GHENT UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

E016350 - Artificial Intelligence

Lecture 7

Machine learning

White-box and black-box ML models

Aleksandra Pizurica

Ghent University

Spring 2024

What is a 'white-box' model?

Black-box: result/decision reached without explaining or showing how.
The internal processes used and the various weighted factors remain unknown.

So, 'white-box' models should do the opposite: give not only the result but also some transparency i.e., posses some level of **interpretability** or at least **explainability**.

We face **accuracy vs. interpretability** trade-off.

Examples of 'black-box' models are deep neural networks and random forests.

But the delineation is not always clear:
– inconsistencies in characterizing some models as white or black-box
– explainable AI (XAI) tools aim at explainability/interpretability for black-box models

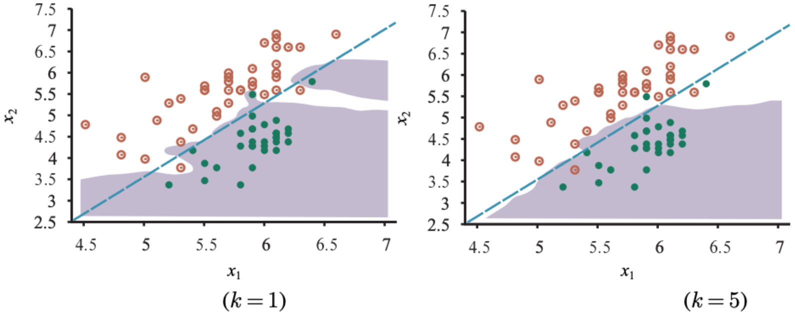
Examples of white-box ML models

- Linear regression
- Logistic regression
- (Some) Nearest-neighbours models
- Support vector machines
- Decision trees
- Generalized additive models (GAMs)

Nonparametric models

- Models that we studied so far (like linear and logistic regression and neural networks) use the training data to estimate a fixed set of parameters $\mathbf{w}$.
- A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model.
- A nonparameteric model cannot be characterized by a bounded set of parameters
  - Nearest-neighbor models
  - Locality-sensitive hashing
  - Non-parametric regression
  - Support Vector Machines (SVM)

Nearest-neighbor models



Determine the class label y of the data point $\mathbf{x}$ as follows:

1. Find k examples that are nearest to $\mathbf{x}$, (k nearest neighbours, thus name k-NN)
2. Take the most common class label in this set

   E.g., if $k = 3$ and the three labels are $\{0, 1, 0\}$ assign $y = 0$

Questions: which **distance metric**? How to set k?

Distance metrics

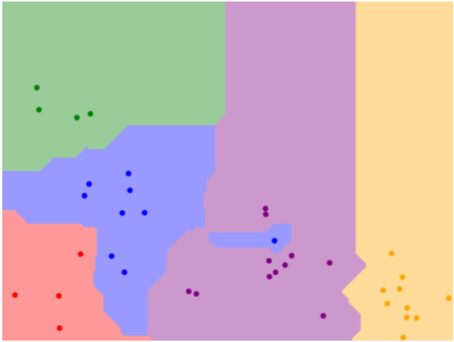
Typically, distances are measured with **Minkowski** distance or L^p norm:

$$L^p(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(\sum_k |x_k^{(i)} - x_k^{(j)}|^p\right)^{1/p}$$

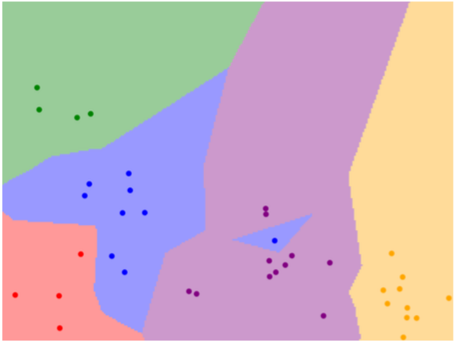
Special cases of particular interest:

