

COMS 4771
Assignment Project Exam Help
Support Vector Machines
<https://powcoder.com>

Add WeChat powcoder

Nakul Verma

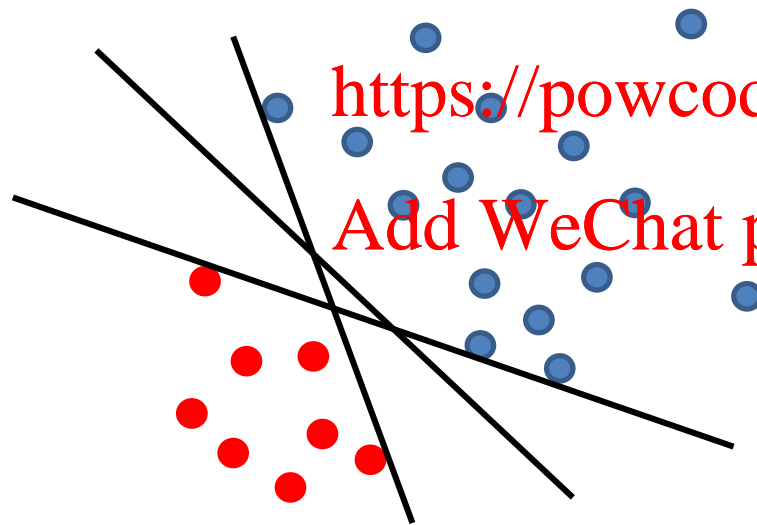
Last time...

- Decision boundaries for classification
- Linear decision boundary (linear classification)
Assignment Project Exam Help
- The Perceptron algorithm
<https://powcoder.com>
- Mistake bound for the perceptron
Add WeChat powcoder
- Generalizing to non-linear boundaries (via Kernel space)
- Problems become linear in Kernel space
- The Kernel trick to speed up computation

Perceptron and Linear Separability

Say there is a **linear** decision boundary which can **perfectly separate** the training data

Which linear separator will the Perceptron algorithm return?



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

*The separator with a **large margin** γ is better for generalization*

How can we incorporate the margin in finding the linear boundary?

Solution: Support Vector Machines (SVMs)

Motivation:

- It returns a **linear classifier** that is **stable** solution by giving a maximum margin solution.
- Slight modification to the problem provides a way to deal with **non-separable** cases.
- It is **kernelizable**, so gives an implicit way of yielding non-linear classification.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

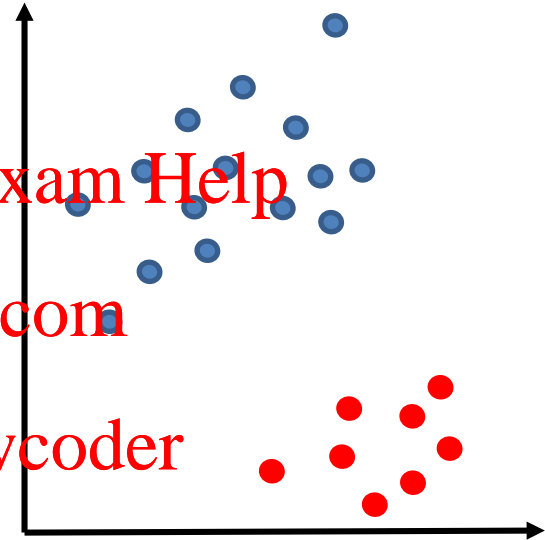
SVM Formulation

Say the training data S is **linearly separable** by some margin (but the linear separator does not necessarily pass through the origin).

Then:

decision boundary: $g(\vec{x}) = \vec{w} \cdot \vec{x} - b = 0$

Linear classifier: $f(\vec{x}) = \text{sign}(g(\vec{x}))$
 $= \text{sign}(\vec{w} \cdot \vec{x} - b)$



*Idea: we can try finding **two** parallel hyperplanes that correctly classify all the points, and **maximize** the distance between them!*

SVM Formulation (contd. 1)

Decision boundary for the two hyperplanes:

$$\vec{w} \cdot \vec{x} - b = +1$$

$$\vec{w} \cdot \vec{x} - b = -1$$

Distance between the two hyperplanes:

$$\frac{2}{\|\vec{w}\|}$$

why?

