Lecture 14

MFML Team

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Agenda

Mathematical preliminaries for Support Vector Machines

- Constrained optimization and Lagrange multipliers.
- Primal and dual problems and how their solutions are related
- Karash-Kuhn-Tucker conditions.
- Definition of Kernel Functions
- Linear Classifiers

Optimization problem

We shall work with the following optimization problem:

$$\min f(\boldsymbol{x}) \text{ subject to}$$
$$g_i(\boldsymbol{x}) \leq 0 \ \forall i \in [m]$$
$$h_j(\boldsymbol{x}) = 0 \ \forall j \in [p]$$

Optimization problem : Lagrangian

The Lagrangian associated with this optimization problem is

- $$\min f(\boldsymbol{x}) + \sum_{i=1}^{i=m} \lambda_i g(\boldsymbol{x}) + \sum_{j=1}^{j=p} \nu_j h_j(\boldsymbol{x})$$

- The λ_i's and h_j's are called Lagrange multipliers.

Quadratic programming

Consider the following primal problem:

- We now consider the case of a quadratic objective function subject to affine constraints:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{subject to } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

- Here $\boldsymbol{A} \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^d$

Quadratic programming

- The Lagrangian $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is given by $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$.
- Rearranging the above we have $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda})^T\boldsymbol{x} - \boldsymbol{\lambda}^T\boldsymbol{b}$
- Taking the derivative of $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ and setting it equal to zero gives $\boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda}) = 0$.

Quadratic programming

We will now derive the dual problem

- If we take $\boldsymbol{Q}$ to be invertible, we have $\boldsymbol{x} = \boldsymbol{Q}^{-1}(\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda})$.
- Plugging this value of $\boldsymbol{x}$ into $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ gives us $\mathfrak{D}(\boldsymbol{\lambda}) = -\frac{1}{2}(\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda})\boldsymbol{Q}^{-1}(\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda}) - \boldsymbol{\lambda}^T\boldsymbol{b}$.
- This gives us the dual optimization problem: $\max_{\boldsymbol{\lambda} \in \mathbb{R}^m} -\frac{1}{2}(\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda})\boldsymbol{Q}^{-1}(\boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\lambda}) - \boldsymbol{\lambda}^T\boldsymbol{b}$ subject to $\boldsymbol{\lambda} \geq \boldsymbol{0}$.

Summary

The original problem is :

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x}$$

$$\text{subject to } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

The dual problem is

$$\max_{\boldsymbol{\lambda} \geq 0} -\frac{1}{2} (\boldsymbol{c} + \boldsymbol{A}^T \boldsymbol{\lambda}) \boldsymbol{Q}^{-1} (\boldsymbol{c} + \boldsymbol{A}^T \boldsymbol{\lambda}) - \boldsymbol{\lambda}^T \boldsymbol{b}$$

Weak duality

- Weak duality establishes an inequality connecting primal and dual problems
- Weak duality condition states that the optimal solution of the primal problem is greater than or equal to that of the dual problem.
- In the Quadratic Optimization problem discussed previously , weak duality exists

Strong duality

- Strong duality condition states that the optimal solution of the primal problem is equal to that of the dual problem
- One can solve the dual problem to get the same solution as solving the primal problem.
- In some optimization problems, solving the dual problem may be easier.
- Question: When does strong duality hold?

Slater's condition

- For a primal optimization problem we say that it obeys Slater's condition if
  1. the objective function f is convex, the constraints g_i are all convex ,the contraint functions h_i are all linear
  2. there exists a point $\bar{\boldsymbol{x}}$ in the interior of the region, i.e $g_i(\bar{\boldsymbol{x}}) < 0$ for all $i \in [m]$ and $h_j(\bar{\boldsymbol{x}}) = 0$ for all $j \in [p]$.
- Suppose Slater's condition holds then we have strong duality.
- Strong duality condition states that the optimal solution of the primal problem is equal to that of the dual problem

Example of Slater's condition

We will consider an optimization problem as given below

$$\min \; x^2 + y^2$$
$$\text{st } x + y - 3 \leq 0$$

- Here $f(x, y) = x^2 + y^2$ is a convex function and $g(x, y) = x + y - 3$ is a convex function
- We can find a point that satisfies the condition $x + y - 3 < 0$
- Slaters condition is satisfied

KKT conditions

$$\min\ f(\boldsymbol{x}) \quad \text{st} \quad g_i(\boldsymbol{x}) \leq 0\ \forall i \in [m], \quad h_j(\boldsymbol{x}) = 0\ \forall j \in [p]$$

We say that $\boldsymbol{x}^*$ and $(\lambda^*, \nu^*) \in \mathbb{R}^m \times \mathbb{R}^p$ respect the Karash-Kuhn-Tucker conditions if:

1. $g_i(\boldsymbol{x}^*) \leq 0\ \forall i \in [m]$, $h_i(\boldsymbol{x}^*) = 0\ \forall i \in [p]$.
2. $\lambda_i^* \geq 0\ \forall i \in [m]$.
3. $\lambda_i^* g_i(\boldsymbol{x}^*) = 0\ \forall i \in [m]$.
4. $\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{i=m} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) + \sum_{i=1}^{i=p} \nu_i^* \nabla h_i(\boldsymbol{x}^*) = 0$.