- with $p = 2$: Euclidean distance
- with $p = 1$: Manhattan distance

k-NN example



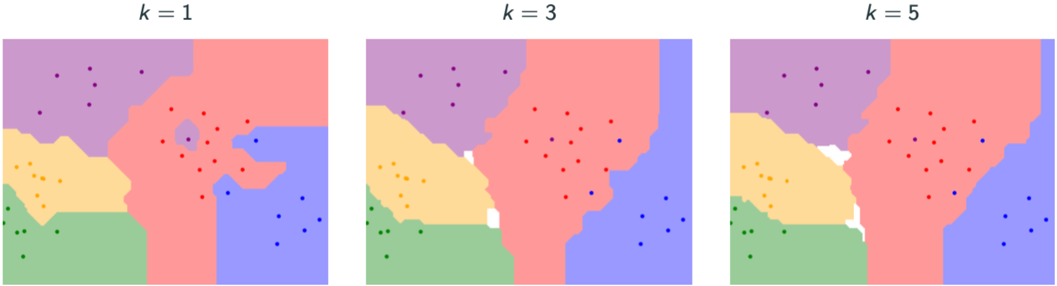
with L^1 distance



with L^2 distance

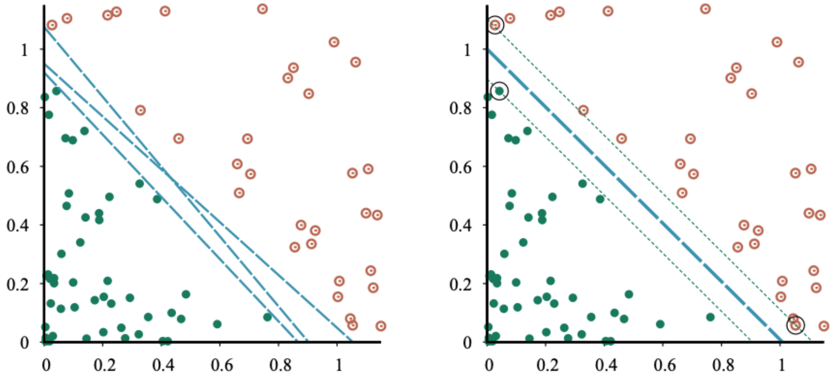
Try it yourself: `http://vision.stanford.edu/teaching/cs231n-demos/knn/`

k-NN example



Try it yourself: `http://vision.stanford.edu/teaching/cs231n-demos/knn/`

Support vector machines (SVM)



Goal: Find the hyperplane with the largest distance to nearest training-data point of any class (largest margin). The examples closest to the separator: support vectors.

The larger the margin, the lower the generalization error $\rightarrow$ less likely overfitting.

Support Vector Machines (SVM)

Huge popularity in the early 2000s.

Now overshadowed by deep learning and random forests.

Three key assets:

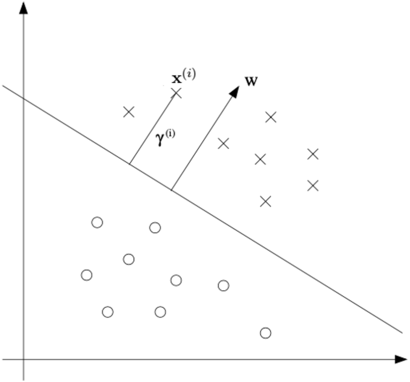
1. Maximum margin separator (largest possible distance to examples)
   – Helps to **generalize well**
2. Kernel trick: hyperplanes in higher dimensions separate non-linear data
3. Non-parameteric and in contrast to K-NN need to keep only a small number of examples – flexible to represent complex functions while **robust to overfitting**

Support Vector Machines (SVM)

Consider labels $y \in \{-1, +1\}$

$$h_{\mathbf{w},b}(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

(Note we dropped the convention $x_0 = 1$ and keep instead b as a separate parameter)



Functional margin

Definition (Functional margin)

Given a training example $(\mathbf{x}^{(i)}, y^{(i)})$, the functional margin of $(\mathbf{w}, b)$ with respect to the training example is

$$\hat{\gamma}^{(i)} = y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)$$

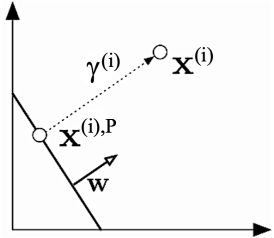
Hence, a **large functional margin** represents a **confident and correct prediction**. But scale invariant (e.g., replacing $(\mathbf{w}, b)$ by $(2\mathbf{w}, 2b)$ no effect on the classifier)

