[AIMLBD] MACHINE LEARNING, BIG DATA, ARTIFICIAL INTELLIGENCE per medicina e chirurgia high tech

L05: Classification

Dott. Giorgio De Magistris

demagistris@diag.uniroma1.it

CORSO DI LAUREA IN MEDICINA E CHIRURGIA HIGH TECH

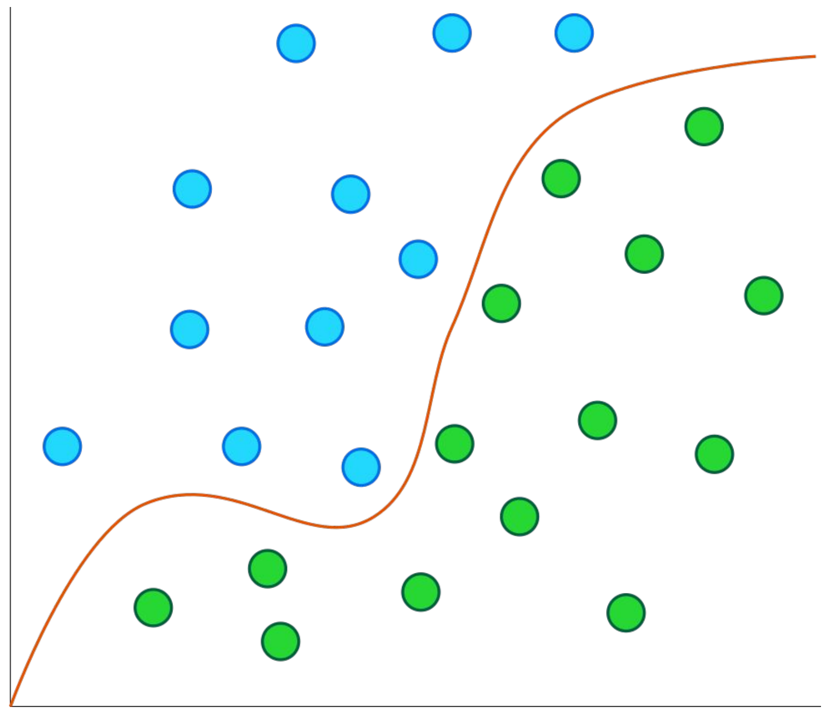
SAPIENZA
UNIVERSITÀ DI ROMA

I3S FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE, INFORMATICA E STATISTICA

