Support Vector Machines

PSC 8185: Machine Learning for Social Science

Iris Malone

March 21, 2022

Materials adapted from Sergio Ballacado

Announcements

- Problem Set 5 Released: Due April 4
- (Virtual) Poster Session April 27
- Final Project Due May 3

Recap

Where We've Been:

- Non-parametric models 'black box' functional form
- Boosting and BART improves over CART, bagging, and RF by sequentially growing trees
- Bayesian models incorporate learning to boost model performance

Recap

Where We've Been:

- Non-parametric models 'black box' functional form
- Boosting and BART improves over CART, bagging, and RF by sequentially growing trees
- Bayesian models incorporate learning to boost model performance

New Terminology:

- Random Forest
- Boosting
- Learning Rate
- Interaction Depth

Agenda

1. Maximal Margin Classifier
2. Support Vector Classifier
3. Support Vector Machines

Maximal Margin Classifier

Recall: KNN Classification

KNN is a classification method which assigns outcomes based on nearest observations

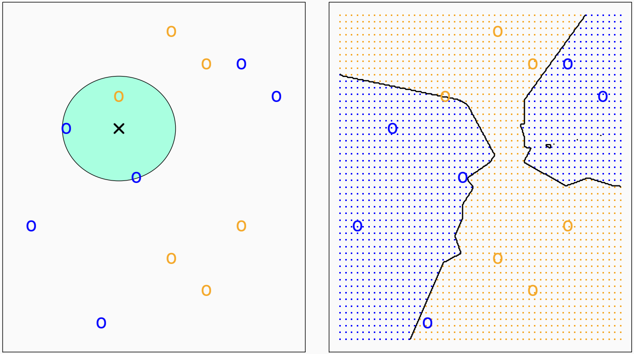
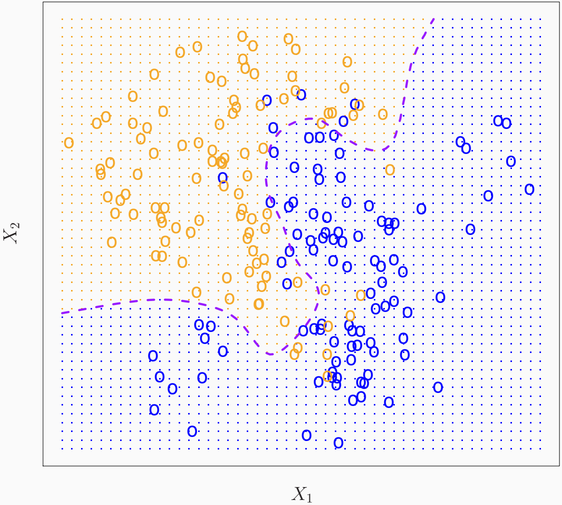


Figure 1: For example, predict input data (x) a color (orange or purple) based on $K = 3$ nearest neighbor colors

KNN has a decision boundary

Bayes Decision Boundary (dashed line) travels through points where probability of belonging to either class is 50%.



Bayesian Decision Boundary

Bayesian decision boundary is a type of **hyperplane**

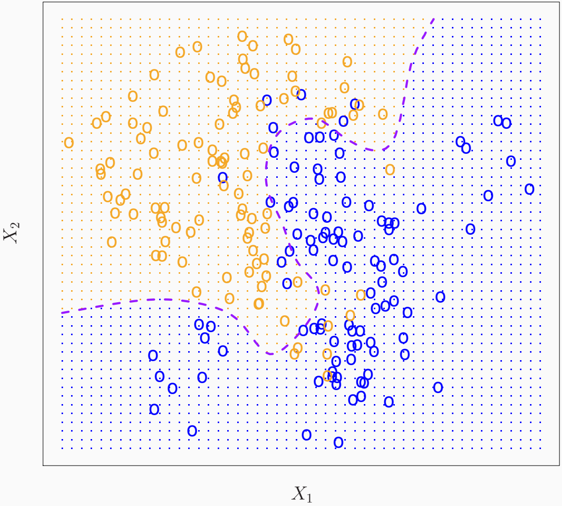


Figure 3: Figure 2.13

Hyperplanes in Nonparametric Modeling

Main Idea: Hyperplanes are a general class of non-parametric classification methods.

