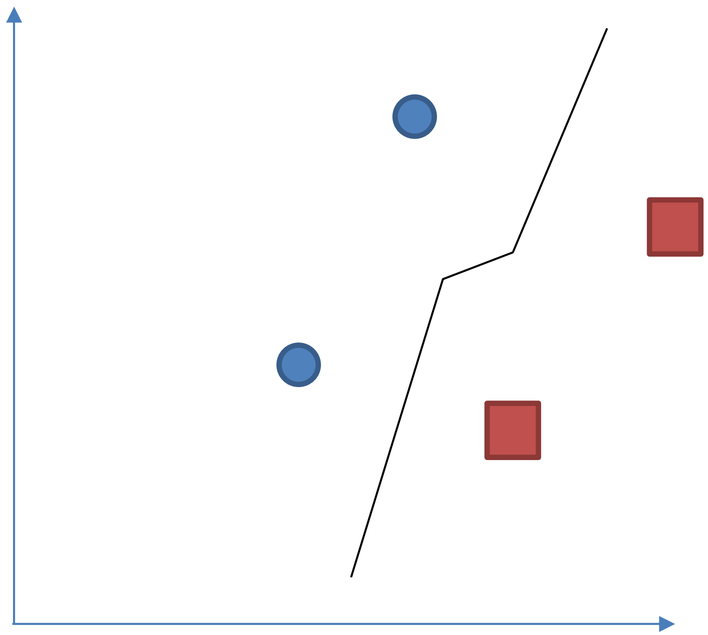
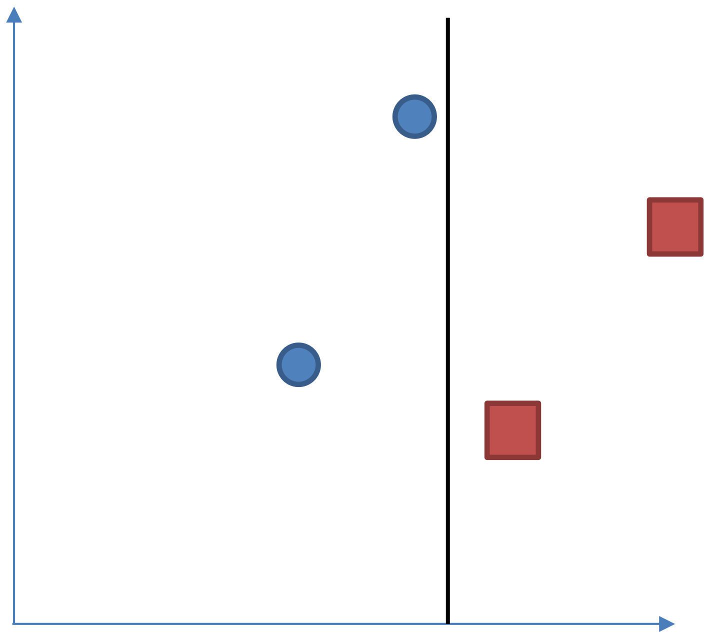
Artificial Intelligence Fundamentals

