SVMs & Good Practices for Machine Learning

Data Science Decal

Hosted by Machine Learning at Berkeley

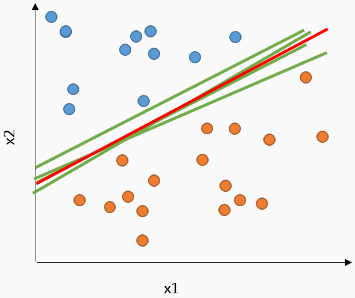
Agenda

SVM Overview

What are SVMs?

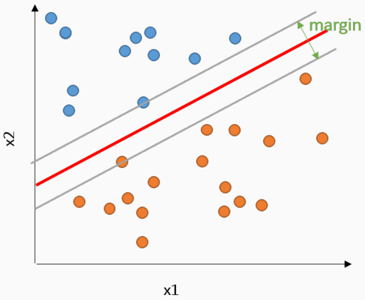
- Supervised classification algorithm
- Finds the optimal decision boundary between training points
- Widely used in practice

Optimal Hyperplane



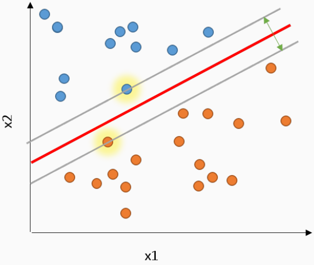
- Correctly separates all data points if possible
- Furthest away from all data points

Margin

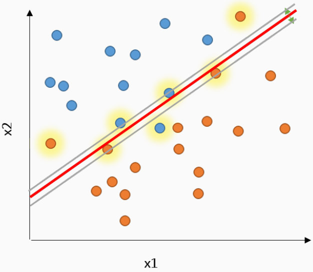


- Empty region with no data points
- Twice the distance from the hyperplane to the closest data point

Support Vectors



(a) Linearly Separable



(b) Nonlinearly Separable

- Data points that lie on margin or violate margin
- Changing support vectors changes the decision boundary

Why Use SVMs?

- SVM picks optimal hyperplane which allows model to generalize well
- Not sensitive to outliers
- Kernel functions allow for efficient computation of nonlinear features

Hard Margin vs Soft Margin SVMs

Hard Margin SVM

- Maximizes margin while requiring that all points of training data are classified correctly - support vectors cannot violate boundary
- If noise in training set, a non-linear kernel may have to be used to classify all points correctly
- Can cause high variance or overfitting

Soft Margin SVM

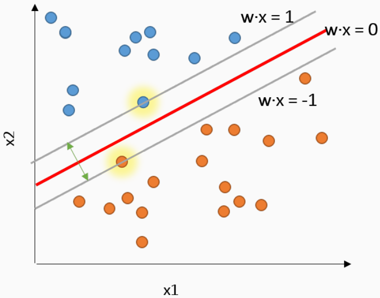
- Doesn't necessarily classify every training point correctly
- Each data point x_i has a slack variable ξ_i associated with it
- Data points that are allowed to violate margin have $\xi_i > 0$
- Slack variable is incorporated into loss term

Which one to use?

- Hard margin SVMs can overfit to training data, especially for non linearly separable data
- Soft margin SVMs have more versatility - we can decide how much of influence slack variables have, which allows us to control bias/variance

SVM Problem - Hard Margin

SVM Constraints



- Decision Boundary: $w \cdot x + b = 0$
- Edge of Margin: $w \cdot x + b = 1$ and $w \cdot x + b = -1$

SVM Constraints

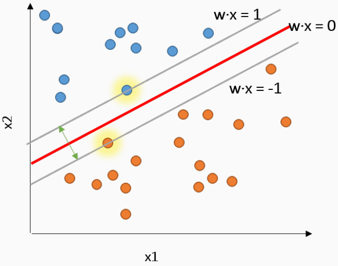


Figure 2: $y = 1$ for blue dots; $y = -1$ for orange dots

Constraint:

- For all $y_i = 1$, $w \cdot x_i + b \geqslant 1$
- For all $y_i = -1$, $w \cdot x_i + b \leqslant -1$

SVM Constraints

Constraints:

- For all $y_i = 1$, $w \cdot x_i + b \geqslant 1$
- For all $y_i = -1$, $w \cdot x_i + b \leqslant -1$