Hyperplanes in Nonparametric Modeling

Main Idea: Hyperplanes are a general class of non-parametric classification methods.

- Given a p-dimensional space of predictors, we can draw a **hyperplane** or flat affine space which separates the space into p-1 regions.
- Draw this hyperplane such that it classifies/demarcates between different observations

Hyperplanes in Nonparametric Modeling

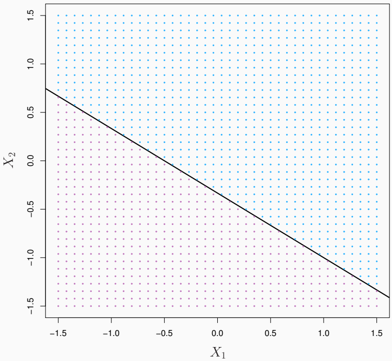
Main Idea: Hyperplanes are a general class of non-parametric classification methods.

- Given a p-dimensional space of predictors, we can draw a **hyperplane** or flat affine space which separates the space into p-1 regions.
- Draw this hyperplane such that it classifies/demarcates between different observations
- Example: 2-dimensional space (2 predictors)
    - Hyperplane is one-dimensional space (a line)
    - Defined by equation $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$
    - Means that for any $X = (X_1, X_2)^T$ there is a point on the hyperplane

Two-Dimensional Hyperplane

Example of Hyperplane: $1 + 2X_1 + 3X_2 = 0$
Blue region is set of points (X_1, X_2) for which $1 + 2X_1 + 3X_2 > 0$
Purple region is set of points (X_1, X_2) for which $1 + 2X_1 + 3X_2 < 0$

Figure 4: Practical example: predict how exercise (X_1) and vegetable intake (X_2) affect health level



Multi-Dimensional Hyperplane

The hyperplane for p-dimensional space is the solution to the equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p = 0$$

Multi-Dimensional Hyperplane

The hyperplane for p-dimensional space is the solution to the equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p = 0$$

Interpretation: For any $X = (X_1, X_2, \ldots, X_p)^T$ there is a point on the hyperplane with equal probability of being assigned to either class. e.g. average exercise and average veg intake $\approx$ border-line healthy

Multi-Dimensional Hyperplane

If there is no defined solution, then hyperplane is function solved by:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots \beta_p X_p < 0$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots \beta_p X_p > 0$$

Multi-Dimensional Hyperplane

If there is no defined solution, then hyperplane is function solved by:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p < 0$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p > 0$$

Interpretation: Tells us that $X = (X_1, X_2, \ldots, X_p)^T$ lies on one side of the hyperplane or the other
(e.g. Low Exercise/Low Veg $<$ border-line healthy)

Practical Example of Hyperplane

Recall: Teach model to predict images of cats versus dogs starting with 1 predictor

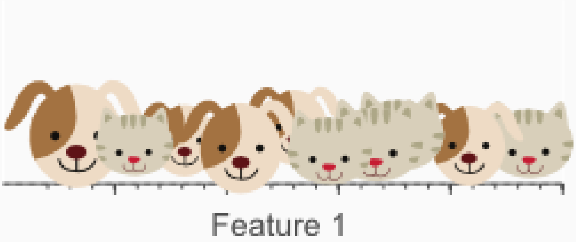


Figure 6: A single feature (5 bins) does not result in a good separation of our training data (5 cats, 5 dogs).

Source: Vision Dummy (2014)

Practical Example of Hyperplane

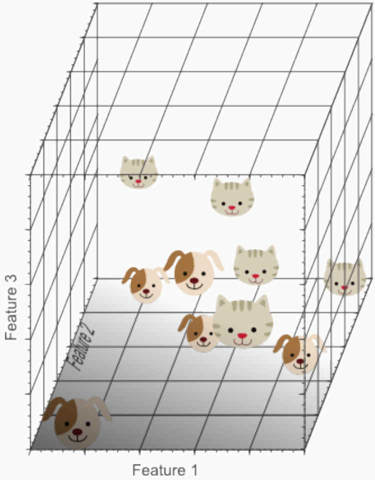


Figure 7: Adding a third feature ($5x5x5 = 125$ bins) results in a linearly separable classification problem in our example. A plane exists that perfectly separates dogs from cats.

