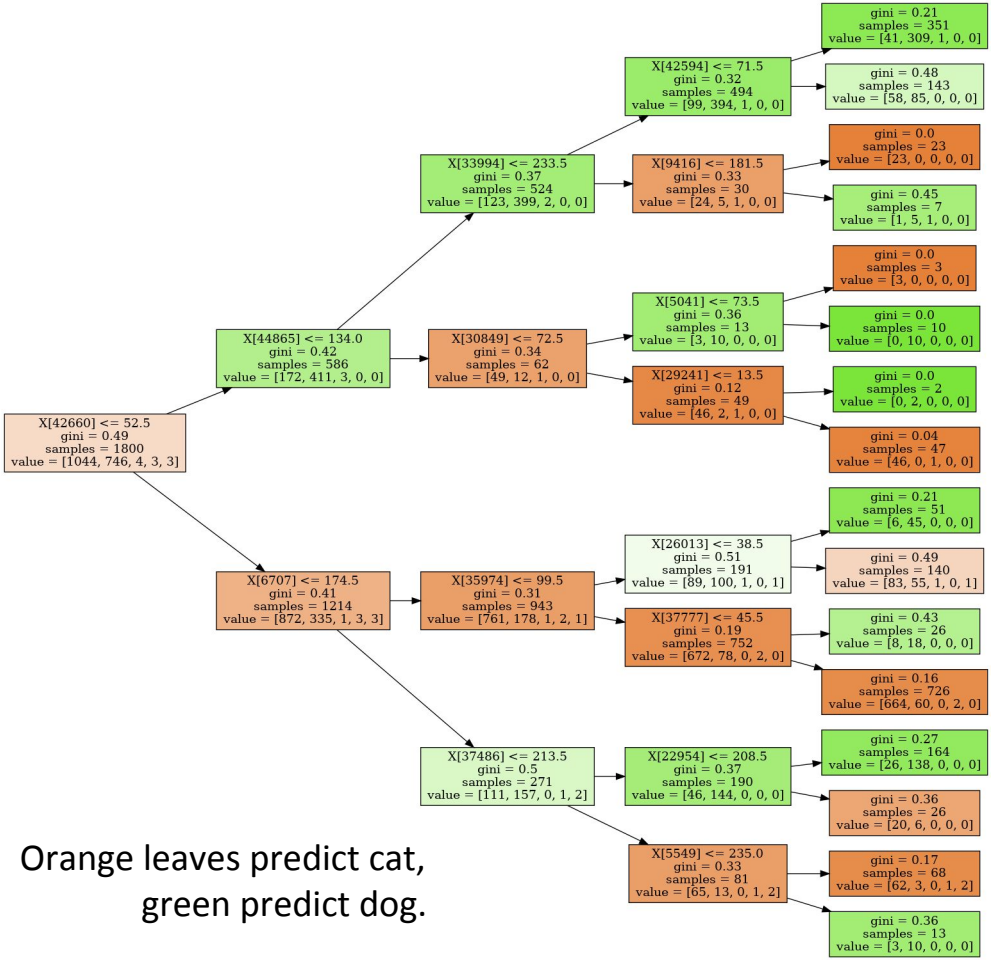
Decision Trees

Decision Trees

- Non-parametric
- Experimentation should be stochastic;

  very order dependent => balance dataset
- Sensitive to small variations in data
  dataset rotation => PCA
- Simple to interpret



Orange leaves predict cat,
green predict dog.

