Support Vector Machines

Malte Schierholz

University of Mannheim, MZES
Institute for Employment Research (IAB)

Malte.Schierholz@iab.de

March 21 and 22, 2018

Introduction

- Very successful in many prediction problems
- Essential tool in the machine learning toolbox
- Complex mathematical theory involved
  - Statistical learning theory
  - Function fitting in *reproducing kernel Hilbert spaces*

More intuitive (and historical) approach taken in the following:

- Build classification model that predicts binary outcome $y \in \{-1, 1\}$
- Extensions exist for regression, ranking, anomaly detection, ... (not discussed here)

How would you build a model for binary prediction?

Why not logistic regression?

Classical approach to predict binary y:

1. Estimate probability model
   $\log(\frac{Pr(y=+1|x)}{Pr(y=-1|x)}) = f_{\text{logit}}(x) = \hat{\beta}_0 + x'\hat{\beta}$ (logistic regression)
2. Predict $y = 1$ if $Pr(y = +1|x) > 0.5$

Issues with logistic regression

- Logistic regression fails $((\beta_0, \beta) \to \infty)$ if data are linearly separable
- If our goal is binary prediction, why use obscure probabilities?

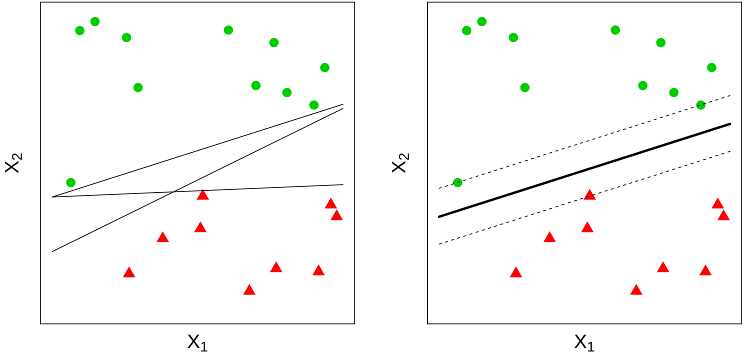
More direct SVM alternative:

1. Find function that is optimized for prediction of categories
   $f_{\text{SVM}}(x) = \hat{\beta}_0 + x'\hat{\beta}$ (Support Vector Classifier)
2. Predict $y = 1$ if $f_{\text{SVM}}(x) > 0$

Separating Hyperplanes

1. Separating Hyperplanes
2. Variable Transformations and the Kernel Trick

Optimal separating hyperplanes



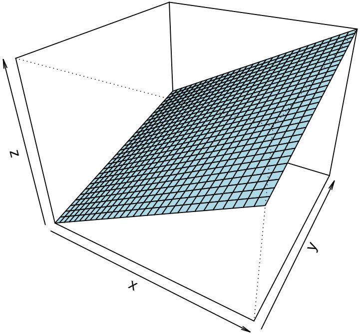
Key idea:

- Find *hyperplane* (= a linear decision boundary)
- that maximizes the *margin* (= distance between hyperplane and its closest points)

Three *support vectors* exist in 2-dimensional space

(developed by Vapnik and Lerner, 1963)

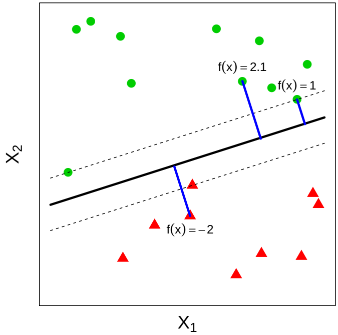
Higher-dimensional spaces



A (hyper-)plane in 3-dimensional space

- If the data have more variables than observations ($p > n$), linear decision boundaries typically exist

Intuition about $f(x)$

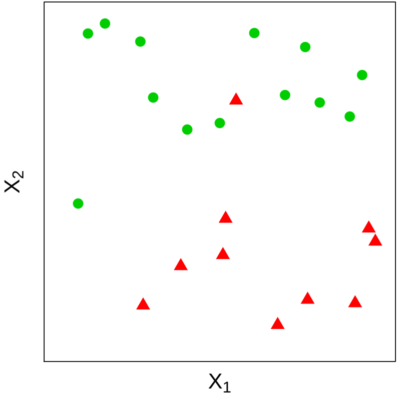


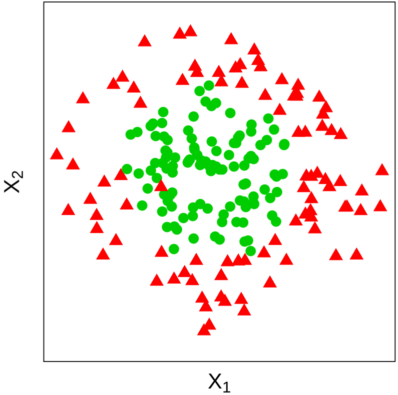
$f(x)$ measures the scaled distance between the hyperplane and point x such that

