Computational Data Analysis

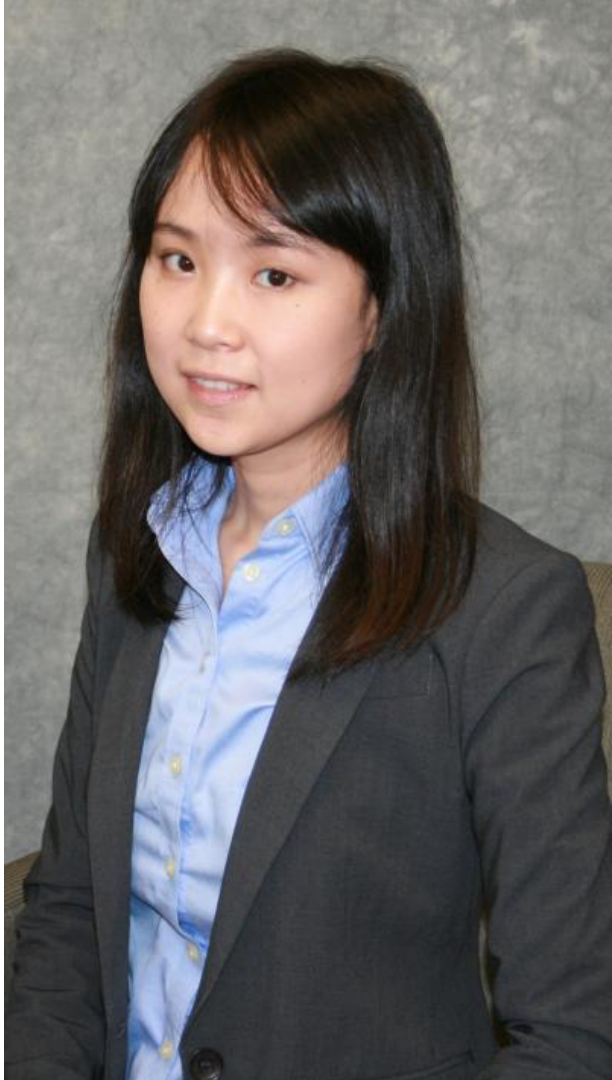
Machine Learning

Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering



Support Vector Machine (SVM)

Main types approaches to design classifiers

- Bayes rule + assumption for $p(x|y=1)$
  - Assume $p(x|y=1)$ is Gaussian
  - Assume $p(x|y=1)$ is fully factorized

- Use geometric intuitions
  - k-nearest neighbor classifier
  - Support vector machine

- Directly go for the decision boundary $\mathrm{h(x)} = -\ln \frac{q_i(x)}{q_j(x)}$
  - Logistic regression
  - Neural networks

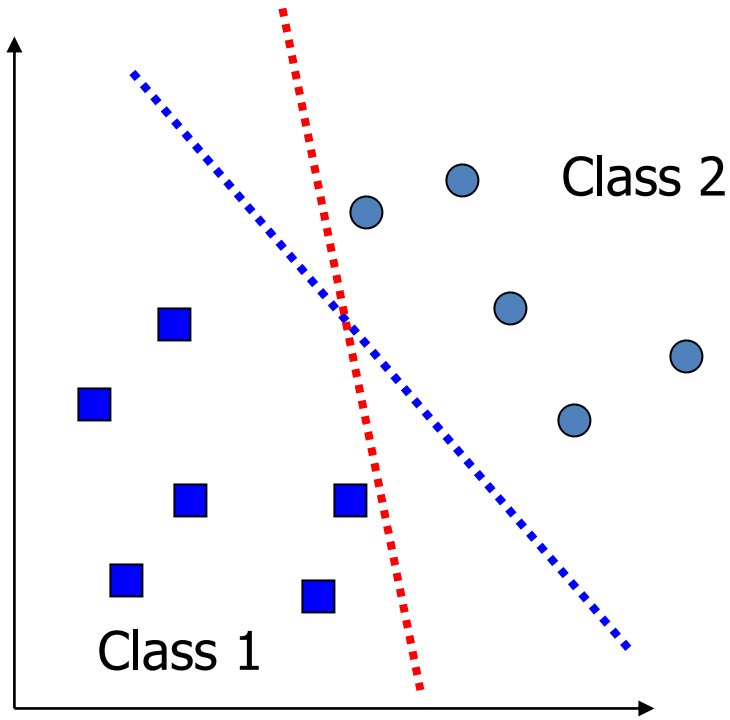
Which decision boundary is better?

