Linear Classifiers and Support Vector Machines

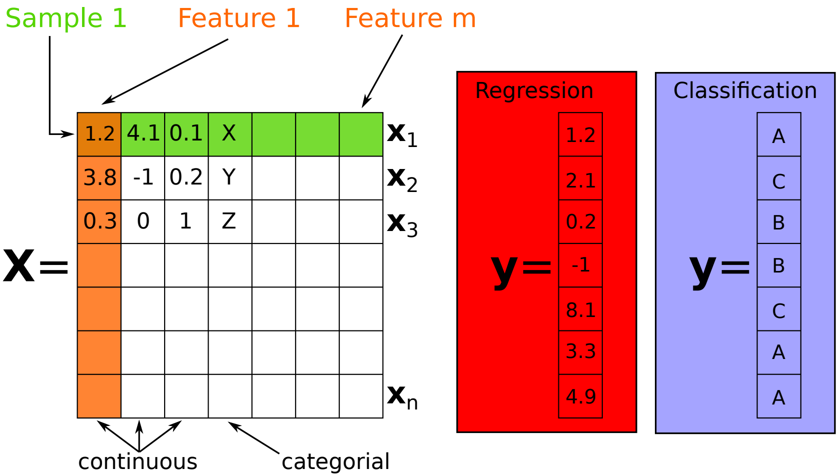
Sebastian Pölsterl

Computer Aided Medical Procedures | Technische Universität München

April 15, 2014

Outline

Definitions



Definitions

- A training **sample** $\mathbf{x}_i$ consists of m **features** $(x_{i1}, \ldots, x_{im})^T$ and is associated with **output** y_i.
- Each feature and the output can either be **continuous** (a number) or **discrete** (from a predefined set of values).
- If the output is continuous, we perform regression and if it is discrete, classification.
- The **training set** $\mathcal{T} = (\mathbf{x}_i, y_i)$ is comprised of n samples $(i = 1, \ldots, n)$.
- Let $\mathbf{X}$ indicate a matrix where the i-th row corresponds to the i-th sample and $\mathbf{y} = (y_1, \ldots, y_n)^T$ the vector of all outputs.

Problem Statement

Assumption

There is a function $f(\mathbf{X})$ that relates the features $x_{i1}, \ldots, x_{im}$ to the output y_i such that $\mathbf{y} = f(\mathbf{X})$ for $i \in \{1, \ldots, n\}$.

Goal

We seek to find a good approximation $\hat{f}(\mathbf{X})$ to the function $f(\mathbf{X})$.

1. Introduction

2. Linear Classifier
   1. Ordinary Least Squares
   2. Logistic Regression
   3. Optimal Separating Hyperplanes

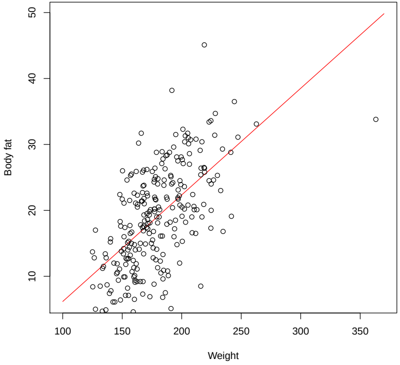
3. Support Vector Machines
   1. Linear SVMs
   2. Non-linear SVMs
   3. Multi-class SVMs

Linear Models

Definition (Linear Model)

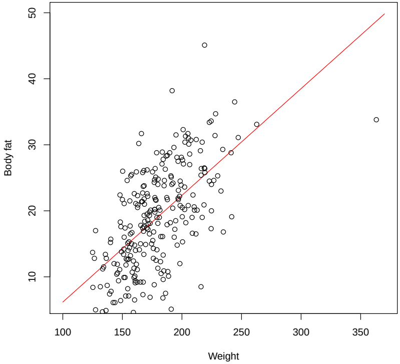
$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_m x_{im} + \varepsilon_i = \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

- The β parameters are **coefficients** or weights of the features.
- βs are to be **estimated** from the training data.
- The errors ε_i are independently and identically distributed (i.i.d.) with $\mathrm{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$.