- $f(x) = 0$ if x is on the hyperplane
- $f(x) = +/-1$ if x is on the margin

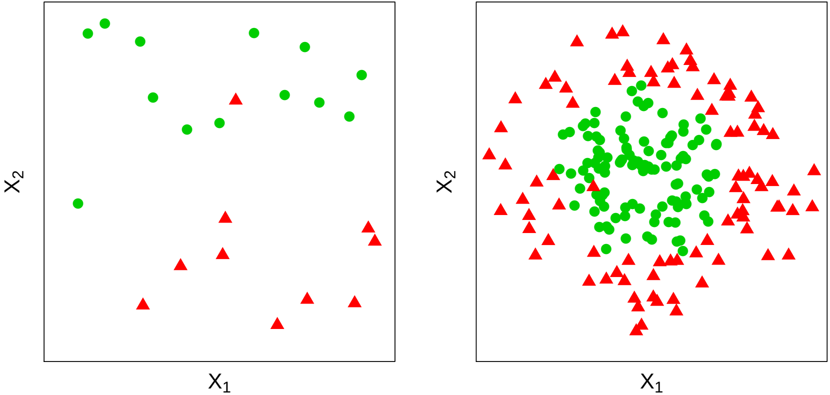
- Predict $+1$ if $f(x)$ positive
- Predict -1 if $f(x)$ negative

Two issues with optimal separating hyperplanes





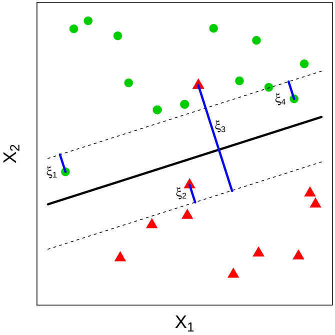
Two issues with optimal separating hyperplanes



Optimal separating hyperplanes often do not exist. Solutions:

- Allow for misclassification in the presence of noise?
- Allow for non-linear decision boundaries?

Soft-Margin Support Vector Classifier



Dealing with noise:

- Allow for margin violations and missclassification
- New constraint: $\sum \xi_i < C$ must not exceed budget C
- Tuning parameter C controls margin width and overfitting

(developed by Cortes and Vapnik, 1995)

Relationship to Logistic Regression

Support Vector Classifier minimizes

$$\min_{\beta_0,\beta}\{\sum_{i=1}^{n}\underbrace{\max(0,1-y_i(\overbrace{\beta_0}^{\text{sometimes left out}}+x_i'\beta))}_{\text{Hinge Loss}}+\lambda(C)\cdot\underbrace{\sum_{j=1}^{p}}_{\text{Penalty}}\beta_j^2\} \quad (1)$$

Compare to logistic regression with ridge penalty, which minimizes

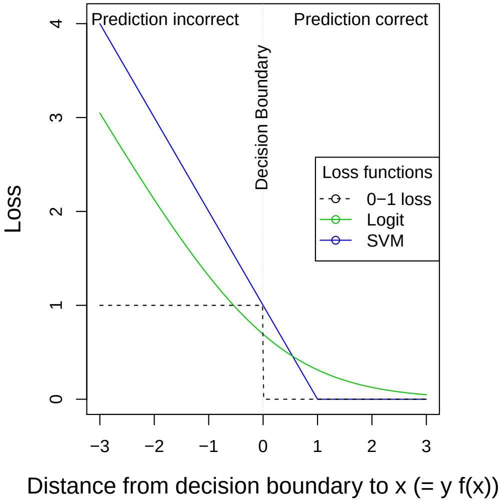
$$\min_{\beta_0,\beta}\{\sum_{i=1}^{n}\underbrace{\log(1+e^{-y_i(\beta_0+x_i'\beta)})}_{\text{Binomial Loss}}+\lambda\cdot\underbrace{\sum_{j=1}^{p}}_{\text{Penalty}}\beta_j^2\} \quad (2)$$

Only the loss functions are different!

