Classification II: SVM variants

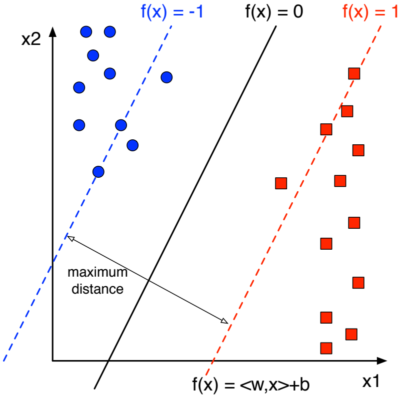
Mario Michael Krell

DFKI Bremen & Universität Bremen

Robotics Innovation Center

Director: Prof. Dr. Frank Kirchner

www.dfki.de/robotics



Outline

Can we apply Regression to binary classification?

Yes we can!

- $Y = \mathbb{R} \rightarrow Y = \{-1, +1\}$

Yes we can!

- $Y = \mathbb{R} \rightarrow Y = \{-1, +1\}$
- Ridge Regression $\rightarrow$ regularized Kernel Fisher Discriminant $\rightarrow$ Least Squares Support Vector Machine

Yes we can!

- $Y = \mathbb{R} \rightarrow Y = \{-1, +1\}$
- Ridge Regression $\rightarrow$ regularized Kernel Fisher Discriminant $\rightarrow$ Least Squares Support Vector Machine
- Support Vector Regression $\rightarrow$ regularized Kernel Fisher Discriminant with ϵ-insensitive loss $\rightarrow$ Support Vector Machine

Yes we can!

- $Y = \mathbb{R} \rightarrow Y = \{-1, +1\}$
- Ridge Regression $\rightarrow$ regularized Kernel Fisher Discriminant $\rightarrow$ Least Squares Support Vector Machine
- Support Vector Regression $\rightarrow$ regularized Kernel Fisher Discriminant with ϵ-insensitive loss $\rightarrow$ Support Vector Machine
- Recap: Support Vector Regression $\stackrel{\epsilon=0}{\rightarrow}$ Ridge Regression

Class labeling

Once the hyperplane is defined, we can use a simple decision function for class labelling:

Decision function

$$d(\mathbf{x}) = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

Note: The decision function is scale invariant, i.e. $\mathbf{w} \to \lambda \mathbf{w}, b \to \lambda b$

Outline

Balanced Relative Margin Machine

regularized Kernel Fisher Discriminant with ϵ-insensitive loss:

$$\begin{aligned} \min_{w,b,t} \quad & \tfrac{1}{2}\|w\|_2^2 + C\|t\|_\epsilon \\ \text{s.t.} \quad & y_j(\langle w, x_j\rangle + b) \quad = 1 - t_j \quad \forall 1 \leq j \leq n. \end{aligned} \tag{1}$$

Consider the hyperparameter mapping:

$$(C', R') = \left(\frac{C}{1-\epsilon}, \frac{1+\epsilon}{1-\epsilon}\right) \tag{2}$$

Balanced Relative Margin Machine

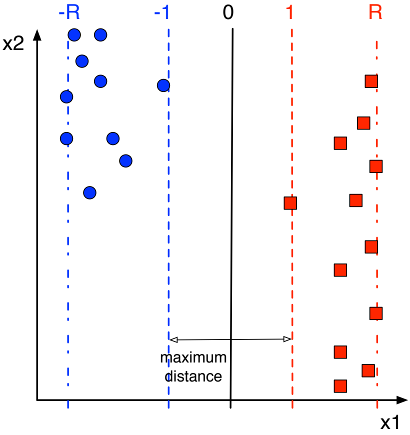
regularized Kernel Fisher Discriminant with ϵ-insensitive loss:

$$\begin{aligned} \min_{w,b,t} \quad & \tfrac{1}{2}\|w\|_2^2 + C\|t\|_\epsilon \\ \text{s.t.} \quad & y_j(\langle w, x_j\rangle + b) = 1 - t_j \quad \forall 1 \leq j \leq n. \end{aligned} \tag{1}$$

New Model with better parameterization:

$$\begin{aligned} \min_{w,b,t} \quad & \tfrac{1}{2}\|w\|_2^2 + C\sum s_j + C\sum t_j \\ \text{s.t.} \quad & R + s_j \geq y_j(\langle w, x_j\rangle + b) \geq 1 - t_j \quad \forall 1 \leq j \leq n \\ & s_j, t_j \geq 0 \quad \forall 1 \leq j \leq n. \end{aligned}$$