Linear Models – Coefficients

- Each **feature** is associated with one **coefficient** β_j.
- In addition, the coefficient β_0 denotes the **intercept**.
- Estimates are denoted by a **hat**: $\hat{\beta}_j$ denotes the estimate of the coefficient of the j-th feature.
- In the example to the right $\beta_0 = -9.995$ (y-intercept) and $\beta_1 = 0.1617$ (slope; coefficient of *weight* feature).

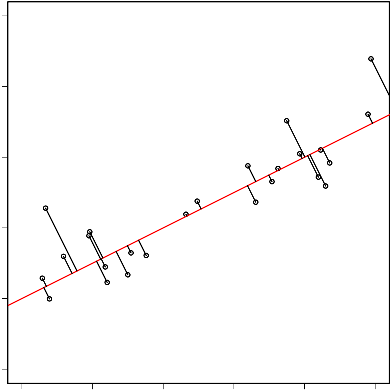


Linear Models – Loss Function

Definition (Estimated Function)

$$\hat{f}(x_1, \ldots, x_m) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_m x_m$$

- We need a way to assess how good the output $\hat{y}_i$ of our estimated model $\hat{f}(\mathbf{x}_i)$ fits the expected output y_i given the current estimates of the coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_m$.
- Hence, define a **loss function** $L(y_i, \hat{f}(\mathbf{x}_i))$.



Linear Models – Loss Function (Examples)

Squared error loss

$$(y_i - \hat{f}(\mathbf{x}_i))^2$$

Binomial Deviance

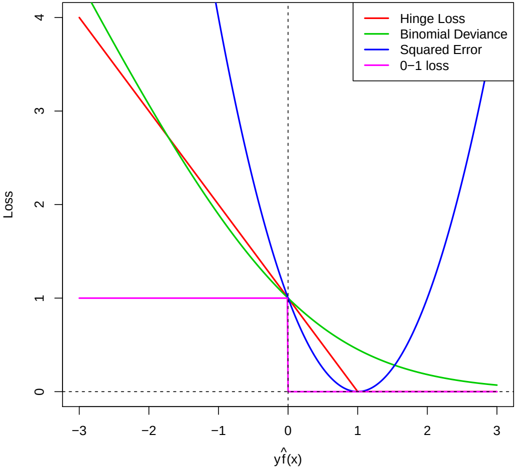
$$\log_2 \left(1 + e^{-y_i \hat{f}(\mathbf{x}_i)}\right)$$

Hinge loss

$$\max(0, 1 - y_i \cdot \hat{f}(\mathbf{x}_i))$$

0-1 loss

$$I(y_i \neq \hat{f}(\mathbf{x}_i))$$



Linear Models – Ordinary Least Squares Estimation

Definition (Residual Sum of Squares; RSS)

$$\mathrm{RSS}(\beta_0, \ldots, \beta_m) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$$

- RSS gives the total loss over the whole training set
- We want to choose the coefficients $\beta_0, \ldots, \beta_m$ such that the total loss according to RSS is **minimized**.
- **How can this be achieved?**

Linear Models – Ordinary Least Squares Estimation

- Set the partial derivative of RSS to zero

$$\frac{\partial \mathrm{RSS}(\beta_0, \boldsymbol{\beta})}{\partial \beta_j} = -2 \sum_{i=1}^{n} x_{ij}(y_i - \beta_0 - \mathbf{x}_i^T \boldsymbol{\beta})$$

- In matrix notation:

$$\mathrm{RSS}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \tag{1}$$

$$\frac{\partial \mathrm{RSS}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \tag{2}$$

- **Note**: $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_m)^T$ and the first column of $\mathbf{X}$ contains only 1 to accommodate the intercept β_0, i.e. $\mathbf{X}$ is a $n \times m + 1$ matrix.

Linear Models – Ordinary Least Squares Estimation

Definition (Ordinary Least Squares Estimate)

$$\hat{\beta} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

- The minimum of the loss function in unique.
- Estimates of the coefficients can be obtained in closed form and therefore no optimization is required.
- $\mathbf{X}$ must have full column rank $\Rightarrow$ $\mathbf{X}^T\mathbf{X}$ is positive definite.
- Prediction (regression) is performed by

$$\hat{f}(x_1, \ldots, x_m) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_m x_m$$

Classification

- Least squares is a linear model for **regression**, i.e. the outcome y_i is **quantitative**.
- We want a linear model for **classification**, i.e. the outcome y_i is **categorial**.
- **Example**: Classify pixels in an image according to the tissue they represent (e.g. fat, muscle, bone, lung).
- Categories are usually represented by coding them as numbers (fat $= 0$, muscle $= 1$, bone $= 2$, lung $= 3$).
- There is a third class where the outcome y_i is **ordered categorical** such as *small*, *medium*, *large* (not discussed here).

