Introduction to Support Vector Machines (SVM)

S.M. Jaisakthi

SCOPE
VIT University

<*jaisakthi.murugaiyan@vit.ac.in*>

February 13, 2018

Overview

Introduction

- SVM is related to statistical learning theory [1]
- SVM was first introduced in 1992 by Vapnik [2]
- SVMs are promising non-linear, non-parametric classification technique
- SVM becomes popular because of its success in many applications
- SVM is regarded as an important example of "kernel methods", one of the key area in machine learning
- SVMs are primarily suitable for binary classification tasks
- SVMs can be easily adopted for multi-class classification and regression tasks too

Overview

Advantages of SVMs

- A principled approach to classification, regression or novelty detection tasks.
- SVMs provide good generalisation.
- Hypothesis has an explicit dependence on the data (via the support vectors). Hence can readily interpret the model.

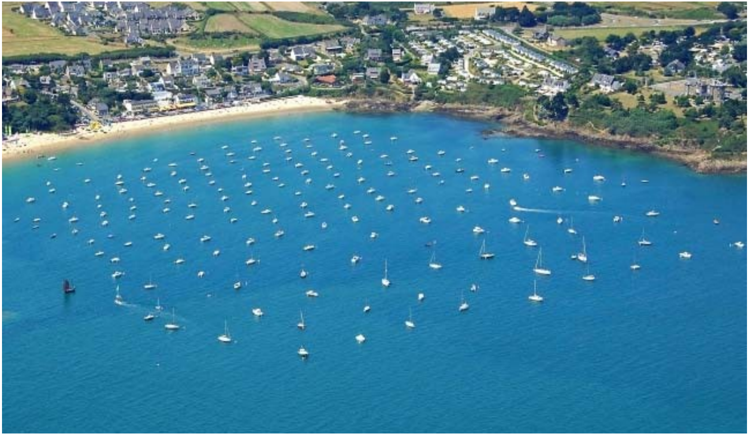
Advantages of SVMs

- Learning involves optimisation of a convex function.
- Few parameters required for tuning the learning machine
- Can implement confidence measures, etc.