Suppose the training samples are linearly separable

We can find a decision boundary which gives zero training error

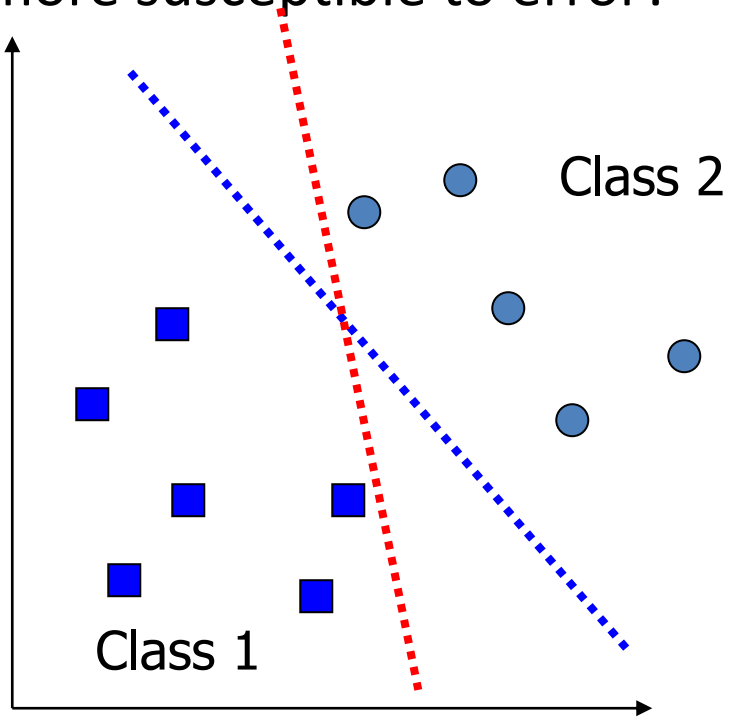
But there are many such decision boundaries

Which one is better?



Compare two decision boundaries

Generalization error: Suppose we perturb data (new test data), which boundary is more susceptible to error?



Support vector machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

An SVM maximize the "gap" of training samples from the decision boundary to be as wide as possible.

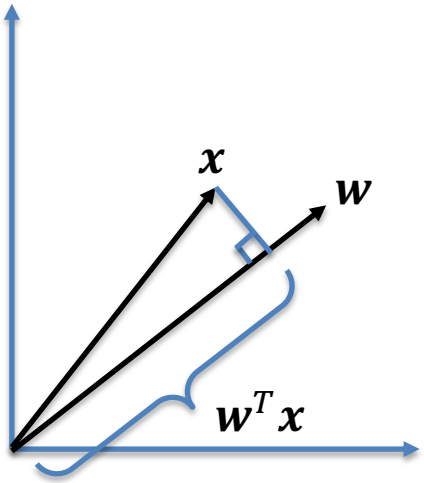
Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297.

Cortes, Corinna

Vladimir Naumovich Vapnik

Background: Projection in vector space

Projection $\boldsymbol{w}^T\boldsymbol{x}$



Projection

$$\boldsymbol{w}^T\boldsymbol{x} = \left(\frac{\boldsymbol{w}}{||\boldsymbol{w}||}\right)^T \boldsymbol{x} \cdot ||\boldsymbol{w}|| = \boldsymbol{w}'^T\,\boldsymbol{x} \cdot ||\boldsymbol{w}||$$