<https://powcoder.com>

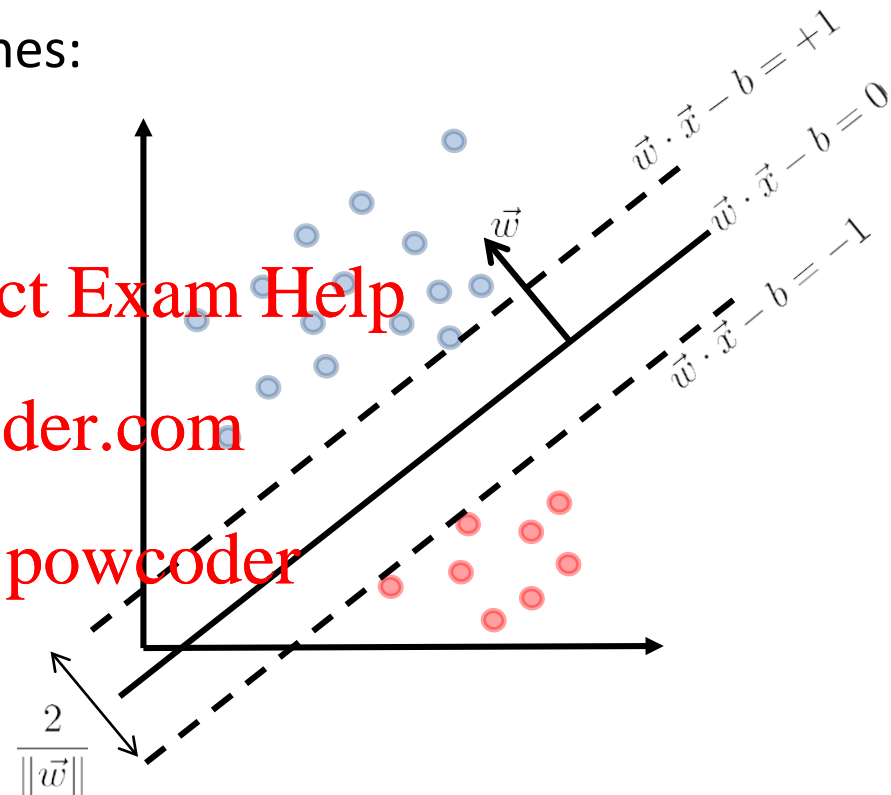
Add WeChat powcoder

Training data is correctly classified if:

$$\vec{w} \cdot \vec{x}_i - b \geq +1 \quad \text{if } y_i = +1$$

$$\vec{w} \cdot \vec{x}_i - b \leq -1 \quad \text{if } y_i = -1$$

Together: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq +1$ for all i



SVM Formulation (contd. 2)