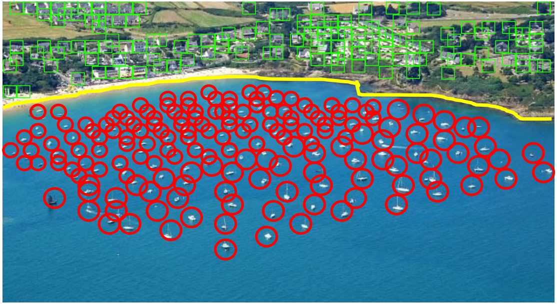
Overview

Basic principles of classification

Classify the objects as boat and house

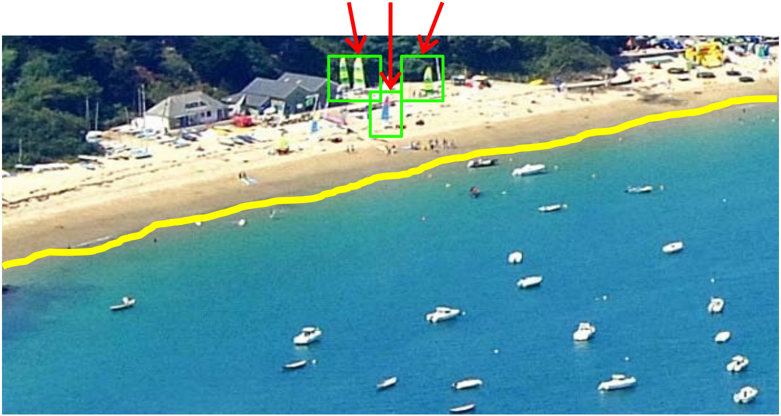


Basic principles of classification

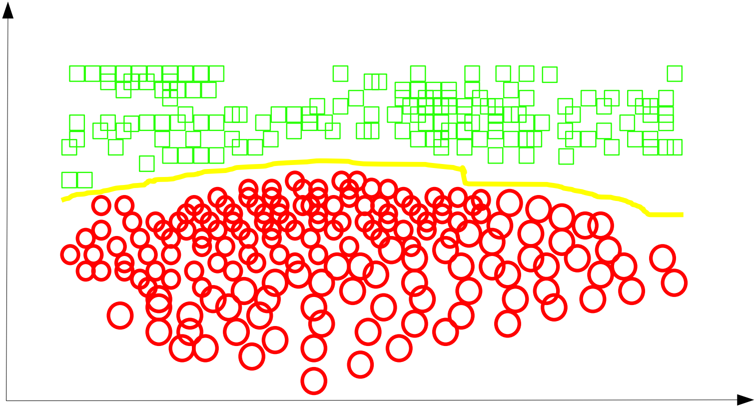


- Objects before the coast line are boats and objects after the coast line are houses.
- Coast line serves as a decision surface that separates two classes

Basic principles of classification



Basic principles of classification



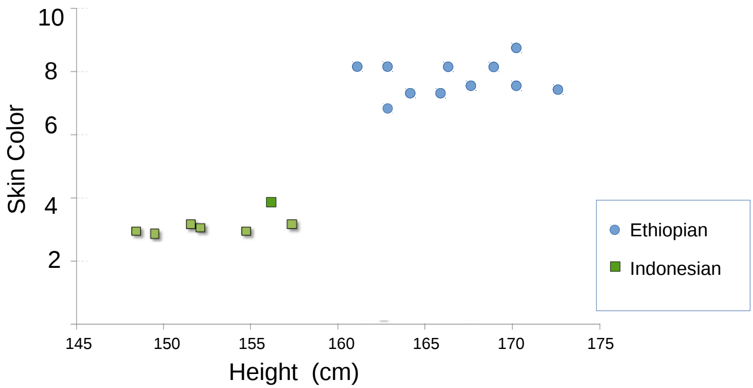
- Classification models (i.e., "classification algorithms") operate very similarly to the previous example.
- All objects are represented geomatrically.

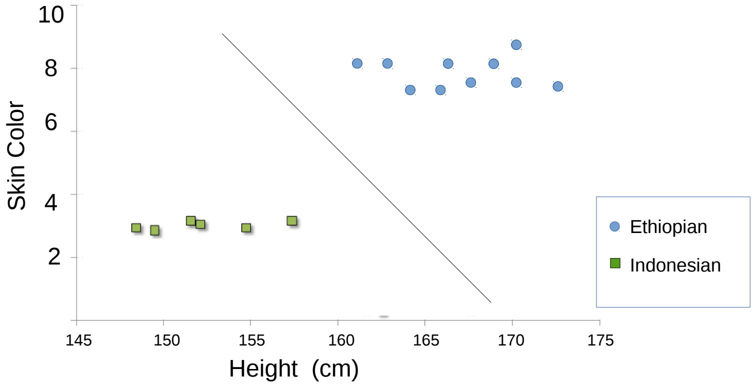
Overview

SVM Approach

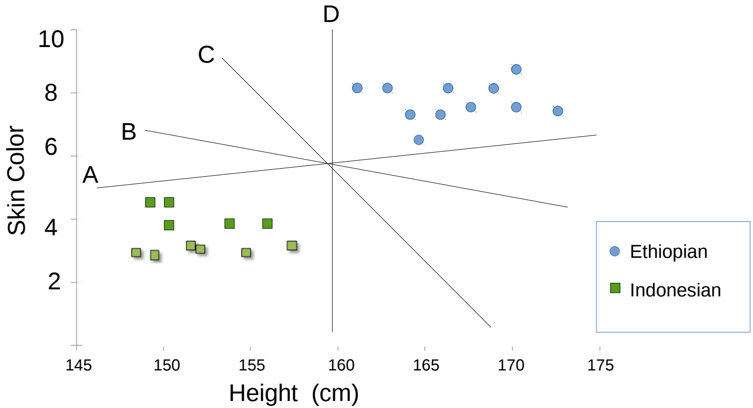
- Support vector machines (SVMs) is a binary classification algorithm
- SVMs are important because
  - Robust to very large number of variables and small samples
  - Can learn both simple and highly complex classification models
  - Employ sophisticated mathematical principles to avoid overfitting

