10-10 | None | Expansion of the training data | 0.27[38] |
| Convolutional neural network (CNN) | 13-layer 64-128(5x)-256(3x)-512-2048-256-256-10 | None | None | 0.25[22] |
| Convolutional neural network | Committee of 35 CNNs, 1-20-P-40-P-150-10 | Elastic distortions | Width normalizations | 0.23[17] |
| Convolutional neural network | Committee of 5 CNNs, 6-layer 784-50-100-500-1000-10-10 | None | Expansion of the training data | 0.21[24][25] |
| Random Multimodel Deep Learning (RMDL) | 10 NN-10 RNN - 10 CNN | None | None | 0.18[27] |
| Convolutional neural network | Committee of 20 CNNS with Squeeze-and-Excitation Networks[39] | None | Data augmentation | 0.17[40] |
| Convolutional neural network | Ensemble of 3 CNNs with varying kernel sizes **2020** | None | Data augmentation consisting of rotation and translation | 0.09[41] |

https://en.wikipedia.org/wiki/MNIST_database

# SVM - Support vector machines

- Vladimir Naumovich Vapnik (* 1936), Rusko, 1990 - AT&T Bell Labs (USA)
- metoda podpůrných vektorů
- Základní metoda – lineární klasifikace do dvou tříd
- Cíl -> nalézt nadrovinu, která prostor příznaků optimálně rozdělí -> trénovací data náležející odlišným třídám leží v opačných poloprostorech
- Optimální nadrovina -> hodnota minima vzdáleností příznaků od roviny je co největší, okolo nadroviny (na obě strany) je co nejširší pruh bez příznaků
- Maximální odstup (*maximal margin*) - pásmo necitlivosti, hraniční pásmo
- Pro popis nadroviny stačí hraniční příznaky, těch bývá malé množství – podpůrné vektory (support vectors)



# SVM - Support vector machines

- Trénovací dataset ($\mathbf{x}_1$, $y_1$) … ($\mathbf{x}_n$, $y_n$) >>> $\mathbf{x} = (x_1, x_2)$, y = 1 nebo -1 (dle třídy)

- Optimální nadrovina -> hodnota minima vzdáleností $\mathbf{x}_i$ od roviny je co největší

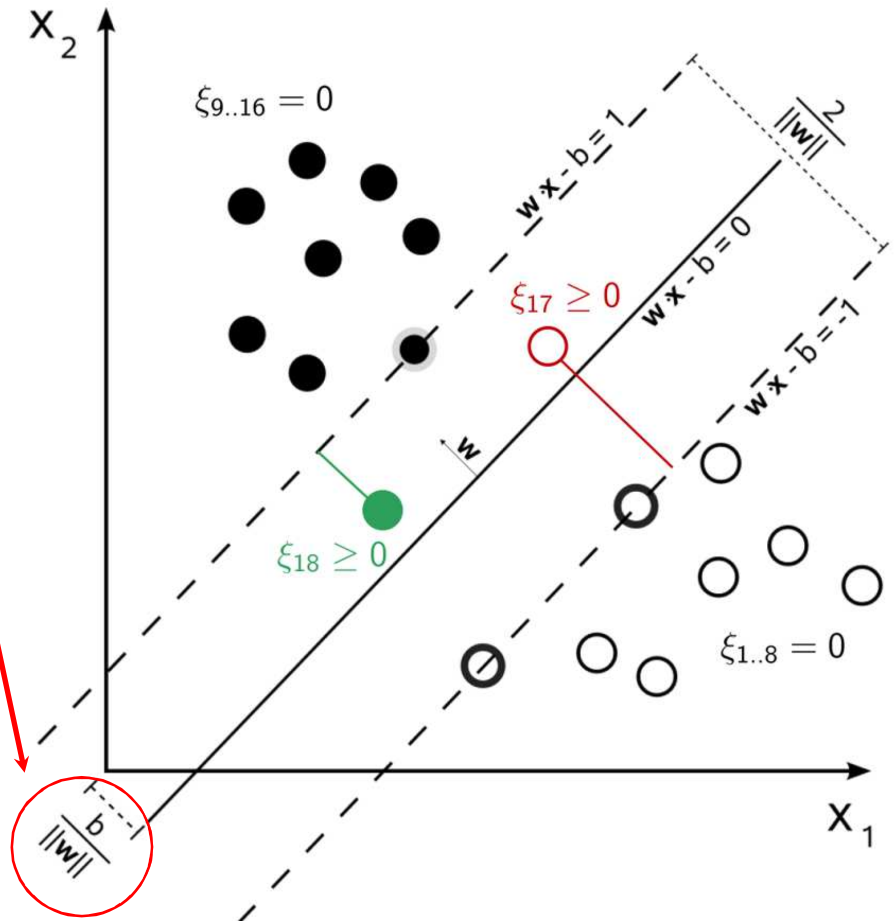
- Nadrovina $\mathbf{w}.\mathbf{x} - b = 0$