Distance between the hyperplanes: $\frac{2}{\|\vec{w}\|}$

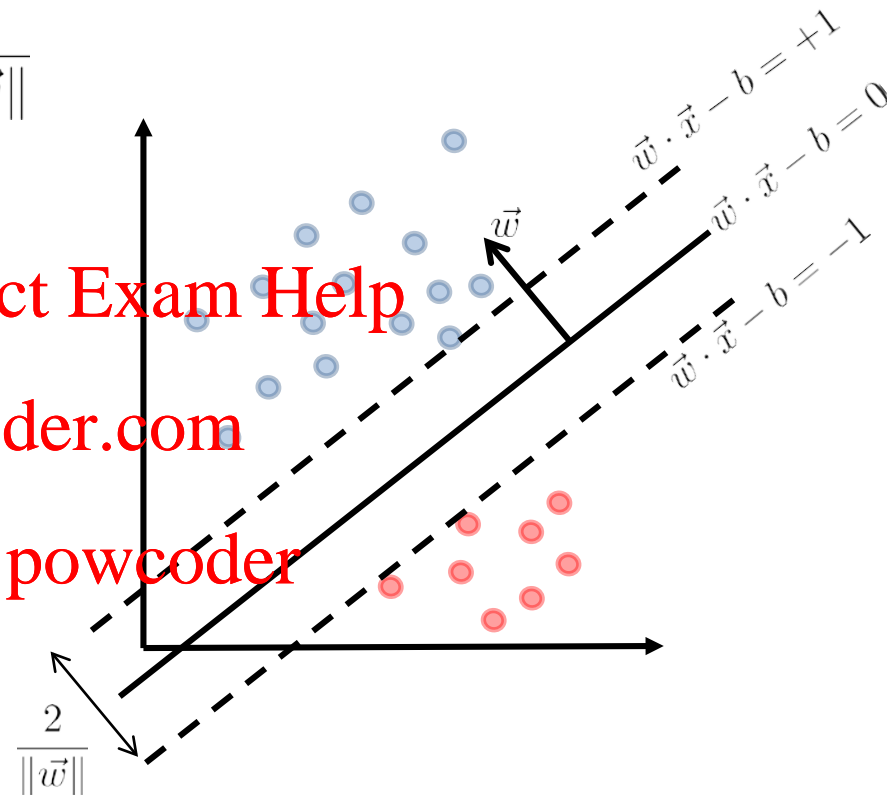
Training data is correctly classified if:

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq +1 \quad (\text{for all } i)$$

Therefore, want:

Maximize the distance: $\frac{2}{\|\vec{w}\|}$

Such that: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq +1$
(for all i)



Let's put it in the standard form...

SVM Formulation (finally!)

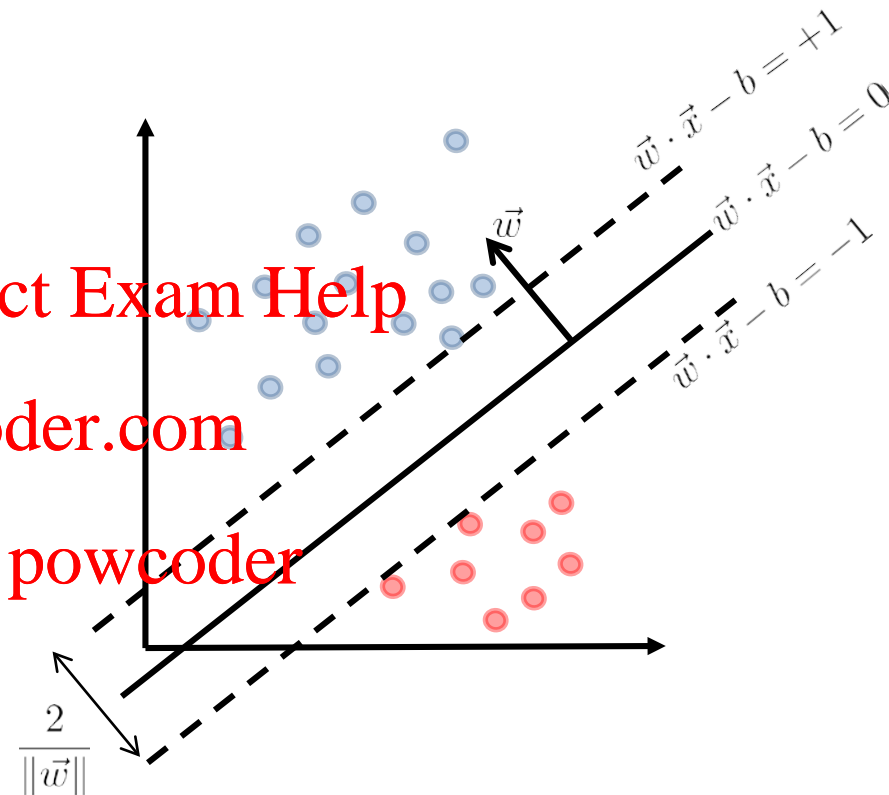
Maximize: $\frac{2}{\|\vec{w}\|}$

Such that: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq +1$
(for all i)

SVM standard (primal) form:

Minimize: $\frac{1}{2} \|\vec{w}\|^2$

Such that: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq +1$
(for all i)



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

What can we do if the problem is not-linearly separable?