Logistic Regression

- Consider a binary classification problem where $y_i \in \{0, 1\}$.
- If $y_i = 1$, the i-th sample belongs to the **positive class**, otherwise to the **negative class**.
- Create a model of the probability of sample $\mathbf{x}_i$ belonging to the positive class

$$\pi_i = P(y_i = 1 | x_{i1}, \dots, x_{im})$$

- Remember that the linear model η_i is defined as

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_m x_{im}$$

- **How to connect the probability π_i to the linear predictor η_i?**

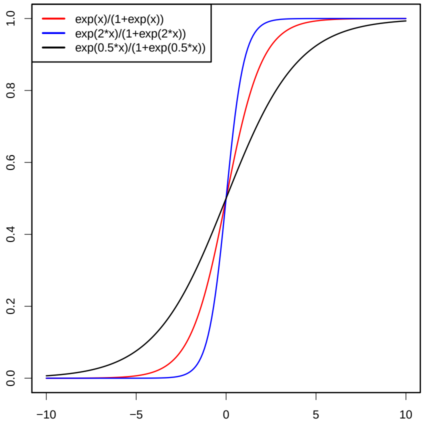
Logistic Regression – Response and link function

- Probability π_i is connected to the linear predictor by the **logistic function** $h(x)$

$$\pi_i = h(\eta_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

- The logistic function is called **response function** and its inverse – the logit function – **link function**

$$h^{-1}(x) = \log\left(\frac{x}{1-x}\right)$$



Logistic Regression – Log-Odds

- The model is linear with respect to the **log-odds**:

$$\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} \Leftrightarrow \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \log\frac{P(y_i = 1|\mathbf{x}_i)}{P(y_i = 0|\mathbf{x}_i)} = \eta_i$$

- Coefficients indicate by how much the odds change when the value of the corresponding feature is increased by 1

$$\frac{P(y_i = 1|x_{i1}, \ldots)}{P(y_i = 0|x_{i1}, \ldots)} \Big/ \frac{P(y_i = 1|x_{i1} + 1, \ldots)}{P(y_i = 0|x_{i1} + 1, \ldots)} = \exp(\beta_1)$$

Logistic Regression – Log-Odds Ratio

Definition (Log-Odds ratio)

The coefficient β_j represents the **log-odds ratio** of the *j*-th feature

- $\beta_j > 0 \Leftrightarrow$ Odds increase
- $\beta_j < 0 \Leftrightarrow$ Odds decrease
- $\beta_j = 0 \Leftrightarrow$ Odds remain unchanged
- This becomes very handy to assess which feature has the largest influence, especially if the goal is to predict which patients are diseased based on clinical features.

Logistic Regression – Example

Birth weight data contains data from 189 births to determine which of these factors were risk factors for low birth weight ($< 2.5\,\text{kg}$) [Hosmer and Lemeshow, 2000].

Feature	β / log-odds ratio	Chance
(Intercept)	0.924910	
Age	-0.042784	decreased
Mother's weight (pounds)	-0.015436	decreased
Race = White	0	
Race = Black	1.168452	increased
Race = Other	0.814620	increased
Previous premature labour	1.333970	increased
History of hypertension	1.740511	increased
Smoking during pregnancy	0.858332	increased

Logistic Regression – Maximum Likelihood Estimation

Definition (Likelihood function)

$$L(\beta_0, \boldsymbol{\beta}) = \prod_{i=1}^{n} P(y_i | \mathbf{x}_i) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1 - y_i}$$

Definition (Log-Likelihood function)