First project along a direction $\boldsymbol{w}'$

Then stretch by magnitude $||\boldsymbol{w}||$

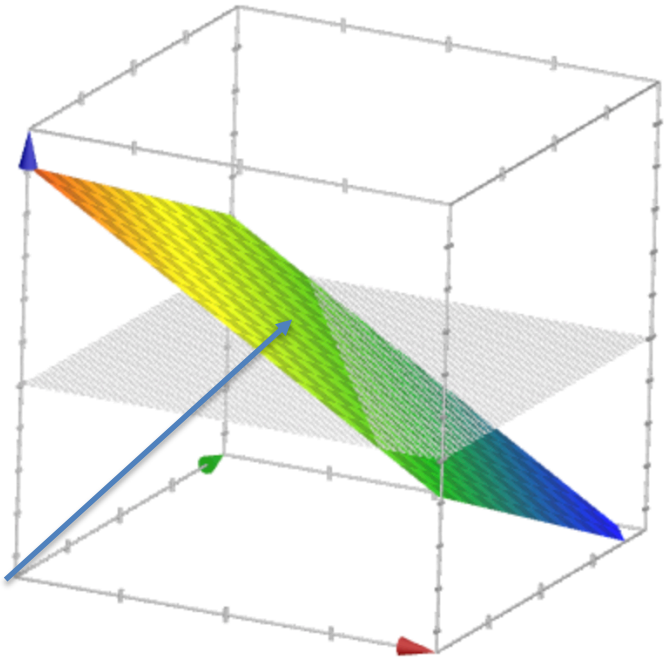
Background: Hyperplane

Hyperplane representation w

$$z = -3x - 2y + 1$$

$$\underbrace{3x + 2y + z}_{\boldsymbol{w}^T \boldsymbol{x}} - 1 = 0 \qquad + b = 0$$

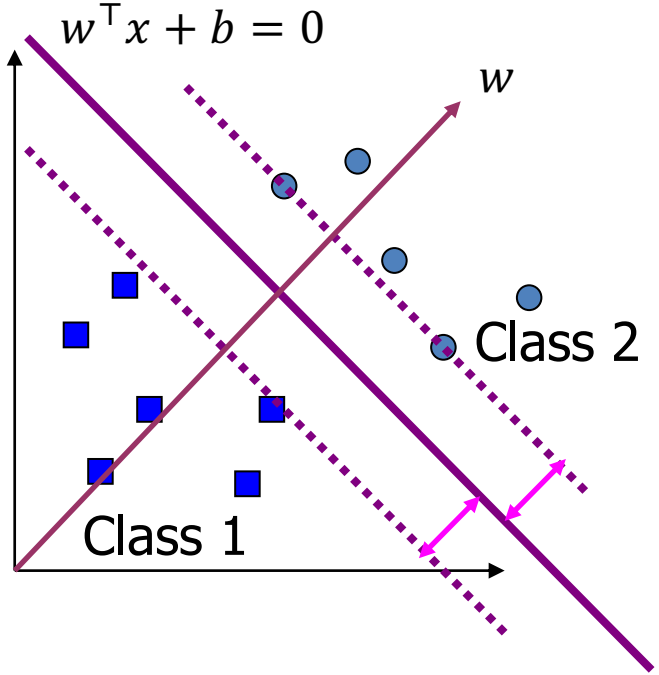
$$\boldsymbol{w} = [3, 2, 1]^{\mathrm{T}}, \boldsymbol{x} = [x, y, z]^{\mathrm{T}}, b = -1$$