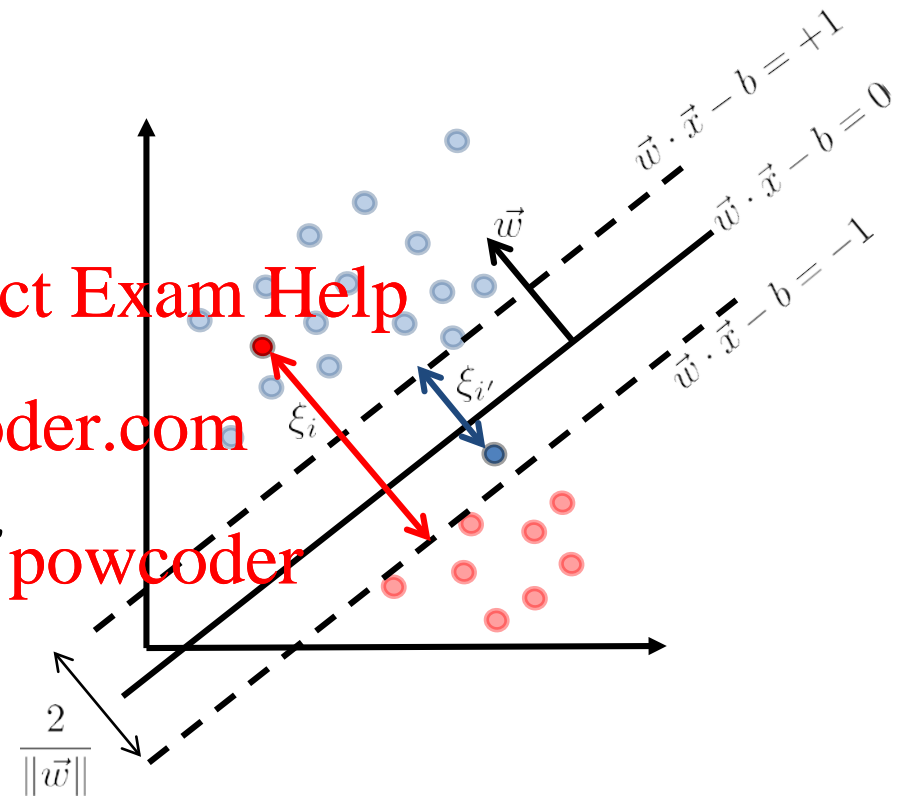
SVM Formulation (non-separable case)

Idea: introduce a **slack** for the misclassified points, and **minimize** the slack!

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



SVM standard (primal) form (with slack):

$$\text{Minimize: } \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n \xi_i$$

$$\text{Such that: } y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 - \xi_i$$

(for all i)

$$\xi_i \geq 0$$

SVM: Question

SVM standard (primal) form (with **slack**):

$$\text{Minimize: } \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n \xi_i$$

Assignment Project Exam Help

$$\text{Such that: } y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 - \xi_i$$

(for all i)

<https://powcoder.com>

$$\xi_i \geq 0$$

Add WeChat powcoder

Questions:

- 1. How do we find the optimal w , b and ξ ?*
- 2. Why is it called “Support Vector Machine”?*

How to Find the Solution?

Cannot simply take the derivative (wrt w , b and ξ) and examine the stationary points...