- **w** normála nadroviny, ||**w**|| Euklidovská norma

- |b| / ||**w**|| vzdálenost nadroviny od počátku souřadnic

- Nejkratší vzdálenost nejbližšího bodu z (1) $d_+$ a (-1) $d_-$, $d_+ + d_-$ je šířka hraničního pásma

- body na hranici hraničního pásma jsou pod-půrné vektory, pokud je odebereme, tak se změní Poloha optimální nadroviny



# SVM - Support vector machines

- Trénovací dataset ($\mathbf{x}_1$, $y_1$) … ($\mathbf{x}_n$, $y_n$) >>> $\mathbf{x} = (x_1, x_2)$, y = 1 nebo -1 (dle třídy)

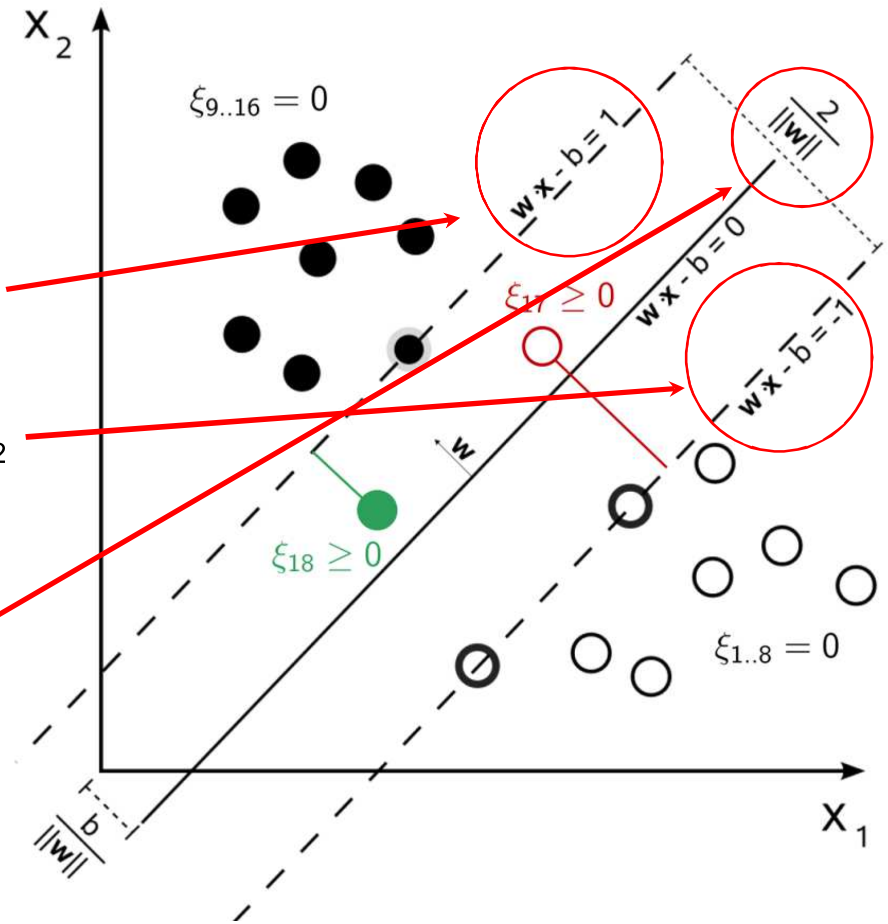
$\mathbf{x_i} \cdot \mathbf{w} - b \geq +1$ pro $y_i = +1$
$\mathbf{x_i} \cdot \mathbf{w} - b \leq -1$ pro $y_i = -1$

>>>
$y_i(\mathbf{x_i} \cdot \mathbf{w} - b) - 1 \geq 0$

- body ($y_i = +1$) kde platí = 1, leží v nadrovině $H_1$
**w**.**x** - b = 1

- body ($y_i = -1$) kde platí = -1, leží v nadrovině $H_2$
**w**.**x** - b = -1

- nadroviny $H_1$ a $H_2$ jsou kolmé na normálový vektor **w,** vzdálenost od rozdělující nadroviny je 1 /||**w**||, dohromady 2 / ||**w**||