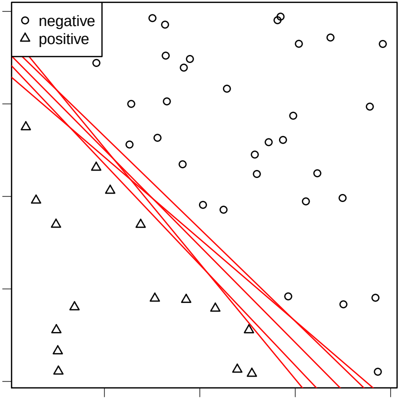
$$l(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i)$$

Definition (Maximum Likelihood Estimate; MLE)

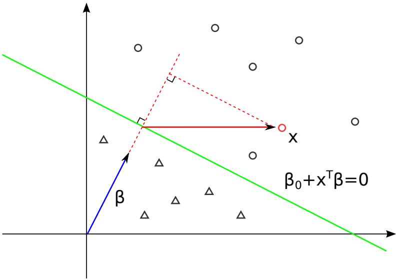
$$\hat{\boldsymbol{\beta}} = \arg\max_{\beta_0, \boldsymbol{\beta}} l(\beta_0, \boldsymbol{\beta})$$

Optimal Separating Hyperplanes

- Consider a binary classification problem where two classes are optimally separable.
- A lot of hyperplanes solve this problem but which one is the best?
- **Intuition**: the **margin** separating both classes has to be **maximized**.



Geometric Margin

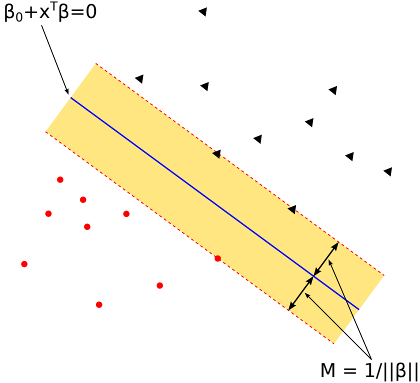


- The linear hyperplane is given by $f(\mathbf{x}) = \beta_0 + \mathbf{x}^T\beta = 0$.
- For any two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^m$ lying on the hyperplane, $(\mathbf{x}_1 - \mathbf{x}_2)^T\beta = 0$ and therefore β is orthogonal to the hyperplane.
- The signed distance of a point $\mathbf{x}_i$ to the hyperplane is given by

$$\frac{\beta_0 + \mathbf{x}_i^T\beta}{\sqrt{\beta^T\beta}} = \frac{\beta_0 + \mathbf{x}_i^T\beta}{\|\beta\|}$$

Optimal Separating Hyperplanes

- The goal is to find a hyperplane that separates the two classes and maximises the distance to the closest point from either class

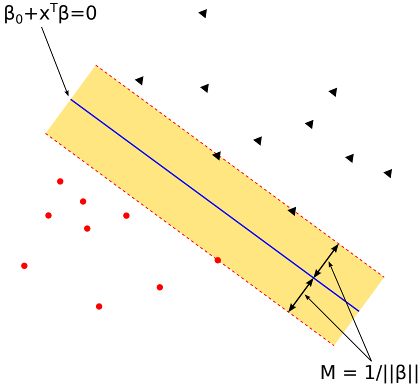


$$\max_{\beta_0, \boldsymbol{\beta}} M$$

$$\text{subject to } \frac{1}{\|\boldsymbol{\beta}\|} y_i(\beta_0 + \mathbf{x}_i^T \beta) \geq M, \quad i = 1, \ldots, n$$

Optimal Separating Hyperplanes

- Since any scaling of β_0 and $\boldsymbol{\beta}$ does not change the margin, we can set $\|\boldsymbol{\beta}\| = 1/M$ and obtain the **convex** optimization problem at the bottom.

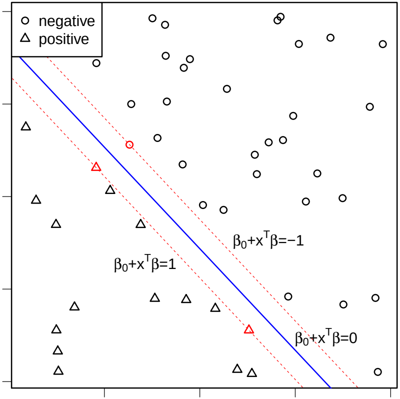


$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|^2$$

$$\text{subject to } y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) \geq 1, \quad i = 1, \ldots, n$$

Optimal Separating Hyperplanes – Support Points

- The hyperplane is defined by a **linear combination** of points lying on the boundary of the margin (**support points**).
- $\beta = \mathbf{X}^T \alpha$, where $\alpha \in \mathbb{R}^n$ is estimated by the classifier and $\alpha_i = 0$ if the i-th sample is **not a support point**.
- Hence, the solution only depends on the support points not on the whole data set.