SVM standard (primal) form:

$$\text{Minimize: } \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n \xi_i$$

$$\text{Such that: } y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 - \xi_i$$

(for all i) $\xi_i \geq 0$

Assignment Project Exam Help

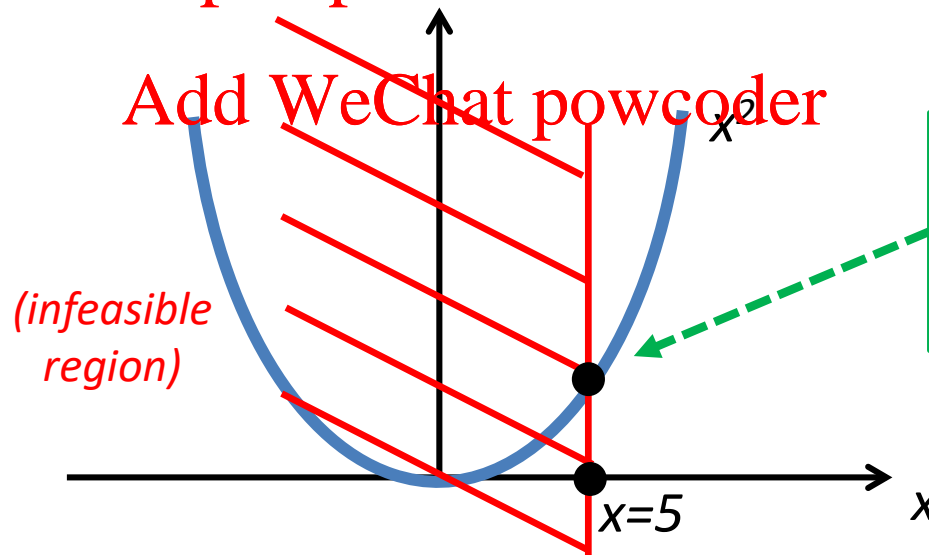
<https://powcoder.com>

Add WeChat powcoder

Why?

Minimize: x^2

Such that: $x \geq 5$



Gradient **not zero** at the function minima (respecting the constraints)!

Need a way to do optimization with constraints

Detour: Constrained Optimization

Constrained optimization (standard form):

$$\begin{array}{ll} \underset{\vec{x} \in \mathbb{R}^d}{\text{minimize}} & f(\vec{x}) \quad (\text{objective}) \\ \text{subject to:} & g_i(\vec{x}) \leq 0 \quad \text{for } 1 \leq i \leq n \quad (\text{constraints}) \end{array}$$

<https://powcoder.com>

What to do?

- Projection methods *Add WeChat powcoder* *We'll assume that the problem is feasible*
 - start with a feasible solution x_0 ,
 - find x_1 that has slightly lower objective value,
 - if x_1 violates the constraints, **project back** to the constraints.
 - iterate.
- Penalty methods
 - use a **penalty function** to incorporate the constraints into the objective
- ...

The Lagrange (Penalty) Method

Consider the augmented function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

(Lagrange function)

(Lagrange variables,
or dual variables)

Optimization problem:

Minimize: $f(\vec{x})$
Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Assignment Project Exam Help

<https://powcoder.com>

Observation:

Add WeChat powcoder

For **any** feasible \vec{x} and **all** $\lambda_i \geq 0$, we have $L(\vec{x}, \vec{\lambda}) \leq f(\vec{x})$

$$\implies \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda}) \leq f(\vec{x})$$

So, the optimal value to the constrained optimization:

$$p^* := \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

The problem becomes
unconstrained in \vec{x} !