If $y_i = -1$, multiply both sides of constraint by y_i:

- $y_i(w \cdot x_i + b) \geqslant -1(y_i) = 1$

If $y_i = 1$, multiply both sides of constraint by y_i:

- $y_i(w \cdot x_i + b) \geqslant 1(y_i) = 1$

SVM Constraints

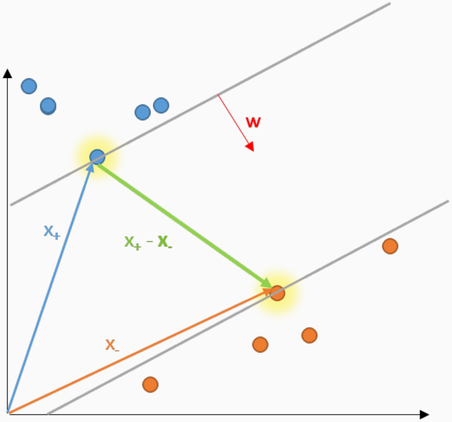
Margin is bounded by hyperplanes where

$$w \cdot x + b = 1 \quad \text{and} \quad w \cdot x + b = -1$$

such that for all data points

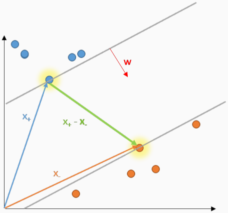
$$y_i(w \cdot x_i + b) \geqslant 1$$

Width of Margin



SVM Constraints

Width of Margin



- Width of margin is $\|proj_w(x_+ - x_-)\|$
- width $= (x_+ - x_-) \cdot \frac{w}{\|w\|}$

SVM Constraints

Width of Margin

- For support vectors, $y_i(w \cdot x_i + b) - 1 = 0$
- When $y_i = 1$, $w \cdot x_i = 1 - b$
- When $y_i = -1$, $w \cdot x_i = -1 - b$
- width $= (x_+ - x_-) \cdot \frac{w}{\|w\|} = \frac{(x_+ - x_-) \cdot w}{\|w\|}$
- width $= \frac{1 - b - (-1 - b)}{\|w\|} = \frac{2}{\|w\|}$

To maximize the margin, minimize $\|w\|$

Solution to SVM Optimization Using Lagrange Multipliers

Problem: minimize $\frac{1}{2}\|w\|^2$ such that $y_i(w \cdot x_i + b) - 1 = 0$ for support vectors

Take Derivative to Find Extremum

- $L = \frac{1}{2}\|w\|^2 - \Sigma\alpha_i[y_i(w \cdot x_i + b) - 1]$ where $\alpha_i = 0$ for non support vectors
- $\frac{\partial L}{\partial w} = w - \Sigma\alpha_i y_i x_i = 0$, so $w = \Sigma\alpha_i y_i x_i$
- $\frac{\partial L}{\partial b} = -\Sigma\alpha_i y_i = 0$, so $\Sigma\alpha_i y_i = 0$

Solution to SVM Optimization Using Lagrange Multipliers

We know there is an extremum at $w = \Sigma \alpha_i y_i x_i$

- $L = \frac{1}{2}\|w\|^2 - \Sigma \alpha_i [y_i(w \cdot x_i + b) - 1]$
- $L = \frac{1}{2}\Sigma \alpha_i y_i x_i \Sigma \alpha_j y_j x_j - \Sigma \alpha_i [y_i(w \cdot x_i + b) - 1]$
- $L = \Sigma \alpha_i - \frac{1}{2}\Sigma\Sigma \alpha_i \alpha_j y_i y_j x_i \cdot x_j$, so optimization problem depends on dot product of pairs of samples

SVM Problem - Soft Margin

SVM Constraints

Constraints: $y_i(w \cdot x_i + b) \geqslant 1 - \xi_i$ for all data points

- Support vectors that violate the margin have $\xi_i > 0$, other points have $\xi_i = 0$
- Support vectors on the margin have $y_i(w \cdot x_i + b) = 1$