Definition (Functional margin with respect to data set)

Given a training set $\mathcal{D}_{train} = \{(\mathbf{x}^{(i)}, y^{(i)}); i = 1, \ldots, n\}$, the functional margin of $(\mathbf{w}, b)$ with respect to $\mathcal{D}_{train}$ is

$$\hat{\gamma} = \min_{i=1,\ldots,n} \hat{\gamma}^{(i)}$$

What is the distance of a point to the separating hyperplane?



Let $\mathbf{x}^{(i),P}$ be the orthogonal projection of the training example $\mathbf{x}^{(i)}$ on the separating hyperplane. Then:

$$\mathbf{x}^{(i),P} = \mathbf{x}^{(i)} - \gamma^{(i)} \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

$$\mathbf{w}^\top \mathbf{x}^{(i),P} + b = 0$$

It follows that $\mathbf{w}^\top(\mathbf{x}^{(i)} - \gamma^{(i)} \frac{\mathbf{w}}{\|\mathbf{w}\|}) + b = 0 \implies \gamma^{(i)} = \frac{\mathbf{w}^\top \mathbf{x}^{(i)} + b}{\|\mathbf{w}\|} = \left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)^\top \mathbf{x}^{(i)} + \frac{b}{\|\mathbf{w}\|}$

Here, $\mathbf{x}^{(i)}$ was the ‘positive’ side of the decision boundary. In general:

$$\gamma^{(i)} = \frac{|\mathbf{w}^\top \mathbf{x}^{(i)} + b|}{\|\mathbf{w}\|} = y^{(i)} \left(\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)^\top \mathbf{x}^{(i)} + \frac{b}{\|\mathbf{w}\|} \right)$$

Geometric margin

Definition (Geometric margin)

Given a training example $(\mathbf{x}^{(i)}, y^{(i)})$, the geometric margin of $(\mathbf{w}, b)$ with respect to the training example is

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \right)^{\top} \mathbf{x}^{(i)} + \frac{b}{\|\mathbf{w}\|} \right)$$

Scale invariant, i.e., invariant to rescaling $(\mathbf{w}, b)$.

Note that if $\|\mathbf{w}\| = 1$ the geometric and functional margin are equal.

Definition (Geometric margin with respect to data set)

Given a training set $\mathcal{D}_{train} = \{(\mathbf{x}^{(i)}, y^{(i)}); i = 1, \ldots, n\}$, the geometric margin of $(\mathbf{w}, b)$ with respect to $\mathcal{D}_{train}$ is

$$\gamma = \min_{i=1,\ldots,n} \gamma^{(i)}$$

The optimal margin classifier

Goal: Find decision boundary $(\mathbf{w}, b)$ that maximizes the (geometric) margin

$$\max_{\gamma,\mathbf{w},b} \gamma \quad \text{such that} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq \gamma, \quad i = 1, \ldots, n$$

$$\|\mathbf{w}\| = 1$$

I.e., maximize γ, subject to each training example having functional margin at least γ.
The constraint $\|\mathbf{w}\| = 1$ ensures the functional margin equals geometric margin.

One can show this problem is equivalent to:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{such that} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \ldots, n$$

Convex quadratic objective with linear constraints

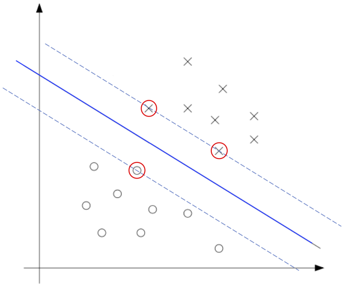
$\rightarrow$ can be solved with commercial quadratic programming (QP) code

– the solution gives the optimal margin classifier

Support vectors

We obtained the separating hyperplane by solving

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{such that} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \ldots, n$$



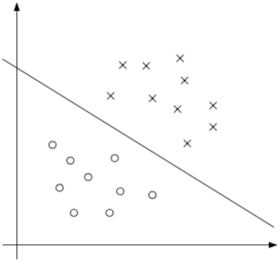
For the optimal $(\mathbf{w}, b)$ pair, some training points will have tight constraints, i.e.