Practice Example of Hyperplane

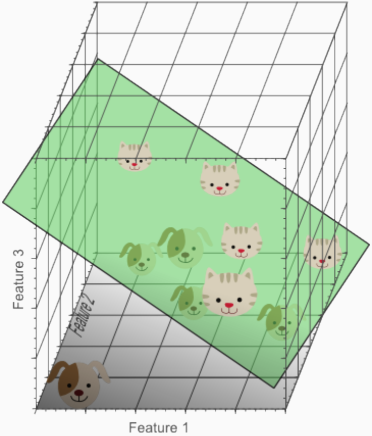


Figure 8: Multi-dimensional space permits perfect separation of classes.

Problem: Perfect Separation of Classes

Suppose we have a classification problem with response $Y = -1$ or $Y = 1$ and functional form:

$$Y = f(X) = \beta_0 + \beta_X + \epsilon \quad (1)$$

If the error ϵ is small enough, then the classes could be perfectly separable.

Problem: Perfect Separation of Classes

If the classes are perfectly separable, then there is an *infinite* number of hyperplanes we could draw.

Problem: Perfect Separation of Classes

If the classes are perfectly separable, then there is an *infinite* number of hyperplanes we could draw.

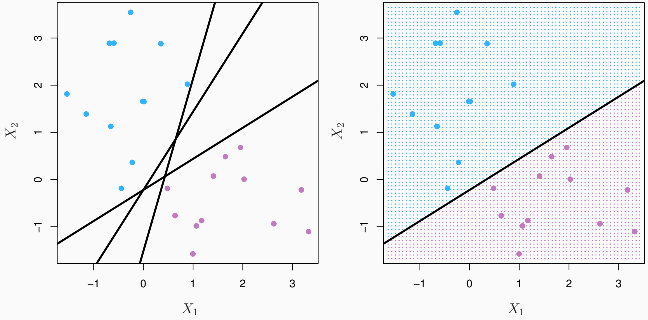


Figure 9: Figure 9.2

Problem: Which hyperplane does the model choose to classify observations?

Maximal Margin Classifier

Problem: Which hyperplane does the model choose to classify observations?

Maximal Margin Classifier

Problem: Which hyperplane does the model choose to classify observations?

Given an infinite number of hyperplanes, pick the **maximal margin classifier** (also known as the maximal margin hyperplane or the optimal separating hyperplane)

Maximal Margin Classifier

Main Idea: The **maximal margin classifier** is the hyperplane with the widest margin M between the two classes

- Draw the largest possible empty margin around each hyperplane
- Out of all possible hyperplanes that separate the 2 classes, pick the one with the <u>widest</u> margin M

Maximal Margin Classifier Loss Function

Solve the optimization problem to find largest margin hyperplane:

$$max_{\beta_0, \beta_1, \dots, \beta_p} M \qquad \text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1$$

$$\mathsf{y}_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M \qquad \forall i = 1, \dots, n$$

$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip})$ tells us how far x_i is from the hyperplane

Interpretation: M is simply the width of the margin in either direction

Maximal Margin Hyperplane

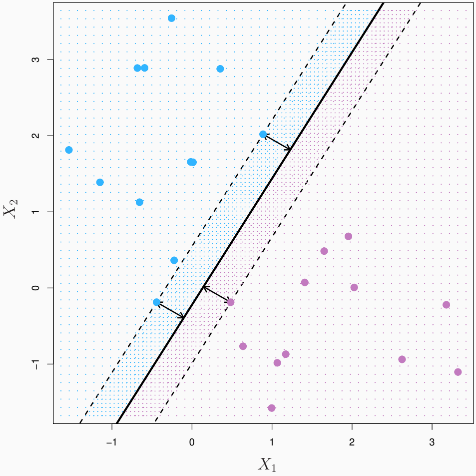


Figure 10: Maximal margin hyperplane shown as solid line. Margin is distance from the solid line to the dashed lines. 2 Blue points and purple point are the support vector. Purple and blue dash indicate decision rule made by classifier based on hyperplane.