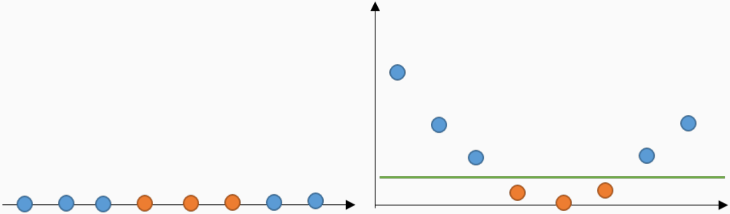
SVM Constraints

Optimization problem: minimize $\frac{1}{2}\|w\|^2 + C\Sigma_{i=1}^{n}\xi_i$ such that $y_i(w \cdot x_i + b) \geqslant 1 - \xi_i$ and $\xi_i \geqslant 0$

- C is regularization hyperparameter
- Similar to hard margin SVM, after solving soft margin SVM problem, we will find that optimization problem depends only on dot product between pairs of samples

Kernels

Nonlinearly separable data can be linearly separable in higher dimension



- Let $\Phi(x)$ be the transformation to a higher space

Kernels with SVMs

- $L = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j x_i \cdot x_j$, so optimization problem depends on $x_i \cdot x_j$
- After applying $\Phi(x)$ optimization problem depends on $\Phi(x_i) \cdot \Phi(x_j)$
- $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$
- Kernels allow us to compute $\Phi(x_i) \cdot \Phi(x_j)$ without computing $\Phi(x_i)$

Example: Polynomial Kernel

- Polynomial Kernel: $K(x, y) = (1 + x \cdot y)^p$
- Let $x =< x_1, x_2 >$, $y =< y_1, y_2 >, p = 2$
- $K(x, y) = (1 + x \cdot y)^2 = (1 + x_1y_1 + x_2y_2)^2$
  $= 1 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 + 2x_1y_1x_2y_2$
  $=< 1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2 > \cdot <$
  $1, y_1^2, y_2^2, \sqrt{2}y_1, \sqrt{2}y_2, \sqrt{2}y_1y_2 >$

Example: Polynomial Kernel

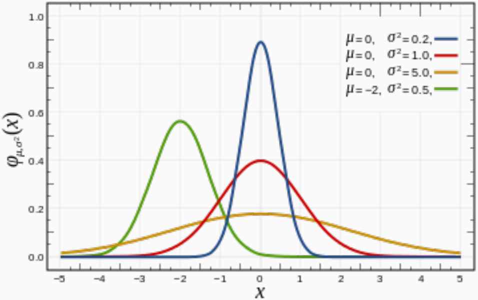
- Computing degree p features of d dimensional input takes $O(d^p)$ time
- Using Polynomial Kernel, $\Phi(x_i) \cdot \Phi(x_j)$ can be computed in $O(d)$ time, even if $\Phi(x_i) \cdot \Phi(x_j)$ has length $O(d^p)$

Decision Function

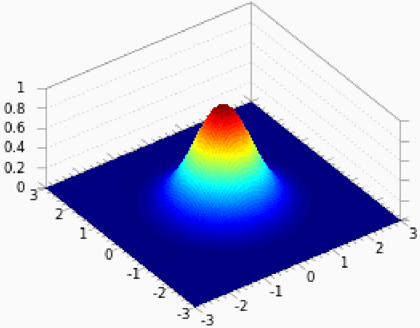
- Without Kernels: $h(z) = \Sigma_{i=1}^{d} w_i z_i$
- With Kernels: $h(z) = \Sigma_{i=1}^{d} w_i K(X_i, z)$

Example: Gaussian Radial Basis Kernel

- $K(x, y) = \exp(-\frac{|x-y|^2}{2\sigma^2}) = \exp(-\gamma|x-y|^2), \gamma > 0$
- $\Phi(x)$ is infinite vector
- $\Phi(x) \cdot \Phi(y)$ converges to $K(x, y)$
- Large γ = small σ, which makes Gaussian narrower $\Rightarrow$ causes high variance, lower bias



Example: Gaussian Kernel as Similarity Function



- $K(x, y)$ assigns high value for points that are near each other

Why Use Gaussian Kernels?