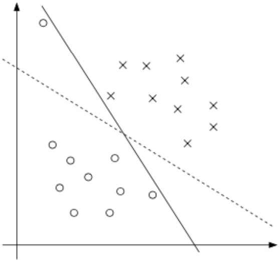
$$\mathbf{w}^\top \mathbf{x}^{(i)} + b = 1$$

These training points are the **support vectors**.

The support vectors define the maximum margin of the hyperplane to the data set and they therefore determine the shape of the hyperplane.

SVM with soft constraints





To be able to deal with non-linearly separable data and to be less sensitive to outliers, reformulate the optimization problem as

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{such that} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i \geq 0, , \quad i = 1, \ldots, n$$

SVM with soft constraints as linear classifier with hinge loss

It is interesting to see the link with the linear classifiers we studied before. For this, consider the value of ξ_i for $C \neq 0$.

$$\xi_i = \begin{cases} 1 - (\mathbf{w}^\top \mathbf{x}^{(i)} + b) & \text{if } \; y_i(\mathbf{w}^\top \mathbf{x}^{(i)} + b) < 1 \\ 0 & \text{if } \; y_i(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 \end{cases}$$

This is equivalent to the following closed form:

$$\xi_i = \max(1 - y_i(\mathbf{w}^\top \mathbf{x}^{(i)} + b), 0)$$

Note that this is hinge loss for the case of $\{-1, +1\}$ labels. Plugging into the objective of soft-constraints SVM:

$$\min_{\mathbf{w},b} \; \frac{1}{2} \underbrace{\|\mathbf{w}\|^2}_{L_2 \text{ reg.}} + C \sum_{i=1}^{n} \underbrace{\max(1 - y_i(\mathbf{w}^\top \mathbf{x}^{(i)} + b), 0)}_{\text{hinge loss}}$$

Optimal margin classifiers: the dual form

Previously posed (primal) optimization problem (P) for the optimal margin classifier:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{such that} \quad y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \ldots, n \qquad (P)$$

Using the Lagrangian: $\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1]$ one can arrive at the dual representation (D), in which the optimal solution is found by solving:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y^{(i)} y^{(j)} \alpha_i \alpha_j (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \quad \text{s.t.} \quad \forall i,\ \alpha_j \geq 0, \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \qquad (D)$$

Once we have found $\boldsymbol{\alpha}$, we can get back to $\mathbf{w}$ with

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)} \quad \text{(and straightforward to find from this also } b\text{)}$$

or we can stay in the dual representation.

Dual formulation for the soft-margin separator

For SVM with soft constraints, the dual problem is

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y^{(i)} y^{(j)} \alpha_i \alpha_j (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \quad \text{s.t.} \quad \forall i,\ 0 \leq \alpha_j \leq C, \ \ \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

Note the only difference with respect to the version without margin softening is that $\alpha_j \geq 0$ is replaced by $0 \leq \alpha_j \leq C$

Still holds that

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)} \quad \text{(and } b \text{ needs to be calculated)}$$

This also means that

$$\mathbf{w}^{\top}\mathbf{x} + b = \left(\sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)} \right)^{\top} \mathbf{x} + b = \sum_{i=1}^{n} \alpha_i y^{(i)} \underbrace{(\mathbf{x}^{(i)} \cdot \mathbf{x})}_{\text{dot product}} + b$$

Why dual form?

