10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Support Vector Machines

Matt Gormley
Lecture 28
April 18, 2018

Reminders

- Homework 8: Reinforcement Learning
  - Out: Tue, Apr 17
  - Due: Fri, Apr 27 at 11:59pm
  - No class on Friday, Apr 20
  - Recitation: Mon, Apr 23 (instead of lecture)

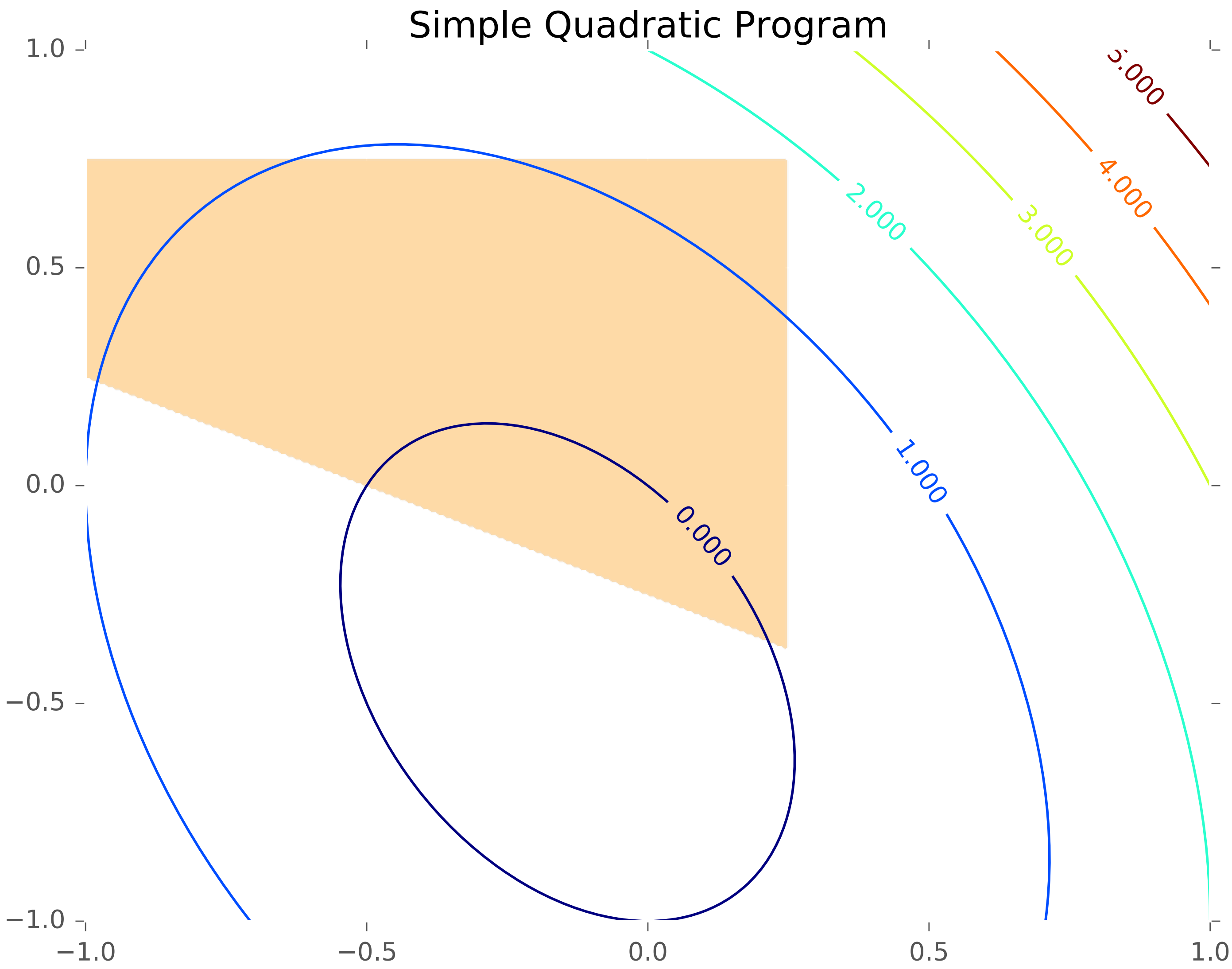
SUPPORT VECTOR MACHINE (SVM)