Main Idea of SVM

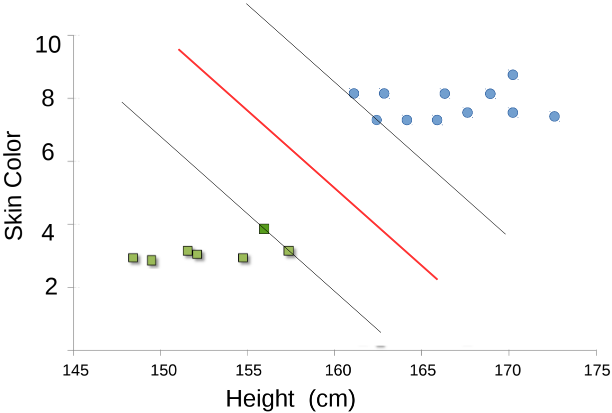




- Classification algorithm seeks to find a decision surface that separates classes of objects.



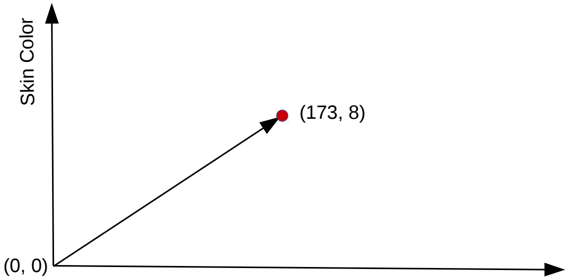
- Select the hyper-plane which separatess the two classes better



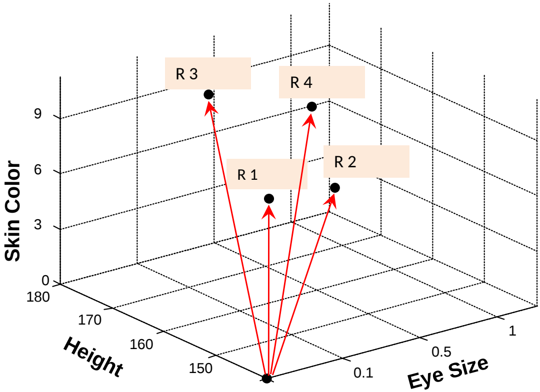
- Find decision boundary that is far away from the data of both classes as possible

How to represent samples geometrically?

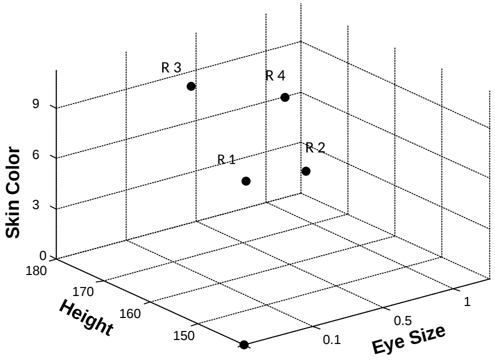
- Assume that a sample is described by n characteristics ("features" or "variables")
- **Representation:** Every sample is a vector in $\Re^n$ with tail at point with 0 coordinates and arrow-head at point with the feature values.
- **Example:** Consider a person described by 2 features: Ethiopian height = 173 and Skin Color = 8.
  This person can be represented as a vector in $\Re^2$:



How to represent samples geometrically?

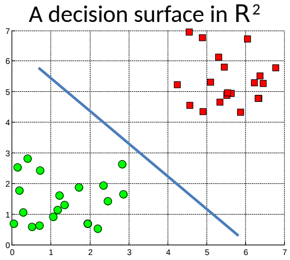


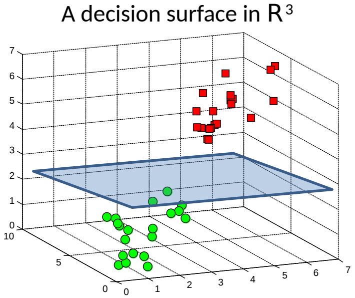
How to represent samples geometrically?



- Since we assume that the tail of each vector is at point with 0 coordinates, we will also depict vectors as points (where the arrow-head is pointing)

How to represent samples geometrically?





- Having represented each sample as a vector allows now to geometrically represent the decision surface that separates two groups of samples.

Overview

SVMs for Binary Classification

- **Preliminaries:**
  Consider a binary classification problem: input vectors are X_i and $y_i = \pm 1$ are the targets or labels. The index i labels the pattern pairs (i = 1, . . . , m).