Support Vectors are Points Nearest Hyperplane

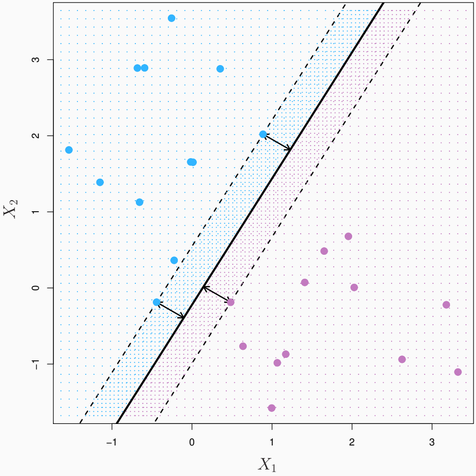


Figure 11: 2 Blue points and purple point are the support vector. Purple and blue dash indicate decision rule made by classifier based on hyperplane.

Support Vectors

Main Idea: **Support vectors** are vectors in p-dimensional space near the maximal margin hyperplane.

Support Vectors

Main Idea: **Support vectors** are vectors in p-dimensional space near the maximal margin hyperplane.

- They “support” the position of the hyperplane
- If these points moved slightly, hyperplane would change as well

Support vectors are important because the hyperplane only depends on the support vectors, not the other observations.

Limits to Maximal Margin Classifier

Problem 1: Perfect separation can lead to overfitting and sensitivity to individual data points → inconsistent hyperplanes.

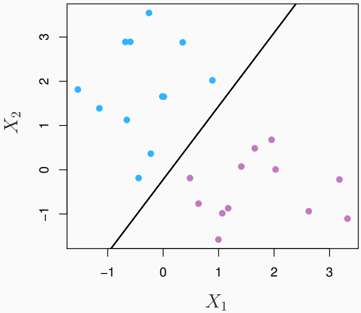


Figure 12: Figure 9.5: Addition of blue dot dramatically shifts hyperplane

Practice Example of Perfect Separation Problem

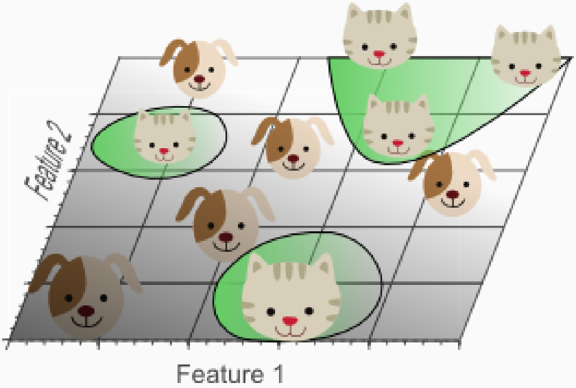
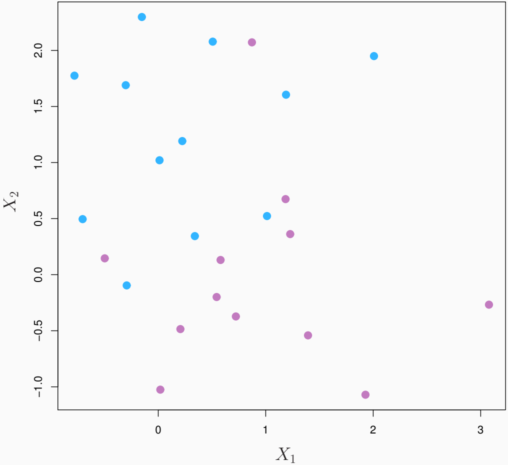


Figure 13: Using too many features results in overfitting due to sparsity of data (10 pets $\leftrightarrow$ 125 bins) The classifier starts learning exceptions that are specific to the training data and do not generalize well when new data is encountered.