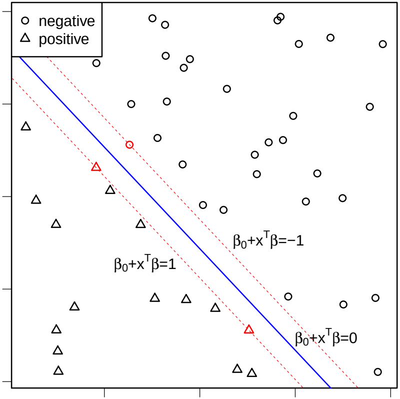


Optimal Separating Hyperplanes – Prediction

- A new sample is classified by

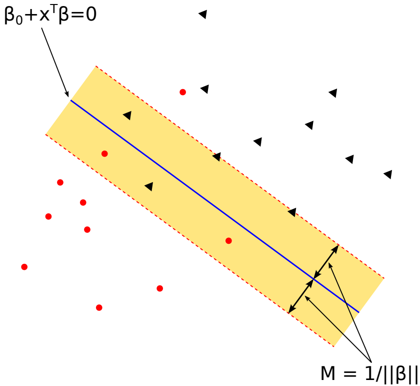
$$\begin{aligned}\mathrm{class}(\mathbf{x}_i) &= \mathrm{sign}\hat{f}(\mathbf{x}_i)\\ &= \mathrm{sign}(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\beta})\end{aligned}$$

- If a sample of the positive class ($y_i = 1$) is misclassified, then $\beta_0 + \beta_1 x_{i1} + \ldots + \beta_m x_{im} < 0$
- The opposite is true if a sample of the negative class ($y_i = -1$) is misclassified.



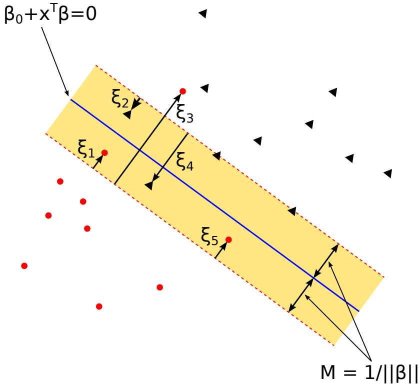
Support Vector Machines

- **Problem**: In real-world applications classes are rarely separated.
- Usually, two classes **overlap** in feature space.
- **Idea**: Still maximise the margin but allow for some points to reside on the wrong side of the margin (**soft margin**).



Support Vector Machines

- Introduce for each sample a **slack variable** $\xi_i \geq 0$ which gives the relative amount, with respect to the margin, by which the prediction falls on the wrong side of its margin.
- If the point is on the correct side, $\xi_i = 0$.
- Points for which $0 < \xi_i \leq 1$ lie between the margin and the correct side of the margin.
- Misclassification occurs if $\xi_i > 1$.



Support Vector Machines

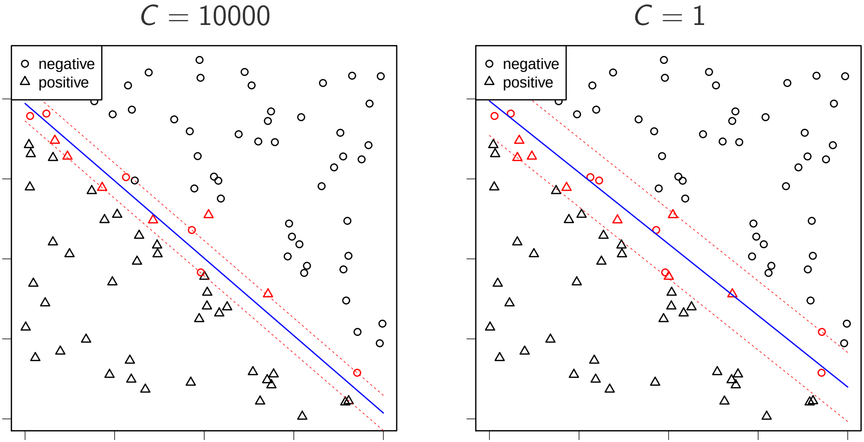
Definition (SVM Optimization)