SVM: Optimization Background

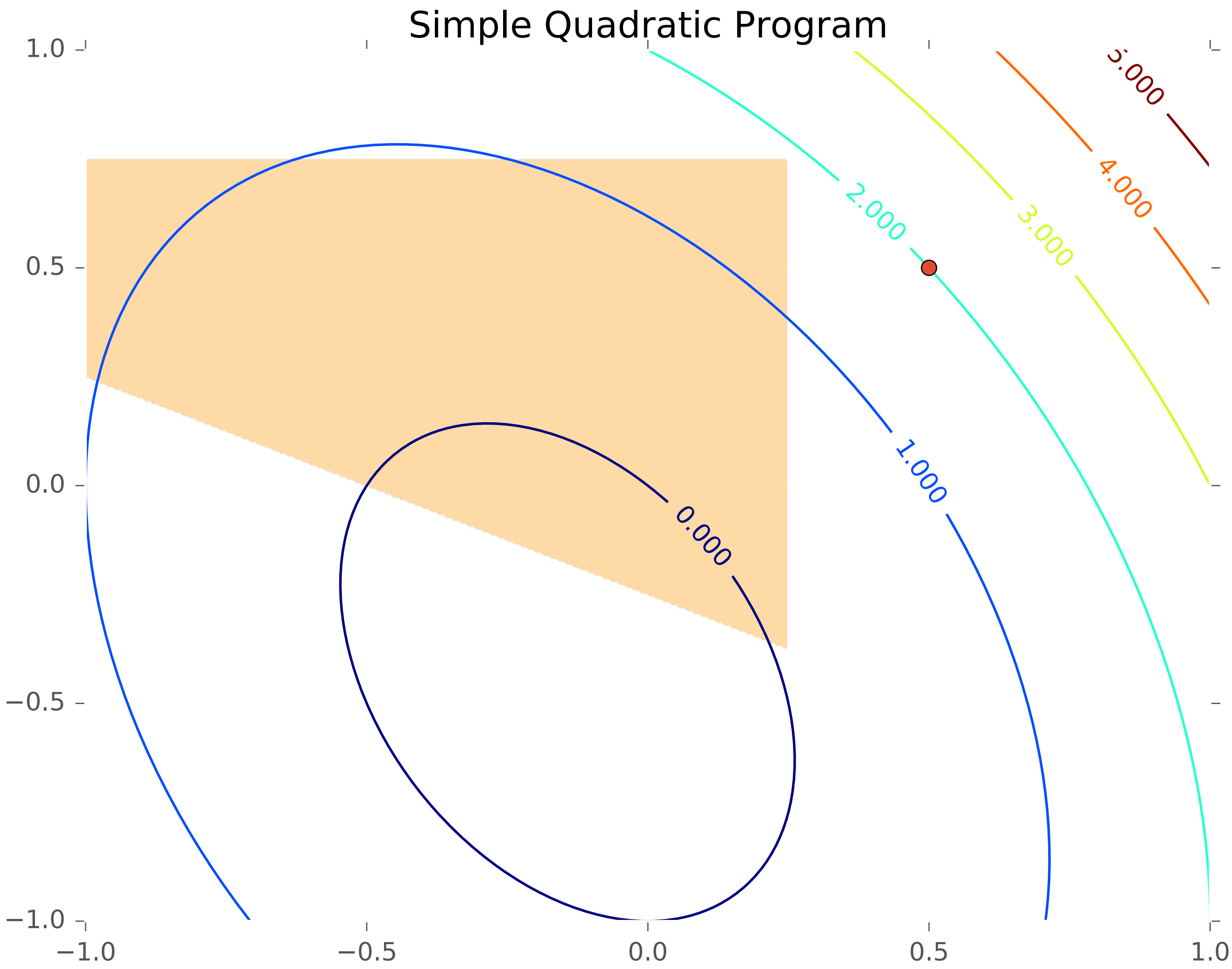
Whiteboard

- Constrained Optimization
- Linear programming
- Quadratic programming
- Example: 2D quadratic function with linear constraints