COUNCIL ON UNDERGRADUATE RESEARCH

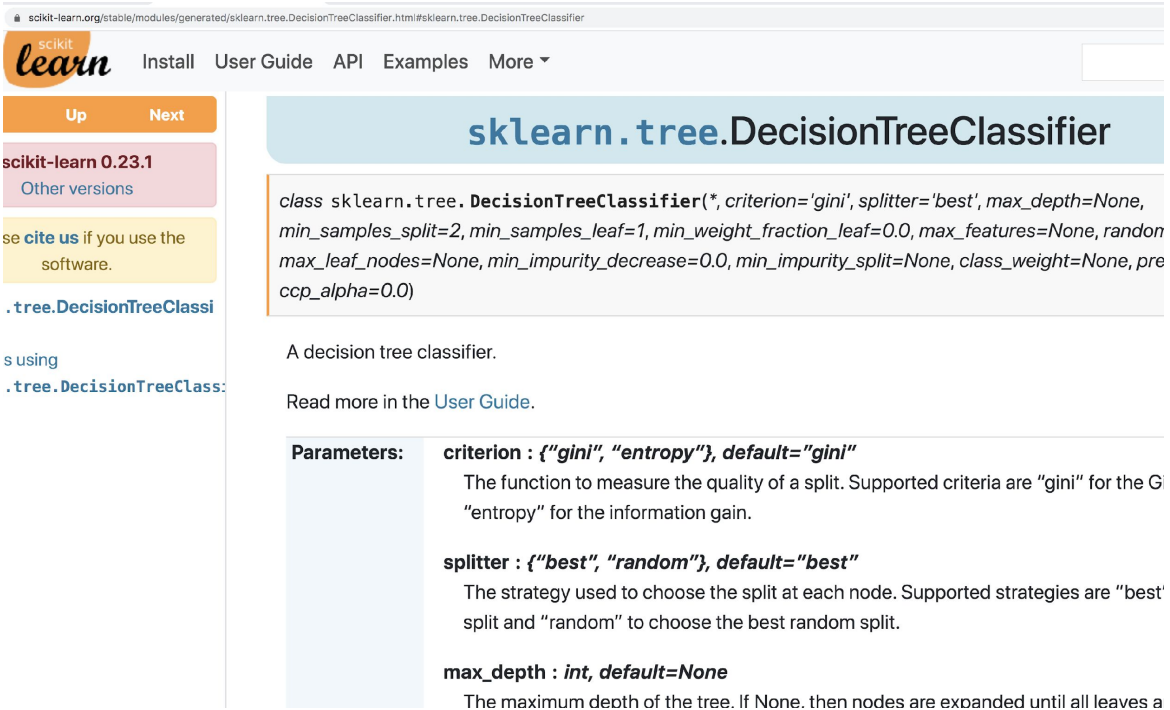
VIRTUAL BIENNIAL CONFERENCE 2020

Scikit's CART Algorithm

Classification and Regression Tree

- Recursively identify (feature,threshold) that best classifies set into two subsets
- Univariate; always creates a binary tree
- Best division minimizes impurity measure

Gini: $H(X_m) = \sum_k p_{mk}(1 - p_{mk})$ Entropy: $H(X_m) = -\sum_k p_{mk} \log(p_{mk})$



(let's examine its parameters on scikit-learn.org)

Support Vector Machines (SVMs)

Machine Learning Models

A learning model should answer several questions:

- What is being learned?
- How is the data being generated? In other words, where does it come from?
- How is the data presented to the learner? For instance, does the learner see all the data at once, or only one example at a time?
- What is the goal of learning in this model?

Supervised vs Unsupervised Learning

- **Supervised learning:**

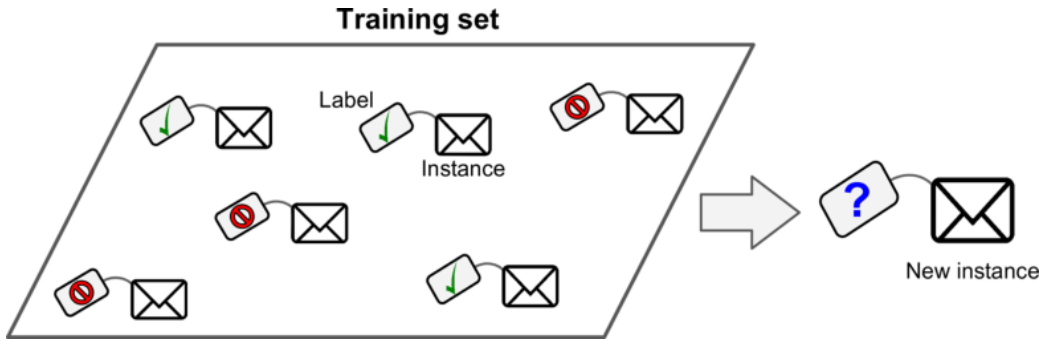
  The training data are accompanied by labels indicating the class of the observations. New data is classified based on the training set.

- **Unsupervised learning:**

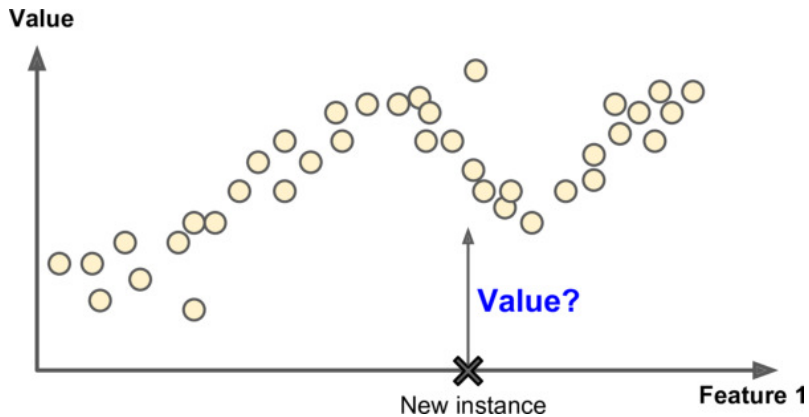
  The class labels of training data is unknown. Given a set of data points, the aim is to establish the existence of classes or clusters in the data.

Supervised Learning

- *Classification* is the process when the data are being used to predict a category (class).
  Two choices - binary classification; more categories - multi-class classification.



- *Compare to Regression:* predicting a value, given an input feature.



(Figures: *Hands-On on ML with Scikit Learn, Keras & TensorFlow*, by A. Géron)

Applications of Classification

- Spam filtering: identify email messages as spam or non-spam
- Medical diagnosis: if a tumor is cancerous or benign
- Fraud detection: identify credit card transactions (for instance) which may be fraudulent in nature
- Weather prediction: predict, for instance, whether or not it will rain tomorrow
- Wilkes Pets dataset: classify animal images. “Is it a cat or a dog?”