Learning: SVM

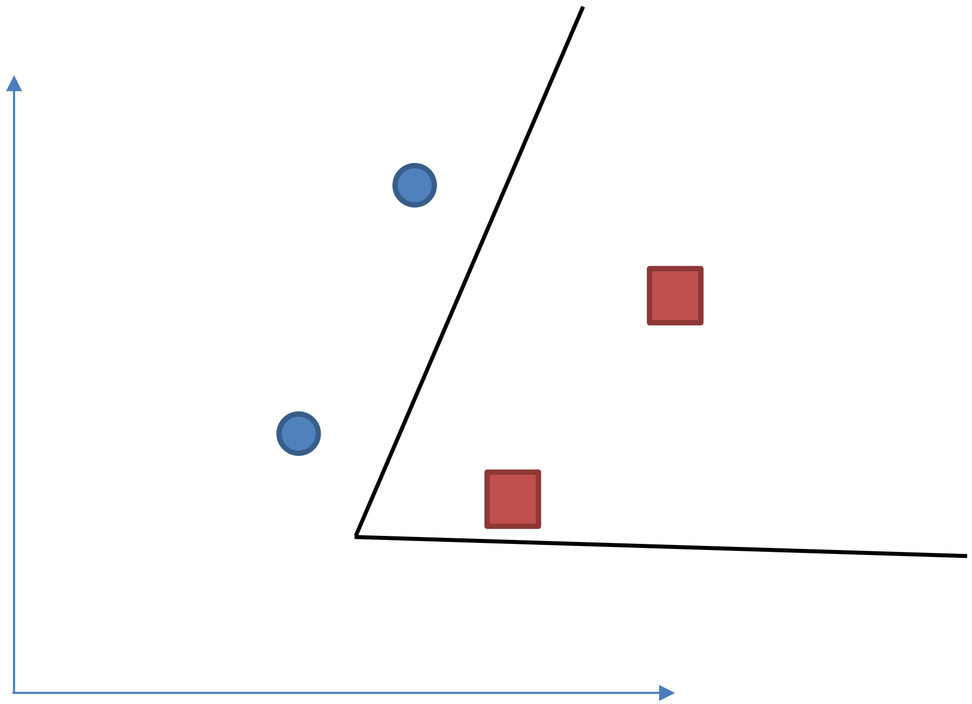
Separate 2 classes - kNN



Separate 2 classes – Decision tree

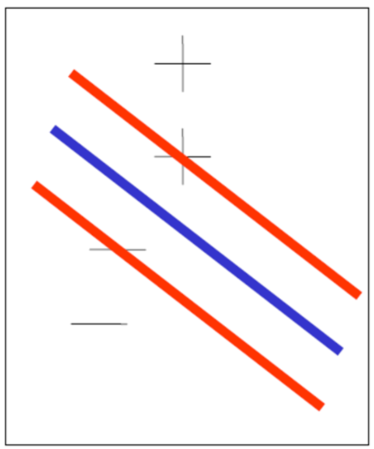


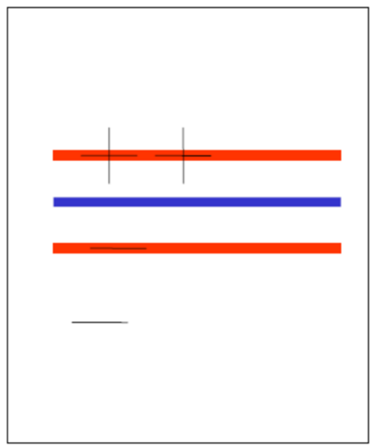
Separate 2 classes – NN



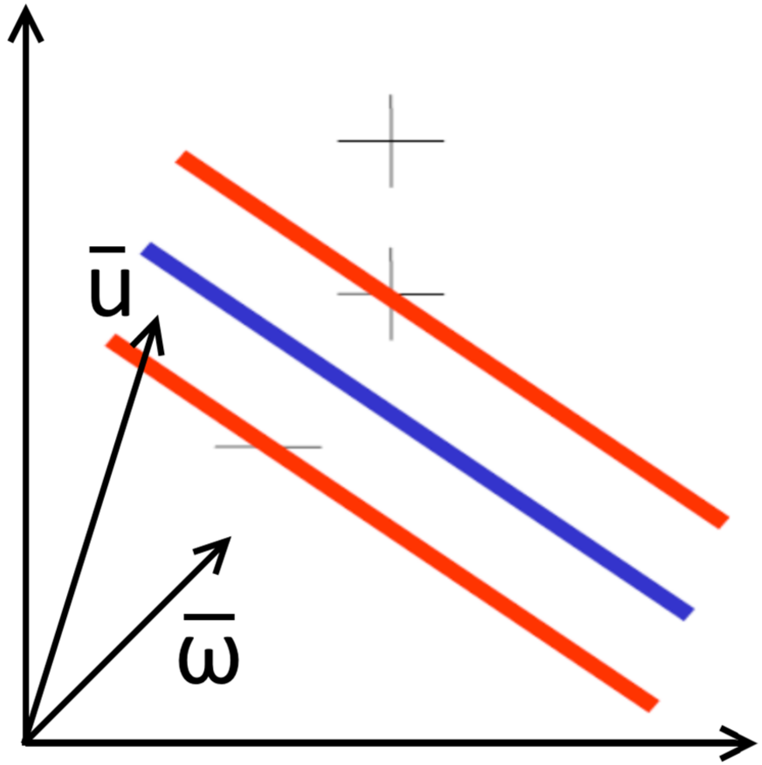
Widest street approach

- Divide a space with decision boundaries (NN, DT, kNN)
- A very sophisticated idea
- Vladimir Vapnik and Alexey Chervonenkis (1963)
- Separating the + from the – as wide as possible





Step 1



- Widest space that separate the + from –
- $\overline{\omega}$ - A vector perpendicular to the median line of that space (street)
- $\overline{u}$ - An unknown example
- Is on the right side of the street?
- Project the $\overline{u}$ on the $\overline{\omega}$
- DECISION RULE : we have a positive example if:

$$\overline{\omega} \bullet \overline{u} \geq c \ or$$
$$\overline{\omega} \bullet \overline{u} + b \geq 0$$