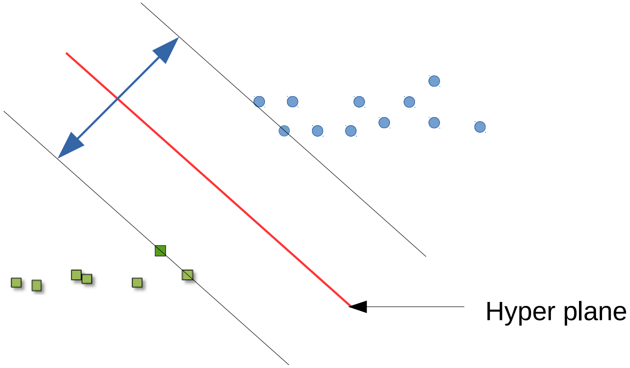
- The X_i define a space of labelled points called input space.

SVMs for Binary Classification

- From the perspective of statistical learning theory the motivation for considering binary classifier SVMs comes from theoretical bounds on the generalization error.

- These generalization bounds have two important features:
  - the upper bound on the generalization error does not depend on the dimensionality of the space.
  - the bound is minimized by maximizing the margin, γ, i.e. the minimal distance between the hyperplane separating the two classes and the closest datapoints of each class.

SVMs for Binary Classification?



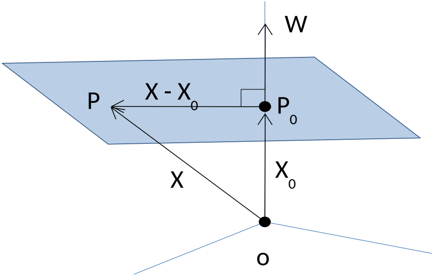
SVMs for Binary Classification

- In an arbitrary-dimensional space a separating hyperplane can be written:

$$W.X + b = 0$$

where b is the bias, and w the weights, etc.

An equation of a hyperplane is defined by a point (P_0) and a perpendicular vector to the plane (W) at that point.



SVMs for Binary Classification

- Thus we will consider a decision function of the form:

$$D(x) = sign(W.X + b)$$

- We note that the argument in D(x) is invariant under a rescaling: $w \rightarrow \lambda w, b \rightarrow \lambda b$. We will implicitly fix a scale with:

$$W.X + b = 1$$

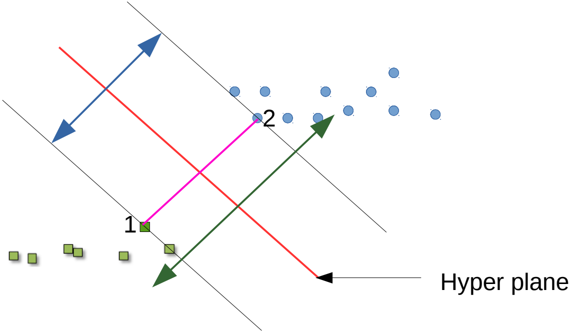
$$W.X + b = -1$$

for the support vectors (canonical hyperplanes).

SVMs for Binary Classification

- Thus
$$w(x_1 - x_2) = 2$$
For two support vectors on each side of the separating hyperplane.
- The margin will be given by the projection of the vector $(x_1 - x_2)$ onto the normal vector to the hyperplane i.e. $w/||w||$ from which we deduce that the margin is given by $\gamma = 1/||w||_2$.

SVMs for Binary Classification?



SVMs for Binary Classification

- Maximisation of the margin is thus equivalent to minimisation of the functional:

$$\Phi(W) = \frac{1}{2}(W.W)$$

subject to the constraints:

$$y_i[(W.X_i + b)] \geq 1$$

- Thus the task is to find an optimum of the primal objective function:

$$L(W, b) = \frac{1}{2}(W.W) - \sum_{i=1}^{m} \alpha_i [y_i((W.X_i) + b) - 1]$$

Solving the saddle point equations $\partial L / \partial b = 0$ gives:

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

SVMs for Binary Classification

and $\partial L / \partial w = 0$ gives:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

which when substituted back in $L(w, \alpha^*, \alpha)$ tells us that we should maximise the functional (the Wolfe dual):

$$w = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i . x_j)$$

subject to the constraints:

$$\alpha_i \geq 0$$

(they are Lagrange multipliers)

and:

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

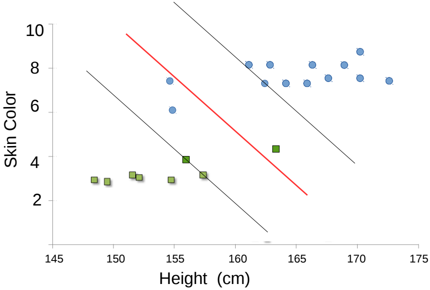
The decision function is then:

$$D(z) = sign[\sum_{j=1}^{m} \alpha_i y_j (X_j, z) + b]$$

Overview

Not Linearly Separable Data : "Soft Margin " Linear SVM

What if the data is not linearly separable? E.g., there are outliers or noisy measurements, or the data is slightly non-linear.



- **Approach:**
  Assign a "slack variable" to each instance $\epsilon_i \geq 0$, which can be thought of distance from the separating hyperplane if an instance is misclassified and 0 otherwise.

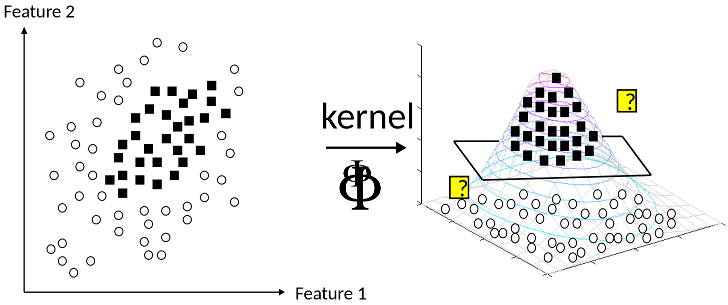
Parameter C in soft-margin SVM

$\frac{1}{2}||W||^2 + c\sum_{i=1}^{N} \epsilon_i$ subject to $y_i(W.X_i + b) \geq 1 - \epsilon_i$

- When C is very large, the soft-margin SVM is equivalent to hard-margin SVM;
- When C is very small, we admit misclassifications in the training data at the expense of having w-vector with small norm;

Overview

Kernal Trick



Data is not linearly separable in the input space

Data is linearly separable in the feature space obtained by a kernel

Kernal Trick

Original data x (in input space)

$$f(x) = sign(w.x + b)$$

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

Data in a higher dimensional feature space $\Phi(x)$

$$f(x) = sign(w.\Phi(x) + b)$$

$$w = \sum_{i=1}^{N} \alpha_i y_i \Phi(x_i)$$

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b)$$

Therefore, we do not need to know Φ explicitly, we just need to define function $K(.,.) : \Re^N \times \Re^N \rightarrow \Re$

Popular Kernals

A kernel is a dot product in some feature space:

$$K(x_i.x_j) = \Phi(x_i).\Phi(x_j)$$

- Linear kernel : $K(X_i, X_j) = X_i.X_j$
- Gaussian kernel : $K(X_i, X_j) = exp(-\gamma||X_i.X_j||^2)$
- Exponential kernel : $K(X_i, X_j) = exp(-\gamma||X_i.X_j||)$
- Polynomial kernel $K(X_i, X_j) = (P + X_i.X_j)^q$
- Hybrid kernel $K(X_i, X_j) = (P + X_i.X_j)^q exp(-\gamma||X_i.X_j||^2)$
- Sigmoidal : $K(X_i, X_j) = \tanh(KX_i.X_j - \delta)$

Overview

Multiclass SVM

- DAG SVM
- One Vs All
  - If there are C classes, then build C independent models
  - Each model m_i is built with positive samples from class C_i and negative samples from all the remaining C-1 classes.
  - A new instance is fed to all the models and the âĂIJwinnerâĂİ decides the class
- One Vs One
  - If there are C classes, $[C \times (C-1)]/2$ models are built For each model m_{ij} , positive samples are from class C_i and negative samples are from C_j
  - Given a new instance, calculate the score using some voting method for a class C_{ii} is sum of results of m_{ix} models minus the sum of results of m_{yi} models for all x and y

References I

1. V. Vapnik. The Nature of Statistical Learning Theory. 2 nd edition, Springer, 1999.
2. B.E. Boser et al . A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, 1992.

THANK YOU