Limits to Maximal Margin Classifier

Problem 2: It is not always possible the separate the points using a hyperplane creating a **non-separable case**



Support Vector Classifier

Support Vector Classifiers

Main Idea: Support vector classifiers resolve **non-separable cases**

Support Vector Classifiers

Main Idea: Support vector classifiers resolve **non-separable cases**

- Relaxation of the maximal margin classifier

Support Vector Classifiers

Main Idea: Support vector classifiers resolve **non-separable cases**

- Relaxation of the maximal margin classifier
- Allows a number of points to be on the wrong side of the margin or even the hyperplane

Support Vector Classifiers

Main Idea: Support vector classifiers resolve **non-separable cases**

- Relaxation of the maximal margin classifier
- Allows a number of points to be on the wrong side of the margin or even the hyperplane
- If the hyperplane accepts *some* misclassifications, then it is a **soft margin classifier**

Example of Soft Margin (Support Vector) Classifier

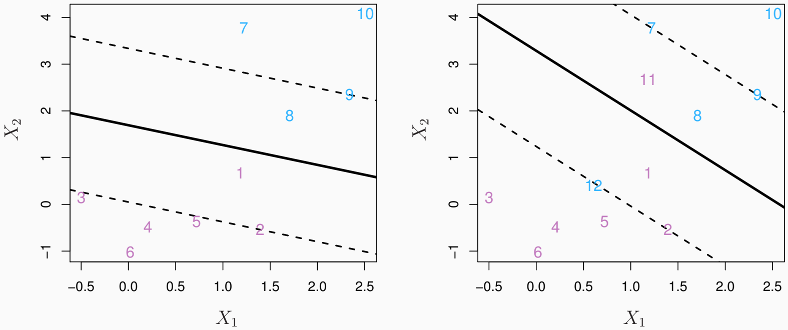


Figure 15: Example of perfect separableand non-separable case. In soft margin case, observations 11 and 12 are on wrong side of the margin

Support Vector Classifier Loss Function

Solve the optimization problem to identify largest margin hyperplane given some number of allowed misclassifications:

$$max_{\beta_0, \beta_1, \ldots, \beta_p, \epsilon} M \qquad \text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1$$

$$\text{y}_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \qquad \forall i = 1, \ldots, n$$

Support Vector Classifier Loss Function

Solve the optimization problem to identify largest margin hyperplane given some number of allowed misclassifications:

$$max_{\beta_0,\beta_1,\ldots,\beta_p,\epsilon} M \qquad \text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1$$

$$\text{y}_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \qquad \forall i = 1, \ldots, n$$

New Equation: Determines Number of Allowable Misclassifications

$$\epsilon_i \geq 0 \qquad \forall i = 1, \ldots, n$$

subject to $\sum_{i=1}^{n} \epsilon_i \leq C$

Key Hyperparameters

- M is the width of the margin in either direction

Key Hyperparameters

- M is the width of the margin in either direction
- $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ are **slack variables**.
  - Allows observations to be on wrong side of the margin
  - If $\epsilon_i > 1$ then it is on the wrong side of the margin

Key Hyperparameters

- M is the width of the margin in either direction
- $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ are **slack variables**.
  - Allows observations to be on wrong side of the margin
  - If $\epsilon_i > 1$ then it is on the wrong side of the margin
- C is called **the budget**
  - “budget” of acceptable violations
  - Determines the number and severity of the violation
  - $\sum_{i=1}^{n} \epsilon_i \leq C$

Interpretation of Budget Parameter C

- Size of C:
  - If $C = 0$ then there is no budget for violation. All $\epsilon = 0 \rightarrow$ maximal margin classifer
  - If $C > 0$, no more than C observations can be on the wrong side of the hyperplace

Interpretation of Budget Parameter C

- Size of C:
  - If $C = 0$ then there is no budget for violation. All $\epsilon = 0 \rightarrow$ maximal margin classifer
  - If $C > 0$, no more than C observations can be on the wrong side of the hyperplace
- As C increases...
  - Tolerance for misclassifications increases
  - Margin widens

Tuning the Budget C

- C controls the bias-variance tradeoff:
  - If C is small → narrow margins ~ low bias, but high variance
  - If C is large → large margins ~ high bias, but low variance
- Tune via cross-validation

Tuning the Budget C (High To Low)