DIAG DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE

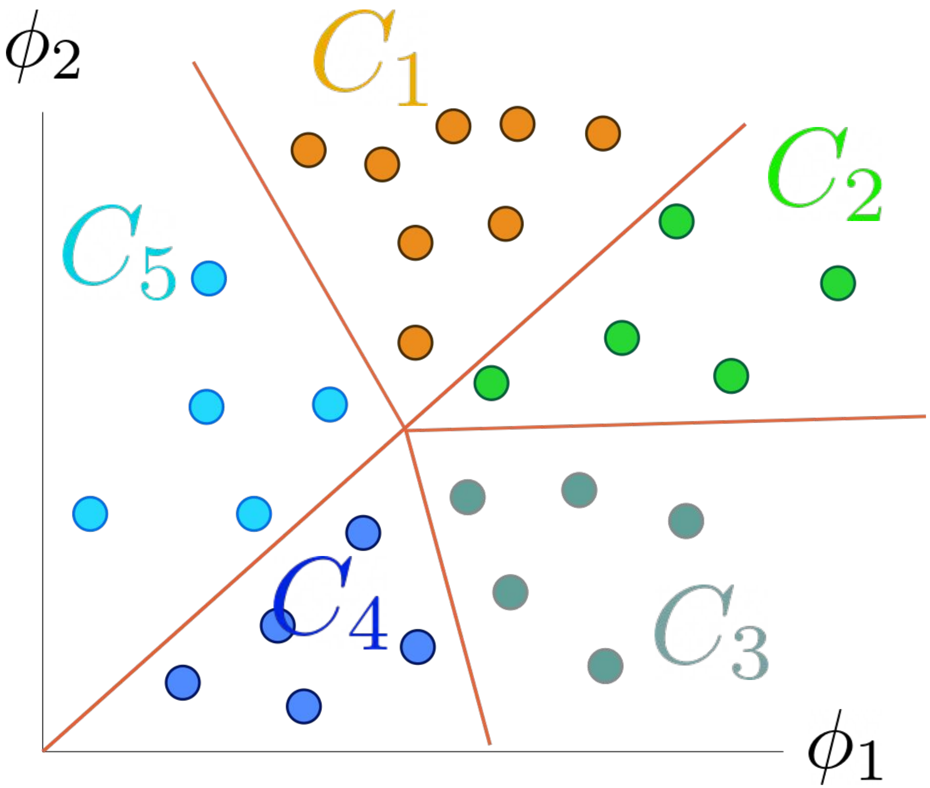
Classification

- Given some data points $X = \{x_1,...,x_n\}$
- Assign to each data point a label Ci from a set of labels $C = \{C_1, \ldots, C_k\}$



Decision Boundaries

- Assume that data points are represented by vectors in an m-dimensional feature space
- The classification task consists in partitioning the feature space into k partitions (where k is the number of categories)



Linear models for classification

- A classification model is linear if the decision boundary a linear function in the feature space, i.e. a line in 2D or a m-1 dimensional hyperplane in an m dimensional feature space
- Note that being linear in the feature space does not imply that the model is linear with respect to the original data space
- We could apply a nonlinear transformation of the input ϕ that transforms the data space into a feature space
- For example we have seen the polynomial expansion of the input, or we can use other transformations like Radial Basis Functions
- In general we will write a data sample as its expansion into the feature space

$$\phi_{\mathbf{i}} = \begin{pmatrix} \phi_0(\mathbf{x_i}) & \phi_1(\mathbf{x_i}) & \dots & \phi_m(\mathbf{x_i}) \end{pmatrix}^T$$

- If no transformation is applied then

$$\phi_{\mathbf{i}} = \begin{pmatrix} \phi_0(\mathbf{x_i}) = 1 & \phi_1(\mathbf{x_i}) = x_1 & \dots & \phi_m(\mathbf{x_i}) = x_m \end{pmatrix}^T = \mathbf{x_i}$$

Binary Classification

- The decision boundary is a linear function of the features

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 \qquad \text{or} \qquad y = \mathbf{w}^T\phi$$