Notation requires $y_i \in \{-1, 1\}$

Relationship to Logistic Regression

- Similar loss functions $\rightarrow$ similar results
- Hinge loss (SVMs) mimics 0-1 loss
- Hinge loss often preferred if data are separated
- Binomial loss often preferred if classes overlap



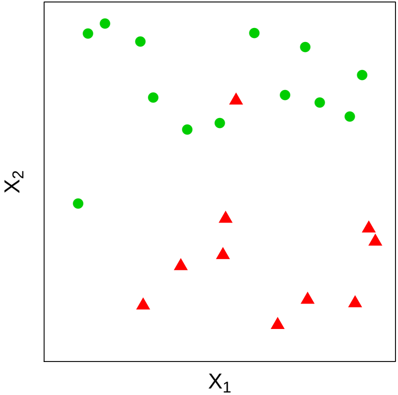
Variable Transformations and the Kernel Trick

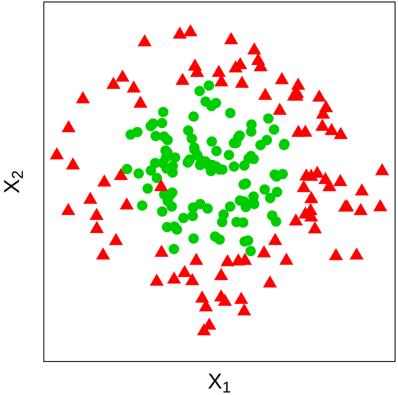
1. Separating Hyperplanes

2. Variable Transformations and the Kernel Trick

From the Support Vector Classifier to Support Vector Machines

Two issues with optimal separating hyperplanes



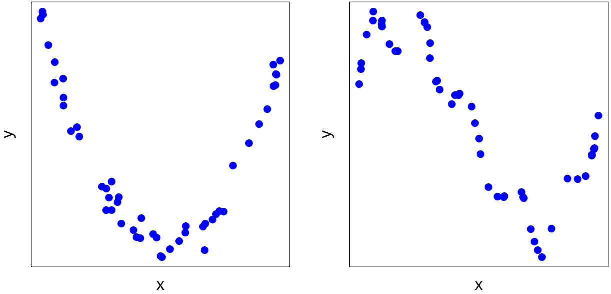


Optimal separating hyperplanes often do not exist. Solutions:

- Solved: Allow for misclassification in the presence of noise
- Now: Allow for non-linear decision boundaries?

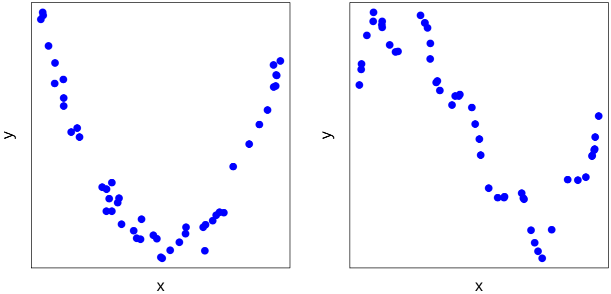
Variable transformations

Same problem as in linear regression with continuous outcome y.



How can we do regression modeling in these situations?

Variable transformations



Transform p-dim. input space $X = (x_1, ..., x_p)$ into Q-dim. feature space

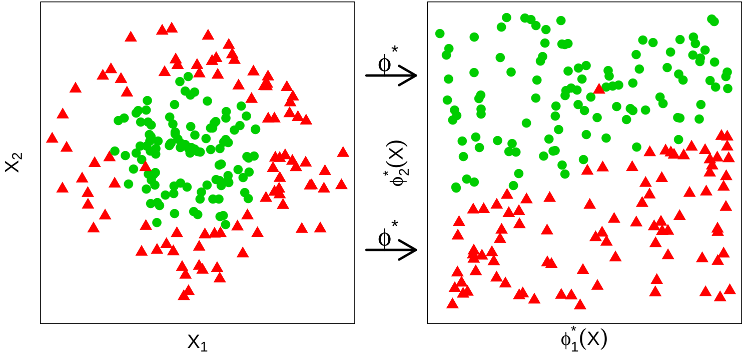
$$\phi(X) = (\phi_1(X), \phi_2(X), \phi_3(X), ..., \phi_Q(X))$$

ϕ can have many possible forms, for example:

$$\phi_j(X) = x_j \qquad \phi_j(X) = \log x_k \qquad \phi_j(X) = \sqrt{x_k}$$
$$\phi_j(X) = x_k^2 \qquad \phi_j(X) = x_k \cdot x_l \qquad \phi_j(X) = I(L_m \leq x_k \leq U_m)$$

(see Hastie et al. (2009), Chapter 5, for an overview on variable transformations)

Variable transformations



- The perfect transformation ϕ^* achieves linear separability in the transformed feature space (see example)
- Problem: ϕ^* is unknown and depends on geometric considerations
- How to find a good transformation ϕ?

The Kernel Trick

The function $f(x)$ has two equivalent representations

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{p=1}^{P} x_p \hat{\beta}_p \tag{3}$$

$$= \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i y_i \langle \phi(x), \phi(x_i) \rangle \tag{4}$$

The second line suggests to calculate $\langle \phi(x), \phi(x_i) \rangle$ separately,

$$K(x, x_i) := \langle \phi(x), \phi(x_i) \rangle = \sum_{q=1}^{Q} \phi_q(x) \phi_q(x_i) \tag{5}$$

x enters only through $K(\cdot, x)$

$\rightarrow$ No need to specify transformation ϕ if one knows kernel K!!