Classification

Labeled training instances (training data)

↓

Prediction rule

↓

New instance (testing data)

↓

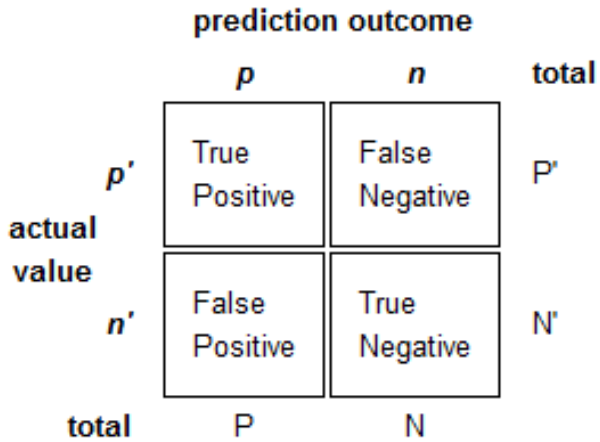
Predicted class of the new instance

- Training instances are the data points we use to create a model.
- The instances are described by a set of attributes also known as features or variables.
- The label is the category (class) we are trying to predict.

Confusion Matrix and Accuracy

- **Confusion Matrix:**

  a specific table that allows visualization of performance of a classification algorithm



		Prediction				
		Class 1	Class 2	Class 3	...	Class n
Actual	Class 1	Accurate				
	Class 2		Accurate			
	Class 3			Accurate		
	...				Accurate	
	Class n					Accurate

- **Accuracy:**

  percentage of correct predictions, e.g.:
  Acc = (TP + TN)/(TP + TN + FP + FN)

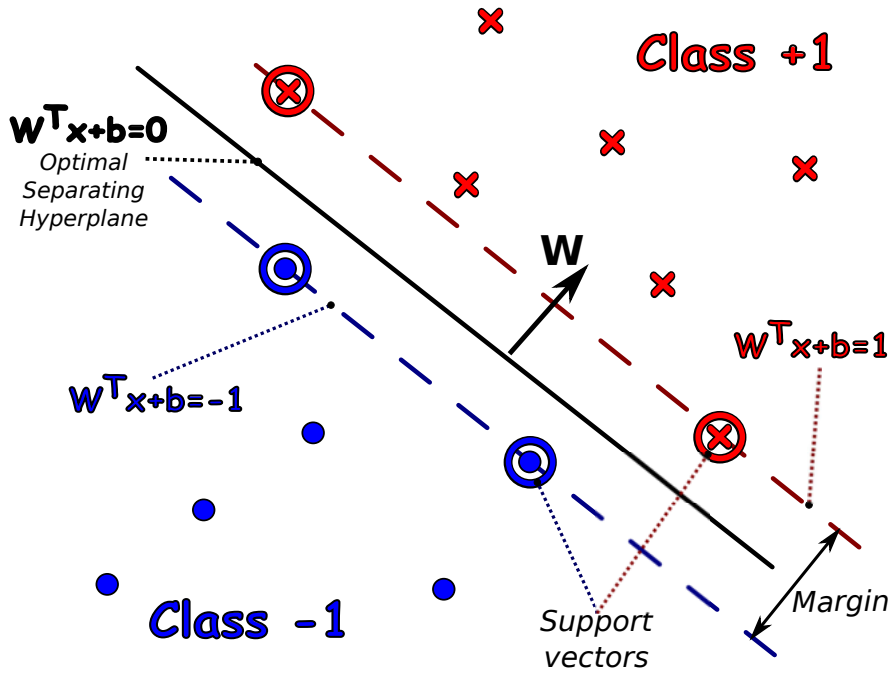
Support Vector Machines (SVMs)

- SVMs are a set of supervised learning methods used for classification and regression.
- Brief history:
  - First mentioned by Vladimir Vapnik in *Estimation of Dependences Based on Empirical Data* (1979).
  - Attracted attention in the 90's: *The Nature of Statistical Learning Theory* by Vapnik, 1995.
  - Since then, SVMs (also called large margin classifiers) have been widely used demonstrating good performance in different applications.

SVM: Linearly Separable Case

Binary problem:

- m vectors $x_i \in \mathbb{R}^N$, each from either class $+1$ or -1.
- m labels $d_i = \{-1, +1\}$.



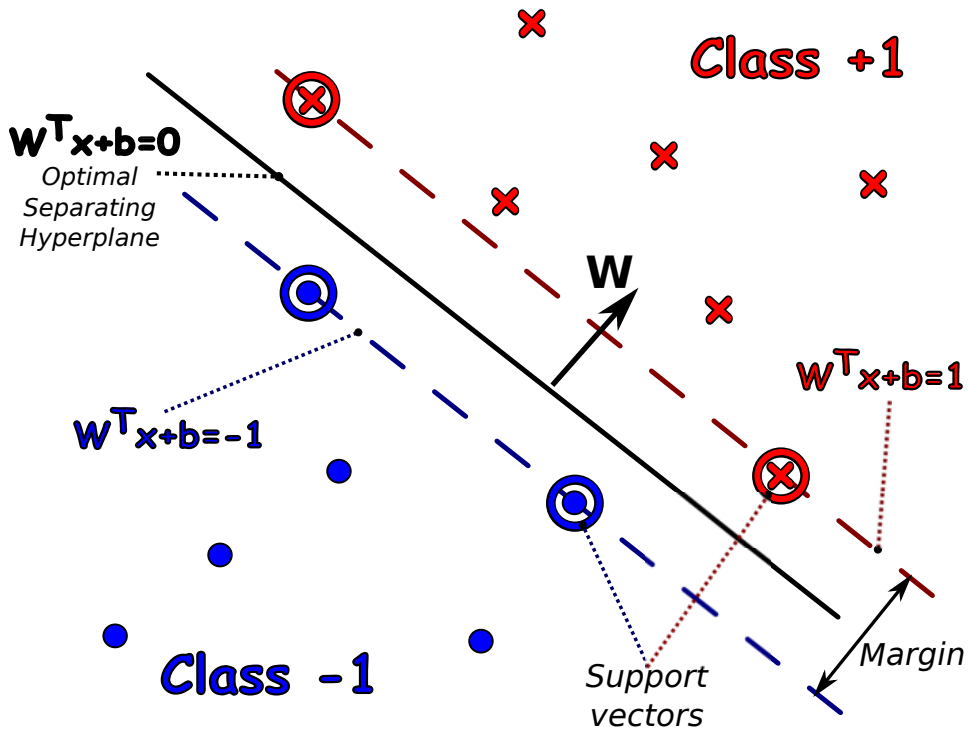
- Separating hyperplane $P = \{x : w^T x + b = 0\}$, $w \in \mathbb{R}^N$ is the normal to P and b is the bias.
- There are many hyperplanes! We need the one for which the smallest perpendicular distance to a training sample is maximized.

