Machine Learning (CS 181): 11. Support Vector Machines

David C. Parkes and Sasha Rush

Spring 2017

Contents

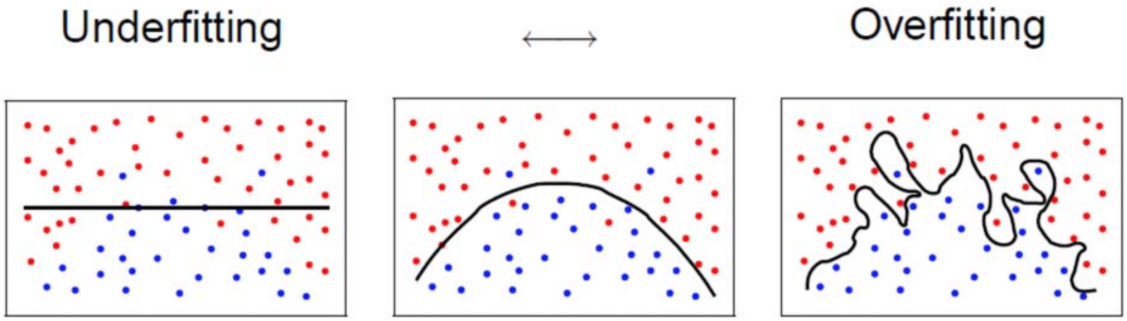
Credit

Credit: A. Zisserman (Oxford) for slides throughout this deck.

Contents

Goal: Good Generalization Performance

Minimize expected 0/1 classification error on unseen data.



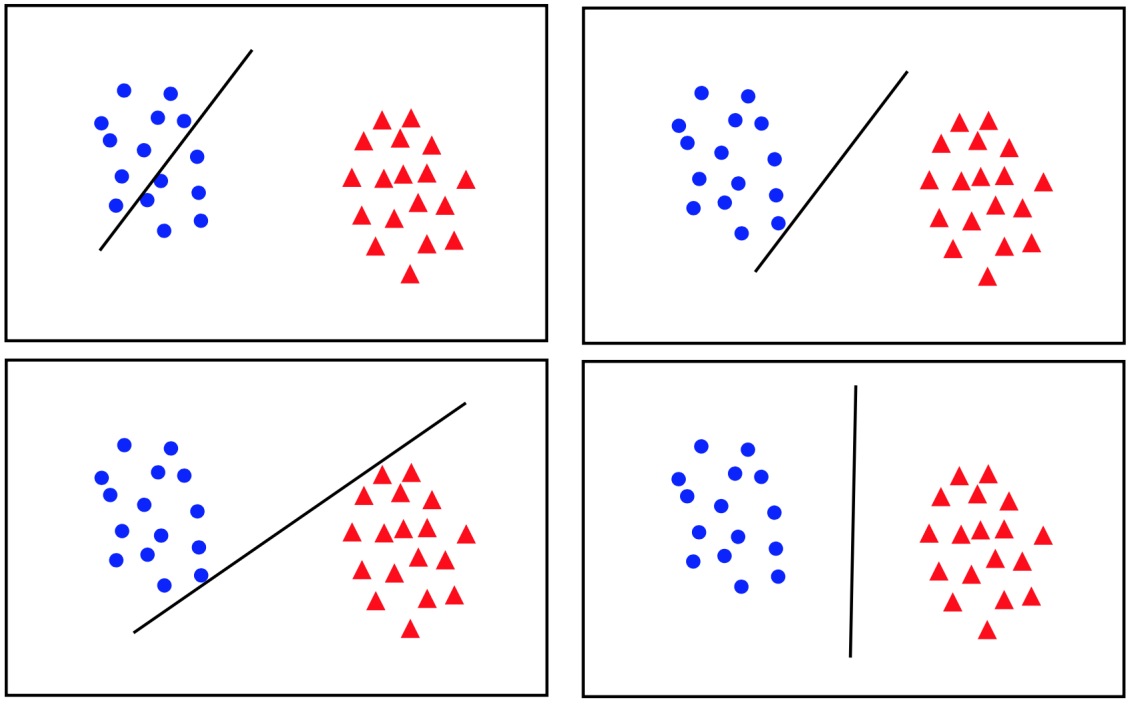
Work with discriminant-based classifiers. Learn function $h(\mathbf{x}; \mathbf{w}, w_0)$ (non-linear when used together with basis functions).

$$\hat{y} = \begin{cases} 1 & \text{if } h(\mathbf{x}; \mathbf{w}, w_0) > 0 \\ \text{-1} & \text{o.w.} \end{cases}$$

Max margin methods

- Convex. Can train via gradient descent (or other standard methods), will find global minimum.
- Coherent theory ("max margin")
- Very good performance.
- (Today) Can readily engineer new basis functions ("kernel engineering")
- (Today) Can obtain a succinct representation (thus, relatively interpretable).

Review: A good decision boundary has a large margin



The Margin of a Classifier

Definition (Margin on an example)

The margin on a correctly classified example $(\mathbf{x}_i, y_i)$ is the absolute, normalized, orthogonal distance to the decision boundary:

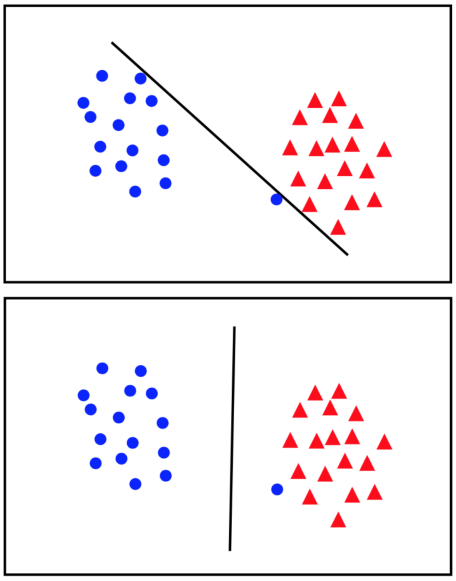
$$margin(\mathbf{x}_i, y_i; \mathbf{w}, w_0) = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{||\mathbf{w}||} \quad (\geq 0)$$

Definition (Margin on data)

The margin of a classifier on data D is the minimum margin over all correctly classified examples.