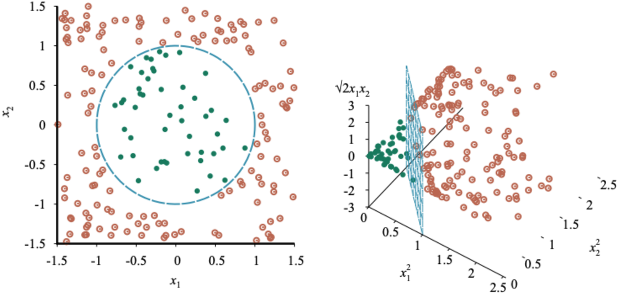
Properties of the margin optimization in the dual form (D)

- Convex problem $\rightarrow$ efficient optimization
- The data enter the expression only as dot product of pairs of data points
  - also true for the separator itself:

$$h_{\mathbf{w},b}(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{n} \alpha_i y^{(i)} (\mathbf{x}^{(i)} \cdot \mathbf{x}) - b\right)$$

- Weights $\alpha_i \neq 0$ only for the support vectors (hence, $\alpha_i = 0$ for most i)

Non-linear decision boundaries with SVM



Example: input space $\mathbf{x} = (x_1, x_2)$, with $y = +1$ inside a circle and $y = -1$ outside.

Take $\phi_1 = x_1^2$, $\phi_2 = x_2^2$, $\phi_3 = \sqrt{2}x_1x_2$

In the dual problem (D), replace $\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}$ with $\phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)})$

What's the big deal? $\phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)}) = (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)})^2 = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ (**kernel** function)

Kernel function and the kernel trick

Definition (Kernel function)

A function that takes as its inputs vectors in the original space and returns the dot product of the vectors in the feature space

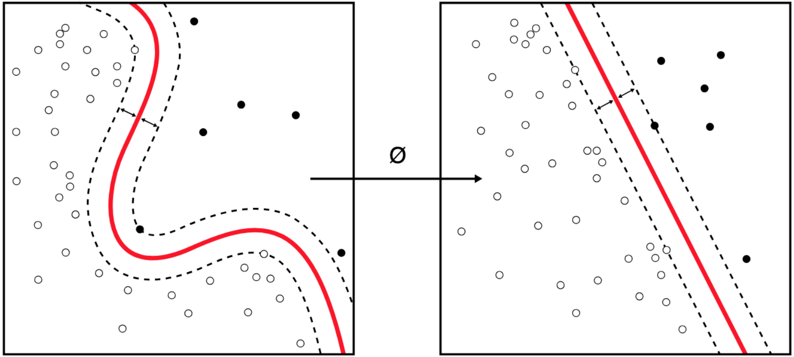
$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)})$$

A common kernel is the Gaussian kernel:

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

Kernel trick: In the optimization problem, replace $(\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)})$ by $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ to obtain $\phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)})$ without calculating (or without even knowing) $\phi(\mathbf{x})$!

Non-linear decision boundaries with SVM through the kernel trick

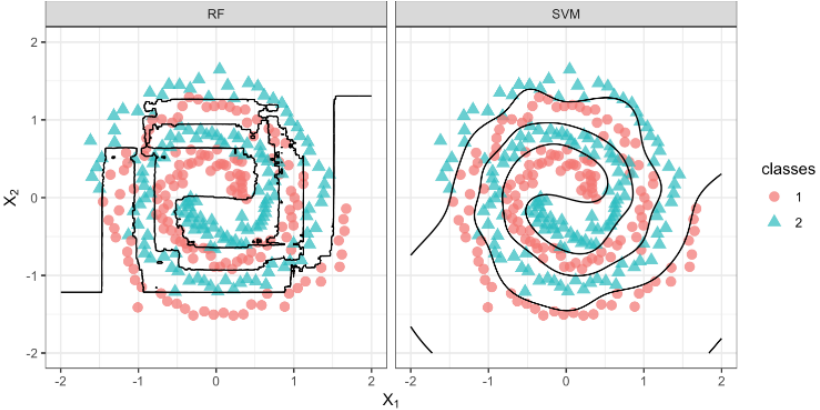


Source: `https://en.wikipedia.org/wiki/Support_vector_machine`

Plugging the kernel into the optimization problem (D), optimal linear separators are found efficiently in feature spaces with billions of parameters

Mapped back to the original space $\rightarrow$ arbitrarily wiggly nonlinear separation

Example: Two spirals benchmark problem



Two spirals benchmark problem. Left: Decision boundary from a random forest. Right: Decision boundary from an SVM with radial basis kernel.

Example from Bradley Boehmke & Brandon Greenwell: Hands-On Machine Learning with R. `https://bradleyboehmke.github.io/HOML/svm.html`.