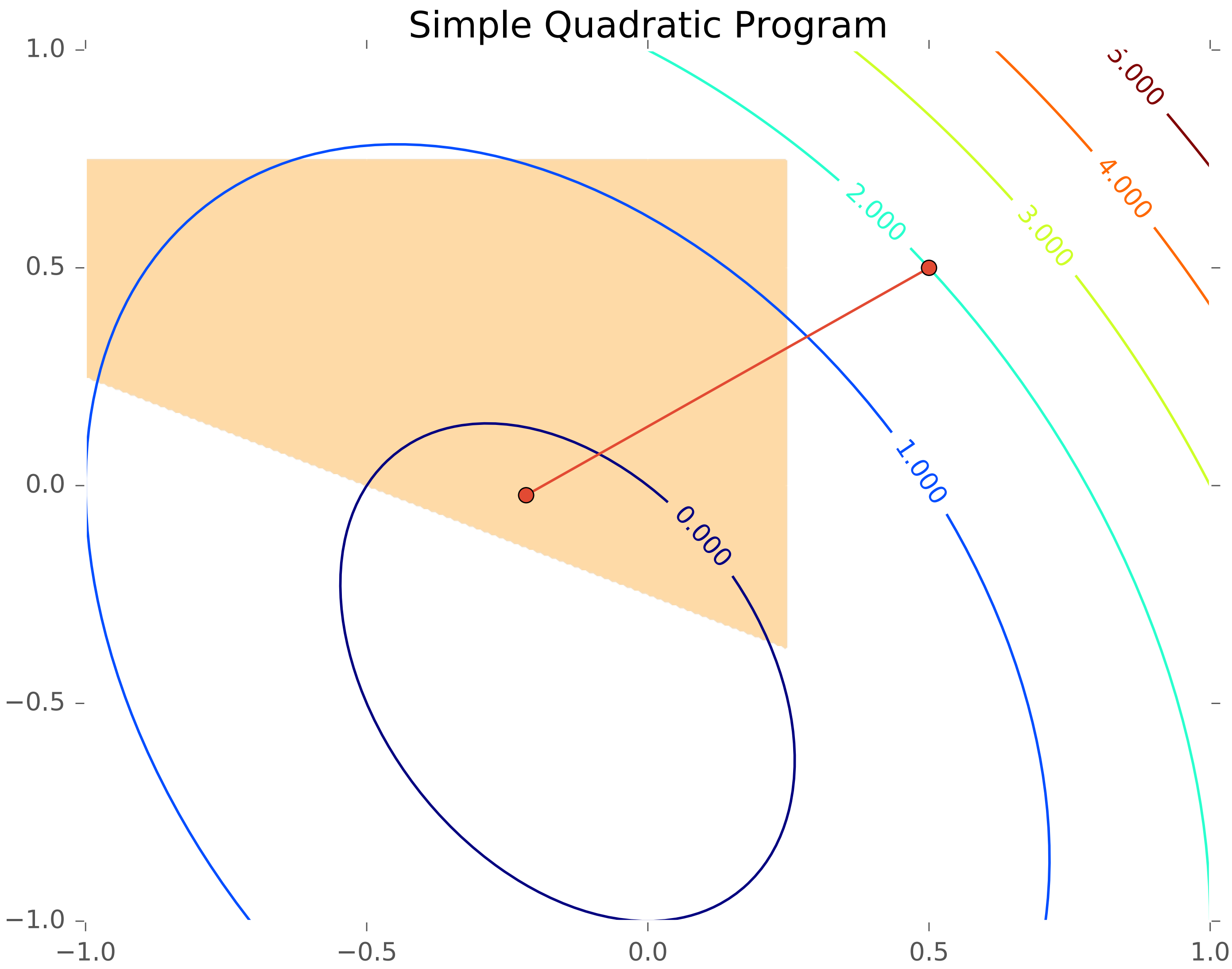
Quadratic Program



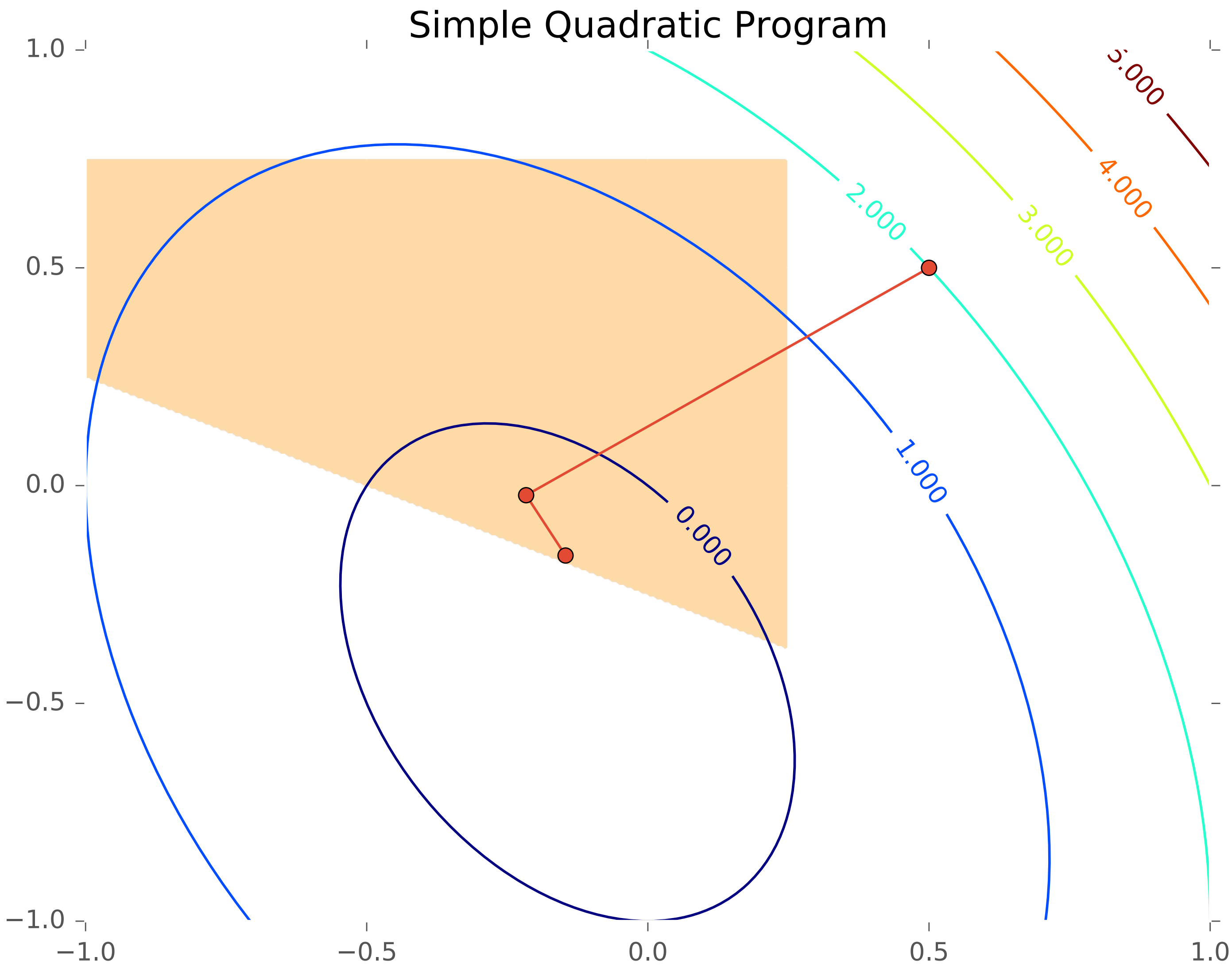
Quadratic Program



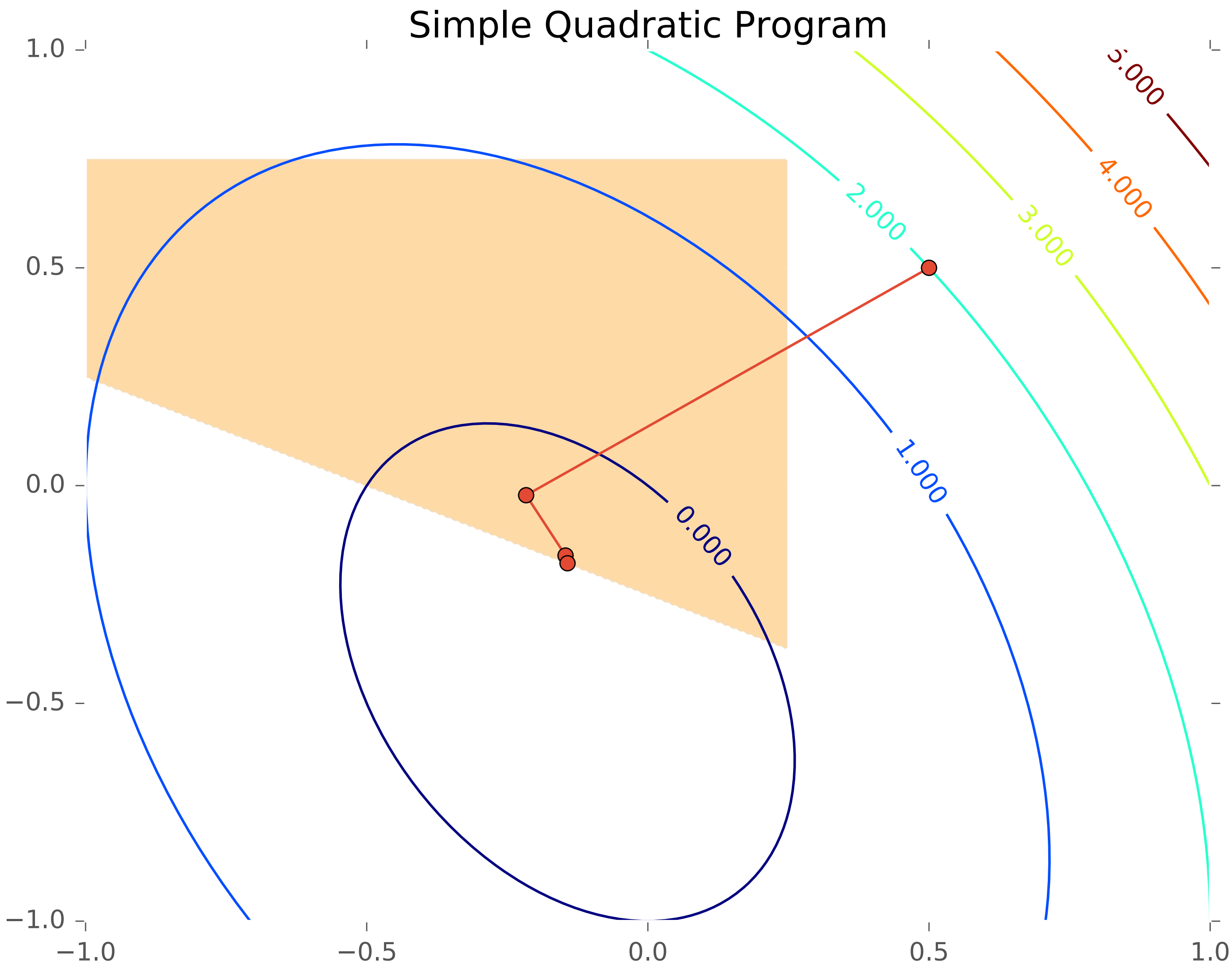
Quadratic Program



Quadratic Program