Balanced Relative Margin Machine

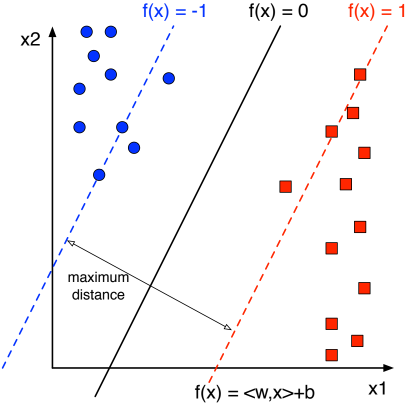


Outline

Support Vector Machine ($R = \infty$)

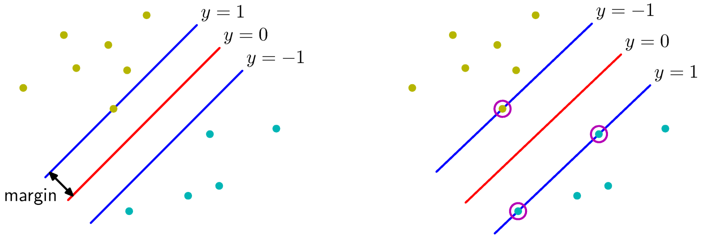
$$
\begin{array}{lrll}
\min\limits_{w,b,t} & \frac{1}{2}\|w\|_2^2 + C\sum t_j & & \\
\text{s.t.} & y_j(\langle w, x_j\rangle + b) & \geq 1 - t_j & \forall 1 \leq j \leq n \\
 & t_j & \geq 0 & \forall 1 \leq j \leq n.
\end{array}
$$

Support Vector Machine ($R = \infty$)



Maximum margin

Computational learning theory aka statistical learning theory motivates a maximum margin classifier:



(a) margin (b) support vectors

The distance

The distance of a point from a hyperplane $h(\mathbf{x})$:

$$dist(x) = \frac{|h(\mathbf{x})|}{||\mathbf{w}||}$$

Now, scale $\mathbf{w}, b$ so that for the nearest(!) points of each class $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)$ the following holds:

Support vectors

$$y_1 : \mathbf{w} \cdot \mathbf{x}_1 + b = +1$$

$$y_2 : \mathbf{w} \cdot \mathbf{x}_2 + b = -1$$

Combining both:

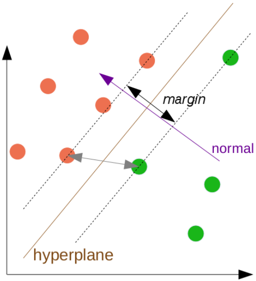
$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$$

The margin

So, using the nearest point to the margin, define the size of the margin m as:

Margin

$$|m| = \frac{1}{||\mathbf{w}||} + \frac{1}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$$



Implication of defining the margin

With the margin being defined by the nearest points, this has some implications on other data points:

Implications

$$\forall(\mathbf{x}, y_1)\ \mathbf{w} \cdot \mathbf{x} + b \geq +1$$

$$\forall(\mathbf{x}, y_2)\ \mathbf{w} \cdot \mathbf{x} + b \leq -1$$

"Combined" representation

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1\ \forall i$$

Maximizing the margin

Goal

(For better generalization) find a parameter set of $\mathbf{w}, b$ for the decision hyperplane that maximizes the margin $\frac{2}{||w||}$.

Equivalent optimization problem

Minimize (the norm) $||\mathbf{w}||$ subject to $y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1$

Note: For reasons of computability $||\mathbf{w}||$ is replaced by $J(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$. This can be solved using *quadratic programming* and does not affect the optimization problem.

Primal and dual optimization problem

Primal optimization problem (Lagrange multipliers **a**,**u**)

$$L(\mathbf{w}, b, \mathbf{a}, \xi, \mathbf{u}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i - \sum_i a_i\left[y_i h(\mathbf{x}_i) - 1 + \xi_i\right] - \sum_i u_i \xi_i$$

Known dual representation of optimization problem

$$L'(\mathbf{a}) = \sum_i a_i - \frac{1}{2}\sum_i\sum_j a_i a_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to (box constraints):

$$0 \leq a_i \leq C \tag{2}$$

$$\sum_i a_i y_i = 0 \tag{3}$$

Non linear classification

So far, we have used SVMs with a linear kernel, i.e.

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$$

To allow non-linear classification the *kernel trick* can be applied, and this linear kernel function can be replaced by a non-linear one.