↘	From	To
x	-10.0000	10.0000
y	-10.0000	10.0000
z	-44.9785	59.6919

Geometric interpretation of a classifier

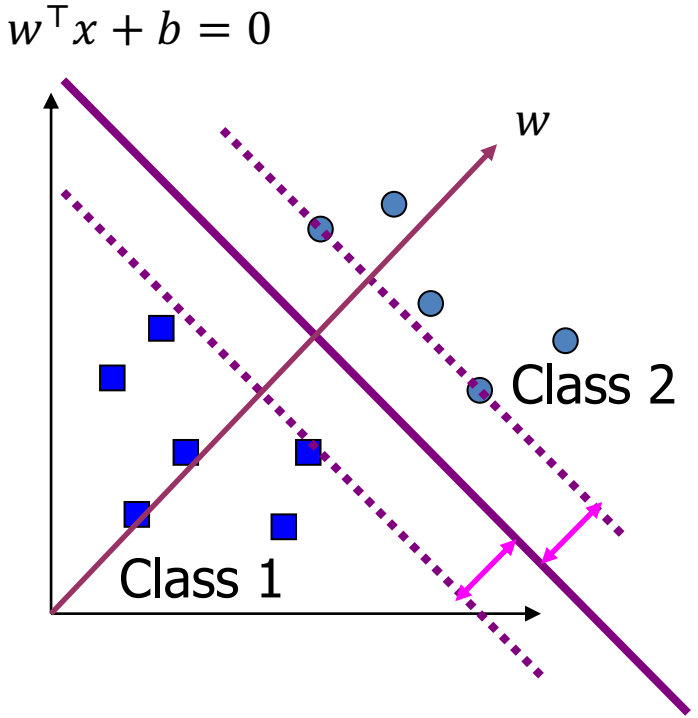
- Parameterizing decision boundary as: $w^\top x + b = 0$
  - w denotes a **vector orthogonal** to the decision boundary
  - b is a **scalar offset** term
- Dash lines are parallel to decision boundary and they just hit the data points



Constraints on data points: correct separation

- Constraints on data points
  - For all x in class 2, $y = 1$
    $$w^\top x + b \geq c$$
  - For all x in class 1, $y = -1$
    $$w^\top x + b \leq -c$$

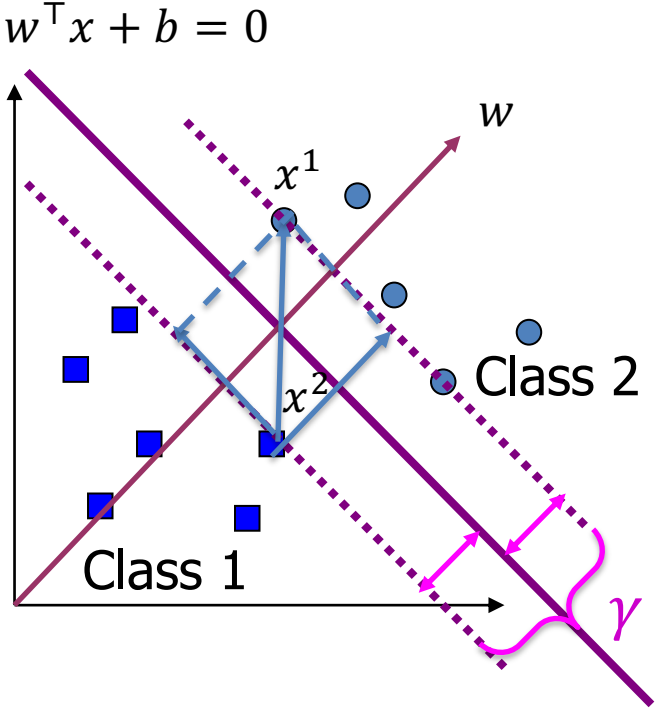
- Or more compactly
  $$(w^\top x + b)y \geq c$$



