Winter Course Statistical Learning

Support Vector Machines

Marjolein Fokkema

Leiden University

Support Vector Machines

Online videos provide a great geometrical view of SVMS:

- "We try and find a plane that separates the classes in feature space."
- However: This (hyper)plane is never actually fitted.
- Instead, *observations' weights* are estimated.
- For a linear kernel, the hyperplane (i.e., β coefficients for the variables) can be *derived* from the estimated observation weights.

Maximum Margin Classifier

Finds the hyperplane with as large as possible distance to the nearest observations.

That is, maximize:

$$M(\beta_0, \beta_1, \ldots, \beta_p)$$

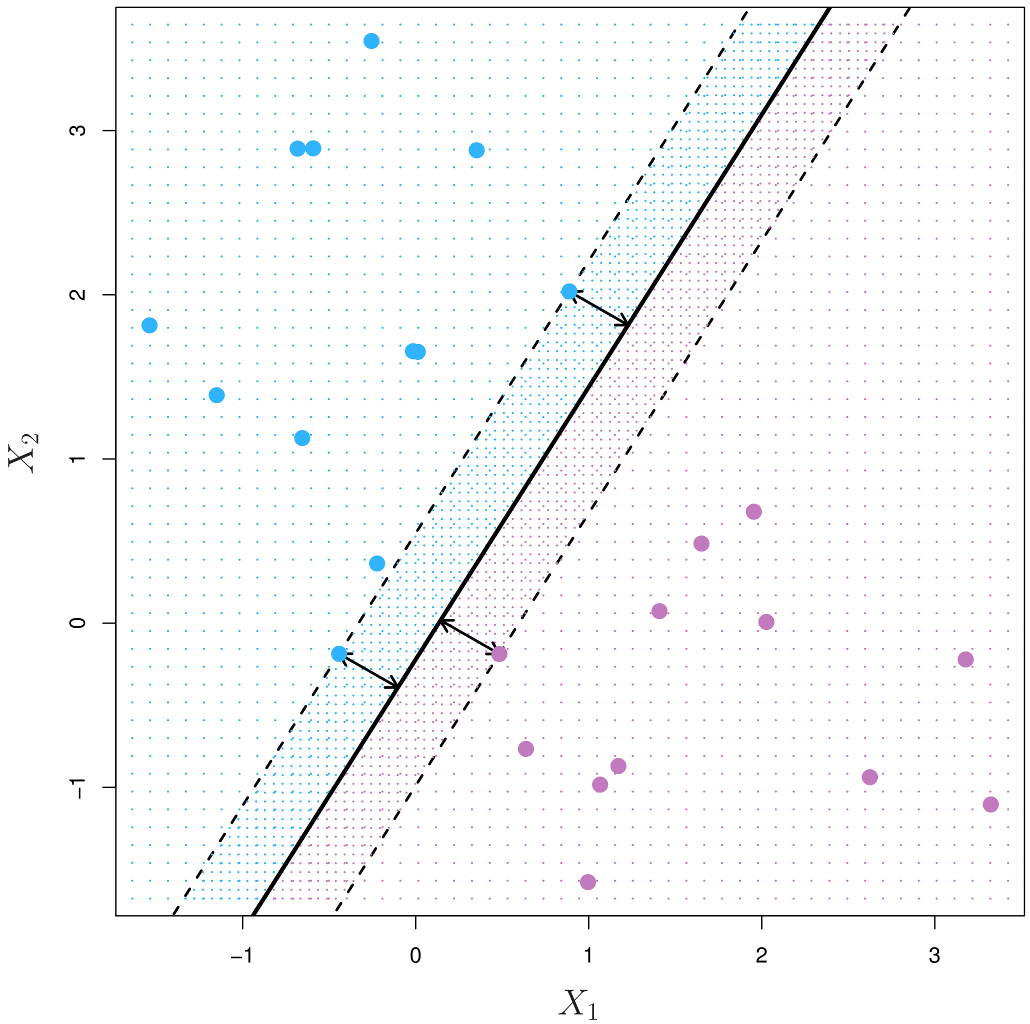
Subject to:

$$\sum_{j=1}^{p} \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M$$

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

Maximum Margin Classifier



Support Vector Classifier

Allows classification errors: $\epsilon_i \geq 0$.

$$\text{Maximize: } M(\beta_0, \beta_1, \ldots, \beta_p, \epsilon_1, \ldots, \epsilon_n).$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i)$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \text{ and } \sum_{i=1}^{n} \epsilon_i \leq C$$

- $\epsilon_i = 0$: observation i is on the correct side of margin and hyperplane.
- $0 < \epsilon_i \leq 1$: observation i is on wrong side of margin, correct side of hyperplane.
- $\epsilon_i > 1$: observation i is on wrong side of the hyperplane.
- Observations with $\epsilon_i > 0$ are the *support vectors*.

Support Vector Classifier: Cost

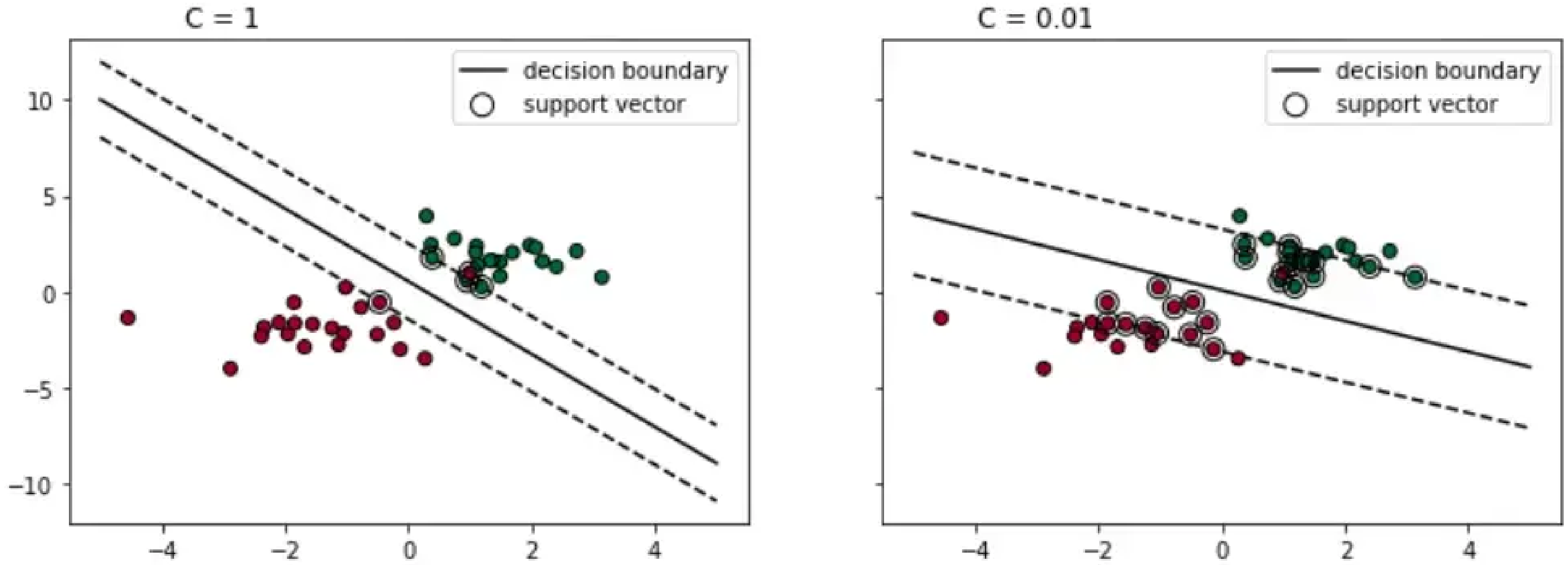
Above, in ISLR book and in online lectures, cost parameter C limits the sum of ϵ_i, functioning like a budget:

- $C = 0$ does not allow errors (hard margin).
- The larger C, the *more* tolerant the classifier becomes of errors.

However, software implementations of SVMs generally have a `cost` argument with opposite effect, it specifies the cost of misclassifications:

- The larger C, the *less* tolerant the classifier becomes of errors.

Support Vector Classifier: Cost



Higher cost yields more narrow margin.

Support Vector Classifier: Estimation and Prediction

- $x_i^\top x$, also written $\langle x_i, x \rangle$, is the *inner product* defined by $\langle x_i, x \rangle = \sum_{j=1}^{p} x_j x_{ij}$
- The inner product is a correlation or similarity measure.
- SVMs do not use the original predictor variables, but only the inner products.

Similarities

Subject	MDD	Consc	Neurot
1	0	-1.10	-0.22
2	0	-0.09	-1.42
3	0	-1.27	-0.33
4	1	-0.93	0.55
5	0	-0.59	-0.43

- Features need to be standardized so they have equal a-priori influence.

- The similarity between a pair of observations is given by the *inner product*:

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

.

- For the first two observations, the similarity is thus: $-1.10 \times -0.09 + -0.22 \times -1.42 = 0.41$.

Support Vector Machine

- We can enlarge the feature space to allow for non-linear boundaries
- Instead of fitting a support vector classifier on

$$X_1, X_2, \ldots X_p$$

We can fit it on

$$X_1, X_1^2, X_2, X_2^2 \ldots X_p, X_p^2$$

and/or include interactions

$$X_1, X_1^2, X_1 X_2, X_2, X_2^2, \ldots, X_{p-1} X_p, X_p, X_p^2$$

- In the enlarged space, the decision boundary is still linear, but in the original space it is non-linear.

Support Vector Machine: The Kernel Trick

- For subjects i and i‘ the polynomial kernel is given by:

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

- for $p = 2$ and $d = 2$

$$K(x_i, x_{i'}) = (1 + x_{i1}x_{i'1} + x_{i2}x_{i'2})^2 =$$

$$1 + 2x_{i1}x_{i'1} + 2x_{i2}x_{i'2} +$$

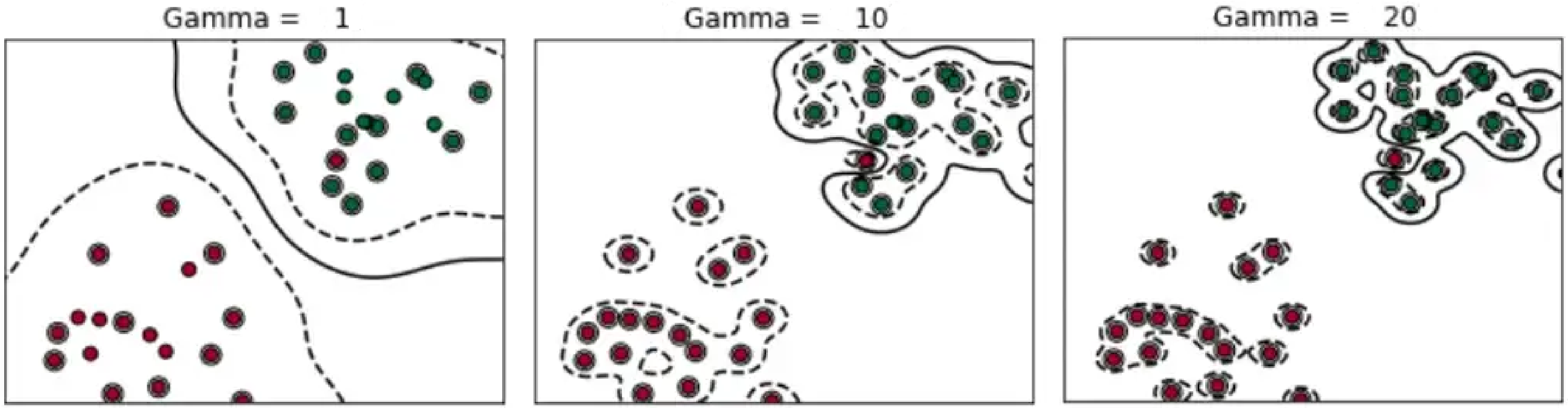
$$(x_{i1}x_{i'1})^2 + (x_{i2}x_{i'2})^2 + 2x_{i1}x_{i'1}x_{i2}x_{i'2}$$

Radial Basis Kernel

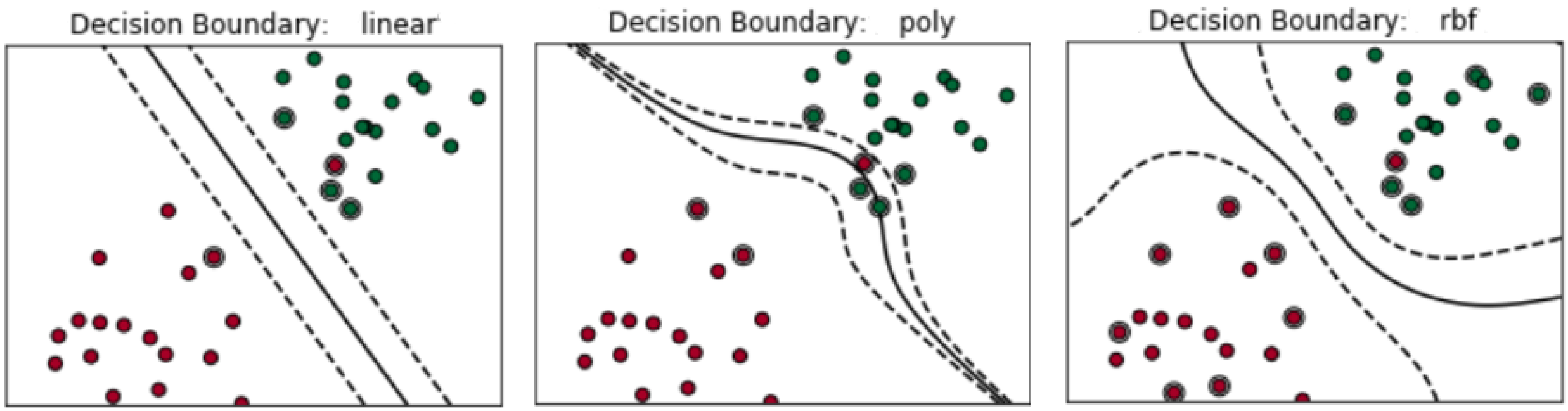
$$K(x_i, x_{i'}) = e^{-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}$$

- For this kernel the *implicit* feature space is infinite dimensional.
- $\gamma > 0$ determines how far the influence of a single observation reaches:
  - Lower γ values yield further reach, with very small values yielding a decision boundary close to linear.
  - Higher γ values yield more localized influence and a more flexible decision boundary.

Radial Basis Kernel



Different Kernels



Predictive model

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

Observations weights are estimated, from which a hyperplane can be derived, but hyperplane is not actually fitted.

Support Vector Machines

- Kernels avoid the need to actually transform to an enlarged space.
  - Only need to compute the kernel and apply the support vector classifier (linear SVM) on this kernel. This is the kernel trick!
- It is important to scale the predictor variables first. Different scalings yield different kernels yield different solutions.
- The most common SVM hyper-parameters to be tuned are C and γ.
- Exponential sequences are typically recommended as candidate values.
- First try only a few values and combinations. This will give a useful indication of whether a broader or more narrow range of values needs to be tried. Using cross validation will guard (but not perfectly) against overoptimism.

Discussion

- SVMs work well for high-dimensional problems: similarity matrix is always $n \times n$.
- For (very) large n, SVMs become computationally very heavy.
- SVMs are great *classifiers*, but do not give probabilities, so not helpful for *statistical* analyses.
- Interpretation difficult or impossible, effects of predictors are not explicitly modeled.