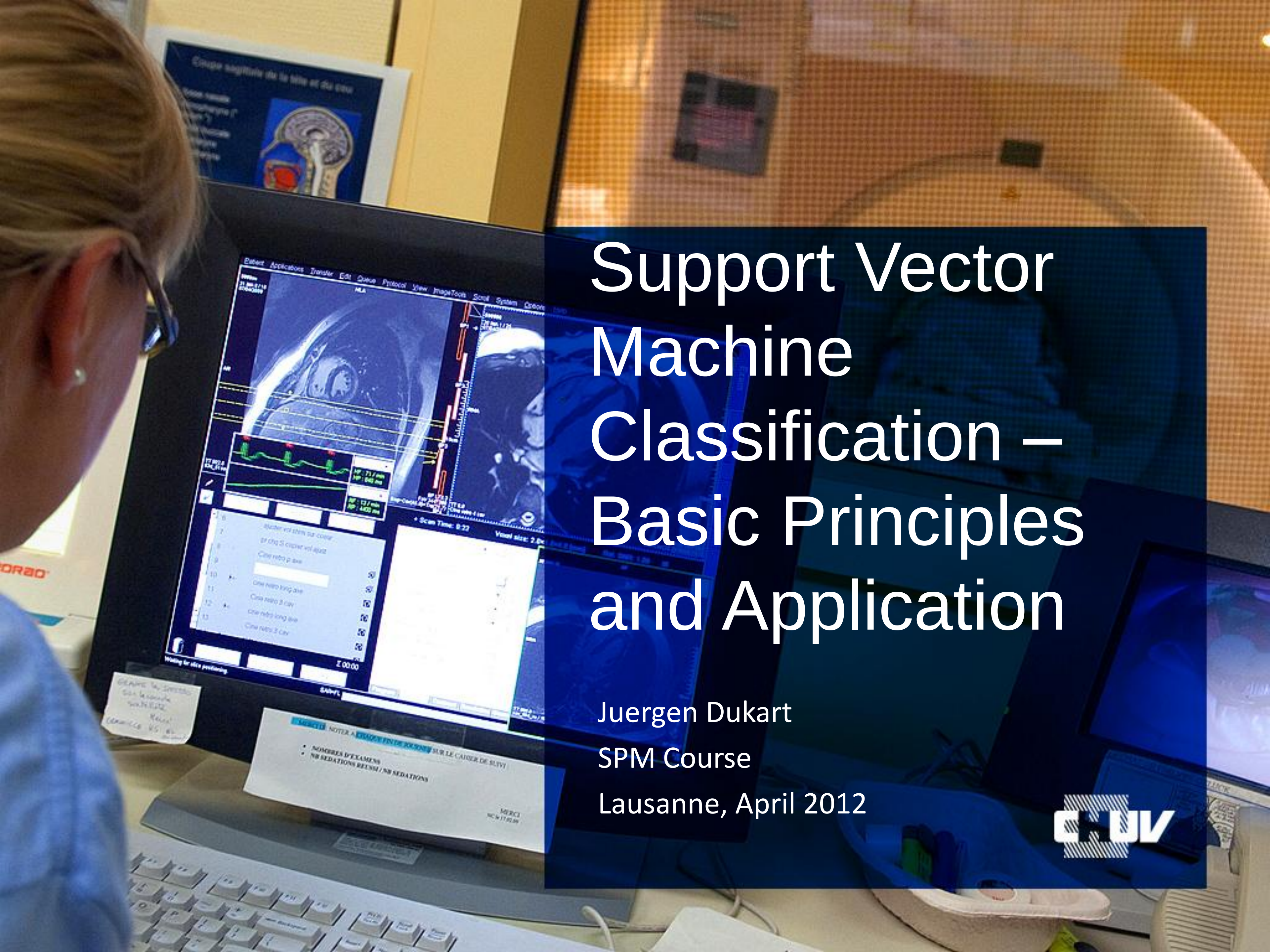


Support Vector Machine Classification – Basic Principles and Application

Juergen Dukart

SPM Course

Lausanne, April 2012

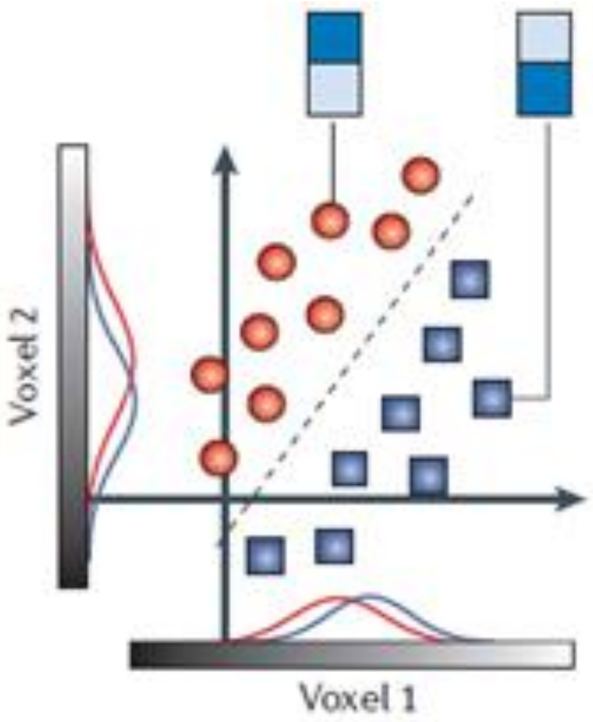
Outline

- Basic principles
- What do we get from SVM?