Classifier margin

- Pick two data points x^1 and x^2 which are on each dash line respectively

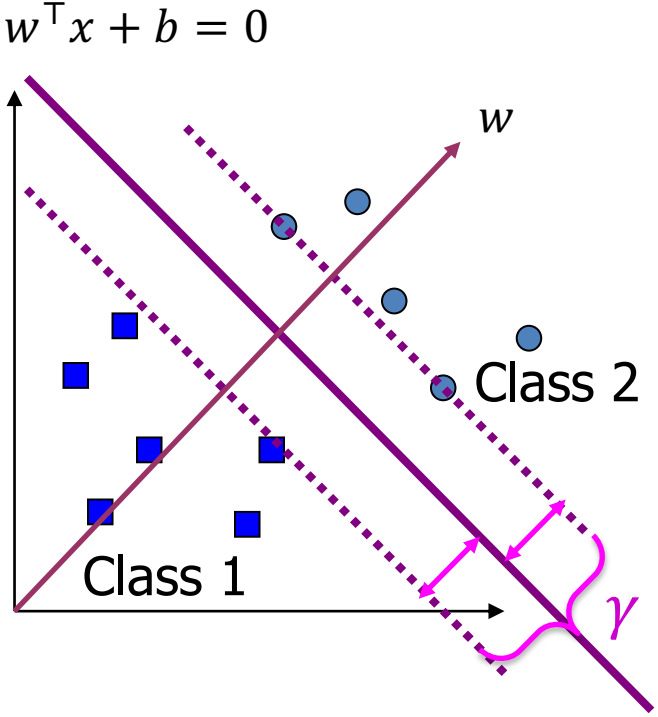
- The unnormalized margin is
$$w^\top (x^1 - x^2) := 2c$$
- On the other hand, we can show
$$w^\top (x^1 - x^2) = \gamma \|w\|$$
- The margin is $\gamma = \frac{2c}{\|w\|}$



Maximum margin classifier

Find decision boundary w as far from data point as possible

$$\max_{w,b} \ \gamma = \frac{2c}{||w||}$$
$$s.t.\ y^i\left(w^\top x^i + b\right) \geq c, \forall i$$



Equivalent form

$$\max_{w,b} \frac{2c}{\|w\|}$$
$$\text{subject to } y^i\left(w^\top x^i + b\right) \geq c, \forall i$$

- Note that the magnitude of c merely scales w and b, and does not change the relative goodness of different classifiers
- Set $c = 1$ (and drop the 2) to get a cleaner problem

$$\max_{w,b} \frac{1}{\|w\|}$$
$$s.t.\ y^i\left(w^\top x^i + b\right) \geq 1, \forall i$$

Support vector machines

- A constrained convex quadratic programming problem

$$\min_{w,b} \ \|w\|^2$$
$$\text{subject to} \ \ y^i\left(w^\top x^i + b\right) \geq 1, \forall i$$

- Only a few of the constraints are relevant → support vectors

- Kernel methods are introduced for nonlinear classification problem

Solving constrained optimization problem

Lagrangian Duality

The primal problem

$$\min_{w} f(w)$$
$$st.\, g_i(w) \leq 0, i = 1, \dots, k$$
$$h_i(w) = 0, i = 1, \dots, l$$

The Lagrangian function

$$L(w, \alpha, \beta) = f(w) + \sum_{i}^{k} \alpha_i g_i(w) + \sum_{i}^{l} \beta_i h_i(w)$$

$\alpha_i \geq 0$, and β_i are called the Lagrangian multipliers

The KKT conditions

If there exists some saddle point of L, then the saddle point satisfies the following "Karush-Kuhn-Tucker" (KKT) conditions:

$$\frac{\partial L}{\partial w} = 0$$

$$\frac{\partial L}{\partial \alpha} = 0$$

$$\frac{\partial L}{\partial \beta} = 0$$

$$g_i(w) \leq 0$$

$$h_i(w) = 0$$

$$\alpha_i \geq 0$$

$$\alpha_i g_i(w) = 0$$

Dual problem of support vector machines

$$\min_{w,b} \left|\left|w\right|\right|^2$$
$$s.t.\ y^i\left(w^\top x^i + b\right) \geq 1, \forall i$$

Convert to standard form

$$\min_{w,b} \frac{1}{2} w^\top w$$
$$s.t.\ 1 - y^i\left(w^\top x^i + b\right) \leq 0, \forall i$$

The Lagrangian function

$$L(w, b, \alpha) = \frac{1}{2} w^\top w + \sum_{i=1}^{m} \alpha_i \left(1 - y^i\left(w^\top x^i + b\right)\right)$$

Deriving the dual problem

$$L(w, b, \alpha) = \frac{1}{2} w^\top w + \sum_{i=1}^{m} \alpha_i \left(1 - y^i \left(w^\top x^i + b\right)\right)$$

Taking derivative and set to zero

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y^i x^i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y^i = 0$$