SVM: Linearly Separable Case

- Want: optimal hyperplane $\underset{w,b}{argmax}(\min_{1\leq i\leq m} d(x_i, P))$.
- The decision function $f(x) = sgn(w^T x + b)$ assigns a class label to a testing sample x.

SVM: Linearly Separable Case

Support vectors (SVs):
x_i that lie on the canonical hyperplanes $w^T x_i + b = \pm 1$



Margin:
the distance between canonical hyperplanes = $\frac{2}{\|w\|}$.

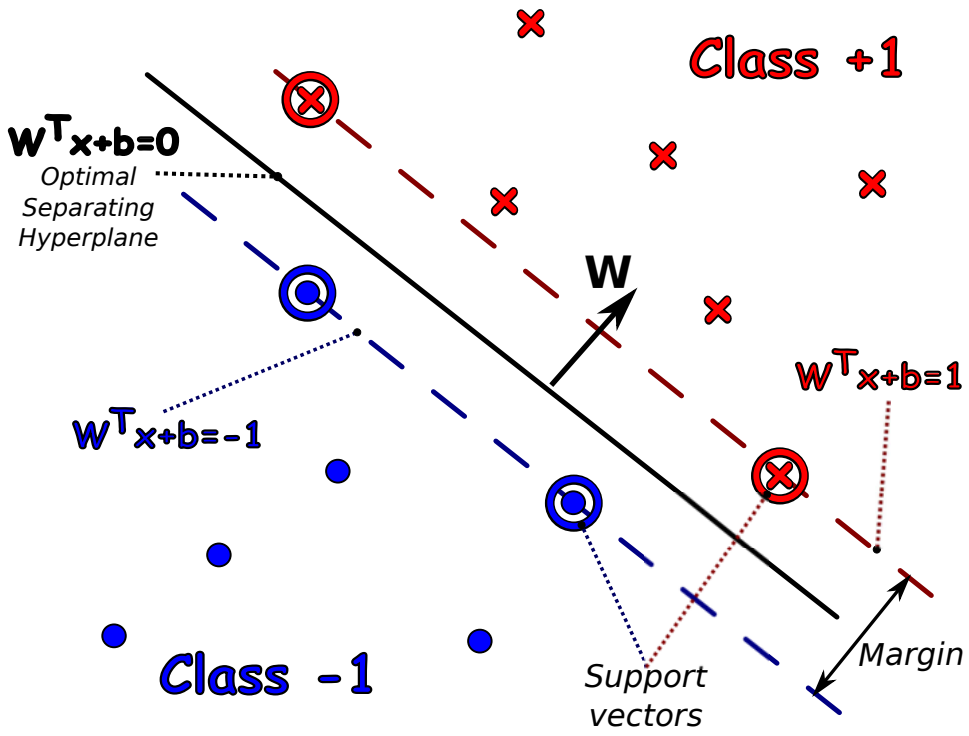
SVM: Linearly Separable Case

Optimization problem:

$$\min_{w,b} \quad \frac{\|w\|_2{}^2}{2}$$
$$\text{subject to } d_i(w^T x_i + b) \geq 1,$$
$$i = 1, \ldots, m$$

In matrix form:

$$\min_{w,b} \quad \frac{\|w\|_2{}^2}{2}$$
$$\text{subject to } D(Xw + be) \geq e$$



Recall: the decision function $f(x) = sgn(w^T x + b)$ assigns a class label to a testing sample x.

Nonseparable Case: Soft-Margin SVM

Optimization problem:

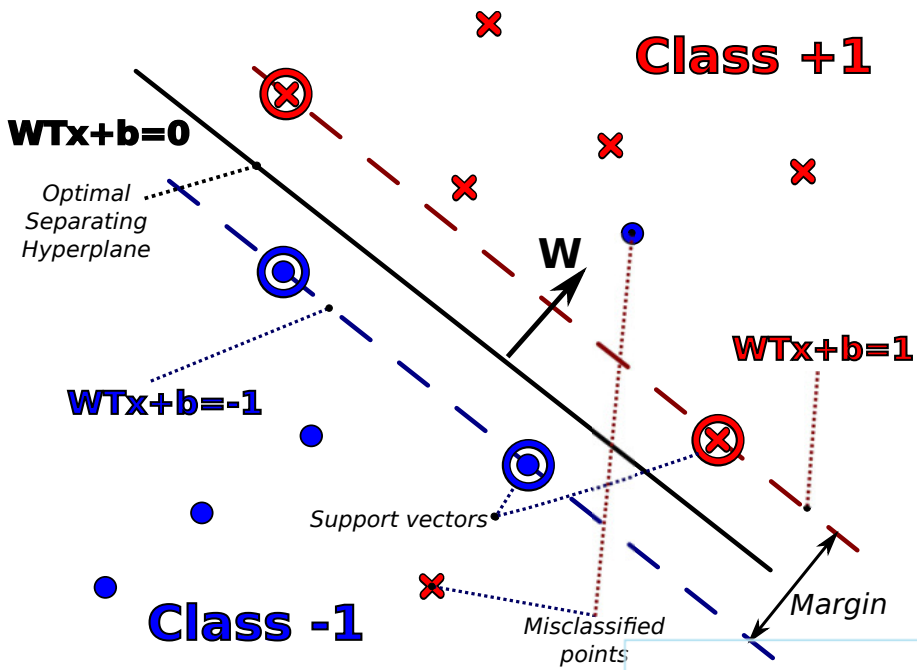
$$\min_{w,b} \quad \frac{\|w\|_2^2}{2} + \boxed{C\sum_{i=1}^{m}\xi_i}$$

subject to $d_i(w^T x_i + b) \geq 1 - \xi_i$,
$\xi_i \geq 0, i = 1, \ldots, m.$

In matrix form:

$$\min_{w,b,\xi} \quad \frac{\|w\|_2^2}{2} + \boxed{Ce^T\xi}$$

subject to $D(Xw + be) \geq e - \xi$,
$\xi \geq 0.$



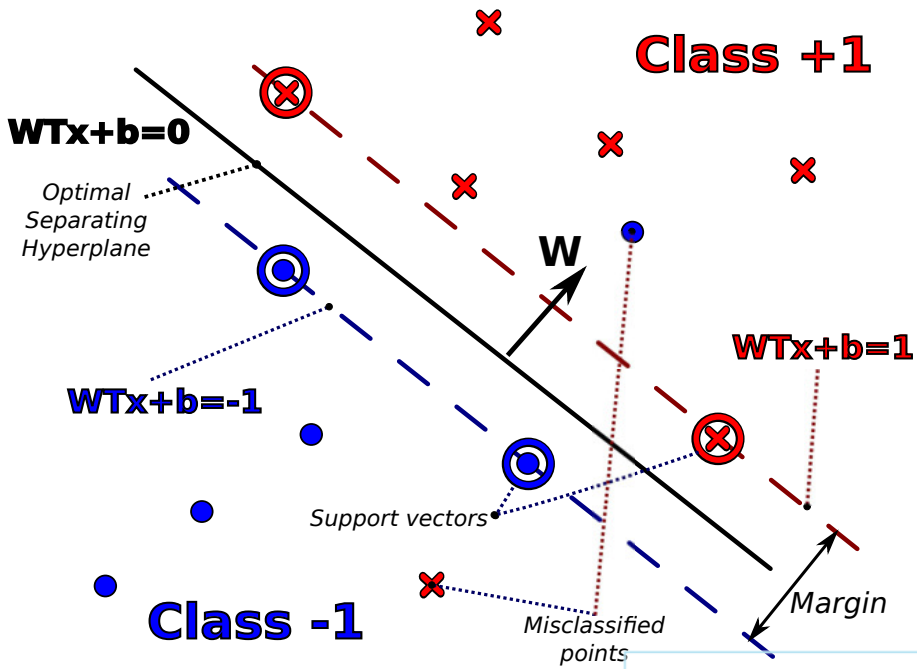
Nonseparable Case: Soft-Margin SVM

Primal Problem:

$$\min_{w,b,\xi} \quad \frac{\|w\|_2{}^2}{2} + Ce^T\xi$$
$$\text{subject to } D(Xw + be) \geq e - \xi,$$
$$\xi \geq 0.$$

Dual Problem:

$$\max_{\alpha} \quad e^T\alpha - \tfrac{1}{2}\alpha^T DXX^T D\alpha$$
$$\text{subject to } e^T D\alpha = 0,$$
$$0 \leq \alpha \leq Ce.$$