- x is assigned to C_1 if y(x) >= 0 and to class C_2 otherwise

- This is the geometrical interpretation for a 2D feature space

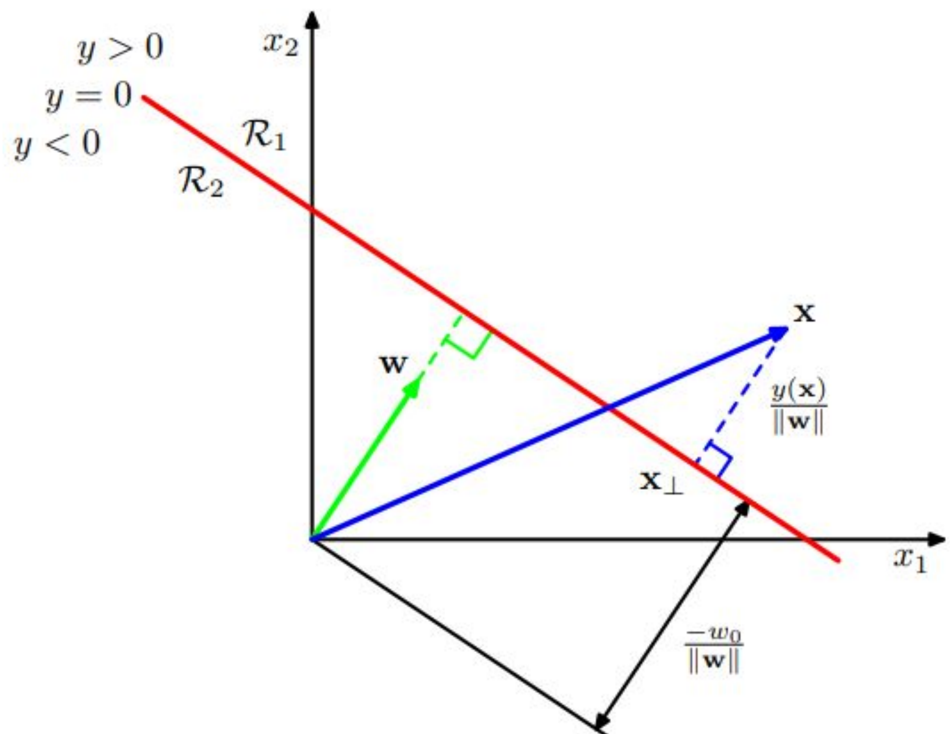


Image Credit: Pattern Recognition and Machine Learning

Multiclass Classification

- I can handle the case of multiple classes using different approaches:
  - *one-versus-the-rest:* for each class I train a classifier to separate points in that class from those that are not in that class
  - *one-versus-one:* for each pair of classes i train a classifier to distinguish the two classes and then apply a majority voting scheme
  - *k-class discriminant:* training k linear function and assign the point to the class that maximizes $y_k(x)$

one-versus-the-rest

- Train k classifiers, one for each class, to classify samples belonging to that class versus samples not belonging to that class