If strong duality holds then any primal optimal solution $\boldsymbol{x}^*$ and dual optimal solution (λ^*, ν^*) satisfy the KKT conditions.

KKT condition

We will consider an optimization problem and will write its KKT conditions

$$\min \; x^2 + y^2$$
$$\text{st } x + y - 3 \leq 0$$

- Here $f(x, y) = x^2 + y^2$ and $g(x, y) = x + y - 3$

1. $x + y - 3 \leq 0$
2. $\lambda \geq 0$
3. $\lambda(x + y - 3) = 0$
4. $\nabla f + \lambda \nabla g = \mathbf{0}$

Classification Problem in Machine Learning

- Classification of data into different classes is one of the primary problems in machine learning
- Binary classification involves classifying data into exactly 2 classes
- There exists different algorithms for binary classification
- We will discuss a model called Support Vector Machine.
- SVM is a linear classifer model for binary classification

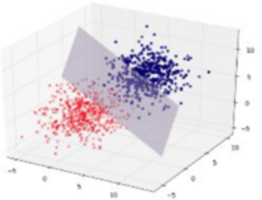
Linear Classifier

$$\mathbf{w}^T\mathbf{x} = 0$$

Hyperplane

$$y = ax + b$$

Line



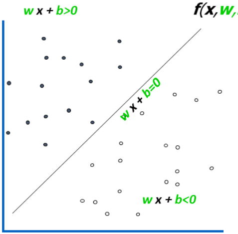
Linear Classifier and Hyperplane

- Consider line $w^T x + b = 0$. Let x_a and x_b lie on this line. So $w^T x_a + b = 0$ and $w^T x_b + b = 0$.
- This means $w^T(x_a - x_b) = 0$. $x_a - x_b$ lies on the line and is directed from x_b to x_a.
- Hence w is orthogonal to $x_a - x_b$ and in turn, to the line.

Linear Classifer for Binary Classification

Linear Classifiers

- denotes +1
- denotes -1

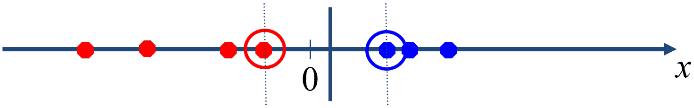


***f*(x,w,b) = *sign*(w x + b)**

How would you classify this data?

Two examples of data

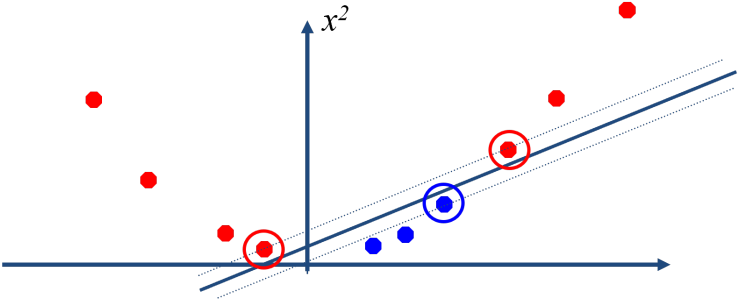
Dataset that are linearly separable with some noise



Dataset is not linearly separable

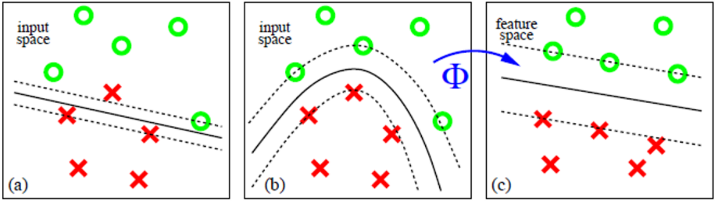
Mapping of Data

mapping data to a higher-dimensional space:



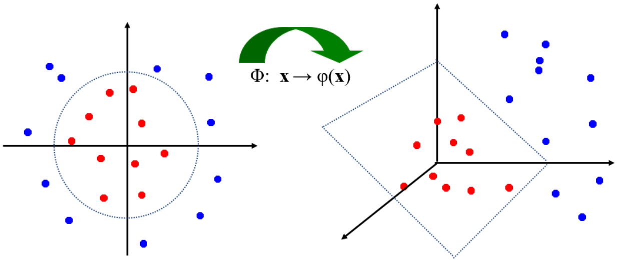
Mapping of Data

Find a feature space



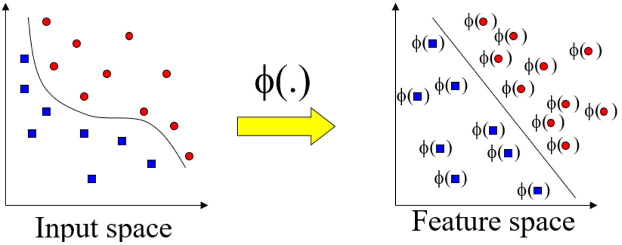
If every data point is mapped into high-dimensional space via some transformation $\phi : x \rightarrow \phi(x)$

Feature spaces



- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable.

Transforming the Data



- Computation in the feature space can be costly because it is high dimensional.
- The feature space is typically infinite-dimensional.
- The kernel trick using kernel functions comes to rescue

Kernel Functions

- Kernel is a continuous function $K(x, y)$
- Kernel takes two arguments x and y
- x and y could be real numbers, functions, vectors, etc
- $K(x, y)$ maps x and y to a real value
- Kernel value is independent of the order of the arguments, i.e.,

$$K(x, y) = K(y, x)$$

Kernel Functions

- A kernel function is some function that corresponds to an inner product in some expanded feature space.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- Linear classifier relies on dot product between vectors $x_i^T x_j$
- If every data point is mapped into high-dimensional space via some transformation $\phi : x \rightarrow \phi(x)$, the dot product becomes: $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$
- For some functions $K(x_i, x_j)$ checking $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is difficult.
- Mercer's theorem: Every positive-semidefinite symmetric function is a kernel function.

Kernel Functions Construction

1) We can *construct kernels from scratch*:

- For any $\varphi : \mathcal{X} \to \mathbb{R}^m$, $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathbb{R}^m}$ is a kernel.
- If $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *distance function*, i.e.
  - $d(x, x') \geq 0$ for all $x, x' \in \mathcal{X}$,
  - $d(x, x') = 0$ only for $x = x'$,
  - $d(x, x') = d(x', x)$ for all $x, x' \in \mathcal{X}$,
  - $d(x, x') \leq d(x, x'') + d(x'', x')$ for all $x, x', x'' \in \mathcal{X}$,

  then $k(x, x') := \exp(-d(x, x'))$ is a kernel.

Kernel Functions Construction

2) We can *construct kernels from other kernels*:

- if k is a kernel and $\alpha > 0$, then αk and $k + \alpha$ are kernels.
- if k_1, k_2 are kernels, then $k_1 + k_2$ and $k_1 \cdot k_2$ are kernels.

Examples of Kernels

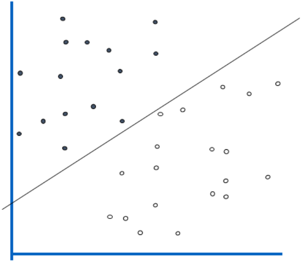
- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polynomial of power p : $K(x_i, x_j) = (1 + x_i^T x_j)^p$
- Sigmoid: $K(x_i, x_j) = tanh(\beta_0 x_i^T x_j + \beta_1)$

Linear Classifiers Revisited

Linear Classifiers

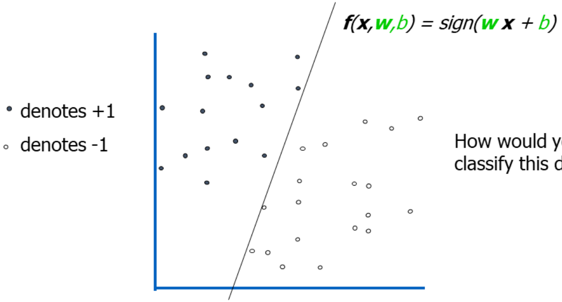
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- • denotes +1
- ∘ denotes -1



How would you classify this data?

Linear Classifiers

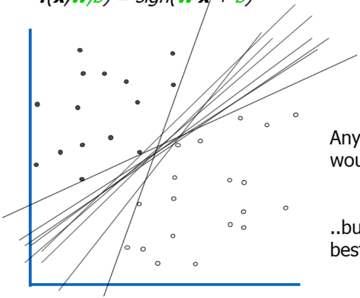


How would you classify this data?

Linear Classifiers

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- • denotes +1
- ○ denotes -1



Any of these would be fine..

..but which is best?

Linear Classifier

Linear Classifiers