Allowing for misclassification, and non-separable data

May want to make a tradeoff between margin and number of mistakes on training data. for example, which decision boundary is best here?



Moreover, training data may not be linearly separable...

Two Max-Margin Formulations

Hard-margin formulation:

$$\min_{\mathbf{w},w_0} \quad \frac{1}{2}||\mathbf{w}||^2 \qquad \text{[P4]}$$
$$\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \text{for all } i$$

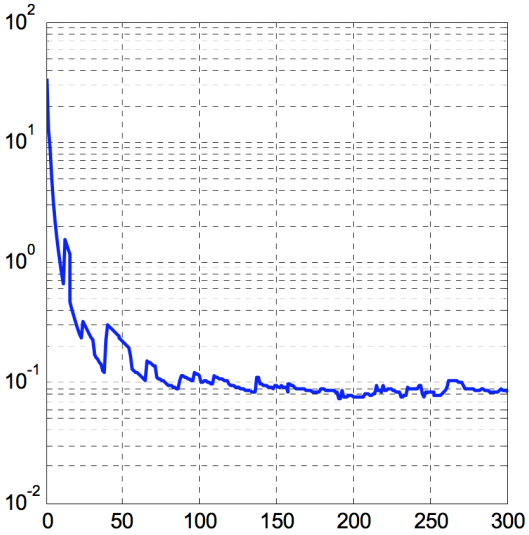
Soft-margin formulation:

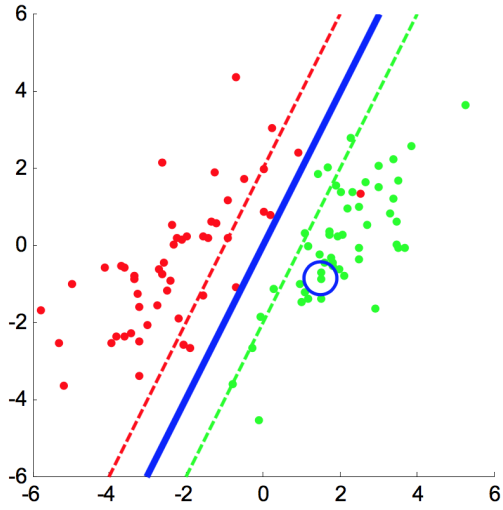
$$\min_{\mathbf{w},w_0,\boldsymbol{\xi}\geq 0} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \qquad \text{[P5]}$$
$$\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 - \xi_i, \quad \text{for all } i$$

By allowing $\xi_i > 0$ on example $\mathbf{x}_i$, we "pretend" the margin on the training data is $1/||\mathbf{w}||$; when $\xi_i > 1$, an example is misclassified. Constant $C > 0$ controls regularization.

Stochastic Gradient Descent

Soft-margin loss function is convex! Can solve via gradient descent.





Remaining Concerns

- If we use basis functions, then dimensionality of weights $\mathbf{w} \in \mathbb{R}^d$ may become very big. Gradient descent becomes slow.
- Two main ideas today:
  1. Support vectors. Use a dual formulation of the training problem, represent a classifier via "support vectors."
  2. The kernel trick. Represent basis functions as "kernel functions." Computational efficiency (scale with n, not d), and provides an elegant way to construct new basis functions.

Contents

Support Vector Machine: Overview

- We have been working with discriminant functions of the form:

$$h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^\top \mathbf{x} + w_0$$

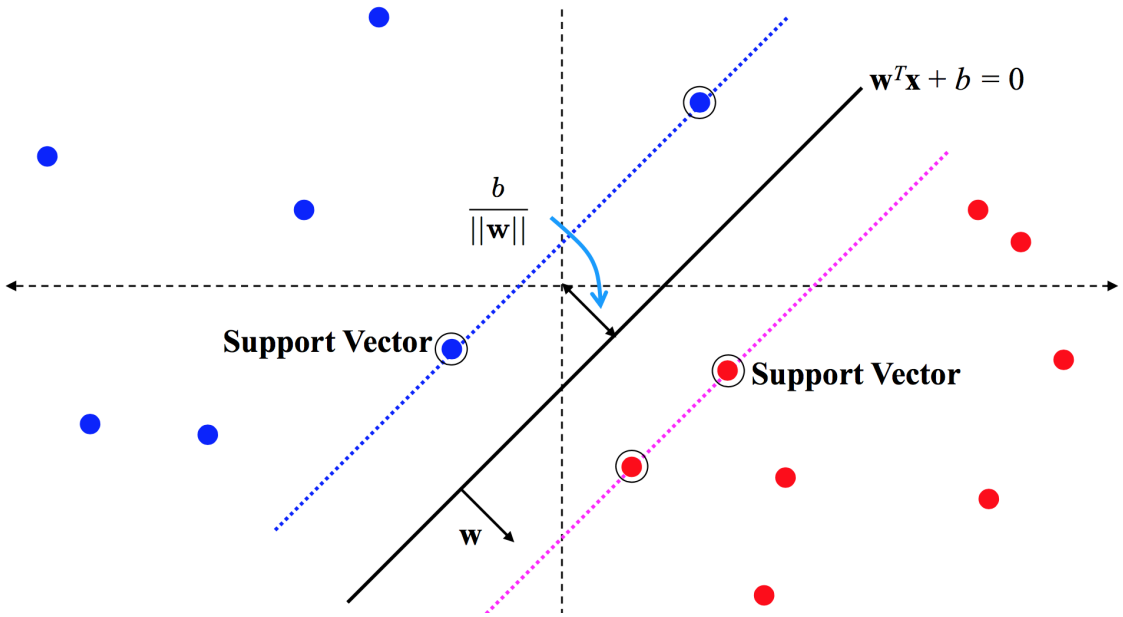
- We can also formulate the max-margin problem as one of learning a classifier of the form

$$h(\mathbf{x}; \boldsymbol{\alpha}, w_0) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + w_0$$

  with $\alpha_i \geq 0$, for all i. This is the dual form of the predictor. We will look at advantages of this formulation.

- Interpretation? (Compare with k-nearest neighbors.)

Illustrating a Support Vector Machine



[read b as w_0.][“support vectors” are examples with $\alpha_i > 0$ (will be on margin boundary, or inside margin region)].

Going from the Primal to the Dual Form (1 of 2)

Equivalent hard-margin formulations:

$$\begin{aligned} \min_{\mathbf{w},w_0} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 \end{aligned} \tag{1}$$

$$\min_{\mathbf{w},w_0} \Big[\max_{\boldsymbol{\alpha} \geq 0} \underbrace{\frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) - 1)}_{\text{Lagrangian function}} \Big] \tag{2}$$

- An optimal solution for (2) must satisfy $y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1$ for all i, otherwise, $\max_{\boldsymbol{\alpha} \geq 0}[\cdot]$ is unbounded.
- An optimal solution for (2) must set $\alpha_i = 0$ for all examples for which $y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) > 1$; thus, it minimizes $\frac{1}{2}||\mathbf{w}||^2$ and solves (1).

Going from the Primal to the Dual Form (2 of 2)

We have:

$$\min_{\mathbf{w},w_0} \left[\max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \boldsymbol{\alpha}, w_0) \right]$$

Suppose we're the "min" player. Adversary plays "max."

We can see the following holds (this is weak duality):

$$\underbrace{\max_{\boldsymbol{\alpha} \geq 0} \left[\min_{\mathbf{w},w_0} L(\mathbf{w}, \boldsymbol{\alpha}, w_0) \right]}_{\text{Dual}} \leq \underbrace{\min_{\mathbf{w},w_0} \left[\max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \boldsymbol{\alpha}, w_0) \right]}_{\text{Primal}}$$

In fact, strong duality holds, and value of optimal dual solution is EQUAL to value of optimal primal solution (out of scope, but because the primal has a quadratic objective and linear constraints.)

Solving the Dual Formulation (1 of 3)

Writing out the dual formulation, we have:

$$\max_{\boldsymbol{\alpha} \geq 0} \left[\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) - 1) \right]$$

For any $\boldsymbol{\alpha}$, we can solve for weights $\mathbf{w}$ analytically:

$$\frac{\partial L(\mathbf{w}, \boldsymbol{\alpha}, w_0)}{\partial w_j} = w_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} = 0 \quad \Leftrightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{3}$$

In regard to w_0, we need:

$$\frac{\partial L(\mathbf{w}, \boldsymbol{\alpha}, w_0)}{\partial w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{4}$$

Solving the Dual Formulation (2 of 3)

In an optimal solution, we have:

$$\begin{aligned}
&\frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) - 1) \\
=&\frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}^\top \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i - w_0 \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \\
=&\frac{1}{2}||\mathbf{w}||^2 - ||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i \qquad \{\text{subst. (3) and (4)}\} \\
=&-\frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\right)^\top \left(\sum_{i'=1}^{n} \alpha_{i'} y_{i'} \mathbf{x}_{i'}\right) + \sum_{i=1}^{n} \alpha_i \\
=&-\frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'} + \sum_{i=1}^{n} \alpha_i
\end{aligned}$$

Solving the Dual Formulation (3 of 3)

The dual form of the hard-margin formulation is:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'} \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& 0 \leq \alpha_i, \quad \text{for all } i
\end{aligned} \tag{5}$$

This has a convex objective and linear constraints! Can solve via gradient methods.

The soft-margin formulation modifies (5) to

$$0 \leq \alpha_i \leq C, \quad \text{for all } i.$$

The dual form of the classifier

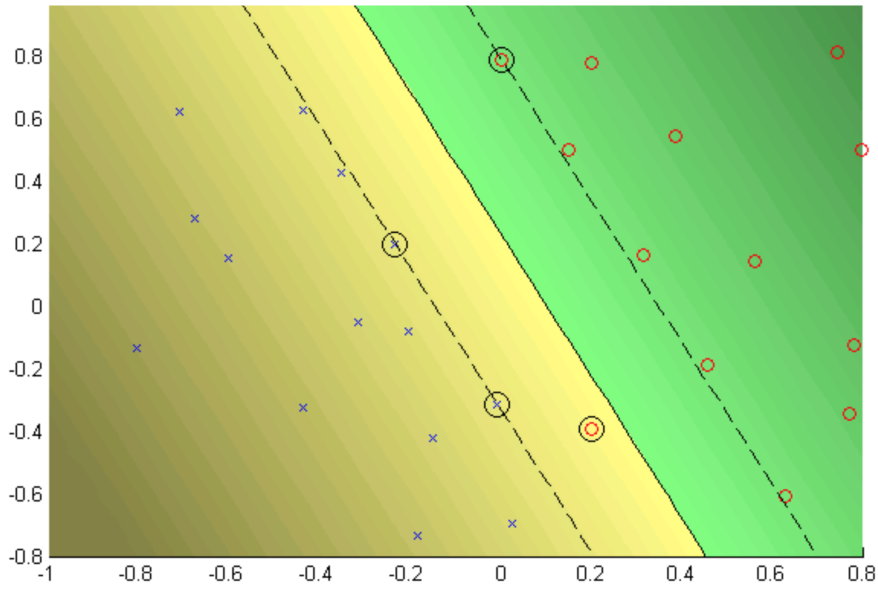
The dual form for the discriminant function is

$$h(\mathbf{x}; \boldsymbol{\alpha}, w_0) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + w_0. \tag{6}$$

- Define $Q = \{i \ : \ \alpha_i > 0\}$ as the index set of support vectors. These examples are on the margin boundary or in the margin region.
- Typically have $|Q| \ll n$ (succinct!), thus like taking a weighted vote on a small set of examples.
- Can compute w_0 by solving $y_i(\mathbf{w}^\top \mathbf{x} + w_0) = 1$ for any i on the decision boundary.

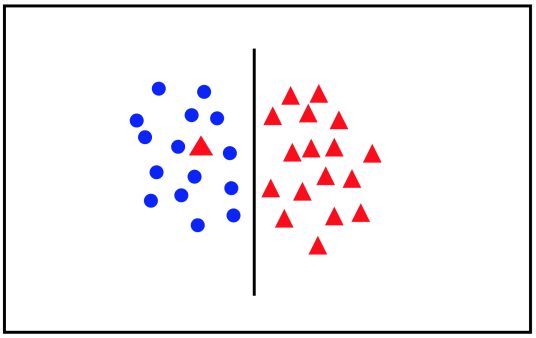
Example: SVM with Regularization (C=10)

In this example there are four "support vectors," including one example that is inside the margin region.

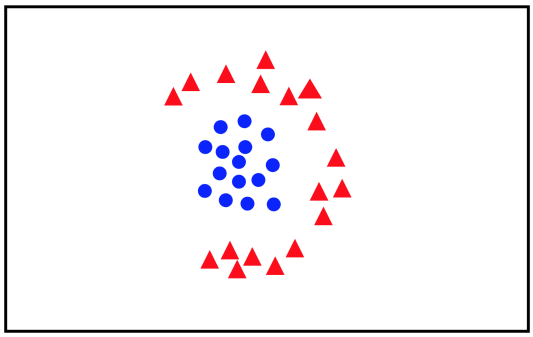


Contents

Handling data that is not linearly separable

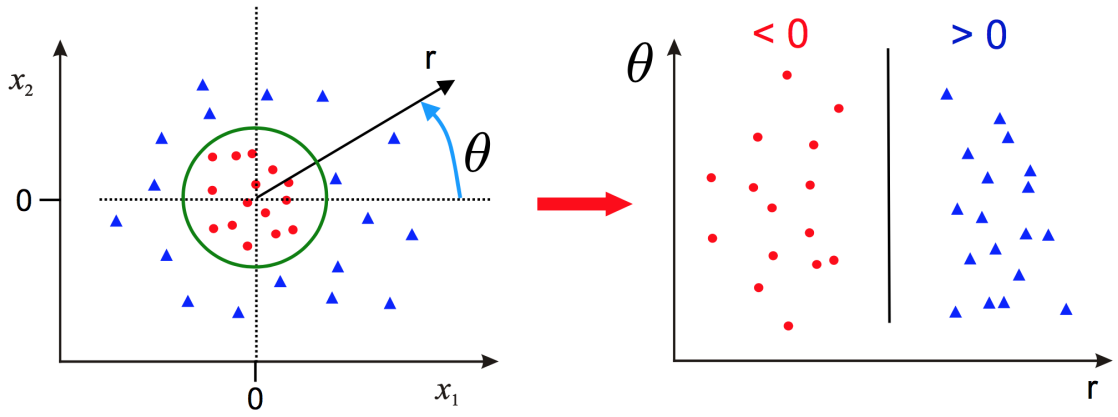


use soft-margin formulation



linear classifier not appropriate

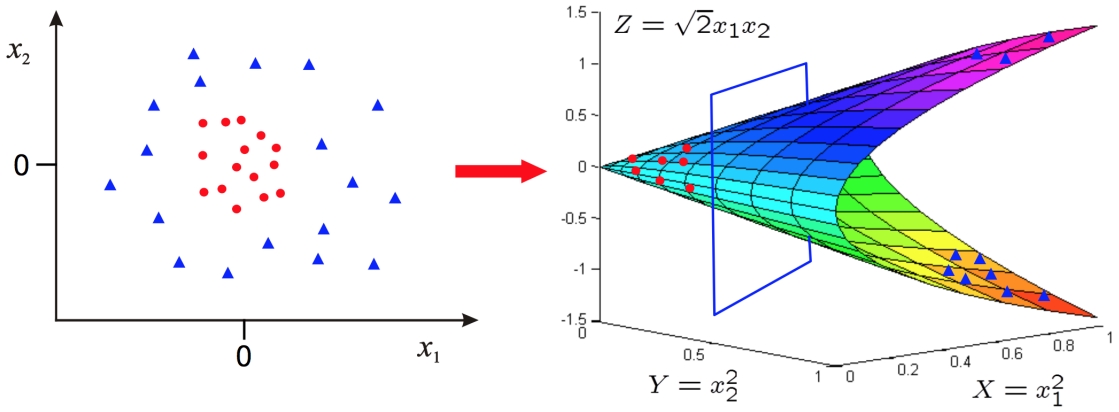
Solution 1: Use Polar coordinates



- Data is linearly separable in polar coordinates

Solution 2: Map Data to Higher Dimension

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \quad \mathbb{R}^2 \to \mathbb{R}^3$$



- Data is linearly separable in 3D (some blue large x_1^2, some large x_2^2, all red have x_1 and x_2 small)

Introducing Basis Functions

Consider $\boldsymbol{\phi} : \mathbb{R}^m \mapsto \mathbb{R}^d$. Discriminant function becomes:

$$h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + w_0$$

Soft-margin training becomes

$$\min_{\mathbf{w}, w_0} \ \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i h(\mathbf{x}_i; \mathbf{w}, w_0)))$$

Solve for $\mathbf{w} \in \mathbb{R}^d$. But, if $d \gg m$, then there are many more parameters to learn. Can this be avoided?

Dual classifier in transformed feature space

Discriminant function becomes

$$h(\mathbf{x}; \boldsymbol{\alpha}, w_0) = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}) + w_0.$$

Soft-margin training becomes

$$\max_{\boldsymbol{\alpha}} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_{i'})$$

s.t.

$$0 \leq \alpha_i \leq C \text{ for all } i, \text{ and } \sum_i \alpha_i y_i = 0.$$

Basis $\boldsymbol{\phi}$ only occurs in pairs $\boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_{i'})$. Once this scalar product computed for every pair of examples, only need to learn $\boldsymbol{\alpha} \in \mathbb{R}^n$.

The Kernel Function

Definition (Kernel function)

A kernel function $K(\cdot, \cdot)$ is an inner product $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ where $\boldsymbol{\phi}$ is a mapping from $\mathbb{R}^m$ to $\mathbb{R}^d$.

Discriminant function becomes

$$h(\mathbf{x}; \boldsymbol{\alpha}, w_0) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0.$$

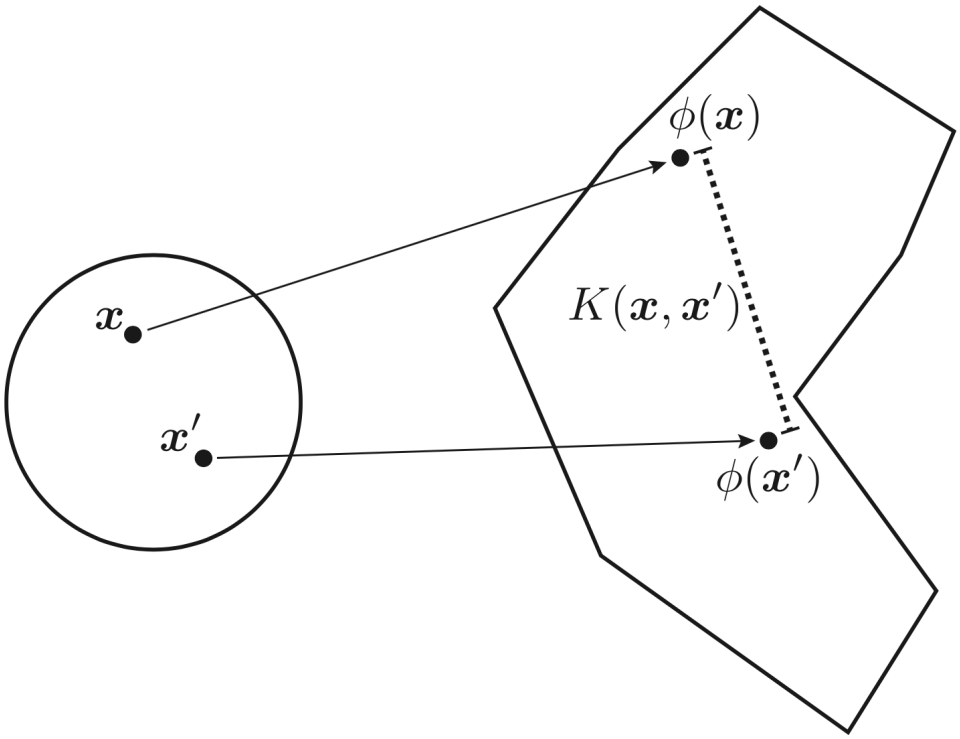
Soft-margin training becomes

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'})$$

s.t. $0 \leq \alpha_i \leq C$, for all i, and $\sum_i \alpha_i y_i = 0$.

Training problem is "kernelized." Basis function only comes in through the scalar product inside $K(\cdot, \cdot)$.

Illustrating a Kernel function



Working with Kernels

Kernel trick: the idea is to compute $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ without computing $\boldsymbol{\phi}(\mathbf{x})$ or $\boldsymbol{\phi}(\mathbf{x}')$. Just compute $K(\cdot, \cdot)$ directly!

Example kernels:

- Linear kernel $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
- Polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^q$, for integer $q \geq 2$
- Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right)$, for variance $\sigma^2 \in \mathbb{R}$

A key point is that we can work with these kernels without describing the basis function explicitly.

Example: The Quadratic Kernel

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^2$$

Suppose $\mathbf{x}, \mathbf{z} \in \mathbb{R}^2$. Then,

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^\top \mathbf{z})^2 = (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= [x_1^2 \; x_1 x_2 \; x_2 x_1 \; x_2^2]^\top [z_1^2 \; z_1 z_2 \; z_2 z_1 \; z_2^2] \end{aligned}$$

The basis function that is implicit in the quadratic kernel is

$$\boldsymbol{\phi}(\mathbf{x}) = [x_1^2 \; x_1 x_2 \; x_2 x_1 \; x_2^2]^\top.$$

For a vector in $\mathbb{R}^m$, for $m > 2$, this basis contains all pairwise terms.

Interpreting other kernel functions

- Polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^q$, for any integer $q \geq 2$
  - contains all polynomial terms up to degree q, and thus the basis function size is exponential in q (!)
- Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right)$, for variance $\sigma^2 \in \mathbb{R}$
  - compares the distance of two examples, with the importance of $\mathbf{x}'$ to $\mathbf{x}$ decaying exponentially with the squared distance
  - corresponds to a basis function that maps to an infinite dimensional space (!)

A Representation Theorem

For training data D, define the kernel matrix (or "gram" matrix) $\mathbf{K}$, an $n \times n$ matrix, with

$$K_{ii'} = \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_{i'})$$

Can just directly ask, without knowing a basis function, whether $\mathbf{K}$ is a valid kernel matrix. Allows for feature engineering!

Theorem (Mercer)

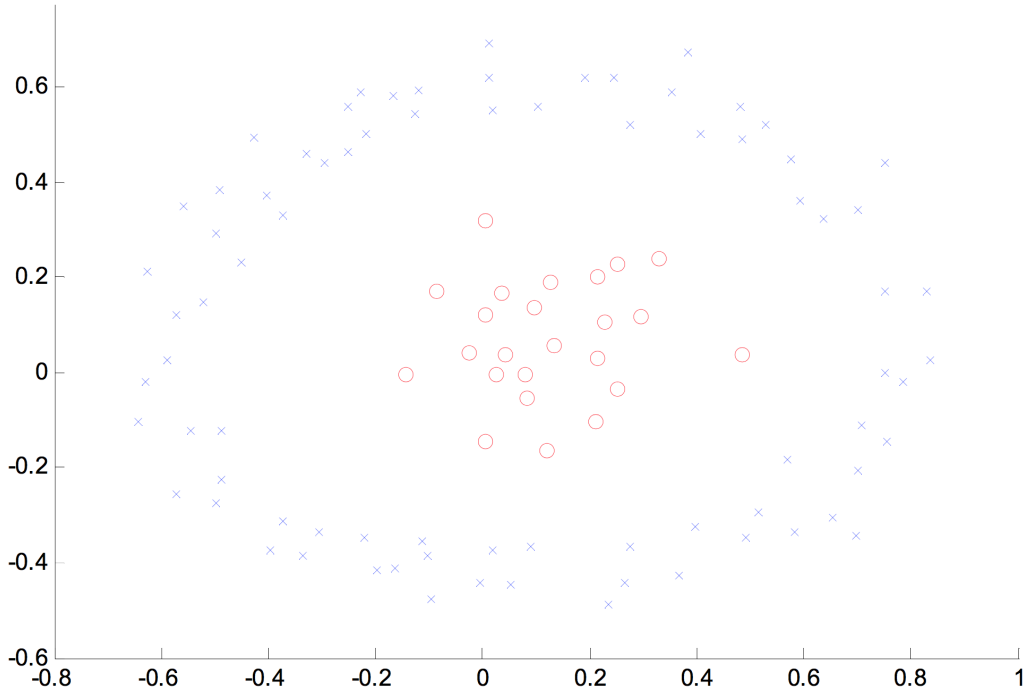
Any positive semi-definite matrix is a valid kernel matrix.

[positive semi-definite $\mathbf{K}$ requires $\mathbf{z}^\top \mathbf{K} \mathbf{z} \geq 0$, for all $\mathbf{z} \in \mathbb{R}^n$.]

Implies a compositional property: if K and K' are valid kernel functions, then $aK + bK'$ is a valid kernel function for $a > 0, b > 0$.

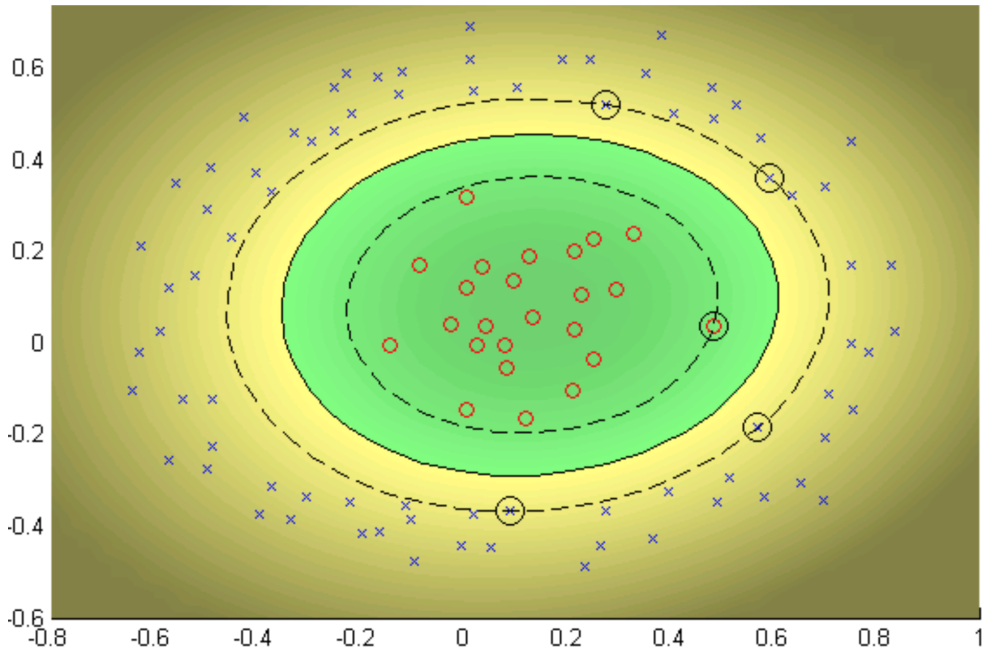
Contents

SVM Gaussian Kernel Example



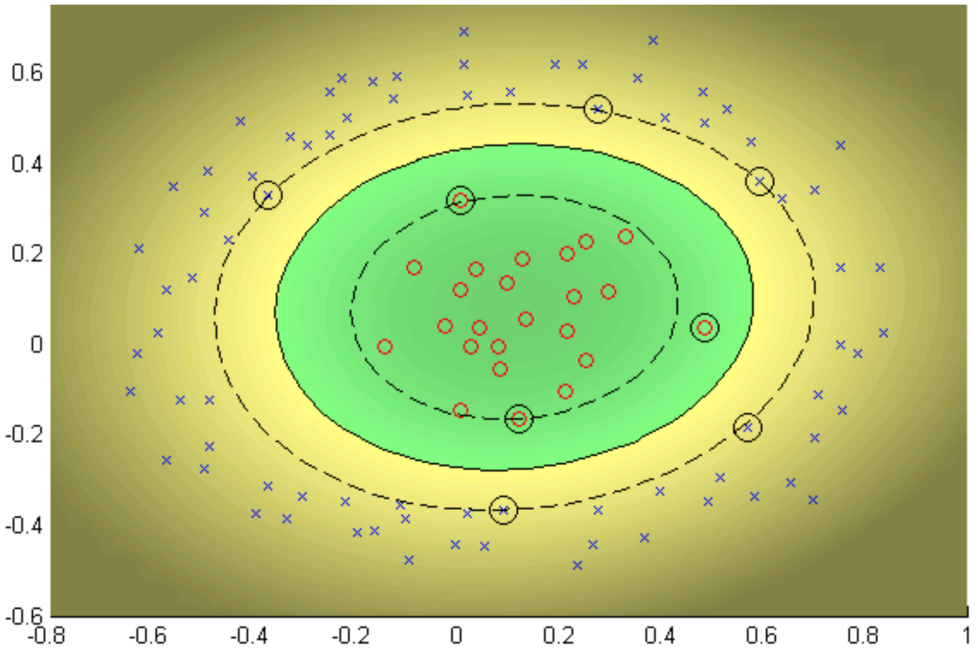
Data is not linearly separable in original feature space

SVM Gaussian Kernel Example $\sigma = 1, C = \infty$



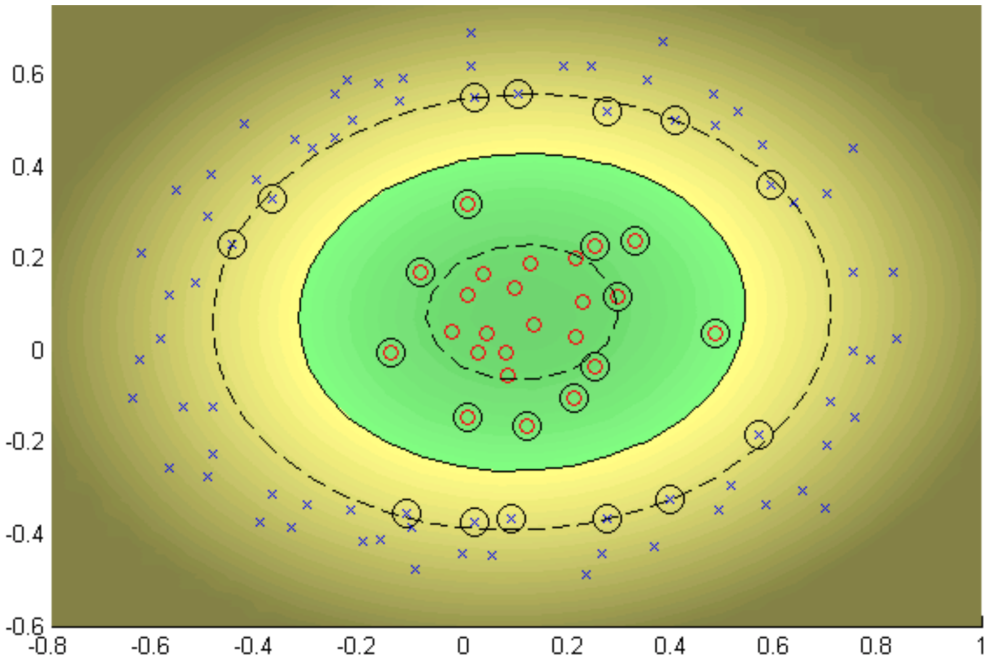
Margin boundary is dashed line, decision boundary is solid line, 5 support vectors.

SVM Gaussian Kernel Example $\sigma = 1, C = 100$



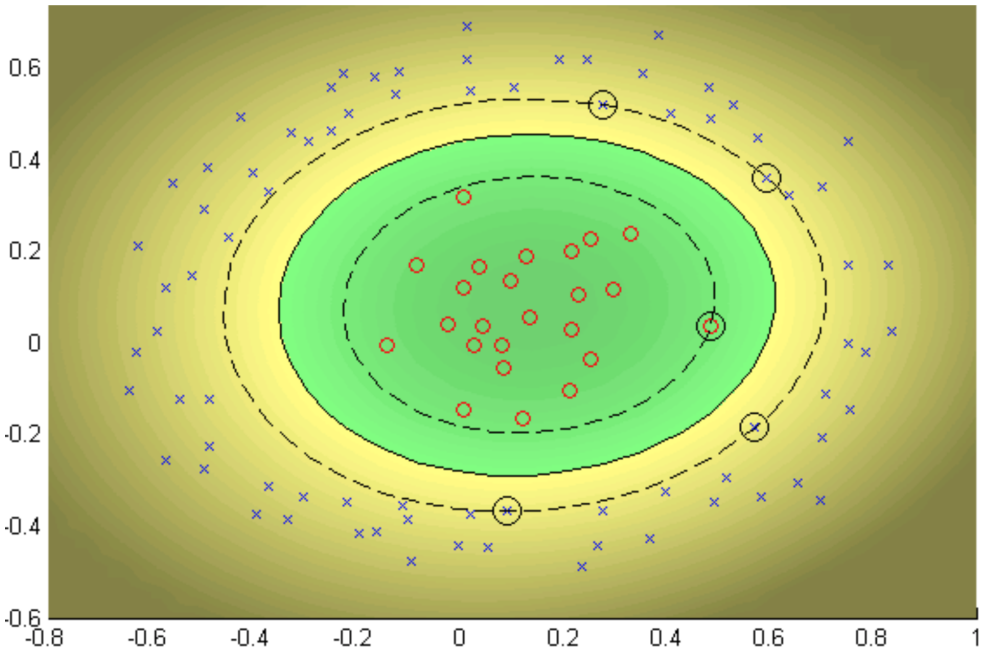
Margin boundary is dashed line, decision boundary is solid line, 8 support vectors (and larger margin, since now with regularization.)

SVM Gaussian Kernel Example $\sigma = 1, C = 10$



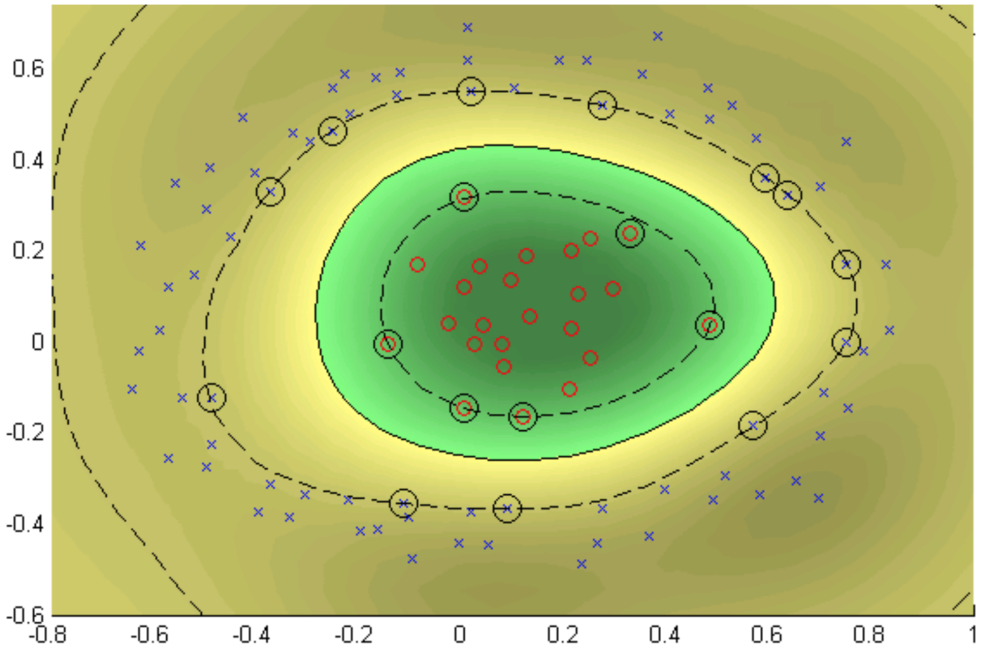
Margin boundary is dashed line, decision boundary is solid line, 24 support vectors.

SVM Gaussian Kernel Example $\sigma = 1, C = \infty$ (again)



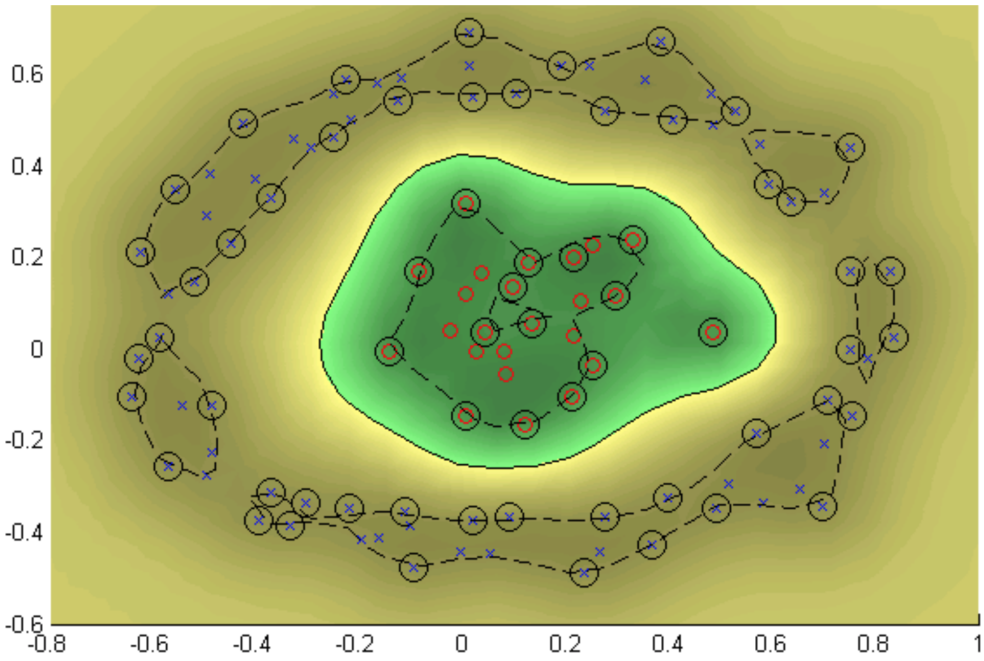
Margin boundary is dashed line, decision boundary is solid line, 5 support vectors.

SVM Gaussian Kernel Example $\sigma = 0.25, C = \infty$



Smaller variance, kernel $\exp(\frac{-||\mathbf{x}-\mathbf{x}_i||^2}{2\sigma^2})$. Only examples with small pairwise distance interact. From 5 to 18 support vectors.

SVM Gaussian Kernel Example $\sigma = 0.1, C = \infty$



Very small variance! Now there are 62 support vectors. Classifier moves towards a k-nearest neighbor classifier.

Contents

Advanced material: Kernel Trick in Primal Space (1 of 3)

- The kernel trick is very nice, but still, dual training can be slow. Tends to scale quadratically in the number of examples because each gradient step works with the kernel matrix.
- It has been possible in recent years to "kernelize" the primal (out of scope of this course!)
- This allows SGD to be used together with the kernel trick, so that work in $\mathbb{R}^m$ not $\mathbb{R}^d$ space. A powerful combination.

Advanced material: Multi-class Classification (2 of 3)

How to extend from binary classification to multi-class classification?

- One vs all. Train c separate SVMs, to predict 1 if class C_k and -1 o.w.; each with their own weight vectors. Classify a new example as the class with the largest $h_k(\mathbf{x}; \{\mathbf{w}_\ell\}_{\ell=1}^c)$.
- Multi-class SVM. Directly formulate the training problem as

$$\min \frac{1}{2} \sum_{\ell=1}^{c} ||\mathbf{w}_\ell||^2$$
$$\text{s.t.} \quad \mathbf{w}_{y_i}^\top \mathbf{x}_i \geq \mathbf{w}_k^\top \mathbf{x}_i + 1, \quad \text{for all } i, \text{ all } k \neq y_i$$

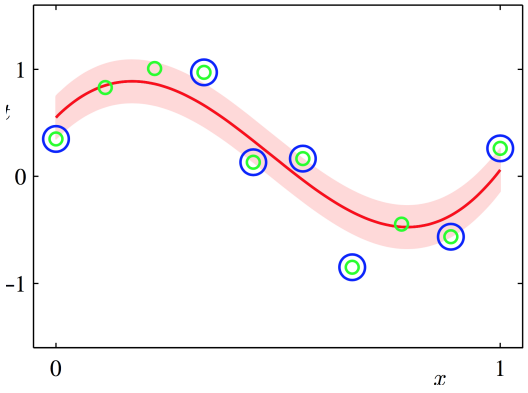
  where y_i denotes the index of the class label of example i.

Advanced material: Regression (3 of 3)

The SVM approach extends nicely to regression problems:

$$h(\mathbf{x}; \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}, w_0) = \sum_{i=1}^{n} (\alpha_i - \hat{\alpha}_i) K(\mathbf{x}_i, \mathbf{x}) + w_0$$

where $\alpha_i, \hat{\alpha}_i$ are support vectors that trace an "ϵ-tube" around the hypothesis. One set of SVs are above curve, one below curve. $\epsilon > 0$ is a parameter to tune, affecting the complexity of the model.



Contents

Summary: Max-margin methods and SVMs

- Convex. Can train via gradient descent, and will find global minimum. Either in primal (via SGD), or in dual via kernel matrix.
- Coherent theory: max margin!
- Can engineer new features without incurring penalty of larger search space ("kernel trick")
- Can obtain succinct representations (via support vectors)
- Very good performance in practice.