The Dual Problem

Optimal value: $p^* = \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$
(also called the primal)

Now, consider the function: $\min_{\vec{x}} L(\vec{x}, \vec{\lambda})$

Observation:

Since, for **any** feasible \vec{x} and **all** $\lambda_i \geq 0$:

$$p^* \geq \min_{\vec{x}} L(\vec{x}, \vec{\lambda})$$

Thus:

$$d^* := \max_{\lambda_i \geq 0} \min_{\vec{x}'} L(\vec{x}', \vec{\lambda}) \leq p^*$$

(also called the dual)

Optimization problem:

Minimize: $f(\vec{x})$
Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Lagrange function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

(Weak) Duality Theorem

Theorem (weak Lagrangian duality):

$$d^* \leq p^*$$

(also called the minimax inequality)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

$$p^* - d^* \quad (\text{called the duality gap})$$

*Under what conditions can we
achieve equality?*

Optimization problem:

Minimize: $f(\vec{x})$
Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Lagrange function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

Primal:

$$p^* = \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

Dual:

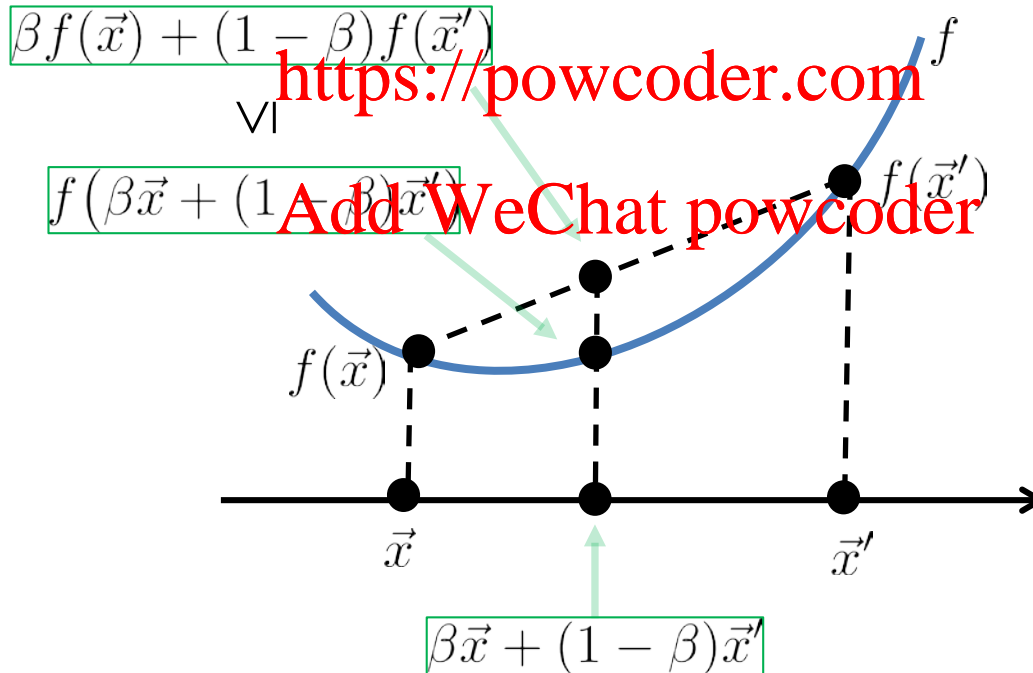
$$d^* := \max_{\lambda_i \geq 0} \min_{\vec{x}} L(\vec{x}, \vec{\lambda})$$

Convexity

A function $f: \mathbf{R}^d \rightarrow \mathbf{R}$ is called convex iff for any two points x, x' and $\beta \in [0,1]$

$$f(\beta \vec{x} + (1 - \beta) \vec{x}') \leq \beta f(\vec{x}) + (1 - \beta) f(\vec{x}')$$

Assignment Project Exam Help



Convexity

A set $S \subset \mathbf{R}^d$ is called convex iff for any two points $x, x' \in S$ and any $\beta \in [0,1]$