$$\min_{\beta_0, \beta} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to } \xi_i \geq 0,\ \ y_i(\beta_0 + \mathbf{x}_i^T \beta) \geq 1 - \xi_i$$

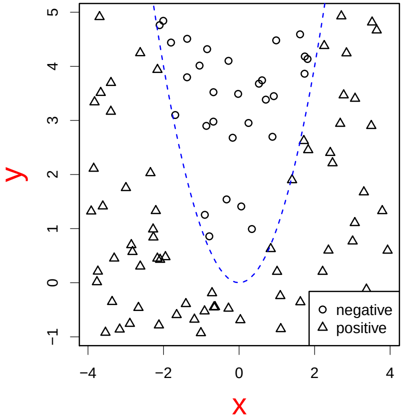
- The parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.
- If $C = \infty$, the result is equal to *optimal separating hyperplanes*.
- $\sum \xi_i$ is an upper bound on the number of misclassified points.

Support Vector Machines – Examples

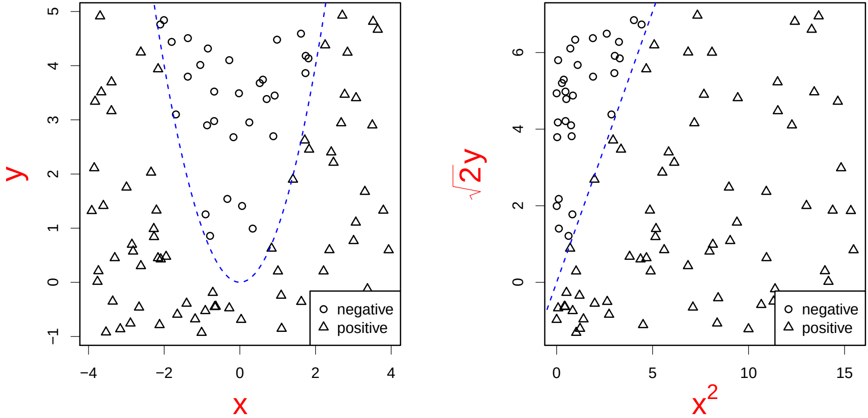


Non-linear SVMs

- **Problem**: In many applications the data is not **linearly separable**.
- **Idea**: Find a non-linear mapping from the input space into a (higher dimensional) feature space in which data are separable.



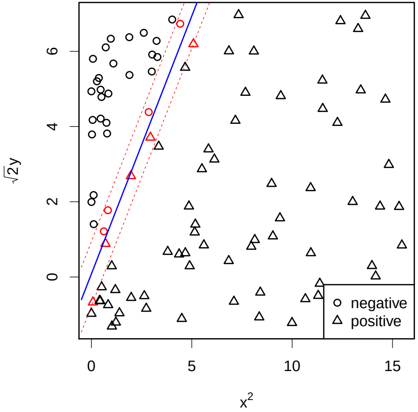
Non-linear SVMs – Transformation



Example: Transform point (x, y) to $(x^2, \sqrt{2}y)$ where the data can be separated linearly.

Non-linear SVMs – Transformation

- Map data from the input space $\mathcal{X} \subseteq \mathbb{R}^d$ to feature space $\mathcal{F} \subseteq \mathbb{R}^D$ using a **non-linear function** $\phi : \mathcal{X} \to \mathcal{F}$, where $d \leq D$.
- Therefore, the decision function becomes $f(\mathbf{x}_i) = \beta_0 + \phi(\mathbf{x}_i)^T \beta$.
- **Example**: Transform data from $\mathbb{R}^2$ into $\mathbb{R}^6$ using ϕ and find a linear hyperplane in the extended space.