Plug back relation of w and b

$$L(w,b,\alpha) = \frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y^i x^i\right)^{\top}\left(\sum_{j=1}^{m}\alpha_j y^j x^j\right)$$
$$+\sum_{i=1}^{m}\alpha_i\left(1 - y^i\left(\left(\sum_{j=1}^{m}\alpha_j y^j x^j\right)^{\top} x^i + b\right)\right)$$

After simplification (plug in the optimal w^* and b^*):

$$L(w^*,b^*,\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y^i y^j\left(x^{i\top} x^j\right)$$

The dual problem of SVM

$$\max_{\alpha} L(w^*, b^*, \alpha) = \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y^i y^j \left(x^{i\top} x^j \right)$$

$$s.t. \alpha_i \geq 0, i = 1, \dots, m$$

$$\sum_{i}^{m} \alpha_i y^i = 0$$

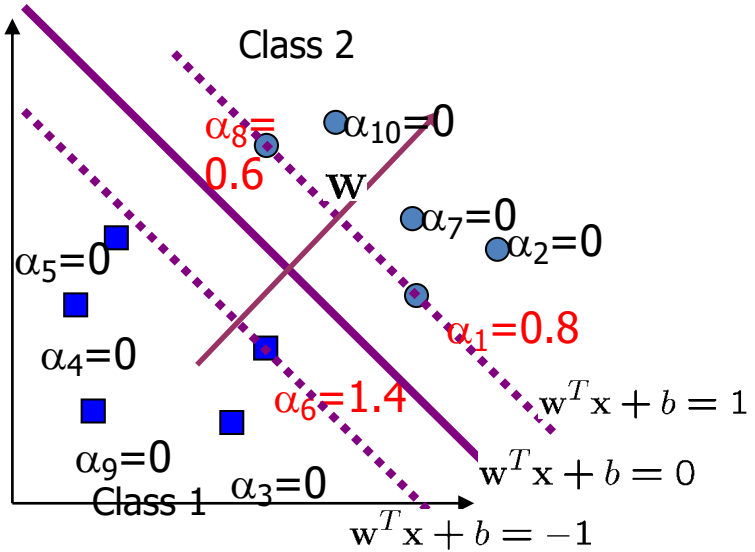
- This is a constrained **quadratic programming**
- Nice and convex, and global maximum can be found
- Very similar to the dual of minimum enclosing ball problem

Properties of SVM

Support vectors

Note that the KKT condition $\alpha_i g_i(w) = 0$: $\alpha_i \left(1 - y^i\left(w^\top x^i + b\right)\right) = 0$

- For data points with $\left(1 - y^i\left(w^\top x^i + b\right)\right) < 0$, $\alpha_i = 0$
- For data points with $\left(1 - y^i\left(w^\top x^i + b\right)\right) = 0$, $\alpha_i \geq 0$



Call the training data points whose α_i's are nonzero the **support vectors** (SV)

Computing b and obtain the classifer

- Pick any data point with $\alpha_i > 0$, solve for b with

$$1 - y^i\left(w^\top x^i + b\right) = 0$$

- One KKT condition: $\frac{\partial L}{\partial w} = 0$

$$w = \sum_{i=1}^{m} \alpha_i y^i x^i$$

- For a new test point z
  - Compute

$$w^\top z + b = \sum_{i \in \text{support vectors}} \alpha_i y^i \left(x^{i^T} z\right) + b$$

  - Classify z as class 1 if the $w^\top z + b > 0$, and class 2 otherwise

Interpretation of support vector machines

The optimal w is a linear combination of a small number of data points. This "sparse" representation can be viewed as data compression

To compute the weights α_i, and to use support vector machines we need to specify only the inner products (or kernel) between the examples $x^{i^\top} x^j$

We make decisions by comparing each new example z with only the support vectors:

$$y^* = sign\left(\sum_{i \in \text{support vectors}} \alpha_i y^i \left(x^{i^T} z\right) + b\right)$$

Comparison

	Bayes classifier (Gaussian)	KNN	Logistic regression	SVM
Number of parameters	$O(n^2)$	1 (non-parametric)	$O(n)$	$O(n)$
Robustness to model mismatch	No	Yes	relatively	relatively
Noisy prediction?	No	Yes	Yes (when logistic function value is close to 0.5)	No

Extensions of SVM

- Kernelized SVM: In addition to performing linear classification, SVMs can perform a non-linear classification (non-linear decision boundary) by kernelized SVM implicitly mapping their inputs into high-dimensional feature spaces.