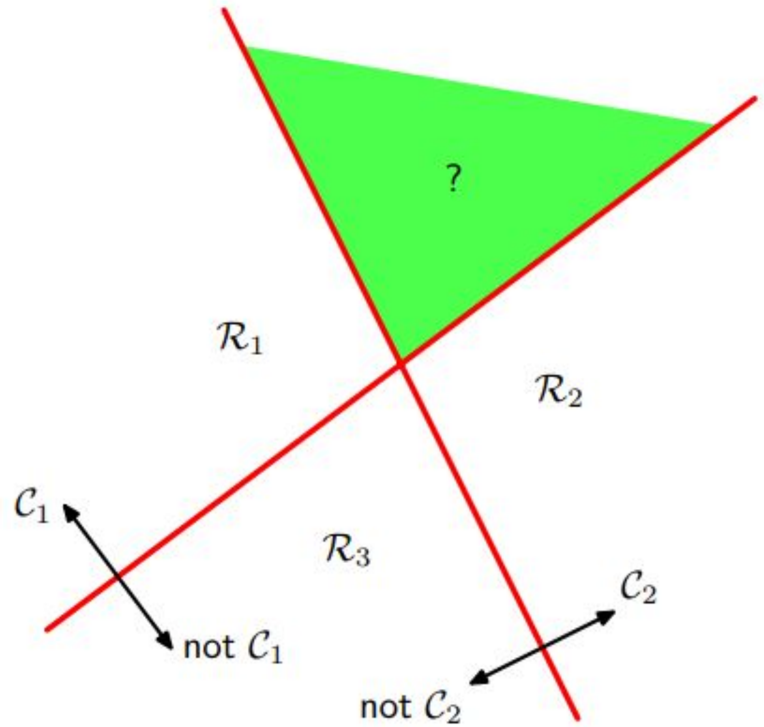
- The classifier

$$y_i(\mathbf{x}) = \mathbf{w_i}^T \mathbf{x} + w_{i0}$$

classifies class Ci versus all other classes

- This approach leads to ambiguous regions, as shown in the figure

- The training is negatively affected by the classes imbalance

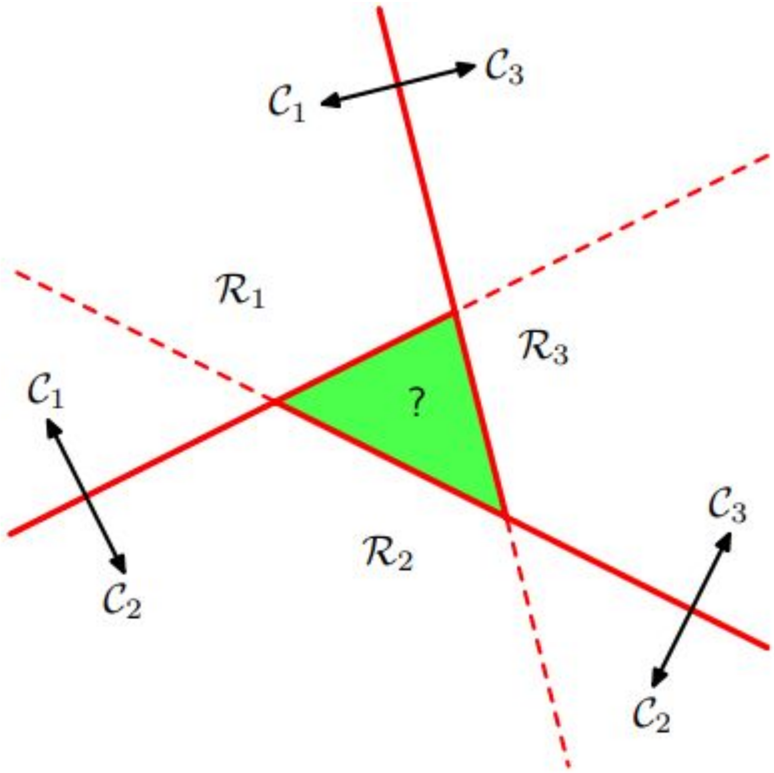


one-versus-one

- Train k(k-1)/2 classifiers, one for each pair of classes
- the classifier

$$y_{ij}(\mathbf{x}) = \mathbf{w_{ij}}^T \mathbf{x} + w_{ij0}$$

  classifies class Ci versus class Cj
- Again some regions of the input space are not labelled correctly, as shown in the figure
- High cost, I've to train a huge number of classifiers



k class discriminant

- A K-class discriminant comprises k linear functions of the form

$$y_k(\mathbf{x}) = \mathbf{w_k}^T \mathbf{x} + w_{k0}$$

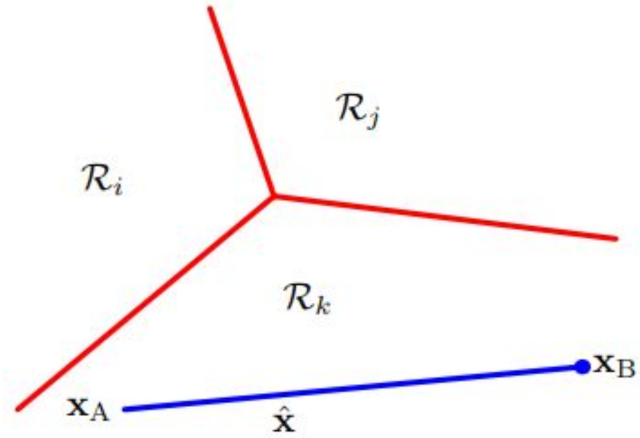
- a point x is assigned to class C_k if $y_k(x) > y_j(x)$ for all $j \neq k$

- the decision boundary between classes Ck and Cj is therefore given by the equation $y_k(x) = y_j(x)$

- the decision boundary corresponds to a D-1 dimensional hyperplane with equation:

$$(\mathbf{w_k} - \mathbf{w_j})^T \mathbf{x} + w_{k0} - w_{j0} = 0$$

k class discriminant



- With this approach there are no ambiguous regions in the input space

- To show this, consider two points x_A and x_B both inside region R_K

- Then consider the line $\hat{\mathbf{x}}$ connecting them:

$$\hat{\mathbf{x}} = \lambda \mathbf{x_A} + (1 - \lambda)\mathbf{x_B} \qquad \text{with} \quad \lambda \in [0, 1]$$

- for the linearity of the discriminant functions we get

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x_A}) + (1 - \lambda) y_k(\mathbf{x_B})$$

- now, knowing that $y_k(\mathbf{x_A}) > y_j(\mathbf{x_A}) \quad \text{and} \quad y_k(\mathbf{x_B}) > y_j(\mathbf{x_B}) \quad \forall j \neq k$
  we can conclude that $y_k(\hat{\mathbf{x}}) > y_j(\hat{\mathbf{x}}) \quad \forall j \neq k$
  and so every point in $\hat{\mathbf{x}}$ also lies in R_K