$$\phi(\mathbf{x}_i) = (x_{i1}^2, x_{i2}^2, \sqrt{2} \cdot x_{i1}, \sqrt{2} \cdot x_{i2}, \sqrt{2} \cdot x_{i1} \cdot x_{i2}, 1)^T$$

Non-linear SVMs – Transformation

- **Problem**: Explicitly computing the non-linear features requires an increased amount of memory.
- Remember, $\boldsymbol{\beta}$ is a linear combination of support points, i.e. $\boldsymbol{\beta} = \mathbf{X}^T \boldsymbol{\alpha}$ and $\beta_j = \sum_{i=1}^n \alpha_i x_{ij}$, where $\alpha_i = 0$ if $\mathbf{x}_i$ is not a support point.
- The decision function can be formulated as

$$f(\mathbf{x}_0) = \beta_0 + \sum_{j=1}^{m} x_{0j} \beta_j = \beta_0 + \sum_{i=1}^{n} \alpha_i \mathbf{x}_i^T \mathbf{x_0}$$

- Applying the transformation function ϕ we obtain

$$f(\mathbf{x}_0) = \beta_0 + \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x_0})$$

Non-linear SVMs – Kernel

Definition (Kernel Function)

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

Definition (Kernel SVM)

$$f(\mathbf{x}_0) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_0)$$

Kernel Trick

- If the Kernel function can be computed efficiently, we can avoid to explicitly transform the data into the extended feature space.
- No explicit representation of ϕ is required.

Kernel Functions

- Linear:

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$$

- d-th degree Polynomial:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

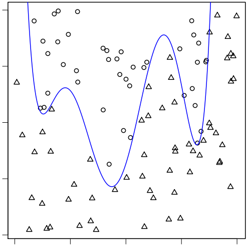
- Radial Basis Function (RBF):

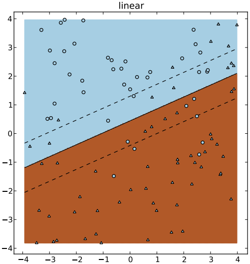
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

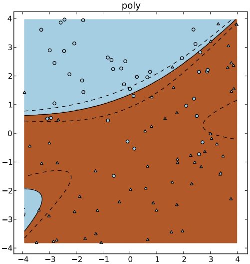
- Sigmoid:

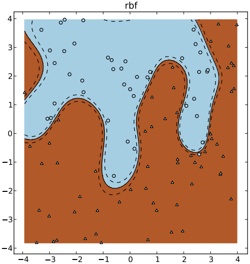
$$K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \cdot \mathbf{x}^T \mathbf{x}' + c)$$

Kernel Functions – Examples









Multi-class SVMs

- SVMs as previously discussed are only applicable to binary classification problems.
- **Idea**: Construct multiple binary SVMs to distinguish $k > 2$ classes from each other.
- **One vs. all**: Train k classifiers where the i-th classifier is given the labels of the i-th class as positives and everything else as negative.
- **One vs. One**: Train $\sum_{i=1}^{k-1} i$ classifiers where each classifier is trained on samples from the i-th and j-th class, respectively.

Summary

- **Least squares** model is simple to construct but yields only good results if relationship is linear, no outliers and no multicollinearity is present.
- **Logistic regression** separates data linearly, yields true probabilities and the notion of log-odds makes it useful in numerous disciplines (e.g. medicine, social science). Can be extended to natively support multiple classes.
- **Optimal separating hyperplanes** can be applied rarely.
- **Support vector machines** can be used both for classification and regression and thanks to the Kernel trick in a wide range of applications. The best choice of Kernel and its parameters is not obvious and requires lots of testing.

References (1)

Ben-Hur, A. and Weston, J. (2009).
A user's guide to support vector machines.
In *Data Mining Techniques for the Life Sciences*. Springer.
http://www.cs.colostate.edu/~asa/pdfs/howto.pdf.

Bishop, C. M. (2006).
Pattern Recognition and Machine Learning.
Springer.
http://research.microsoft.com/~cmbishop/PRML.

Hastie, T., Tibshirani, R., and Friedman, J. (2009).
The Elements of Statistical Learning.
Springer, second edition.
http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

References (2)

Hosmer, D. W. and Lemeshow, S. (2000).
Applied Logistic Regression.
John Wiley & Sons, second edition.