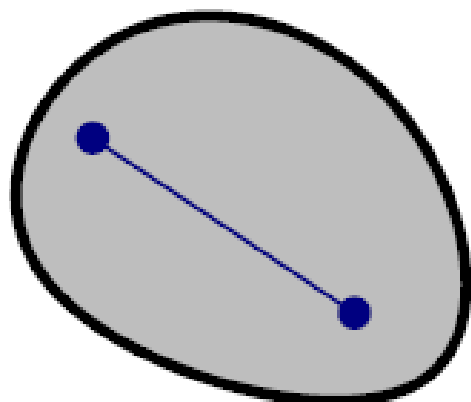
$$\beta \vec{x} + (1 - \beta) \vec{x}' \in S$$

Examples:

Assignment Project Exam Help

<https://powcoder.com>

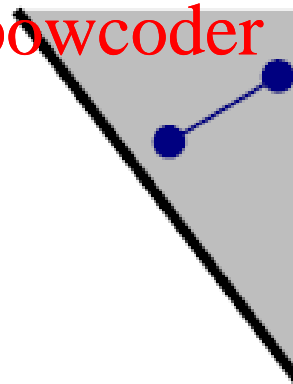
Add WeChat powcoder



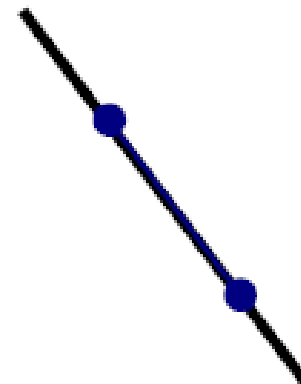
convex



not convex



convex



convex

Convex Optimization

A constrained optimization

$$\begin{array}{ll} \underset{\vec{x} \in \mathbb{R}^d}{\text{minimize}} & f(\vec{x}) \quad (\text{objective}) \\ \text{subject to:} & g_i(\vec{x}) \leq 0 \quad \text{for } 1 \leq i \leq n \quad (\text{constraints}) \end{array}$$

Assignment Project Exam Help

<https://powcoder.com>

is called convex a convex optimization problem

Add WeChat powcoder

If:

the objective function $f(\vec{x})$ is convex function, and

the feasible set induced by the constraints g_i is a convex set

Why do we care?

*We and find the optimal solution for convex problems **efficiently!***

Convex Optimization: Niceties

- Every local optima is a **global optima** in a convex optimization problem.

Example convex problems:

Linear programs, quadratic programs,

Conic programs, semi-definite program.

Several **solvers exist** to find the optima.

CVX, SeDuMi, C-SALSA, ...

- We can use a **simple** 'descend-type' algorithm for finding the minima!

Gradient Descent (for finding local minima)

Theorem (Gradient Descent):

Given a smooth function $f : \mathbb{R}^d \rightarrow \mathbb{R}$

Then, for any $\vec{x} \in \mathbb{R}^d$ and $\vec{x}' := \vec{x} - \eta \nabla_x f(\vec{x})$

For sufficiently small $\eta > 0$, we have: $f(\vec{x}') \leq f(\vec{x})$

<https://powcoder.com>

Can derive a **simple algorithm** (the projected Gradient Descent):

Add WeChat powcoder

Initialize \vec{x}^0

for $t = 1, 2, \dots$ do

$$\vec{x}'^t := \vec{x}^{t-1} - \eta \nabla_x f(\vec{x}^{t-1}) \quad (\text{step in the gradient direction})$$

$$\vec{x}^t := \Pi_{g_i}(\vec{x}'^t) \quad (\text{project back onto the constraints})$$

terminate when no progress can be made, ie, $|f(\vec{x}^t) - f(\vec{x}^{t-1})| \leq \epsilon$

Back to Constrained Opt.: Duality Theorems

Theorem (weak Lagrangian duality):

$$d^* \leq p^*$$

Optimization problem:

Minimize: $f(\vec{x})$
Such that: $g_i(\vec{x}) \leq 0$
(for all i)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Theorem (strong Lagrangian duality):

If f is convex and for a feasible point \vec{x}^*

$$g_i(\vec{x}^*) < 0, \text{ or}$$

$$g_i(\vec{x}^*) \leq 0 \text{ when } g \text{ is affine}$$

Then $d^* = p^*$

Lagrange function:

$$L(\vec{x}, \vec{\lambda}) := f(\vec{x}) + \sum_{i=1}^n \lambda_i g_i(\vec{x})$$