Step 2

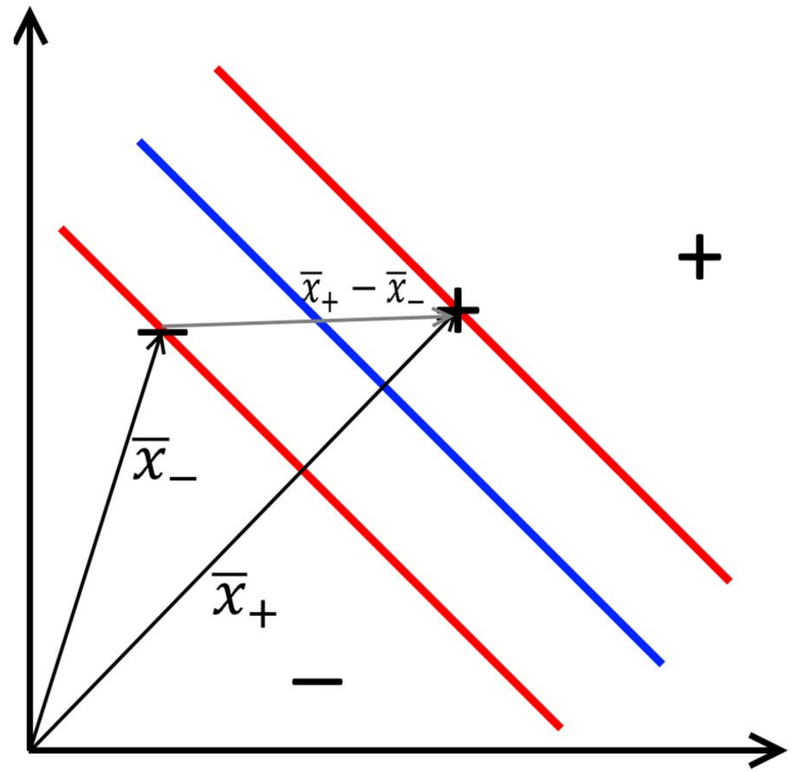
$$\overline{\omega} \cdot \overline{x_{i_+}} + b \geq 1$$
$$\overline{\omega} \cdot \overline{x_{i_-}} + b \leq -1$$

$$y_i(\overline{\omega} \cdot \overline{x_i} + b) \geq 1$$
$$y_i(\overline{\omega} \cdot \overline{x_i} + b) \geq 1$$

- Introducing a new variable for mathematical convenient
- y_i such that y_i = +1 for positive samples and -1 for negative samples

$$y_i(\overline{\omega} \cdot \overline{x_i} + b) - 1 \geq 0$$
$$y_i(\overline{\omega} \cdot \overline{x_i} + b) - 1 = 0 \text{ for } x_i \text{on the boundary}$$

Step 3 – width of the street



Width of the street is:

$$(\overline{x_+} - \overline{x_-}) \cdot \frac{\overline{\omega}}{\|\omega\|} =$$

$$\frac{1 - b + 1 + b}{\|\omega\|} = \frac{2}{\|\omega\|}$$

- Maximize that expression $\frac{2}{\|\omega\|}$, maximize $\frac{1}{\|\omega\|}$, minimize $\|\omega\|$, minimize $\frac{1}{2}\|\omega\|^2$

Step 4 – development

- Minimize with the help of Lagrange multipliers
- $L = \frac{1}{2}\|\overline{\omega}\|^2 - \sum \alpha_i [y_i(\overline{\omega} \cdot \overline{x_i} + b) - 1]$
- Derivatives and set it to 0

$$\frac{\partial L}{\partial \overline{\omega}} = \overline{\omega} - \sum_i \alpha_i y_i \overline{x_i} \qquad \overline{\omega} = \sum_i \alpha_i y_i \overline{x_i}$$

(Linear combination of the inputs)

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i \qquad \sum_i \alpha_i y_i = 0$$

- $L = \frac{1}{2}\left(\sum_i \alpha_i y_i \overline{x_i}\right) \cdot \left(\sum_j \alpha_j y_j \overline{x_j}\right) - \sum_i \alpha_i y_i \overline{x_i} \cdot \left(\sum_j \alpha_j y_j \overline{x_j}\right)$ - $\sum_i \alpha_i y_i b + \sum_i \alpha_i$

- maximum of the following expression

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \overline{x_i} \cdot \overline{x_j}$$

- optimization depends only on the dot product of pairs of samples

Step 5 – Switch to another perspective

$$\sum \alpha_i y_i \overline{x_i} \cdot \bar{u} + b \geq 0 \text{ then } +$$

- Decision rule depends only on the dot product of samples vectors and unknown
- ϕ - a transformation that will take us from the space we're in into a space where things are more convenient

$$\Phi(\overline{x_i}) \cdot \Phi\left(\overline{x_j}\right) \text{ to maximize}$$

- Find a transformation K - *kernel*

$$K\left(\overline{x_i}, \overline{x_j}\right) = \Phi(\overline{x_i}) \cdot \Phi\left(\overline{x_j}\right)$$

- I don’t need to know the transformation into another space, just to know the K – the dot product of transformations into another space

$$(\bar{u} \cdot \bar{v} + 1)^n - polinomial$$

$$e^{-\frac{(\bar{u} - \bar{v})}{\sigma}} - gaussian$$

Conclusions

- Produces global solutions
- Kernel functions – convex
- Immune against local minima
- Not immune against overfitting