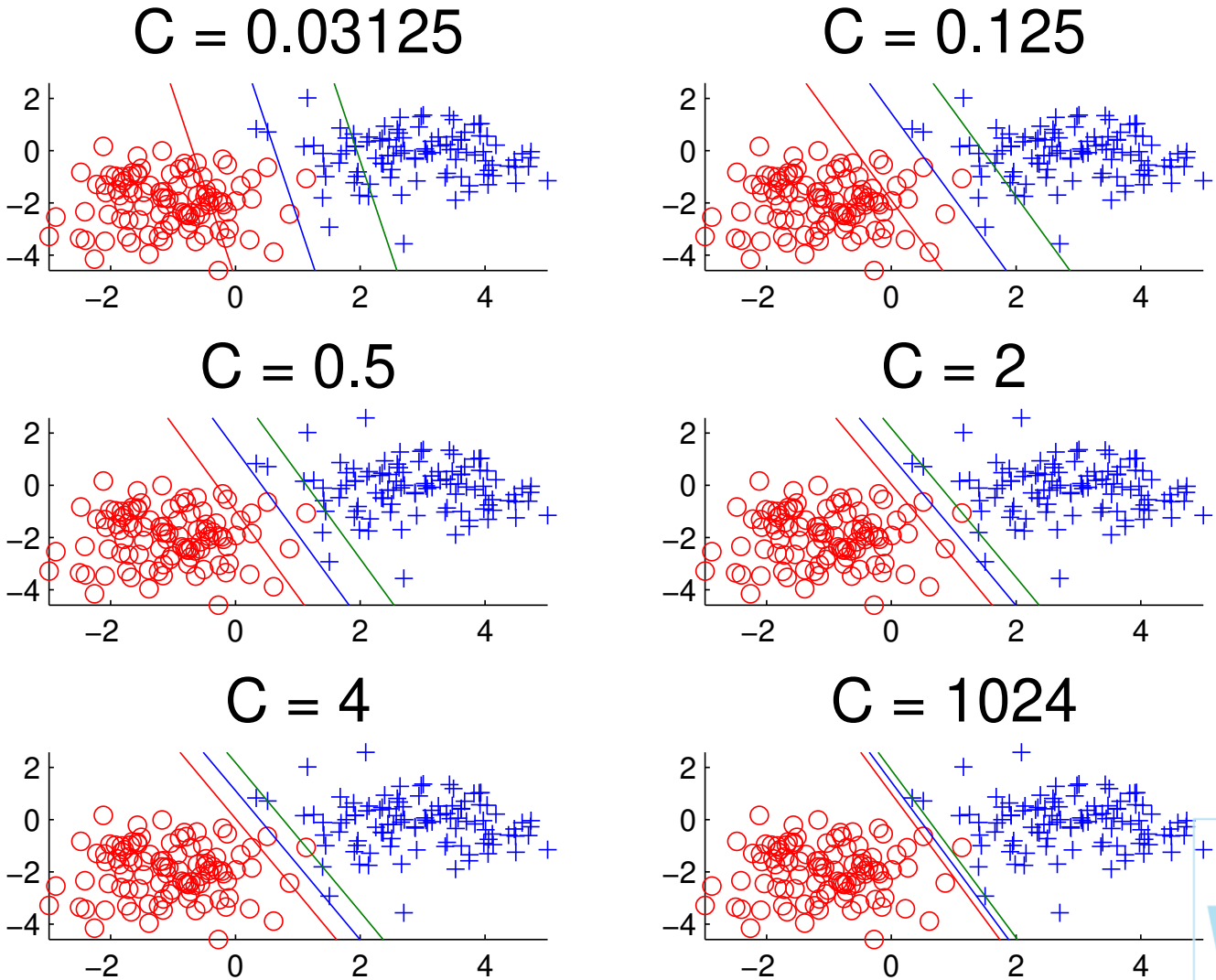


Nonseparable Case: Soft-Margin SVM

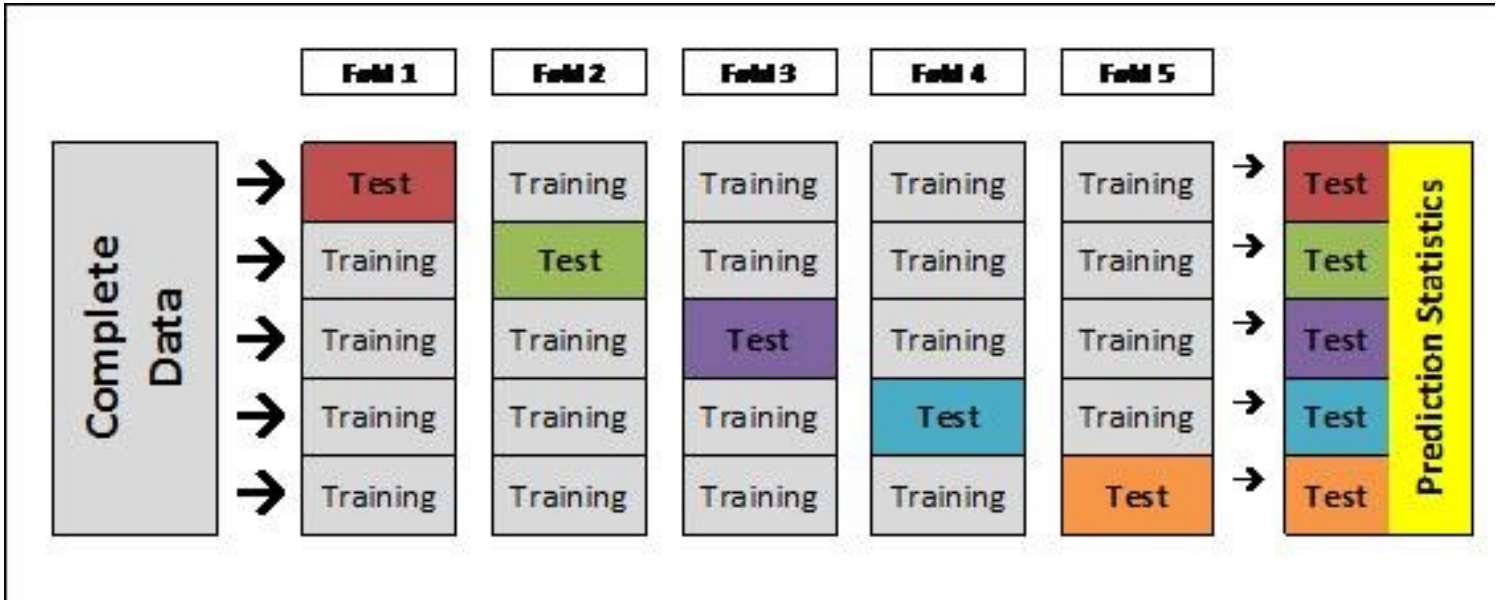
- **Support vectors:**
  - The solution is given by $w = \sum_{i=1}^{m} \alpha_i d_i x_i$ (b can be computed using w and any training point).
  - Support vectors: $\xi_i = 0$ and $0 \leq \alpha_i \leq C$.