Kernel examples $k(\mathbf{x}, \mathbf{x}')$

polynomial:	$(\lambda \mathbf{x} \cdot \mathbf{x}' + r)^d$, where $\lambda > 0$
radial basis:	$\exp(-\lambda \|\|\mathbf{x} - \mathbf{x}'\|\|^2)$, where $\lambda > 0$
sigmoid:	$\tanh(\lambda \mathbf{x} \cdot \mathbf{x}' + r)$

Evaluation

Advantages:

- 'easy' basic concept
- good generalization
- decision surface, i.e. hypothesis has explicit dependence on data
- optimization of a convex function, i.e. no dealing with local minima
- few parameters (with linear kernel)
- confidence measures can be included
- relatively fast in delivering the classifier and the classification
- general application as Kernel machine

Evaluation

Drawbacks:

- only directly applicable for binary classification
- does provide a decision, but not a likelihood
- sensitivity to asymmetric class distributions
- numerous variants of SVMs using different kernels
- parameters not trivial to optimize

Outline

Solving the SVM Dual

Dual representation of the SVM optimization problem

$$\max \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j - \sum_i a_i k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to:

$$0 \leq a_i \leq C \tag{4}$$

$$\sum_i a_i y_i = 0 \tag{5}$$

Where is the problem?

Solving the SVM Dual

Dual representation of the SVM optimization problem

$$\max \frac{1}{2}\sum_i \sum_j a_i a_j y_i y_j - \sum_i a_i k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to:

$$0 \leq a_i \leq C \tag{4}$$

$$\sum_i a_i y_i = 0 \tag{5}$$

Where is the problem?

$$\sum_i a_i y_i = 0$$

The Trick: Offset Tweaking

Before:

$$\begin{aligned} \min_{w,b,t} \quad & \tfrac{1}{2}\|w\|_2^2 + C\sum t_j \\ \text{s.t.} \quad & y_j(\langle w, x_j\rangle + b) \geq 1 - t_j \quad \forall 1 \leq j \leq n \\ & t_j \geq 0 \quad \forall 1 \leq j \leq n. \end{aligned}$$

After:

$$\begin{aligned} \min_{w,b,t} \quad & \tfrac{1}{2}\|(w,b)\|_2^2 + C\sum t_j \\ \text{s.t.} \quad & y_j \langle (w,b),(x_j,1)\rangle \geq 1 - t_j \quad \forall 1 \leq j \leq n \\ & t_j \geq 0 \quad \forall 1 \leq j \leq n. \end{aligned}$$

The Benefit

$$
\begin{aligned}
\min_{w,b,t} \quad & \tfrac{1}{2}\left\|(w,b)\right\|_2^2 + C\sum t_j \\
\text{s.t.} \quad & y_j \left\langle (w,b),(x_j,1)\right\rangle \geq 1 - t_j \quad \forall 1 \leq j \leq n \\
& t_j \geq 0 \quad \forall 1 \leq j \leq n.
\end{aligned}
$$

Dual of the modified SVM optimization problem

$$\max \frac{1}{2}\sum_i \sum_j a_i a_j y_i y_j - \sum_i a_i k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to: $0 \leq a_i \leq C$

Solution algorithm

Dual of the modified SVM optimization problem

$$\max \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j - \sum_i a_i k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to: $0 \leq a_i \leq C$

Direction Search:

1. start with $\alpha = 0$
2. iterate over all indices
   $\rightarrow$ for each index i determine optimal α_i and update it
3. To many iterations? $\rightarrow$ STOP
4. To small change in the loop? $\rightarrow$ STOP
5. GOTO 2
6.

Online version? – Single Iteration – Passive-Aggressive Algorithm

INPUT: aggressive parameter $C > 0$
INITIALIZE: $w_1 = (0, \dots, 0)$
For $t = 1, 2, \dots$

- receive instance: $x_t \in \mathbb{R}^m$
- predict: $\hat{y}_t = \operatorname{sign} \langle w_t, x_t \rangle$
- receive correct label: $y_t \in \{-1, +1\}$
- suffer loss: $l = \max \{0,\ 1 - y_t \langle w_t, x_t \rangle\}$
- update:
    1. set: determine α_t
    2. update: $w_{t+1} = w_t + \alpha_t y_t x_t$

FINISHED

$$\alpha_t = \frac{l_t}{\|x_t\|^2} \quad \text{(PA)}$$

$$\alpha_t = \min \left\{ C, \frac{l_t}{\|x_t\|^2} \right\} \quad \text{(PA-I)}$$

$$\alpha_t = \frac{l_t}{\|x_t\|^2 + \frac{1}{2C}} \quad \text{(PA-II)}$$

different losses:
hard margin (PA), **hinge loss** (PA-I, online SVM), squared hinge loss (PA-II)

Outline

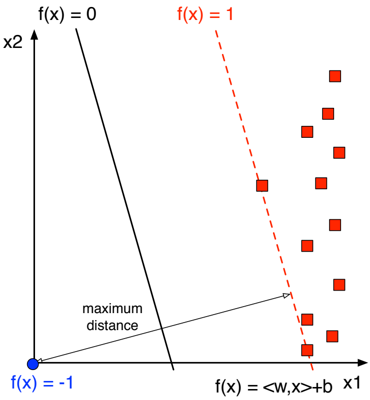
Why unary classification?

