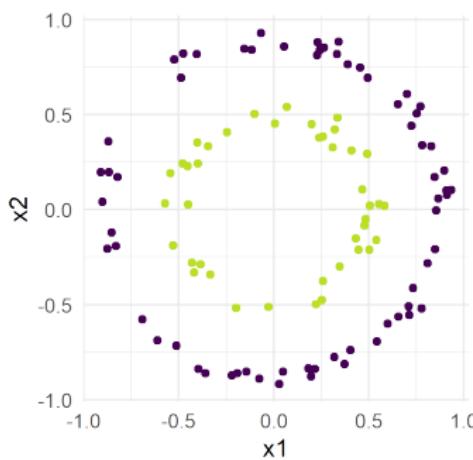


Introduction to Machine Learning

Nonlinear Support Vector Machines Details on Support Vector Machines



Learning goals

- Know that SVMs are non-parameteric models
- Understand the concept of universal consistency
- Know that SVMs with an universal kernel (e.g. Gaussian kernel) are universally consistent

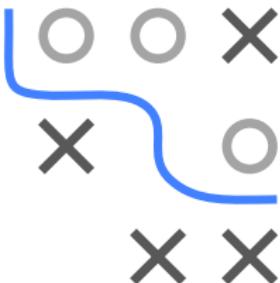


SVMs as Non-Parametric Models



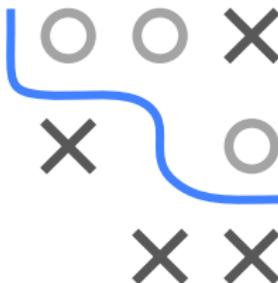
SVMS AS NON-PARAMETRIC MODELS

- In contrast to linear models, for an SVM we do not have to decide the number of coefficients of the decision function before training.
- The number of coefficients depends on the size of the dataset, or on the number of support vectors.
- Such models are called **non-parametric**.
- The big advantage of non-parametric models is that their modeling capacity is not *a priori* restricted to a finite-dimensional subspace of a function space.
- It turns out that SVMs do even better: There exist kernels so that an SVM can model all continuous functions arbitrarily well. It is also known that the SVM learning algorithm can approximate the Bayes optimal decision function arbitrarily well in the limit of infinite data.
- This property is known as **universal consistency**.



ASYMPTOTIC PERFORMANCE

- Convergence of the risk to the Bayes risk for all distributions is called **universal consistency**.
- A universally consistent learning machine can solve all problems optimally, provided enough data.
- Parametric models are too inflexible for this property. They can model only a finite-dimensional subspace (manifold) of decision functions.
- Thus, in the limit of infinite data, they will systematically underfit.
- Universal consistency requires more than infinite-dimensional modeling power: We also need a learning rule that uses the flexibility wisely and avoids overfitting.
- The existence of universally consistent learners is one of the most exciting facts from non-parametric statistics.



SVM – PRO'S & CON'S

Advantages

- Often **sparse** solution (w.r.t. observations)
- Robust against overfitting (**regularized**); especially in high-dimensional space
- **Stable** solutions (w.r.t. changes in train data)
→ Non-SV do not affect decision boundary
- Convex optimization problem
→ local minimum $\hat{=}$ global minimum

Disadvantages

- **Long** training times
 $\rightarrow O(n^2p + n^3)$
- Confined to **linear model**
- Restricted to **continuous features**
- Optimization can also fail or get stuck



Kernels on Infinite-Dimensional Vector Spaces



KERNELS ON INFINITE-DIMENSIONAL VECTOR SPACES

- Note that the input space \mathcal{X} does not need to be a finite-dimensional vector space.
- \mathcal{X} could be the set of all character strings (of unlimited length) or of graphs, or of trees.
- Such data structures are natural representations for, e.g., HTML documents.
- There are many examples of data that do not naturally come in vector form.
- Most often meaningful and cheap-to-compute kernels can be defined directly on the input data structures – they simply define a similarity measure over these data.
- SVMs (and other kernel methods) allow to learn and predict directly on these spaces.