Popular Kernels

$$K_d(x, x') = (1 + \sum_{p=1}^{P} x_p x'_p)^d \qquad (d\text{th degree polynomial})$$
$$K_\gamma(x, x') = \exp(-\gamma \sum_{p=1}^{P} (x_p - x'_p)^2) \qquad (\text{radial basis, distance based!})$$
$$K_\kappa(x, x') = \tanh(\kappa_1 + \kappa_2 \sum_{p=1}^{P} x_p x'_p) \qquad (\text{neural network})$$

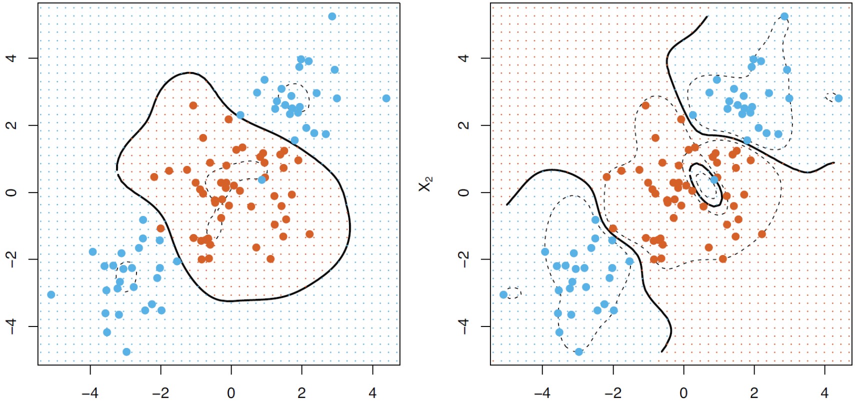
For example, for 2 predictors (x_1, x_2), mapped into a 6-dimensional feature space with ϕ,

$$\phi_1(x_1, x_2) = 1 \qquad \phi_4(x_1, x_2) = x_1^2$$
$$\phi_2(x_1, x_2) = \sqrt{2}x_1 \qquad \phi_5(x_1, x_2) = x_2^2$$
$$\phi_3(x_1, x_2) = \sqrt{2}x_2 \qquad \phi_6(x_1, x_2) = \sqrt{2}x_1 x_2$$

the inner product and the 2-degree polynomial kernel are identical,

$$\langle \phi(x_1, x_2), \phi(x'_1, x'_2) \rangle = K_2((x_1, x_2), (x'_1, x'_2)) \tag{6}$$

Illustration



- SVM classifier with radial kernel (most popular choice)
- Tuning parameters C differ $\rightarrow$ overfitting in the right panel

(Source: Efron and Hastie, 2016, p. 383)

Kernel Summary

Things to know:

- Think about kernels as a similarity measure between points x and x'
- Kernels are useful beyond binary outcomes and SVMs
  - Requires a model with linear term $x'\beta$ and ridge penalty $\sum \beta_j^2$

Key advantages of kernel methods:

- Alternative way to specify variable transformations
- May speed up computation
  - Matters if we have 1,000s or 1,000,000s of predictors
- Input objects x can be different than numbers (e.g. text)
  - Requires a definition of similarity between objects

Summary

- Binary classification is approached with geometric arguments only and without reference to probability models
- Kernels can be used for a wide range of prediction problems (not only binary classification)
  - Provides an alternative to transform variables into a feature space
- Most successful if the prediction problem
  - has few observations relative to the number of input variables (e.g., genetics, engineering, document classification) and
  - all input variables are believed to be relevant for prediction (no variable selection)
- Application requires
  - appropriate preprocessing,
  - parameter tuning, and
  - (in the most difficult situations) the development and programming of new kernels

Software Resources

Resources for R

- Interface to libsvm: Package `e1071`
- Support for additional kernels: Package `kernlab`

Other

- `http://www.kernel-machines.org/software`

References

Efron, Bradley and Hastie, Trevor (2016)
Support-Vector Machines and Kernel Methods. In: *Computer Age Statistical Inference*. Cambridge University Press, p. 375–393

Hastie, Trevor, Tibshirani, Robert & Friedman, Jerome (2009)
The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. Chapter 9.

James, Gareth, Witten, Daniela, Hastie, Trevor & Tibshirani, Robert (2013)
An Introduction to Statistical Learning. Springer. Chapter 9.

Moguerza, Javier and Muñoz, Alberto (2006)
Support Vector Machines with Applications. *Statistical Science* **21**(3), p. 322–336.

Steinwart, Ingo and Christmann, Andreas (2008)
Support Vector Machines, Springer.