As C increases, the tolerance for misclassified observations increases

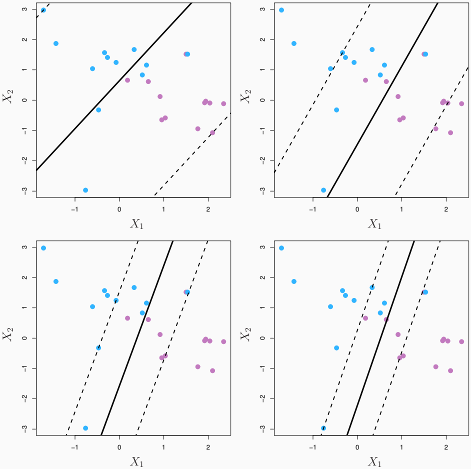


Figure 16: Largest value of C in top-left; smallest value of C in bottom-right.

Budget C Affects Number of Support Vectors

Recall: The loss function means that only observations that either lie on the margin or that violate the margin affect the hyperplane.

- Large C → large number of support vectors
- Small C → small number of support vectors

Budget C Affects Number of Support Vectors

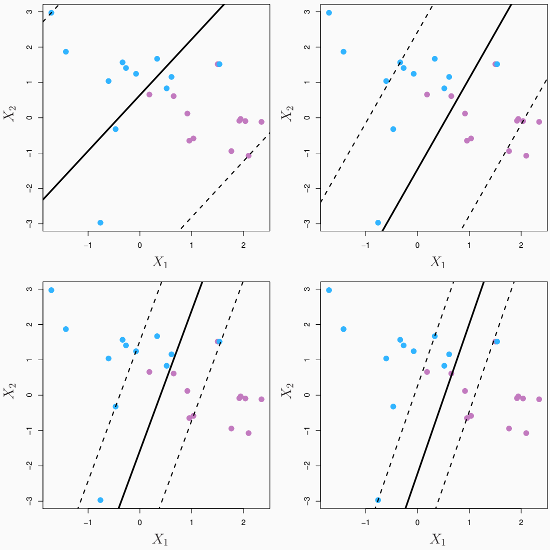


Figure 17: Large C → More Support Vectors

Advantages and Disadvantages to Support Vector Classifiers

Advantages:

Disadvantages:

Advantages and Disadvantages to Support Vector Classifiers

Advantages:

- Handles non-separated data
- Great robustness to individual observations
- Better overall classification of most training observations

Disadvantages:

Advantages and Disadvantages to Support Vector Classifiers

Advantages:

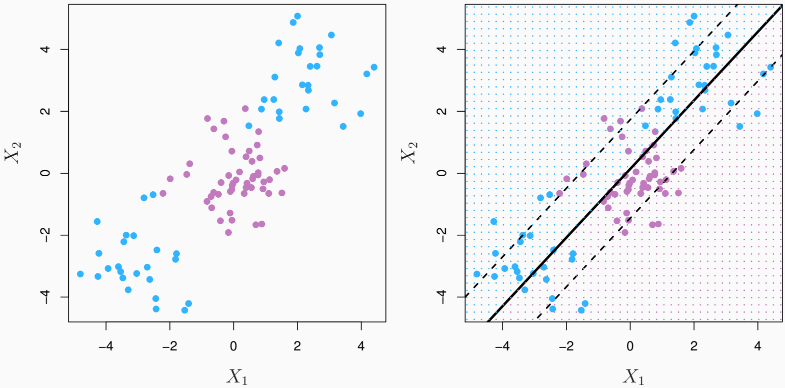
- Handles non-separated data
- Great robustness to individual observations
- Better overall classification of most training observations

Disadvantages:

- Can't handle non-linear boundaries

Non-Linear Boundaries Don't Work

Problem: Support vector classifier seeks a linear boundary, but can't find one.