- Gives a smooth decision function
- Behaves like smoother k-nearest-neighbors
- Oscillates less than polynomial kernels, depending on value of σ
- Sample points closer to z have greater impact on prediction of z

SVMs in Practice

Pros:

- Finds optimal decision boundary between data
- Can capture complex nonlinear relationship between data with more efficiency than manually calculating features, while still maintaining simplicity of model

Cons:

- Calculating many higher dimensional features still takes long time, especially if input size is large
- Data transformation and boundary after kernel trick is hard to interpret $\Rightarrow$ SVMs are often treated like black box

SVMs in Practice

- Challenging to implement from scratch efficiently
- Better use of time to know how to use an SVM well rather than know how to code an SVM from scratch

Hyperparameter C

Soft Margin SVM: minimize $\frac{1}{2}\|w\|^2 + C\Sigma_{i=1}^{n}\xi_i$ such that $y_i(w \cdot x_i + b) \geqslant 1 - \xi_i$ and $\xi_i \geqslant 0$

- C represents how unacceptable it is to misclassify **training** data points

Hyperparameter C

Large C:

- Similar to hard margin SVM - goal is to misclassify few training points
- Often results in small margins
- Very sensitive to outliers
- Risk of overfitting

Small C:

- Maximizes margin at cost of misclassifying training data points
- Risk of underfitting

Hyperparameter γ

- γ applies for polynomial, RBF, and sigmoid kernels in sklearn

Small γ:

- Larger variance in Gaussian RBF kernel, so each support vector has a greater influence on class of points far away from it
- Leads to high variance models with risk of overfitting

Large: γ

- Leads to high bias models with risk of underfitting

SVM Demo

Debugging ML Algorithms

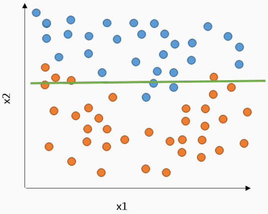
Common Issue 1 - Bias vs Variance Tradeoff

- Problem: Classifier has a test error that is too high

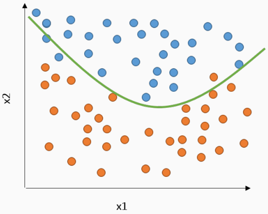
Common Issue 1 - Bias vs Variance Tradeoff

- Problem: Classifier has a test error that is too high
- Solution: Check if classifier is overfitting or underfitting

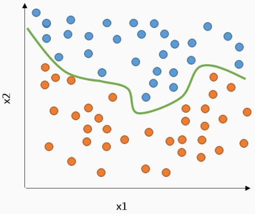
Common Issue 1 - Bias vs Variance Tradeoff



(a) High Bias

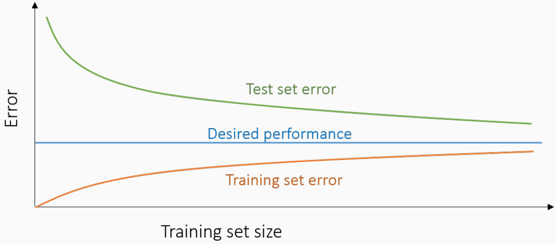


(b) Balanced



(c) High Variance

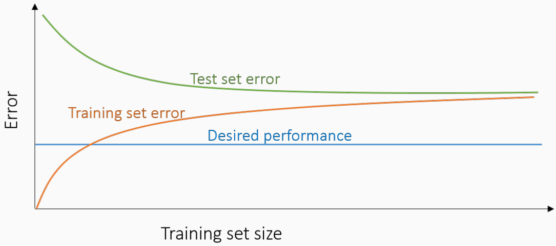
Characteristics of High Variance (Overfitting)



- Training error less than test error
- Test error decreases as training set size increases

Common Issue 1 - Bias vs Variance Tradeoff

Characteristics of High Bias (Underfitting)



- Training error similar to test error
- Test error plateaus as training set size increases

Common Issue 1 - Bias vs Variance Tradeoff

Mini Quiz: Which of the following will fix high bias? Which will fix high variance?

- Obtain more training examples
- Reduce number of features
- Increase number of features
- Use regularization for linear or logistic regression

ML@B

Mini Quiz: Which of the following will fix high bias? Which will fix high variance?

- Obtain more training examples - High Variance
- Reduce number of features
- Increase number of features
- Use regularization for linear or logistic regression

Common Issue 1 - Bias vs Variance Tradeoff

Mini Quiz: Which of the following will fix high bias? Which will fix high variance?