- **Hyperparameter C:**
  - $C = \infty$ corresponds to the hard-margin SVM (separable case).
  - C is finite: larger C produces less errors, margin shrinks; smaller C maximizes the margin, more errors.
  - C can be found by cross validation procedure.

Example: Hyperparameter C for Toy Data

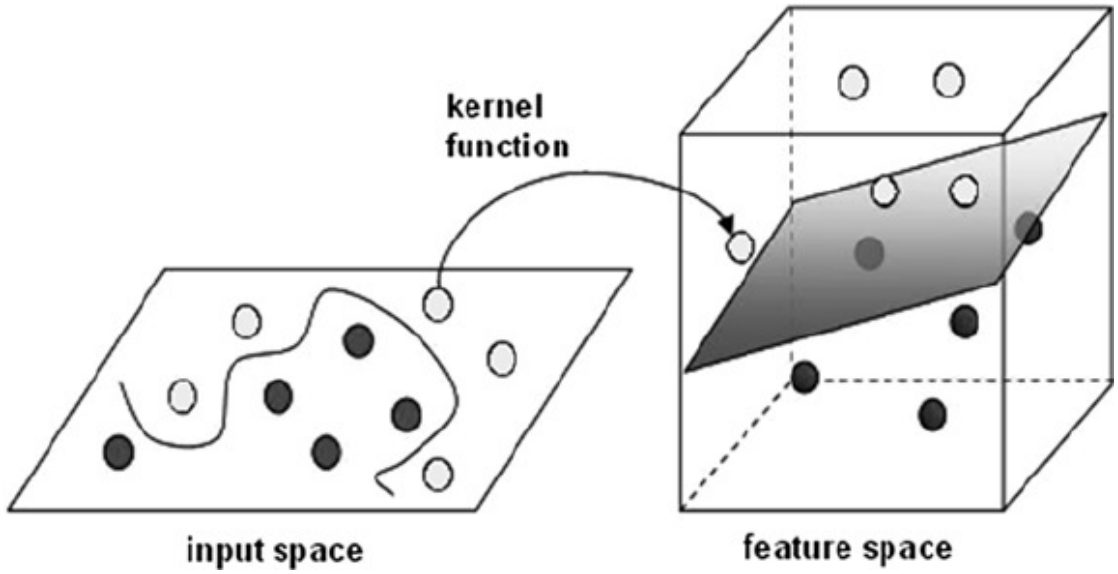


Cross Validation



Set a range of feasible values for C, for instance, take C values in the interval $[1, 15]$. Then make a search in laps of 1: 1,2,3,4,...,15. Look for the average error (accuracy) using a 5- or 10-fold cross validation and keep the best SVM parameter => your model.

Nonlinear SVM: Kernel Trick



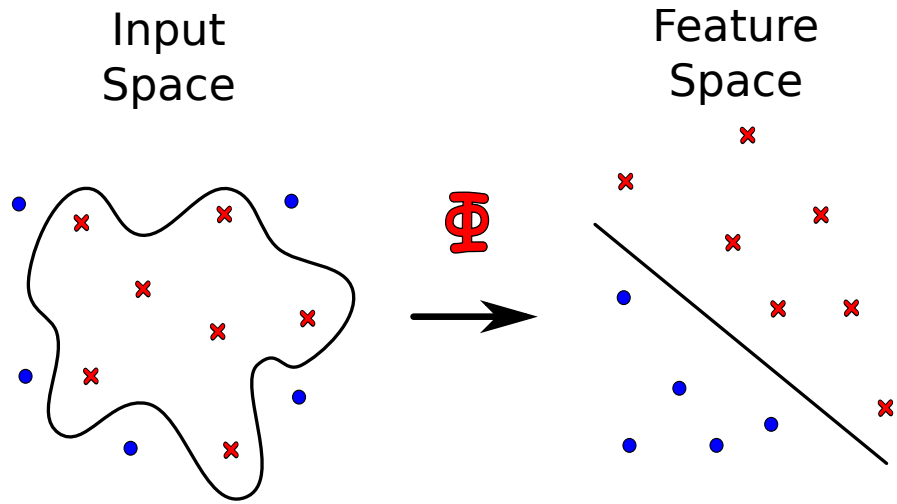
Kernel trick: a way of computing the dot product of two vectors x and y in some (possibly very high dimensional) feature space, so often a kernel function is called "generalized dot product" $K(x, y) = \Phi(x)^T \Phi(y)$, where ϕ is a mapping from the original data space to a feature space.