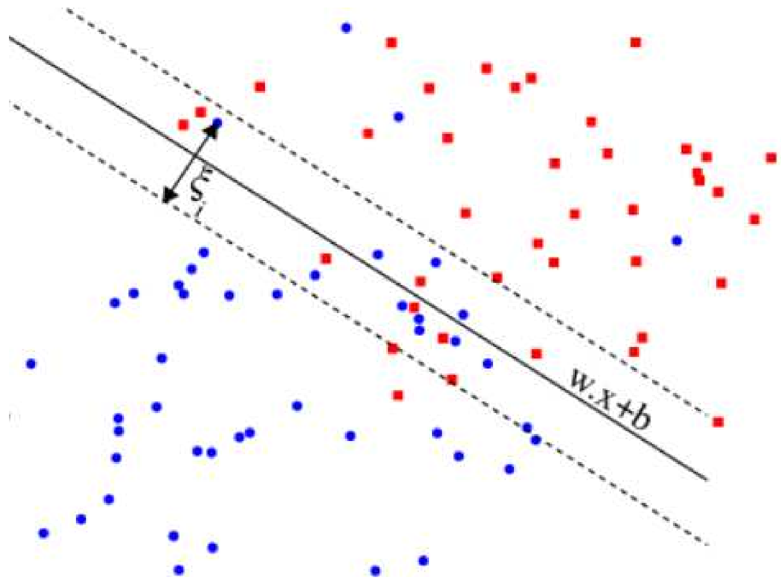
# SVM - Support vector machines

- Trénovací dataset ($\mathbf{x}_1$, $y_1$) … ($\mathbf{x}_n$, $y_n$) >>> $\mathbf{x} = (x_1, x_2)$, y = 1 nebo -1 (dle třídy)

**Neseparabilní případ - Soft-margin (Hard-margin)**
>>> minimalizace chyby, E ≥ 0

$\mathbf{x_i} \,.\, \mathbf{w} - b \geq +1 - E_i$ pro $y_i = +1$
$\mathbf{x_i} \,.\, \mathbf{w} - b \leq -1 - E_i$ pro $y_i = -1$

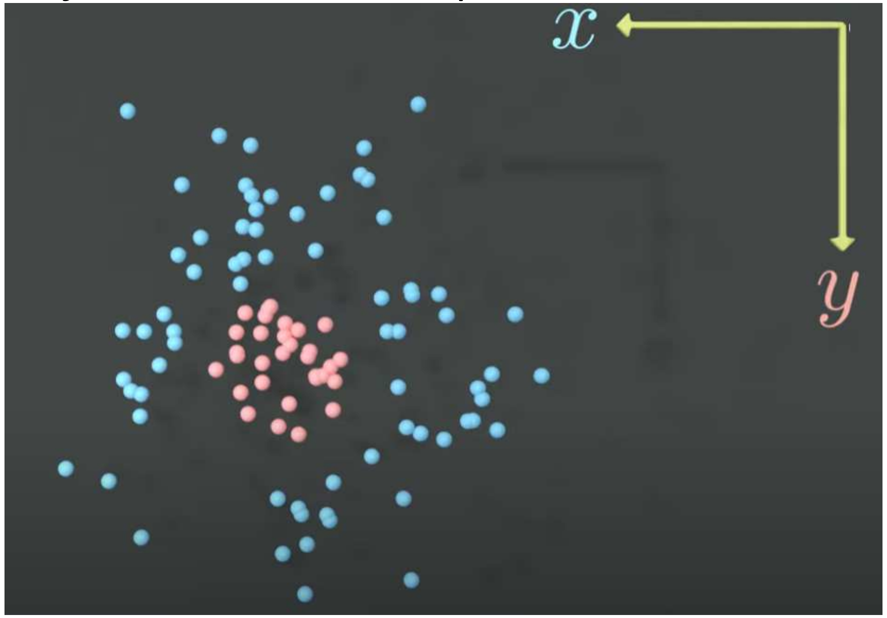
$$\min_{\mathbf{w},\xi} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\}$$