Quadratic Program

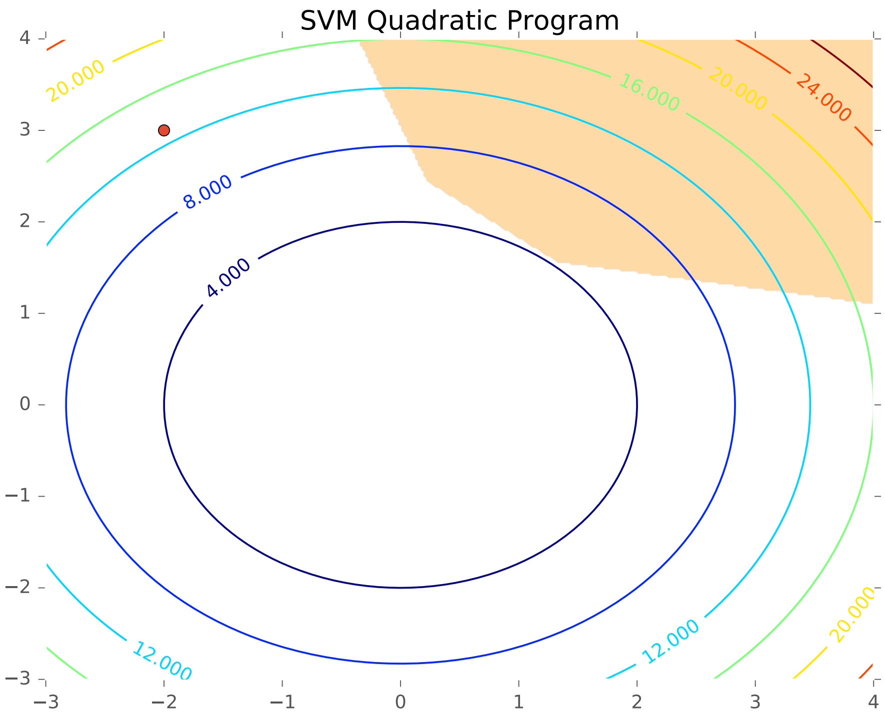


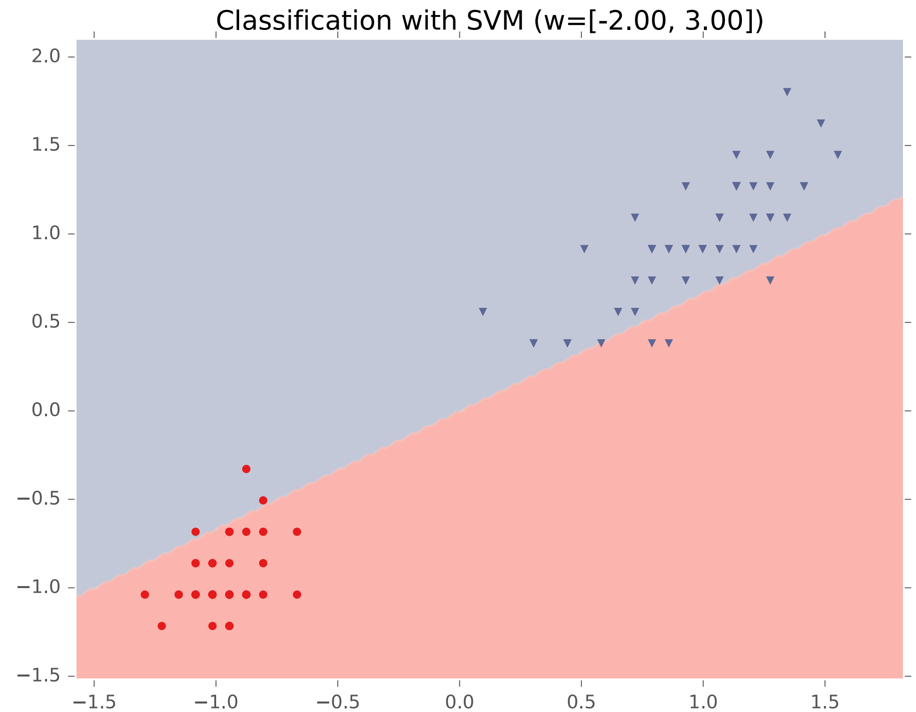
SVM

Whiteboard

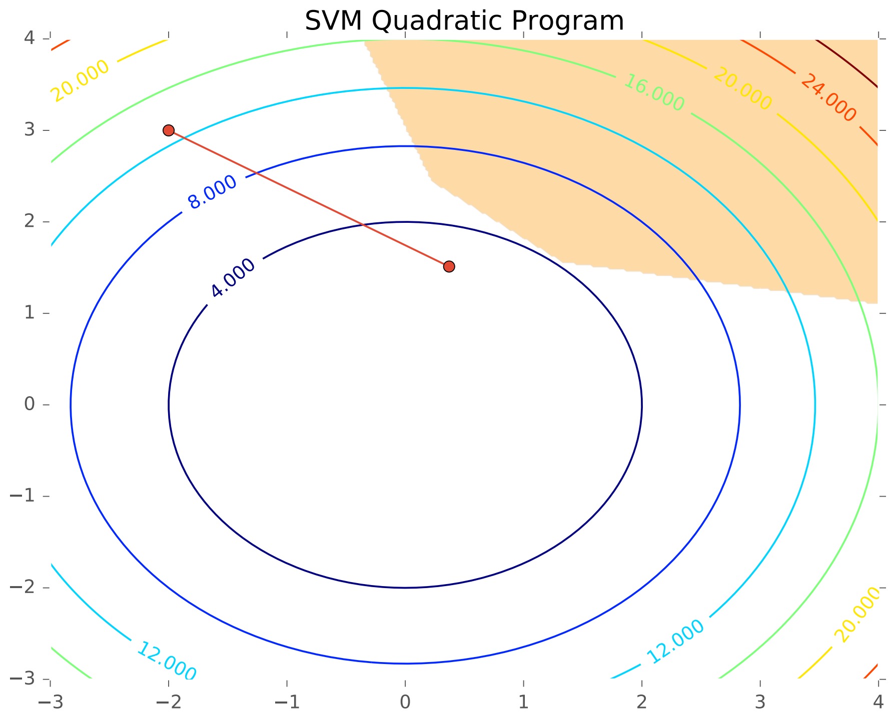
- SVM Primal (Linearly Separable Case)
- SVM Dual (Linearly Separable Case)

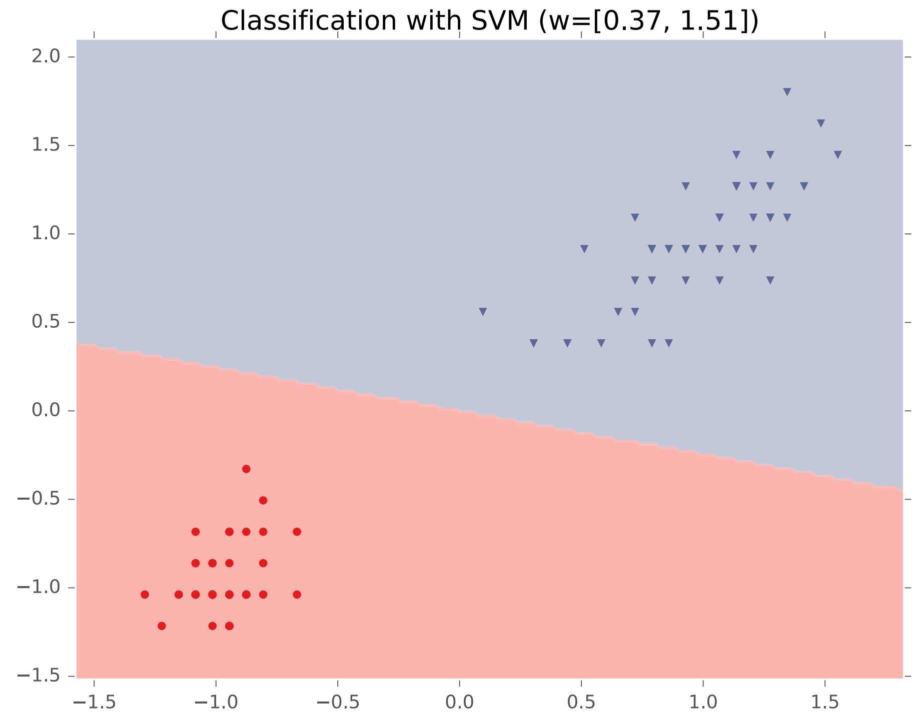
SVM QP


SVM Quadratic Program
4.000
8.000
12.000
16.000
20.000
24.000

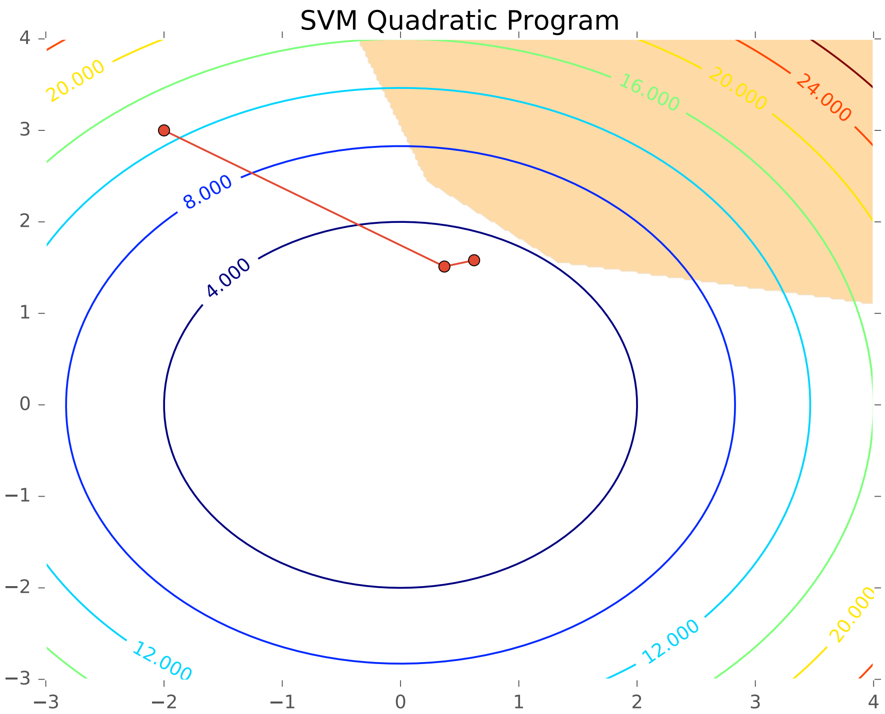

Classification with SVM (w=[-2.00, 3.00])

SVM QP

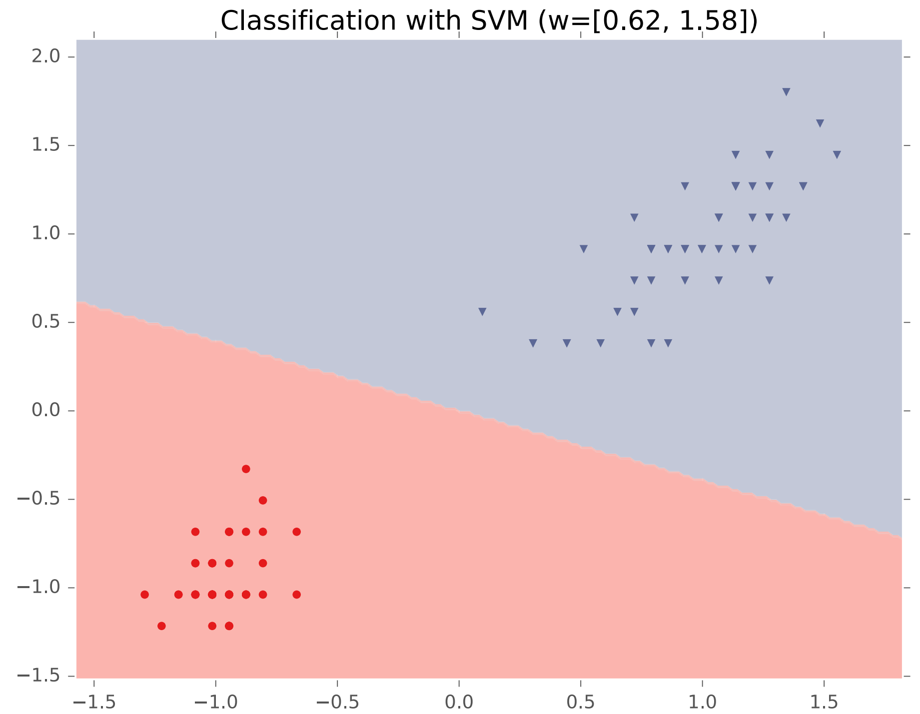

SVM Quadratic Program
4.000
8.000
12.000
16.000
20.000
24.000


Classification with SVM (w=[0.37, 1.51])

SVM QP

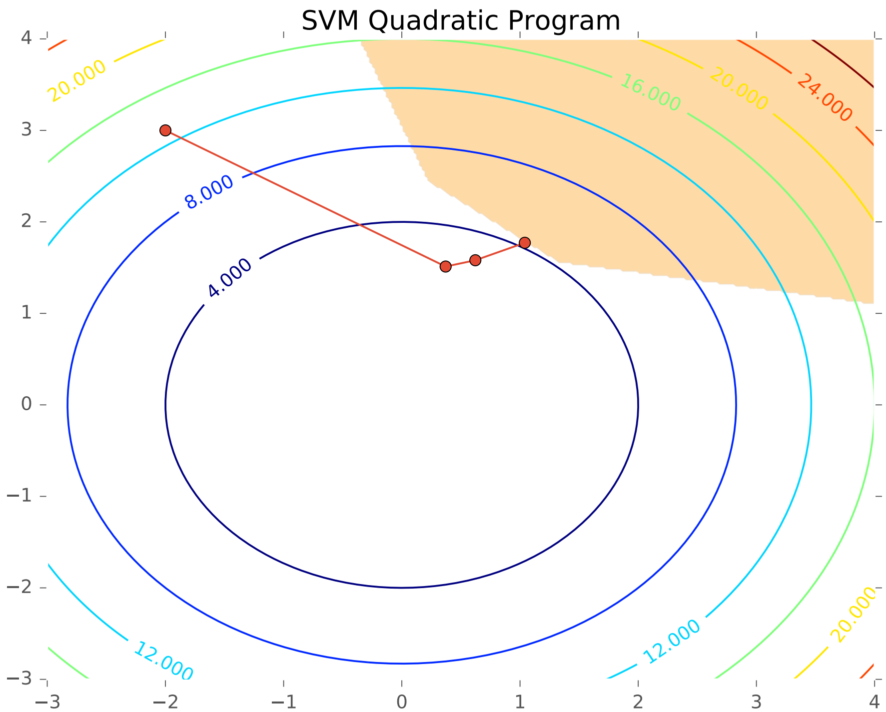


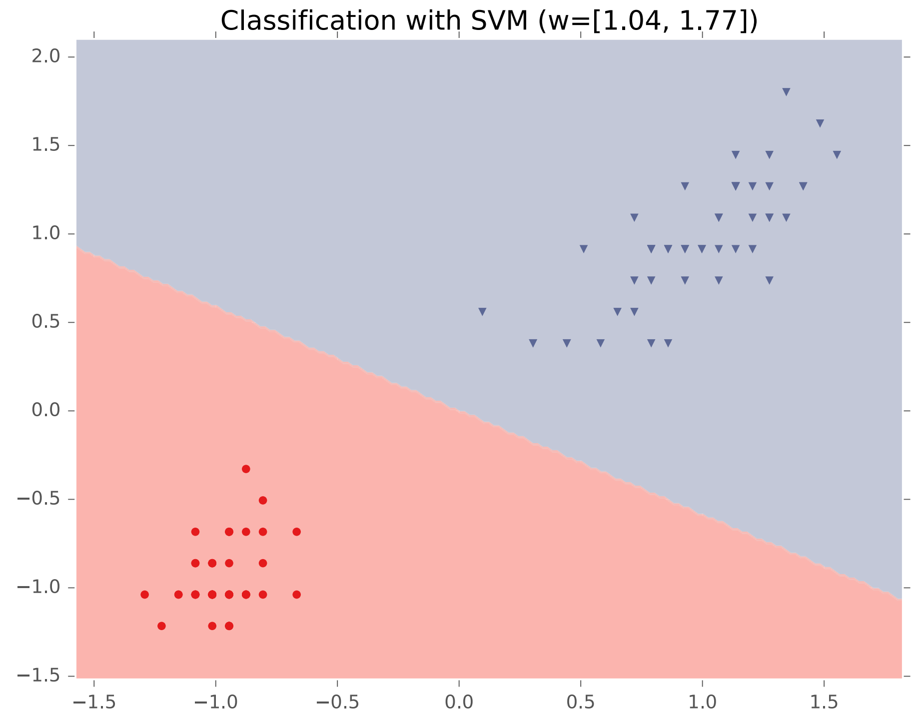
SVM Quadratic Program
4.000
8.000
12.000
16.000
20.000
24.000
−3
−2
−1
0
1
2
3
4



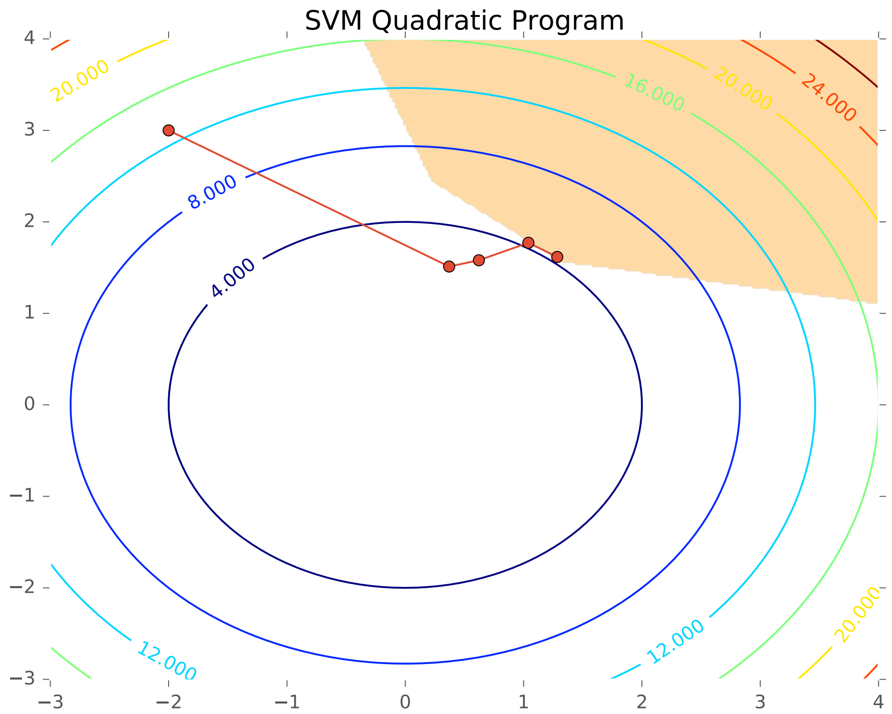
Classification with SVM (w=[0.62, 1.58])
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

SVM QP

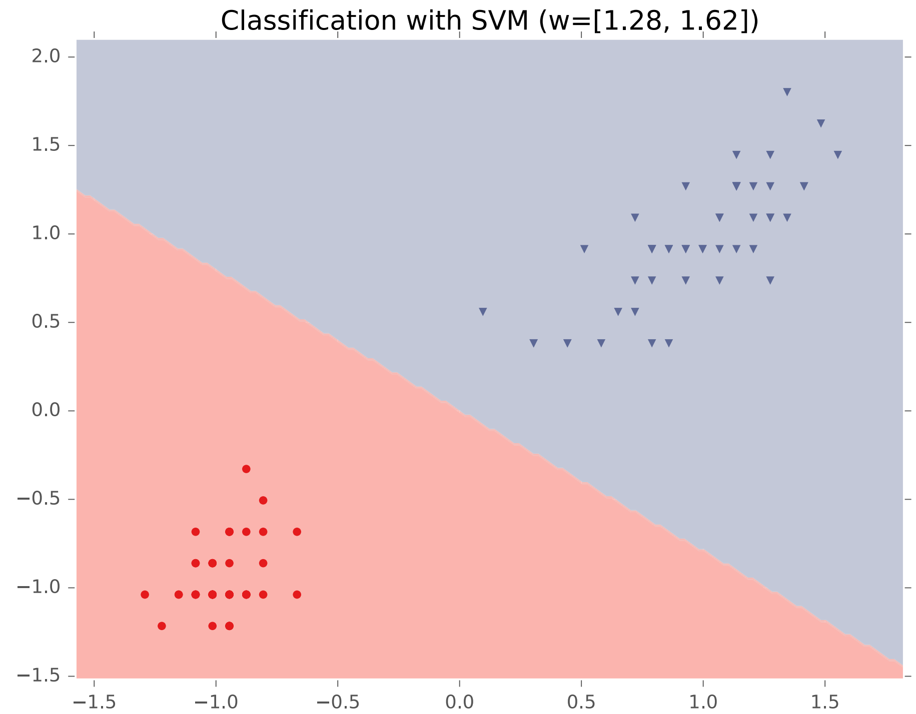

SVM Quadratic Program
4.000
8.000
12.000
16.000
20.000
24.000
−3
−2
−1
0
1
2
3
4


Classification with SVM (w=[1.04, 1.77])
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5

SVM QP

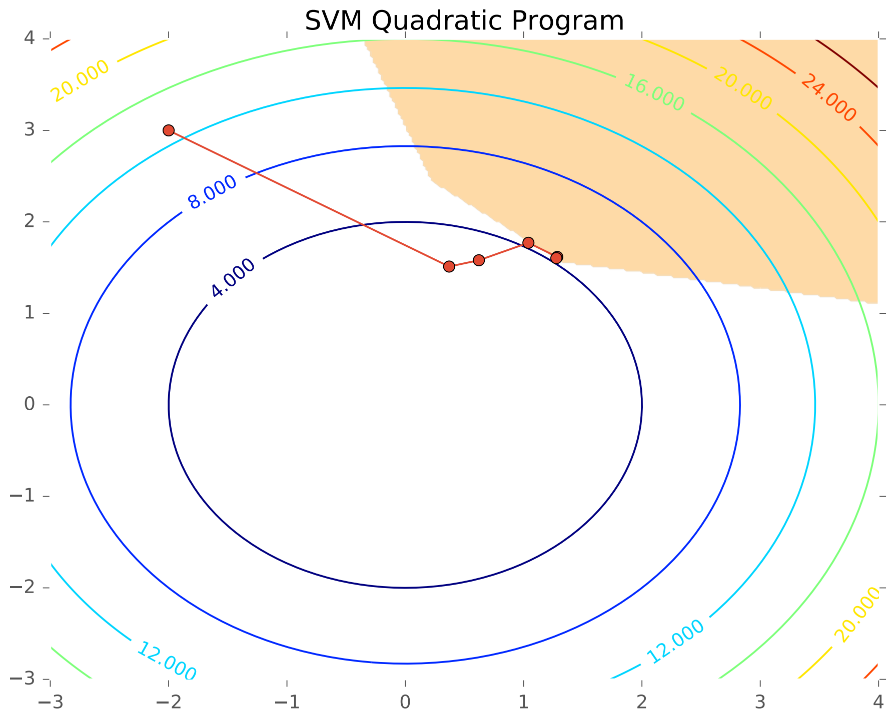


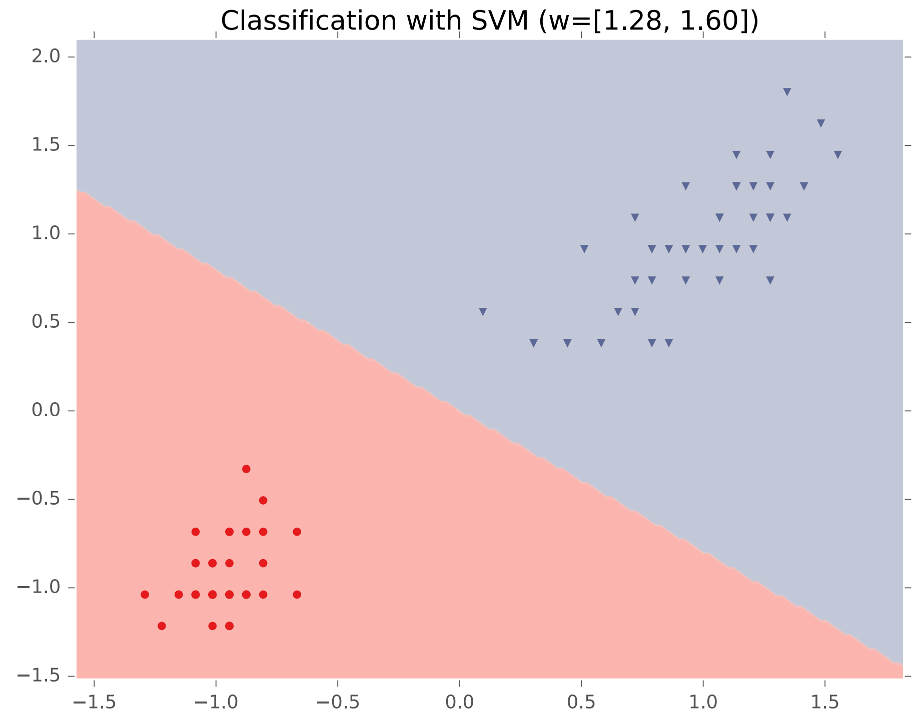
SVM Quadratic Program
4.000
8.000
12.000
16.000
20.000
24.000



Classification with SVM (w=[1.28, 1.62])

SVM QP


SVM Quadratic Program
20.000
16.000
20.000
24.000
8.000
4.000
12.000
12.000
20.000


Classification with SVM (w=[1.28, 1.60])

Support Vector Machines (SVMs)

Hard-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$

$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, N$$

Hard-margin SVM (Lagrangian Dual)

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}$$

$$\text{s.t. } \alpha_i \geq 0, \quad \forall i = 1, \ldots, N$$

$$\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

- Instead of minimizing the primal, we can maximize the dual problem
- For the SVM, these two problems give the same answer (i.e. the minimum of one is the maximum of the other)
- *Definition*: **support vectors** are those points $x^{(i)}$ for which $\alpha^{(i)} \neq 0$

SVM

Support Vector Machines (SVMs)

Hard-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, N$$

Hard-margin SVM (Lagrangian Dual)

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

- Instead of minimizing the primal, we can maximize the dual problem
- For the SVM, these two problems give the same answer (i.e. the minimum of one is the maximum of the other)
- *Definition*: **support vectors** are those points $x^{(i)}$ for which $\alpha^{(i)} \neq 0$

Not Covered

SVM EXTENSIONS

Soft-Margin SVM

Not Covered

Hard-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, N$$

Soft-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\left(\sum_{i=1}^{N} e_i\right)$$
$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i, \quad \forall i = 1, \ldots, N$$
$$e_i \geq 0, \quad \forall i = 1, \ldots, N$$

- **Question:** If the dataset not linearly separable, can we still use an SVM?
- **Answer:** Not the hard-margin version. It will never find a feasible solution.

  In the soft-margin version, we add **“slack variables”** that **allow some points to violate** the large-margin constraints.

  The constant C dictates **how large** we should allow the slack variables to be

Soft-Margin SVM

Not Covered

Hard-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$

$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, N$$

Soft-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\left(\sum_{i=1}^{N} e_i\right)$$

$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i, \quad \forall i = 1, \ldots, N$$

$$e_i \geq 0, \quad \forall i = 1, \ldots, N$$

Soft-Margin SVM

Not Covered

Hard-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, N$$

Hard-margin SVM (Lagrangian Dual)

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

Soft-margin SVM (Primal)

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\left(\sum_{i=1}^{N} e_i\right)$$
$$\text{s.t. } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - e_i, \quad \forall i = 1, \ldots, N$$
$$e_i \geq 0, \quad \forall i = 1, \ldots, N$$

Soft-margin SVM (Lagrangian Dual)

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

We can also work with the dual of the soft-margin SVM

Multiclass SVMs

Not Covered

The SVM is **inherently** a **binary** classification method, but can be extended to handle K-class classification in many ways.

1. ***one-vs-rest*:**
   - build K binary classifiers
   - train the k^{th} classifier to predict whether an instance has label k or something else
   - predict the class with largest score
2. ***one-vs-one*:**
   - build (K choose 2) binary classifiers
   - train one classifier for distinguishing between each pair of labels
   - predict the class with the most "votes" from any given classifier

Learning Objectives

Support Vector Machines

You should be able to…

1. Motivate the learning of a decision boundary with large margin
2. Compare the decision boundary learned by SVM with that of Perceptron
3. Distinguish unconstrained and constrained optimization
4. Compare linear and quadratic mathematical programs
5. Derive the hard-margin SVM primal formulation
6. Derive the Lagrangian dual for a hard-margin SVM
7. Describe the mathematical properties of support vectors and provide an intuitive explanation of their role
8. Draw a picture of the weight vector, bias, decision boundary, training examples, support vectors, and margin of an SVM
9. Employ slack variables to obtain the soft-margin SVM
10. Implement an SVM learner using a black-box quadratic programming (QP) solver