Solution: More flexible support vector models

Recall: Non-Linear Model Solutions

1. Transform the explanatory variable
2. More flexible regressions
    - Polynomial function
    - Stepwise function (Piecewise Function)
3. Semi-parametric Models
4. Non-parametric models

Solutions to Non-Linear Boundaries

We can apply two similar solutions to hyperplane models …

- **Polynomial support vector classifiers**: Add higher order polynomials (polynomial support vector classifiers)
- **Support Vector Machines**: Add more flexible model functions

Polynomial Support Vector Classifiers

Main Idea: Instead of fitting $X_1, X_2, \ldots, X_p$, fit support vector classifier using $2p$ features: $X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2$

Polynomial Support Vector Classifiers

Main Idea: Instead of fitting $X_1, X_2, \dots, X_p$, fit support vector classifier using $2p$ features: $X_1, X_1^2, X_2, X_2^2, \dots, X_p, X_p^2$

Polynomial Loss Function:

$$max_{\beta_0, \beta_1, \dots, \beta_p, \epsilon} M \qquad \text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1$$

$$y_i(\beta_0 + \textstyle\sum_{j=1}^{p} \beta_{j1} x_{ij}) + \cdots + \textstyle\sum_{j=1}^{p} \beta_{jp} x_{ij} \geq M(1 - \epsilon_i) \qquad \forall i = 1, \dots, n$$

$$\sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1$$

Support Vector Machines

Support Vector Machines

Main Idea: SVM provide a more flexible approach for non-separable and non-linear data.

Main Idea: SVM provide a more flexible approach for non-separable and non-linear data.

- Enlarge the feature space to accommodate non-linearities using **kernels**
- Kernels quantify the similarity of two observations

$$K(x_i, x_i') = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

- Common Types of Kernels:
    - **Linear Kernel**
    - **Polynomial Kernel**
    - **Radial Kernel**

Linear Kernels

- **Linear Kernel:** Classifier is linear in features

$$K(x_i, x_i') = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

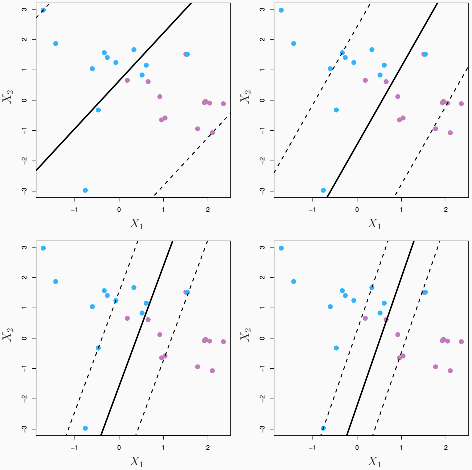
- Solution to inner product for support vectors defines margin
- $\langle x, x_i \rangle$ is the inner product of two observations

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$

- Linear kernel quantifies similarity of pair of observations (correlation)

Relationship to Support Vector Classifiers

Linear Kernel is regular support vector classifier



Polynomial Kernel

Polynomial Kernel: Adding degree d to kernel creates more flexible decision boundary

$$K(x_i, x_i') = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$

Classifier Function:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle$$

Radial Kernel

Radial Kernel:

- Focus on local observations to draw decision boundary
- Only nearby observations affect classification

$$K(x_i, x_i') = exp(-\gamma + \sum_{j=1}^{p}(x_{ij} - x_{i'j}))$$

- Decision boundary
  - Drawn differently than KNN Bayesian decision boundary
  - Drawn based on **Euclidean distance** between observations

Euclidean Distance

- Euclidean Distance:

$$d\left(x_{i'j}, x_{ij}\right) = \sqrt{\sum_{i=1}^{n} \left(x_{ij} - x_{i'j}\right)^2}$$

- If a given test observation $x^* = (x_1^*, \ldots x_p^*)$ is far from training observations x_i, then $\sum_{j=1}^{p}(x_{ij} - x_{i'j})$ is very small
- When distance is small, training observations have very little influence on test observation