Primal:

$$p^* = \min_{\vec{x}} \max_{\lambda_i \geq 0} L(\vec{x}, \vec{\lambda})$$

Dual:

$$d^* := \max_{\lambda_i \geq 0} \min_{\vec{x}} L(\vec{x}, \vec{\lambda})$$

Ok, Back to SVMs

Observations:

- object function is **convex**
- the constraints are **affine**, inducing a polytope constraint set

SVM standard (primal) form:

$$\text{Minimize: } \frac{1}{2} \|\vec{w}\|^2$$

$$\text{Such that: } y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$$

(for all i)

So, SVM is a convex optimization problem

(in fact a **quadratic program**)

Moreover, **strong duality holds**.

Let's examine the dual... the Lagrangian is:

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \|\vec{w}\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i(\vec{w} \cdot \vec{x}_i - b))$$

SVM Dual

Lagrangian:

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \|\vec{w}\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i(\vec{w} \cdot \vec{x}_i - b))$$

Primal: $p^* = \min_{\vec{w}, b} \max_{\alpha_i \geq 0} L(\vec{w}, b, \vec{\alpha})$

Dual: $d^* = \max_{\alpha_i \geq 0} \min_{\vec{w}, b} L(\vec{w}, b, \vec{\alpha})$

SVM standard (primal) form:

Minimize: $\frac{1}{2} \|\vec{w}\|^2$
(w,b)

Such that: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$
(for all i)

<https://powcoder.com>
Add WeChat powcoder

$$\frac{\partial}{\partial \vec{w}} L(\vec{w}, b, \vec{\alpha}) = \vec{w} - \sum_{i=1}^n \alpha_i y_i \vec{x}_i \quad \implies \quad \vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i$$

- when $\alpha_i > 0$, the corresponding x_i is the support vector
- w is only a function of the support vectors!

$$\frac{\partial}{\partial b} L(\vec{w}, b, \vec{\alpha}) = \sum_{i=1}^n \alpha_i y_i \quad \implies \quad \sum_{i=1}^n \alpha_i y_i = 0$$

SVM Dual (contd.)

Lagrangian:

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \|\vec{w}\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i(\vec{w} \cdot \vec{x}_i - b))$$

Primal: $p^* = \min_{\vec{w}, b} \max_{\alpha_i \geq 0} L(\vec{w}, b, \vec{\alpha})$

Dual: $d^* = \max_{\alpha_i \geq 0} \min_{\vec{w}, b} L(\vec{w}, b, \vec{\alpha})$

SVM standard (primal) form:

Minimize: $\frac{1}{2} \|\vec{w}\|^2$
(w,b)

Such that: $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$
(for all i)

<https://powcoder.com>
Add WeChat powcoder

$$\min_{\vec{w}, b} L(\vec{w}, b, \vec{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

So:

$$d^* = \max_{\alpha_i \geq 0} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

subject to $\sum_{i=1}^n \alpha_i y_i = 0$

SVM Optimization Interpretation

SVM standard (primal) form:

$$\text{Minimize: } \frac{1}{2} \|\vec{w}\|^2$$

(w, b)

$$\text{Such that: } y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$$

(for all i)

$$\text{Maximize } \gamma = 2/\|\vec{w}\|$$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

SVM standard (dual) form:

$$\text{Maximize: } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

(α_i)

Such that:

(for all i)

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad \alpha_i \geq 0$$

Kernelized version

*Only a function of
"support vectors"*

What We Learned...

- Support Vector Machines
- Maximum Margin formulation
- Constrained Optimization
- Lagrange Duality Theory
- Convex Optimization
- SVM dual and Interpretation
- How get the optimal solution

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Questions?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Next time...

Parametric and non-parametric Regression

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder