18-661 Introduction to Machine Learning

SVM – III

Spring 2023

ECE – Carnegie Mellon University

Outline

1. A Dual View of SVMs
2. Kernel SVM
3. Midterm Review

A Dual View of SVMs

Three SVM Formulations

Hard-margin (for separable data)

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1,\ \xi_n \geq 0,\ \forall\ n$$

Soft-margin (add slack variables)

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \ \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n,\ \xi_n \geq 0,\ \forall\ n$$

Hinge loss (define a loss function for each data point)

$$\min_{\boldsymbol{w},b} \ \sum_n \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

What Is Duality?

Duality is a way of transforming a constrained optimization problem.

It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Under strong duality condition (the details is beyond the scope...), primal and dual problems are equivalent.
- Further, due to complementary slackness, dual variables tell us whether constraints are met with $=$ or $<$
- The strong duality condition is not always true for all optimiztion problems, but is true for the soft-margin SVM problem.

What Is Duality?

Consider optimization problem with single constraint

$$\min f(x) \text{ s.t. } g(x) \leq 0$$

Define Lagrangian $L(x, \lambda) = f(x) + \lambda g(x)$, where you can think of $\lambda g(x)$ as "penalty" for constraint violation.

The above (known as primal) is equivalent to $\min_x \max_{\lambda \geq 0} L(x, \lambda)$

- If $g(x) \leq 0$, $\max_{\lambda \geq 0} L(x, \lambda) = f(x)$
- If $g(x) > 0$, $\max_{\lambda \geq 0} L(x, \lambda) = +\infty$
- Effectively enforces constraint $g(x) \leq 0$.

Dual problem: swapping the order of min and max

$$\max_{\lambda \geq 0} \underbrace{\min_x L(x, \lambda)}_{\text{known as dual function}}$$

What Is Duality?

Consider the following problem with optimizer $x^* = -1$, optimal value $\frac{1}{2}$.

$$\min \frac{1}{2}x^2 \text{ s.t. } x + 1 \leq 0$$

Lagrangian $L(x, \lambda) = \frac{1}{2}x^2 + \lambda(x + 1)$

Dual problem:

$$\max_{\lambda \geq 0} \quad \underbrace{\min_x L(x, \lambda)}_{\text{known as dual function } D(\lambda)}$$

$D(\lambda) = \min_x L(x, \lambda)$ - how to compute?

- Set $\nabla_x L(x, \lambda) = x + \lambda = 0 \Rightarrow x^*(\lambda) = -\lambda$
- $D(\lambda) = L(x^*(\lambda), \lambda) = -\frac{1}{2}\lambda^2 + \lambda$

Can show $\max_{\lambda \geq 0} D(\lambda) = \frac{1}{2}$ (achieved at $\lambda^* = 1$), same as the optimal value of primal problem). Further, $x^*(\lambda^*) = -1$, recovers optimal primal solution.

What Is Duality?

Recap: for the following problem with optimizer

$$\min \frac{1}{2}x^2 \text{ s.t. } x + 1 \leq 0$$

- Primal solution $x^* = -1$ satisfies constraint $x + 1 \leq 0$ with $=$.
- Dual solution $\lambda^* = 1$ is non-zero.

Slightly change the problem:

$$\min \frac{1}{2}x^2 \text{ s.t. } x - 1 \leq 0$$

- Primal solution $x^* = 0$ satisfies constraint $x - 1 \leq 0$ with $<$.
- Can show dual solution λ^* is zero.

This is known as complimentary slackness: suppose the constraint is $g(x) \leq 0$, then $\lambda^* g(x^*) = 0$, i.e. $\lambda^* > 0$ only when the constraint is met with $=$.

What Is Duality?

Duality is a way of transforming a constrained optimization problem.

It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Under strong duality condition (the details is beyond the scope...), primal and dual problems are equivalent.
- Further, due to complementary slackness, dual variables tell us whether constraints are met with $=$ or $<$
- The strong duality condition is not always true for all optimiztion problems, but is true for the soft-margin SVM problem.

Instead of solving the max margin (primal) formulation, we solve its dual problem which will have certain advantages we will see.

Derivation of the Dual

Here is a skeleton of how to derive the dual problem.

Recipe

1. Formulate the generalized Lagrangian function (we'll define this on the next slide) that incorporates the constraints and introduces dual variables
2. Minimize the Lagrangian function over the primal variables
3. Plug in the primal variables from the previous step into the Lagrangian to get the dual function
4. Maximize the dual function with respect to dual variables
5. Recover the solution (for the primal variables) from the dual variables

Deriving the Dual for SVM

Primal SVM

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \ n$$

The constraints are equivalent to the following canonical forms:

$$-\xi_n \leq 0 \quad \text{and} \quad 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n \leq 0$$

Lagrangian

$$L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \{1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n\}$$

under the constraints that $\alpha_n \geq 0$ and $\lambda_n \geq 0$.

Deriving the Dual of SVM

Lagrangian

$$L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \{1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n\}$$

under the constraints that $\alpha_n \geq 0$ and $\lambda_n \geq 0$.

- Primal variables: $\boldsymbol{w}$, $\{\xi_n\}$, b; dual variables $\{\lambda_n\}$, $\{\alpha_n\}$
- Minimize the Lagrangian function over the primal variables by setting $\frac{\partial L}{\partial \boldsymbol{w}} = 0$, $\frac{\partial L}{\partial b} = 0$, and $\frac{\partial L}{\partial \xi_n} = 0$.
- Substitute primal variables from the above into the Lagrangian to get the dual function.
- Maximize the dual function with respect to dual variables
- After some further maths and simplifications, we have...

Dual Formulation of SVM

Dual is also a convex quadratic program

$$\max_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{x}_m^\top \boldsymbol{x}_n$$

$$\text{s.t.} \quad 0 \le \alpha_n \le C, \quad \forall\ n$$

$$\sum_n \alpha_n y_n = 0$$

- There are N dual variables α_n, one for each data point
- Independent of the size d of $\mathbf{x}$: SVM scales better for high-dimensional feature.
- May seem like a lot of optimization variables when N is large, but many of the α_n's become zero. α_n is non-zero only if the n^{th} point is a support vector

Why Do Many α_n's Become Zero?

$$\max_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{x}_m^\top \boldsymbol{x}_n$$

$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n$$

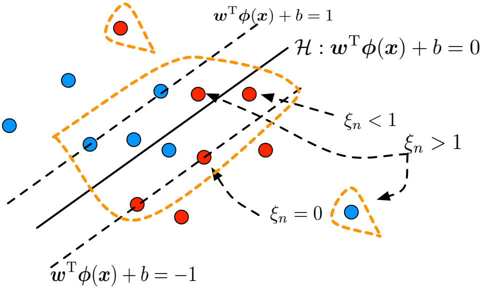
$$\sum_n \alpha_n y_n = 0$$

- By complementary slackness:

$$\alpha_n\{1 - \xi_n - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]\} = 0 \quad \forall n$$

- This tells us that $\alpha_n > 0$ only when $1 - \xi_n = y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$, i.e. (x_n, y_n) is a support vector. So most of the α_n is zero, and the only non-zero α_n are for the support vectors.
- Further, $\alpha_n < C$ only when $\xi_n = 0$. (The derivation of this is beyond the scope of today's lecture)

Visualizing the Support Vectors



- $\alpha_n = 0$: non-support vector.
- $0 < \alpha_n < C$: support vector with $\xi_n = 0$, i.e. $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$, distance to boundary $\frac{1}{\|w\|}$.
- $\alpha_n = C$: support vector with $\xi_n > 0$, hence $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] < 1$.

How to Get w and b?

Lagrangian

$$L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \{1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] - \xi_n\}$$

Recovering **w**

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_n \alpha_n y_n \boldsymbol{x}_n$$

Only depends on support vectors, i.e., points with $\alpha_n > 0$!

Recovering b

Find a sample (x_n, y_n) such that $0 < \alpha_n < C$. Using $y_n \in \{-1, 1\}$,

$$\begin{aligned} y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] &= 1 \\ \Rightarrow b &= y_n - \boldsymbol{w}^\top \boldsymbol{x}_n \\ \Rightarrow b &= y_n - \sum_m \alpha_m y_m \boldsymbol{x}_m^\top \boldsymbol{x}_n \end{aligned}$$

Summary of Dual Formulation

Primal Max-Margin Formulation

$$\begin{aligned}
\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\
\text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \ n \\
& \xi_n \geq 0, \quad \forall \ n
\end{aligned}$$

Dual Formulation

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{x}_m^\top \boldsymbol{x}_n \\
\text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad \forall \ n \\
& \sum_n \alpha_n y_n = 0
\end{aligned}$$

- In dual formulation, the # of variables is independent of dimension.
- Most of the dual variables are 0, and the non-zero ones are the support vectors.
- Can easily recover the primal solution $\boldsymbol{w}, b$ from dual solution.

Advantages of SVM

We have shown SVM:

1. Maximizes distance of training data from the boundary
2. Only requires a subset of the training points.
3. Is less sensitive to outliers.
4. Scales better with high-dimensional data.
5. Generalizes well to many nonlinear models.

The last thing left to consider is non-linear decision boundaries, or kernel SVMs, which we will cover next.

Kernel SVM

Non-linear Basis Functions in SVM

- What if the true decision boundary is not linear?
- Similar to linear regression, we can transform the feature vector **x** using non-linear basis functions. For example,

$$\phi(\boldsymbol{x}) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$$

- Replace **x** by $\phi(\mathbf{x})$ in both the primal and dual SVM formulations

Primal and Dual SVM Formulations: Kernel Versions

Primal

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \phi(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad \forall \; n$$

$$\xi_n \geq 0, \quad \forall \; n$$

Dual

$$\max_{\boldsymbol{\alpha}} \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \phi(\boldsymbol{x}_m)^\top \phi(\boldsymbol{x}_n)$$

$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall \; n$$

$$\sum_n \alpha_n y_n = 0$$

IMPORTANT POINT: In the dual problem, we only need $\phi(\boldsymbol{x}_m)^\top \phi(\boldsymbol{x}_n)$.

Dual Kernel SVM

We replace the inner products $\phi(\boldsymbol{x}_m)^\top \phi(\boldsymbol{x}_n)$ with a kernel function

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) \\
\text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad \forall\ n \\
& \sum_n \alpha_n y_n = 0
\end{aligned}
$$

What is kernel function?

- $k(\mathbf{x}_m, \mathbf{x}_n)$ is a scalar valued function that measures the similarity of $\mathbf{x}_m$ and $\mathbf{x}_n$.
- $k(\mathbf{x}_m, \mathbf{x}_n)$ is a valid kernel function if it is symmetric and positive-definite.

Why we can use kernel function to replace $\phi(\boldsymbol{x}_m)^\top \phi(\boldsymbol{x}_n)$? Each valid kernel $k(\mathbf{x}_m, \mathbf{x}_n)$ will implicitly define a $\phi(\mathbf{x})$ in the sense $k(\mathbf{x}_m, \mathbf{x}_n) = \phi(\mathbf{x}_m)^\top \phi(\mathbf{x}_n)$.

Examples of Popular Kernel Functions

Here are some example kernel functions and the corresponding feature.

- Dot product:

$$k(\mathbf{x}_m, \mathbf{x}_n) = \mathbf{x}_m^\top \mathbf{x}_n, \text{ corresponding } \phi(\mathbf{x}) = \mathbf{x}$$

- Dot product with PD matrix $\mathbf{Q}$:

$$k(\mathbf{x}_m, \mathbf{x}_n) = \mathbf{x}_m^\top \mathbf{Q} \mathbf{x}_n, \text{ corresponding } \phi(\mathbf{x}) = \mathbf{Q}^{1/2} \mathbf{x}$$

- Polynomial kernels (corresponding $\phi(\mathbf{x})$ complicated):

$$k(\mathbf{x}_m, \mathbf{x}_n) = (1 + \mathbf{x}_m^\top \mathbf{x}_n)^d, \quad d \in \mathbb{Z}^+$$

- Radial basis kernel (corresponding $\phi(\mathbf{x})$ complicated):

$$k(\mathbf{x}_m, \mathbf{x}_n) = \exp\left(-\gamma \left\|\mathbf{x}_m - \mathbf{x}_n\right\|^2\right) \text{ for some } \gamma > 0$$

and many more.

The Kernel Trick

In dual SVM, we can use any of the kernel functions discussed in the previous slide.

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) \\ \text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad \forall\ n \\ & \sum_n \alpha_n y_n = 0 \end{aligned}$$

Each choice of kernel function will correspond to doing SVM using the transformed data $\phi(\boldsymbol{x})$, but we do not need to know what exactly is $\phi(\boldsymbol{x})$.

This is allows us using more complicated $\phi(\boldsymbol{x})$ (like the $\phi(\boldsymbol{x})$ associated with radial basis function) to boost performance - without knowing what $\phi(\boldsymbol{x})$ is! This is known as "kernal trick".

Test Prediction

Learning $\boldsymbol{w}$ and b:

$$\boldsymbol{w} = \sum_n \alpha_n y_n \phi(\boldsymbol{x}_n),$$

$$b = y_n - \boldsymbol{w}^\top \phi(\boldsymbol{x}_n) = y_n - \sum_m \alpha_m y_m k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

But for test prediction on a new point $\mathbf{x}$, do we need the form of $\phi(\mathbf{x})$ in order to find the sign of $\boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$? Fortunately, no!

Test Prediction:

$$h(\boldsymbol{x}) = \text{SIGN}(\sum_n y_n \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b)$$

At test time it suffices to know the kernel function! So we really do not need to know ϕ.

Summary of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Select a kernel. In general, you don't need to concretely define $\phi(\mathbf{x})$ and can just use one of the popular kernel functions (polynomial kernel or radial kernel).

Training

$$\max_{\boldsymbol{\alpha}} \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\mathbf{x}_m, \mathbf{x}_m)$$

$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n$$

$$\sum_n \alpha_n y_n = 0$$

Prediction

$$h(\boldsymbol{x}) = \text{SIGN}(\sum_n y_n \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b)$$

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n) \text{ for } n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

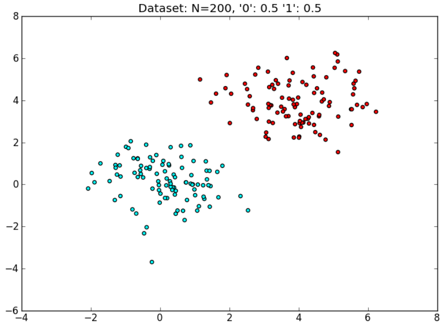


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Here is the decision boundary with linear soft-margin SVM

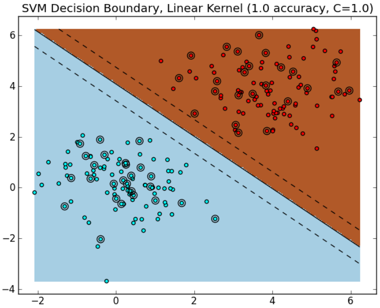


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

What if the data is not linearly separable?

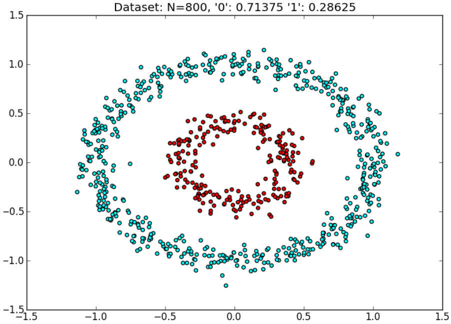


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n) \text{ for } n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

The linear decision boundary is pretty bad...

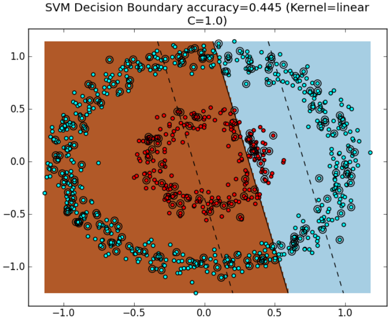
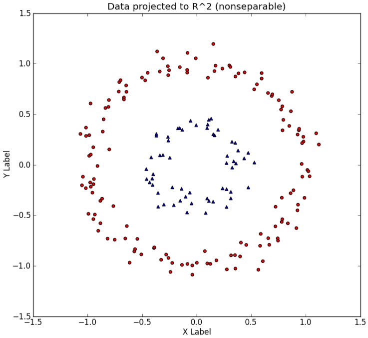


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Use feature $\phi(x) = [x_1, x_2, x_1^2 + x_2^2]$ to transform the data in a 3D space



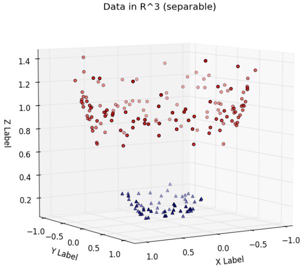


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Then find the decision boundary. How? Solve the dual problem!

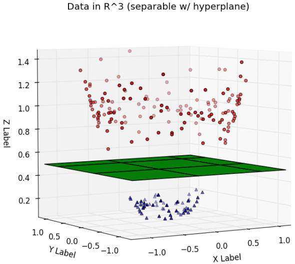
$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^\top \boldsymbol{\phi}(\boldsymbol{x}_n) \\
\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n \\
\sum_n \alpha_n y_n = 0
\end{aligned}
$$

Then find $\mathbf{w}$ and b. Predict $y = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}) + b)$.

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Here is the resulting decision boundary



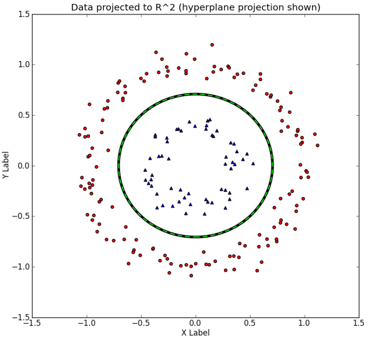


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

In the previous example, we manually defined a $\phi(\mathbf{x})$.

As mentioned in the "kernel trick" slides, in general you don't need to concretely define $\phi(\mathbf{x})$. We could select a kernel function $k(\mathbf{x}_m, \mathbf{x}_n)$ and solve the following dual SVM.

$$\max_{\boldsymbol{\alpha}} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\mathbf{x}_m, \mathbf{x}_n)$$

$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n$$

$$\sum_n \alpha_n y_n = 0$$

Test Prediction also only uses kernel:

$$h(\boldsymbol{x}) = \text{SIGN}\left(\sum_n y_n \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b\right)$$

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n)$ for $n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Effect of the choice of kernel: Polynomial kernel (degree 4)

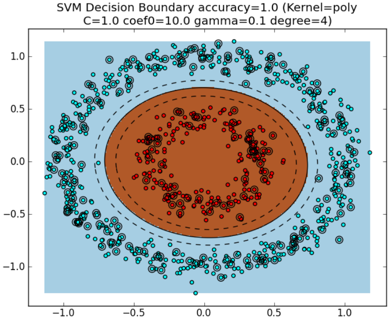


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Example of Kernel SVM

Given a dataset $\{(\boldsymbol{x}_n, y_n) \text{ for } n = 1, 2, \ldots, N\}$, how do you classify it using kernel SVM ?

Effect of the choice of kernel: Radial Basis Kernel

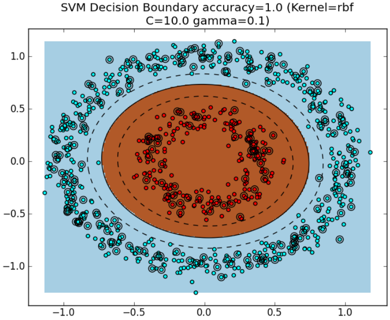


Image Source: `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Advantages of SVM

Now we have shown all of the below.

1. Maximizes distance of training data from the boundary
2. Only requires a subset of the training points.
3. Is less sensitive to outliers.
4. Scales better with high-dimensional data.
5. Generalizes well to many nonlinear models.

Midterm Review

Midterm: Concepts That You Should Know

This is a quick overview of the most important concepts/methods/models that you should expect to see on the midterm.

- MLE/MAP: how to find the likelihood of one or more observations given a system model, how to incorporate knowledge of a prior distribution, how to optimize log-likelihood functions
- Linear regression: how to formulate the linear regression optimization problem, how it relates to MLE/MAP, ridge regression, overfitting and regularization, gradient descent, bias-variance trade-off
- Naïve Bayes: Bayes' rule, naïve classification rule, why it is naïve
- Logistic regression: how to formulate logistic regression, how it relates to MLE, comparison to naïve Bayes, sigmoid function, softmax function for multi-class classification, cross-entropy function
- SVMs: hinge loss formulation, max-margin formulation, support vectors