Why unary classification?

- data description
- novelty detection
- outlier detection
- one-vs-rest classifier in multi-class scenario

How?

Origin Separation



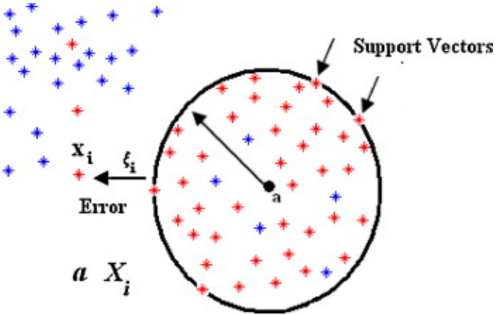
One-Class Support Vector Machine (simplified)

$$\begin{array}{ll} \min\limits_{w,t} & \frac{1}{2}\|w\|_2^2 + C\sum t_i \\ \text{s.t.} & \langle w, x_i\rangle \geq 2 - t_i \\ \text{and} & t_i \geq 0 \; \forall i\ . \end{array}$$

One-Class Balanced Relative Margin Machine

$$\begin{array}{ll} \min\limits_{w,t} & \frac{1}{2}\|w\|_2^2 + C\sum s_i + C\sum t_i \\ \text{s.t.} & 1 + R + s_i \geq \langle w, x_i\rangle \geq 2 - t_i \\ \text{and} & s_i, t_i \geq 0 \; \forall i\ . \end{array}$$

Support Vector Data Description



$$
\begin{array}{ll}
\min\limits_{R,a,t'} & R^2 + C' \sum t'_i \\
\text{s.t.} & \|a - x_i\|_2^2 \leq R^2 + t'_i \\
\text{and} & t'_i \geq 0 \ \forall i \ .
\end{array}
$$

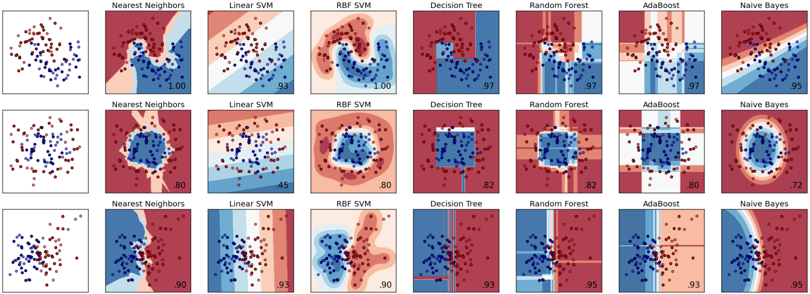
Outline

Wrap up

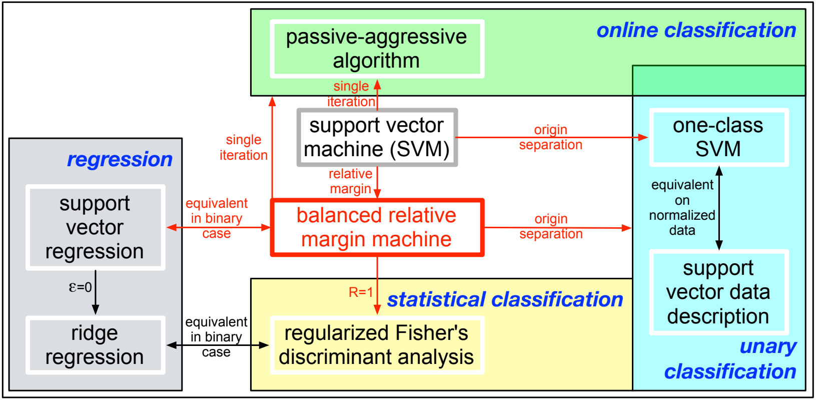
Classifiers:

1. nearest neighbor
2. naive Bayes
3. logistic regression
4. decision trees
5. regularized kernel Fisher discriminant
6. balanced relative margin machine
7. support vector machines
8. online passive aggressive algorithms (single iteration)
9. one-class SVM (zero separation)
10. support vector data description

Wrap Up: Classifier Comparison with Scikit-learn



Wrap Up: Classifier Connections



The End *… is just the beginning*