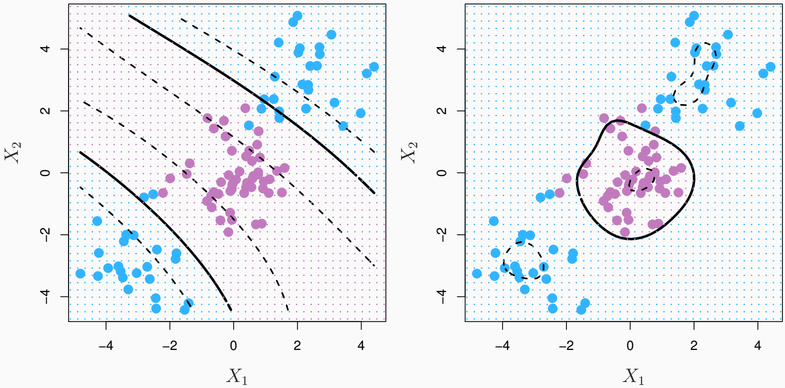
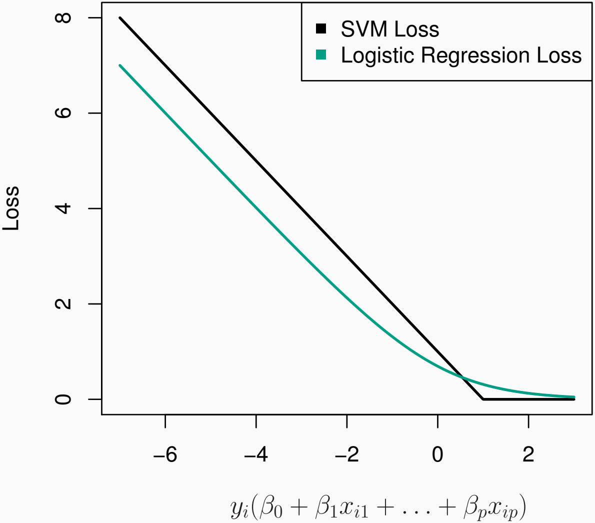


Figure 18: Left: SVM with polynomial kernel degree 3; Right: SVM with radial kernel.

Kernels vs Classifiers

- Polynomial/radial kernels faster than polynomial classifiers
- Kernels provide more flexibility in boundary space
- Kernels can look at global or local boundaries

Comparison to Logit and LDA



Comparison to Logit and LDA

Example: Predict heart disease using information about 13 possible characteristics

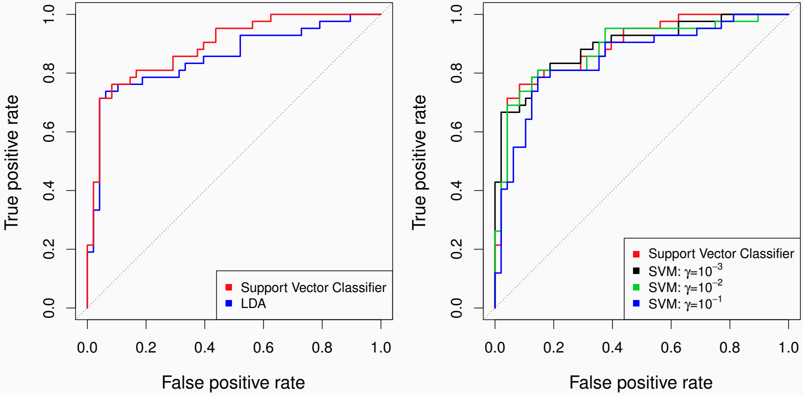


Figure 20: SVM performs better than LDA on Heart Disease test data

Advantages and Disadvantages to SVM

Advantages:

Disadvantages:

Advantages and Disadvantages to SVM

Advantages:

- Works for separable and non-separable data
- Works for linear and non-linear data
- Performs better than LDA
- Performs better than parameteric approaches if true f unknown
- Performs better than logit when classes well-separated

Disadvantages:

Advantages and Disadvantages to SVM

Advantages:

- Works for separable and non-separable data
- Works for linear and non-linear data
- Performs better than LDA
- Performs better than parameteric approaches if true f unknown
- Performs better than logit when classes well-separated

Disadvantages:

- Poor performance with multi-class outcomes
- Performs worse than logit with overlapping classes
- Not as popular/fast as logit

Conclusion

- Hyperplanes are type of decision boundary for classification problems
- Maximal margin classifers discriminate between perfectly separated data
- Support vector classifiers allow for misclassifications when data non-separable
- SVM provide highly flexible and well-performing approach