- SVM Modifications
- Applications in Neuroimaging
- Available software packages

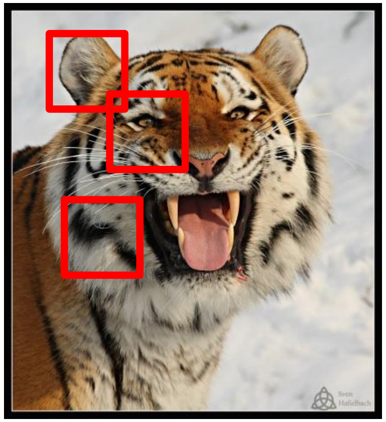
Idea behind

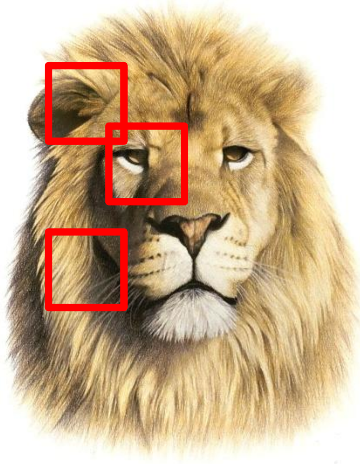
- To recognize multivariate patterns
- Supervised learning
- Providing predictions for new data

Basic principles



Basic principles



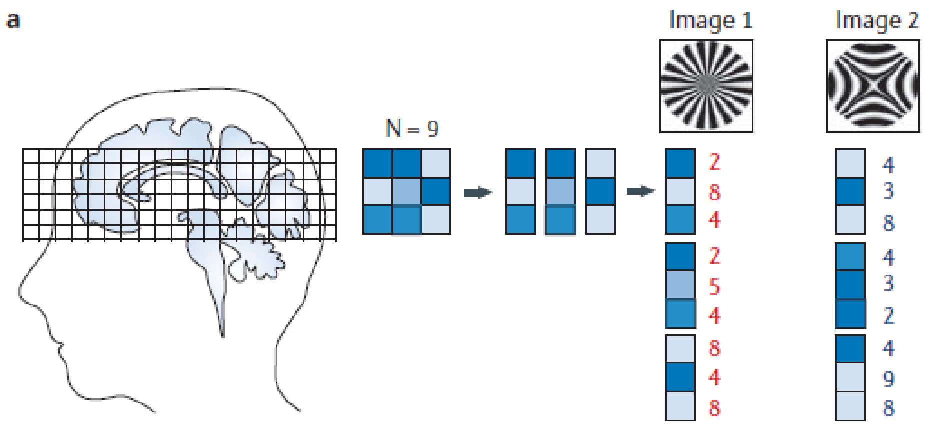


Training data

Decision boundary

Testing data

Basic principles

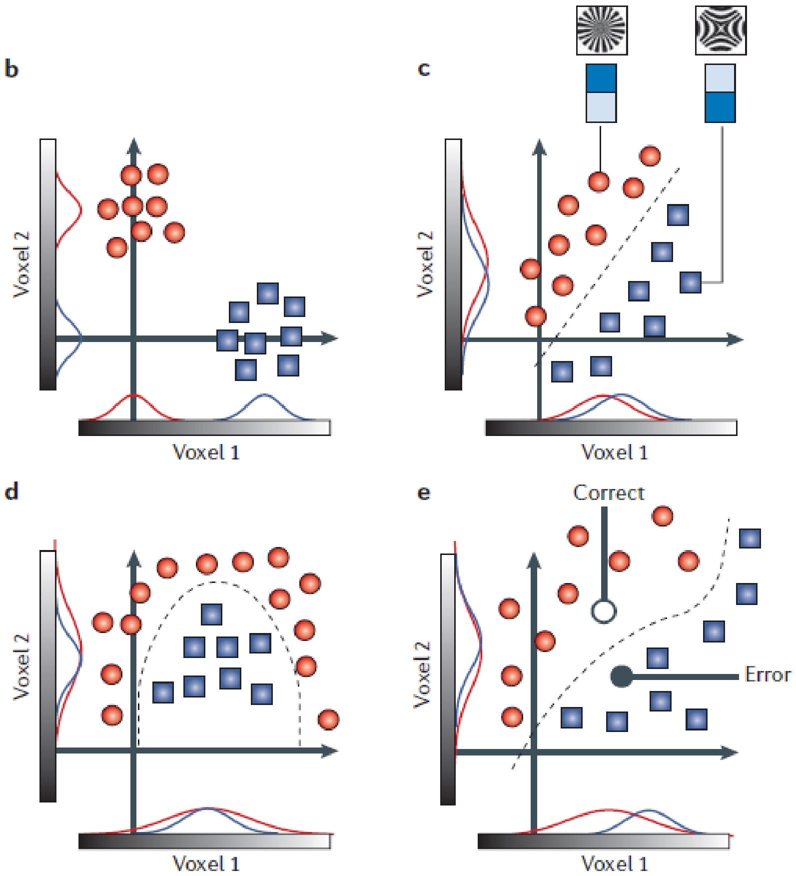


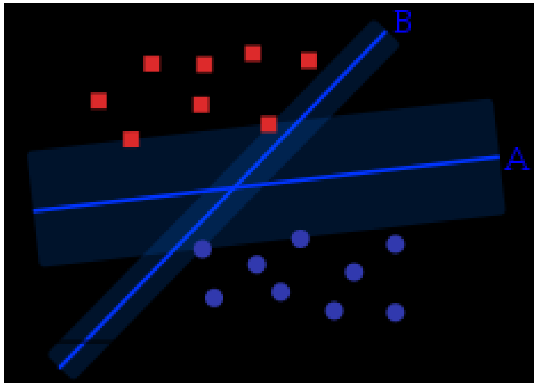
N-dimensional space

In Neuroimaging up to several million features

Basic principles

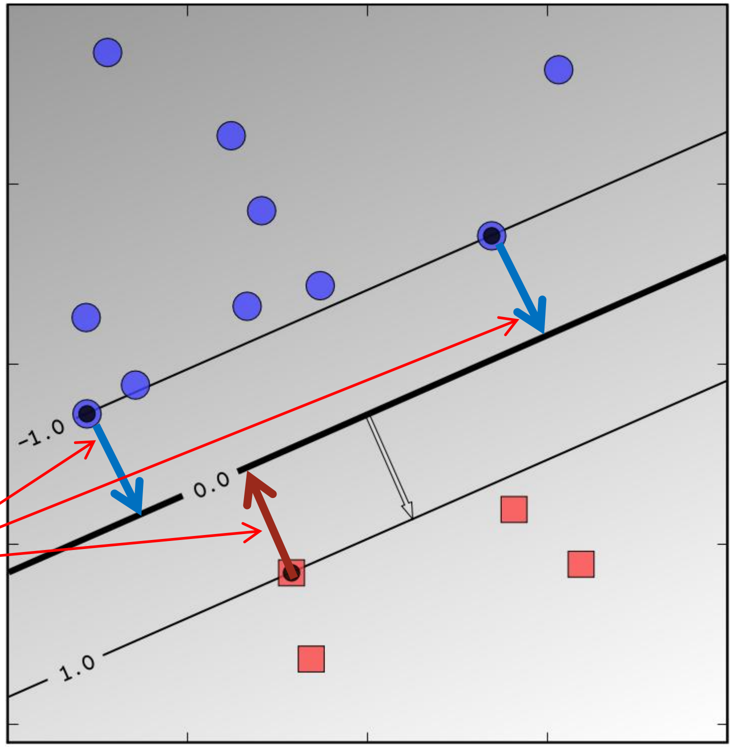
For N = 2





Basic principles

Support vectors



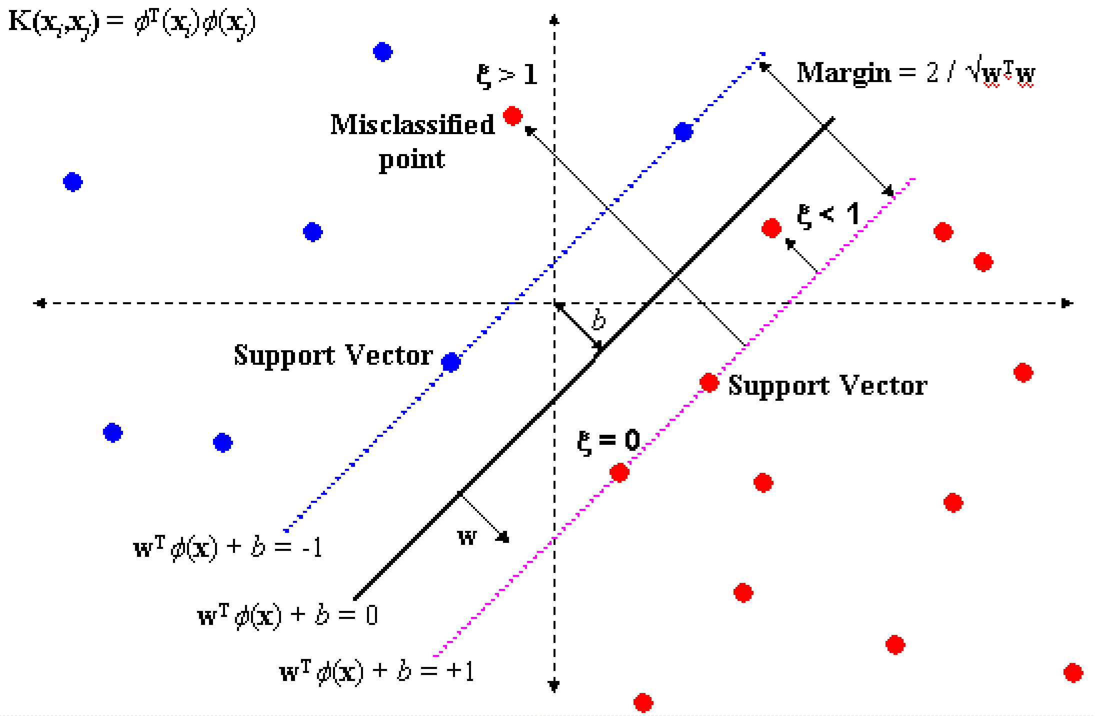
Basic principles

Φ: implicit embedding function

($\mathbf{x}_i$, y_i): training data from 2 classes such that $y_i = \pm 1$

A Support Vector Machine finds a hyperplane
$\mathbf{w}^T \Phi(\mathbf{x}) + b = 0$
that best separates the two classes.

→ maximises the margin
→ while minimising some measure of loss on the training data.



Basic principles

Primal problem:

User defined penalty

$$\min_{\mathbf{w},\boldsymbol{\xi}} \quad \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\,\mathbf{1}^t\boldsymbol{\xi} \qquad (1)$$

$$\text{subject to} \quad y_i(\mathbf{w}^t\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \qquad (2)$$

$$\boldsymbol{\xi} \geq 0 \qquad (3)$$

$$\text{where} \quad \phi^t(\mathbf{x}_i)\phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \qquad (4)$$

Dual problem:

$$\max_{\boldsymbol{\alpha}} \quad \mathbf{1}^t\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^t\mathbf{Y}\mathbf{K}\mathbf{Y}\boldsymbol{\alpha} \qquad (5)$$

$$\text{subject to} \quad 0 \leq \boldsymbol{\alpha} \leq C, \;\; \mathbf{1}^t\mathbf{Y}\boldsymbol{\alpha} = 0 \qquad (6)$$

Diagonal matrix with labels on the diagonal

Perpendicular to the hyperplane

$w = \Sigma_i\, y_i\, \alpha_i \Phi(x_i)$

What do we get from SVM?

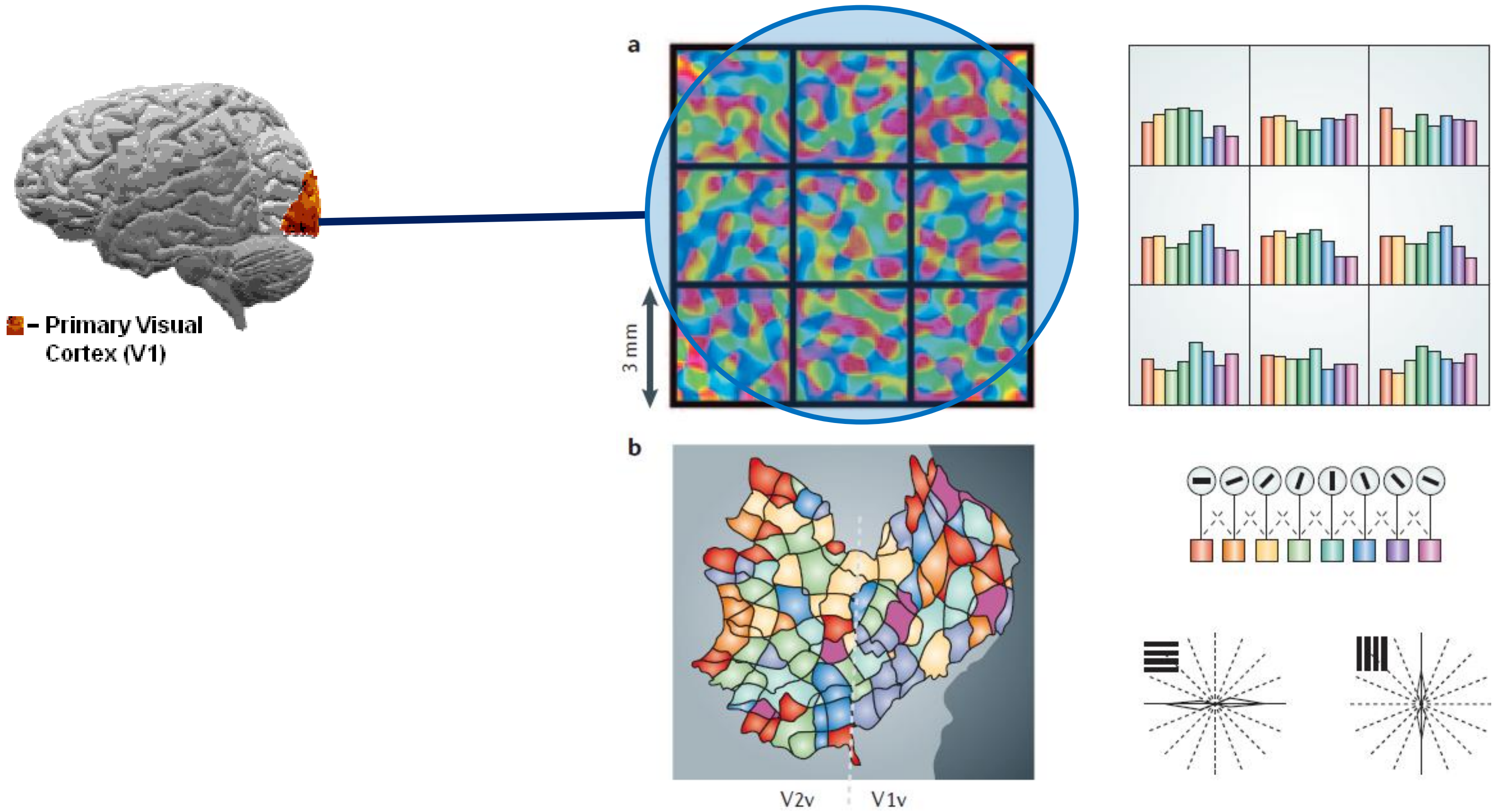
- We get a model allowing us to classify new data
- We get the importance of each feature for separation (weights*)
- We get support vectors (Subjects – information close to the boundary)
- We get not only a binary decision but also a probability/certainty for a specific decision

*Feature weights for each feature = Support vectors * support vector coefficients

SVM Modifications

- Relevance Vector Machines → uses a sparse feature set extracted from SVM
- Combination with Markov random fields using neighborhood information as features
- Multi-class SVM
- Support vector regression→ predicts continuos variables

Applications in Neuroimaging

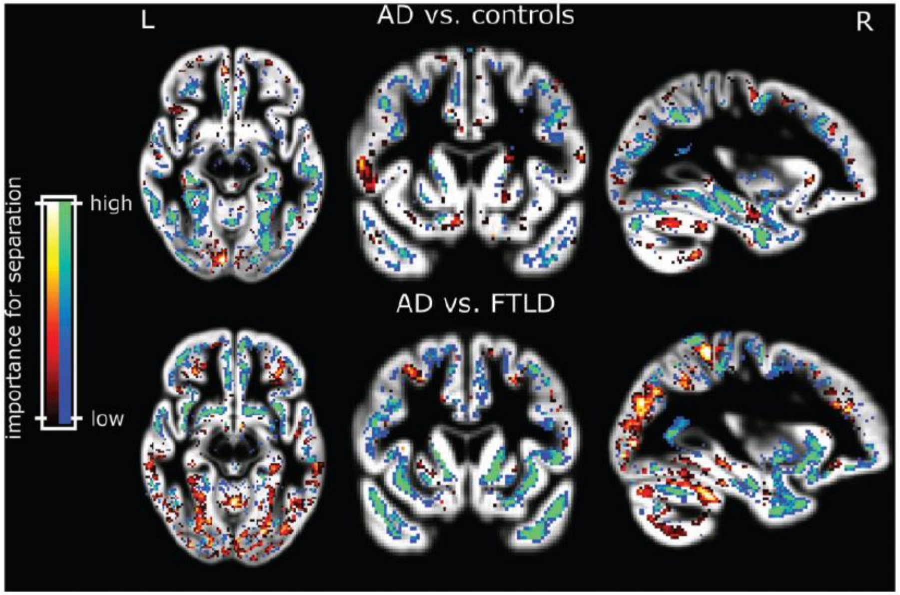


Applications in Neuroimaging

Table 2 Results of SVM classification using grey matter from the whole brain for image analysis

Group	Correctly classified (%)	Sensitivity (%)*	Specificity (%)*
AD and controls Group I	95.0	95.0	95.0
AD and controls Group II	92.9	100	85.7
AD and controls Group III	81.1	60.6	93.0
Dataset I for training, set II for testing	96.4	100	92.9
Dataset II for training, set I for testing	87.5	95.0	80.0
Group I + II	95.6	97.1	94.1
AD from Dataset II and FTLD Group IV	89.2	83.3	94.7

*Considering a correctly identified AD case as a true positive.



Applications in Neuroimaging

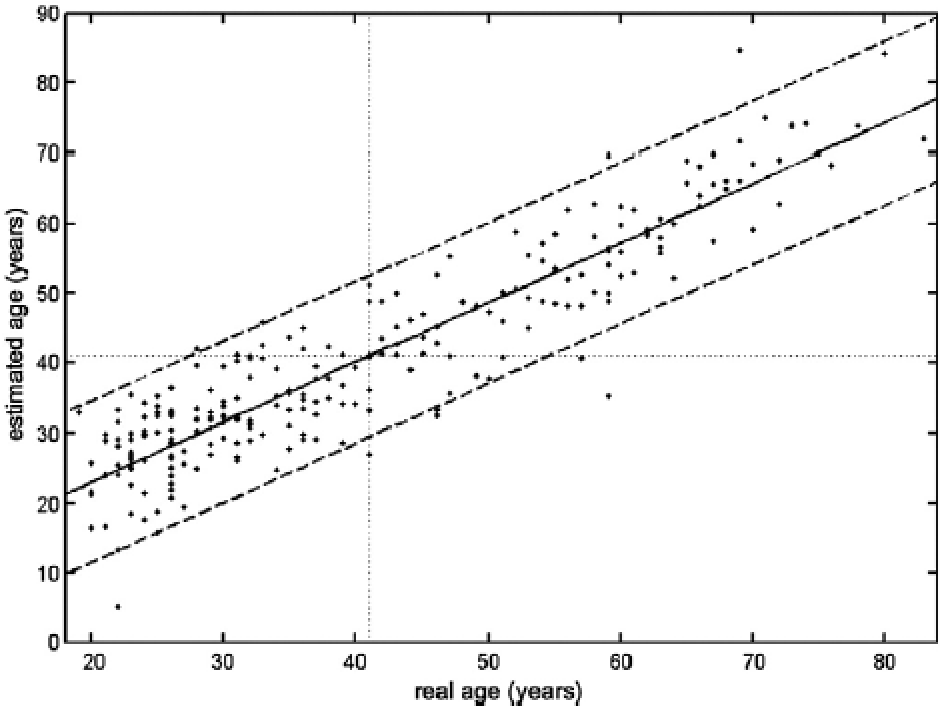
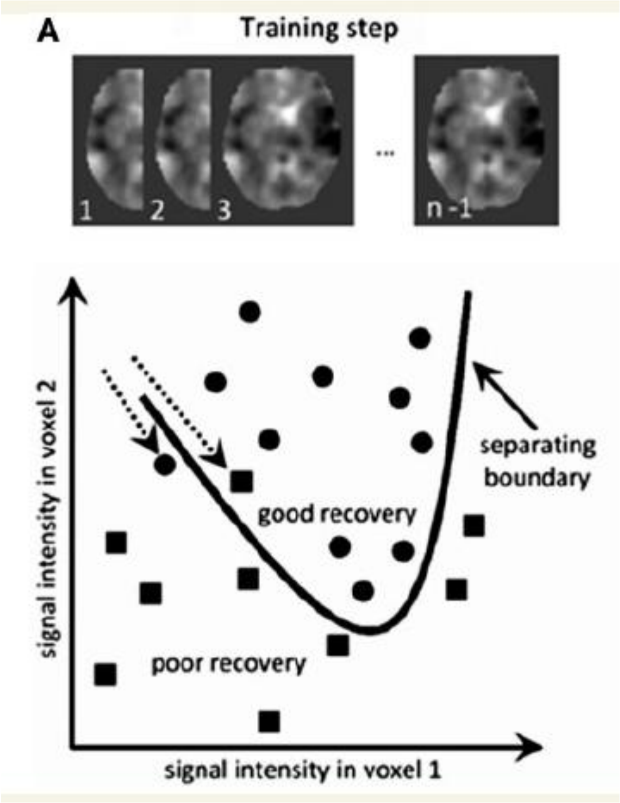
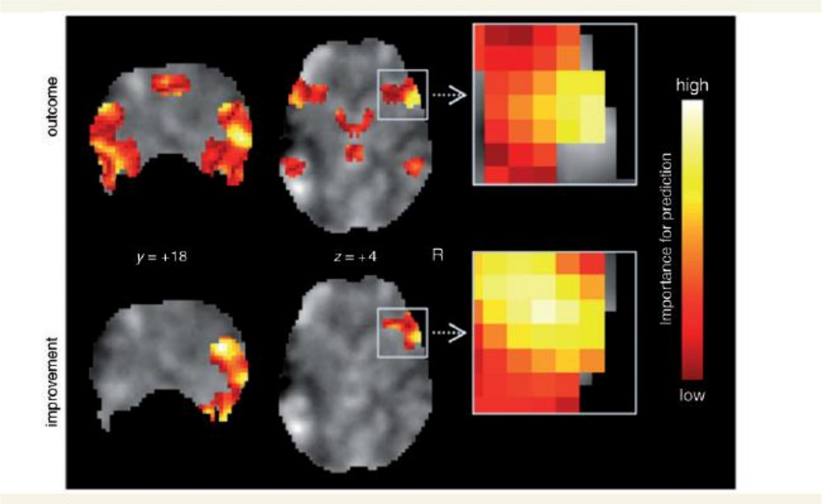


Fig. 3. Estimated age and real age are shown for the whole test sample (TEST1-3 + TEST4) with the confidence interval (dashed lines) at a real age of 41 years of ± 11.5 years. The overall correlation between estimated and real age is $r = 0.92$, and the overall MAE = 4.98 years.

Applications in Neuroimaging

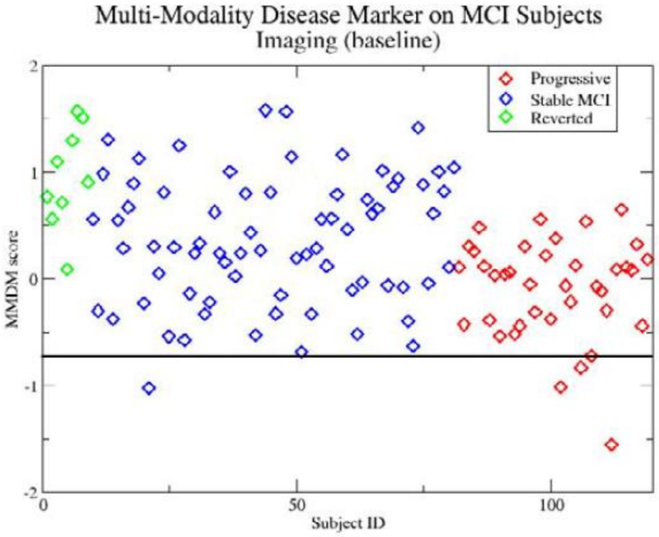




Applications in Neuroimaging

Classification problem: Alzheimer's disease vs healthy aging

Modalities used	Accuracy	Sensitivity	Specificity	Area under ROC
Imaging modalities	0.876	0.789	**0.938**	0.944
Biological measures	0.704	0.581	0.794	0.767
Cognitive scores	**0.912**	**0.892**	0.926	**0.983**
All modalities	0.924	0.867	0.966	0.977



Available software packages

- www.shogun-toolbox.org (*Shogun (toolbox)* *contains about 20 different implementations of SVMs)*
- libsvm libsvm is a library of SVMs which is actively patched
- liblinear liblinear is a library for large linear classification including some SVMs
- flssvm flssvm is a least squares svm implementation written in fortran
- Shark Shark is a C++ machine learning library implementing various types of SVMs
- dlib dlib is a C++ library for working with kernel methods and SVMs
- SVM light is a collection of open-source software tools for learning and classification using SVM.

Thank you for attention!