- Obtain more training examples - High Variance
- Reduce number of features - High Variance
- Increase number of features
- Use regularization for linear or logistic regression

Common Issue 1 - Bias vs Variance Tradeoff

Mini Quiz: Which of the following will fix high bias? Which will fix high variance?

- Obtain more training examples - High Variance
- Reduce number of features - High Variance
- Increase number of features - High Bias
- Use regularization for linear or logistic regression

Common Issue 1 - Bias vs Variance Tradeoff

Mini Quiz: Which of the following will fix high bias? Which will fix high variance?

- Obtain more training examples - High Variance
- Reduce number of features - High Variance
- Increase number of features - High Bias
- Use regularization for linear or logistic regression - High Variance

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Optimization Objective

- Suppose for a particular dataset a SVM gives a higher accuracy than logistic regression
- When minimizing the same loss function $J(\theta)$, if $J(\theta_{LR}) \leqslant J(\theta_{SVM})$, logistic regression is the wrong optimization objective to use
- Even if logistic regression is fully optimized, it cannot beat accuracy of SVM

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Optimization Algorithm

- Suppose for a particular dataset a SVM gives a higher accuracy than logistic regression
- When minimizing the same loss function $J(\theta)$, if $J(\theta_{LR}) > J(\theta_{SVM})$, logistic regression is not fully optimized

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Mini Quiz: Which of the following will fix a bad optimization objective? Which will fix a bad optimization algorithm?

- Run more iterations of gradient descent
- Try using Newton's method to optimize the function
- Use different values for hyperparameters
- Use a more expressive model, such as a neural network

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Mini Quiz: Which of the following will fix a bad optimization objective? Which will fix a bad optimization algorithm?

- Run more iterations of gradient descent - Optimization Algorithm
- Try using Newton's method to optimize the function
- Use different values for hyperparameters
- Use a more expressive model, such as a neural network

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Mini Quiz: Which of the following will fix a bad optimization objective? Which will fix a bad optimization algorithm?

- Run more iterations of gradient descent - Optimization Algorithm
- Try using Newton's method to optimize the function - Optimization Algorithm
- Use different values for hyperparameters
- Use a more expressive model, such as a neural network

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Mini Quiz: Which of the following will fix a bad optimization objective? Which will fix a bad optimization algorithm?

- Run more iterations of gradient descent - Optimization Algorithm
- Try using Newton's method to optimize the function - Optimization Algorithm
- Use different values for hyperparameters - Optimization Objective
- Use a more expressive model, such as a neural network

Common Issue 2 - Issues with Optimization Objective or Optimization Algorithm

Mini Quiz: Which of the following will fix a bad optimization objective? Which will fix a bad optimization algorithm?

- Run more iterations of gradient descent - Optimization Algorithm
- Try using Newton's method to optimize the function - Optimization Algorithm
- Use different values for hyperparameters - Optimization Objective
- Use a more expressive model, such as a neural network - Optimization Objective

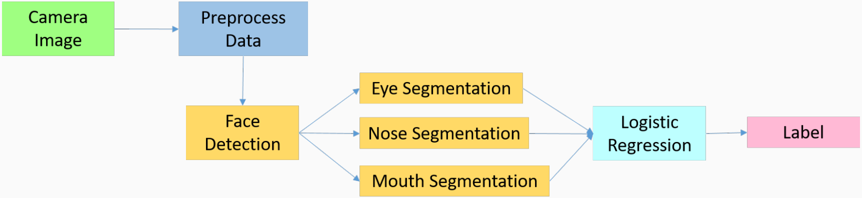
Applying ML Algorithms

Applying ML Algorithms

When using an ML algorithm:

1. Design various components of algorithm architecture
   - Benefit: Allows for more scalable algorithm
   - Issue: Hard to predict design for each component and understand what hardest components are
2. Try to come up with a quick implementation and then optimize
   - Benefit: Often application will work more quickly - time is spent only on components that are broken

Error Analysis Example: Face Recognition



Error Analysis Example: Face Recognition

- Plug in true values as input to each component and see how each component affects accuracy

Component	Accuracy
Overall System	85%
Preprocess Data	85.1%
Eye Segmentation	95%
Nose Segmentation	96%
Mouth Segmentation	97%
Logistic Regression	100%

- Most room for improvement in eye segmentation

Questions

Questions?