Nonlinear SVM: Kernel Trick

- $\Phi : x \in \mathbb{R}^N \mapsto \Phi(x) \in \mathbb{R}^{N'}, N' > N.$
- Kernel function $K_{ij} = K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j).$

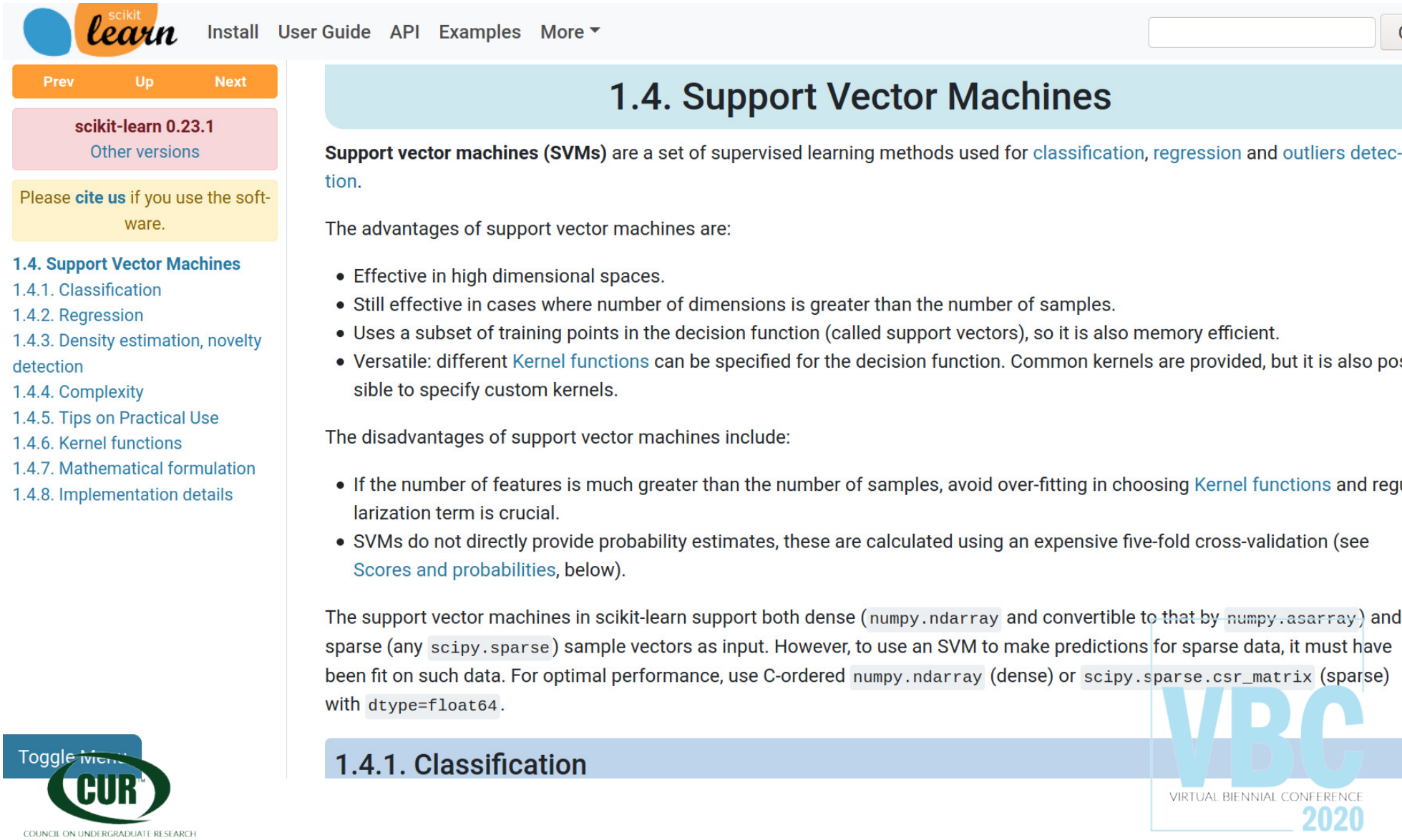
Dual Problem:

$$\begin{aligned} &\max_\alpha \quad e^T\alpha - \tfrac{1}{2}\alpha^T DKD\alpha \\ &\text{subject to } e^T D\alpha = 0, \\ &0 \le \alpha \le Ce. \end{aligned}$$



- the decision function is $f(x) = sgn(\sum_i^m \alpha_i d_i K(x_i, x) + b).$
- Gaussian RBF $K(x_i, x) = \exp(-\gamma \|x_i - x\|^2)$,
  polynomial $K(x_i, x) = (ax_i^T x + c)^d.$

SVM in Scikit-Learn



scikit learn Install User Guide API Examples More

Prev Up Next

scikit-learn 0.23.1
Other versions

Please cite us if you use the software.

1.4. Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

The support vector machines in scikit-learn support both dense (`numpy.ndarray` and convertible to that by `numpy.asarray`) and sparse (any `scipy.sparse`) sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data. For optimal performance, use C-ordered `numpy.ndarray` (dense) or `scipy.sparse.csr_matrix` (sparse) with `dtype=float64`.

1.4.1. Classification

Toggle Menu

References

- "A Tutorial on Support Vector Machines for Pattern Recognition" by Christopher Burges
- "A User's Guide to Support Vector Machines" by Asa Ben-Hur and Jason Weston
- Chapter 7 of "Pattern Recognition" by Christopher Bishop
- SVMs in Python (Scikit Learn): `http://scikit-learn.org/stable/modules/svm.html`
- Cross validation in Python (Scikit Learn): `https://scikit-learn.org/stable/modules/cross_validation.html`

Live Session III

(pause video here)

CUR™
COUNCIL ON UNDERGRADUATE RESEARCH

VBC
VIRTUAL BIENNIAL CONFERENCE
2020