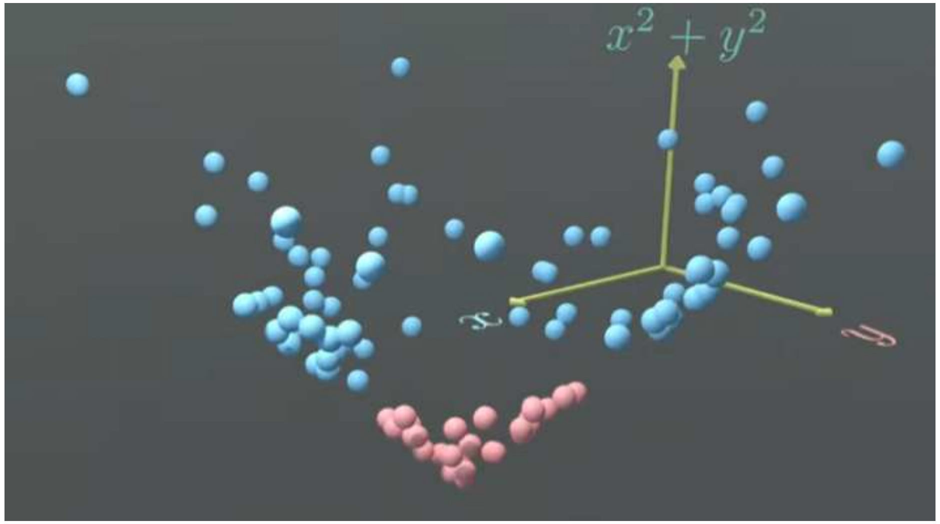


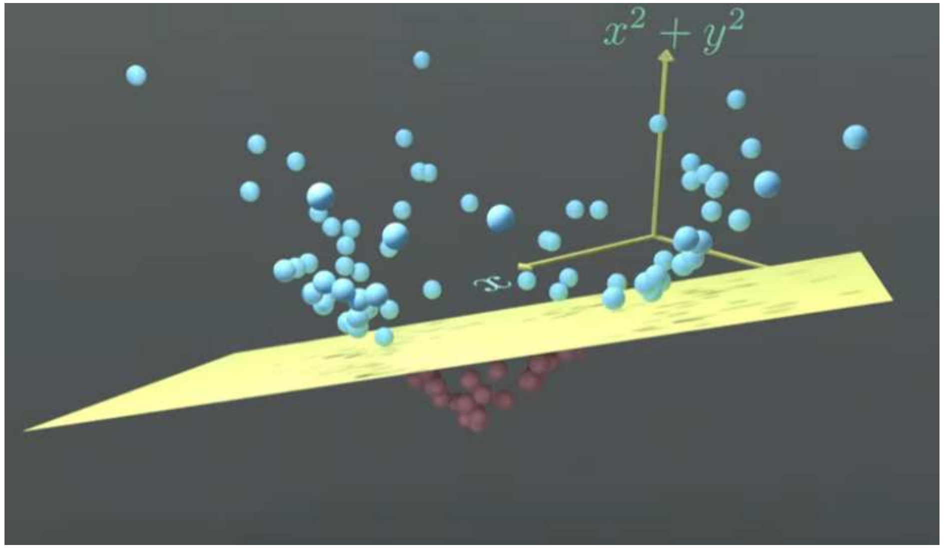

- C volitelný parametr, větší C = větší penalizace chyb

# SVM - Support vector machines

**Nelineární** SVM >>> jádrový trik (kernel trick) transformace dat do prostoru vyšší dimenze, kde jsou datalineárně separabilní




Support Vector Machine (SVM) in 2 minutes: https://youtu.be/_YPScrckx28

# SVM - Support vector machines

- Řešení (**trénování**)>>> analytické řešení >>> jen pokud máme málo dat, nebo jsme schopni určit, které příznaky budou podpůrnými vektory

- Data jsou většinou zašuměná - Neseparabilní případ - Soft-margin >>> numerické řešení + speciální metody programování (kvadratického programování s lineárními vazbovými podmínkami, "hroudová" metoda (chunking method)

- Lagrangeova formulace SVM – namísto $y_i$ ($\mathbf{x}_i$ . $\mathbf{w}$ - b) - 1 ≥ 0 jsou využity Lagrangeovy multiplikátory

- Karushovy-Kuhnovy-Tuckerovy (KKT) podmínky >>> pro optimální řešení v úloze nelineárního programování

- **Testování** (klasifikace, identifikace, rozpoznávání) – neznámý objekt – parametry x =($x_1$, $x_2$) spočítáme sgn(**w** . **x** + b) a dle znaménka zařadíme do jedné ze tříd

- Využití SVM: dobře parametrizovatelné objekty, identifikace obličejů, OCR, bioinformatika (klasifikace proteinů), regresní analýza

- **+** oproti ANN vždy po trénování nalezeno řešení, malý počet parametrů, poměrně rychlé
- **-** u rozsáhlé úlohy (miliony podpůrných vektorů) >>> velké nároky na paměť a rychlost

https://scikit-learn.org/stable/modules/svm.html