- Soft-margin SVM: allow a few data points to be mis-classified.

- Multi-class SVM

Kernelized SVM

$$\max_{\alpha} L(w^*, b^*, \alpha) = \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y^i y^j \left(x^{i\top} x^j \right)$$

$$s.t. \alpha_i \geq 0, i = 1, \dots, m$$

$$\sum_{i}^{m} \alpha_i y^i = 0$$

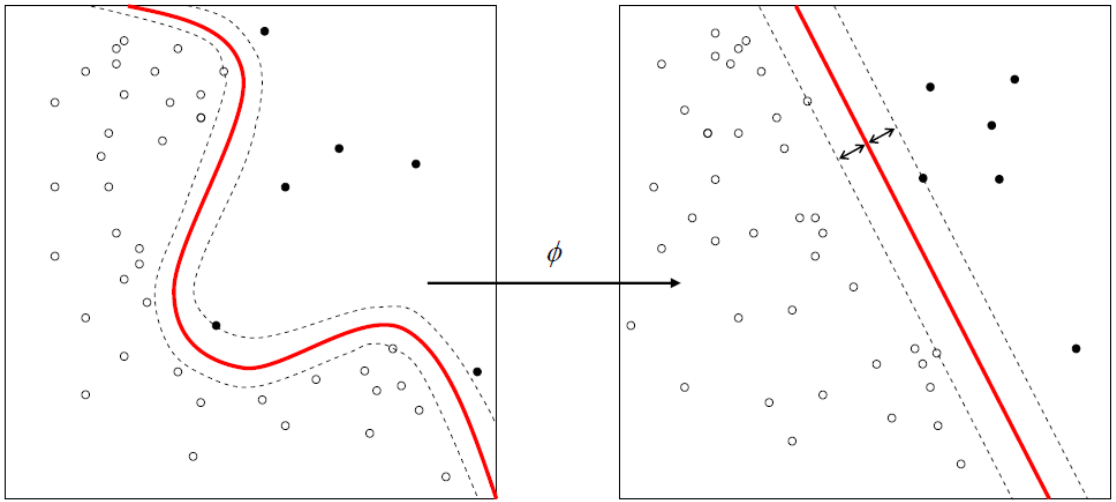
Replaced by $k(x^i, x^j)$

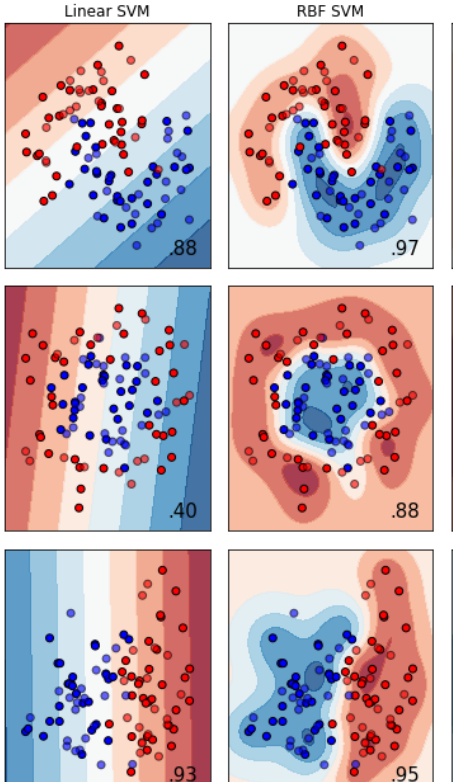
e.g. Gaussian RBF kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2} \right)$$

- Implicit map data to non-linear feature space

Implicitly map data to non-linear feature space



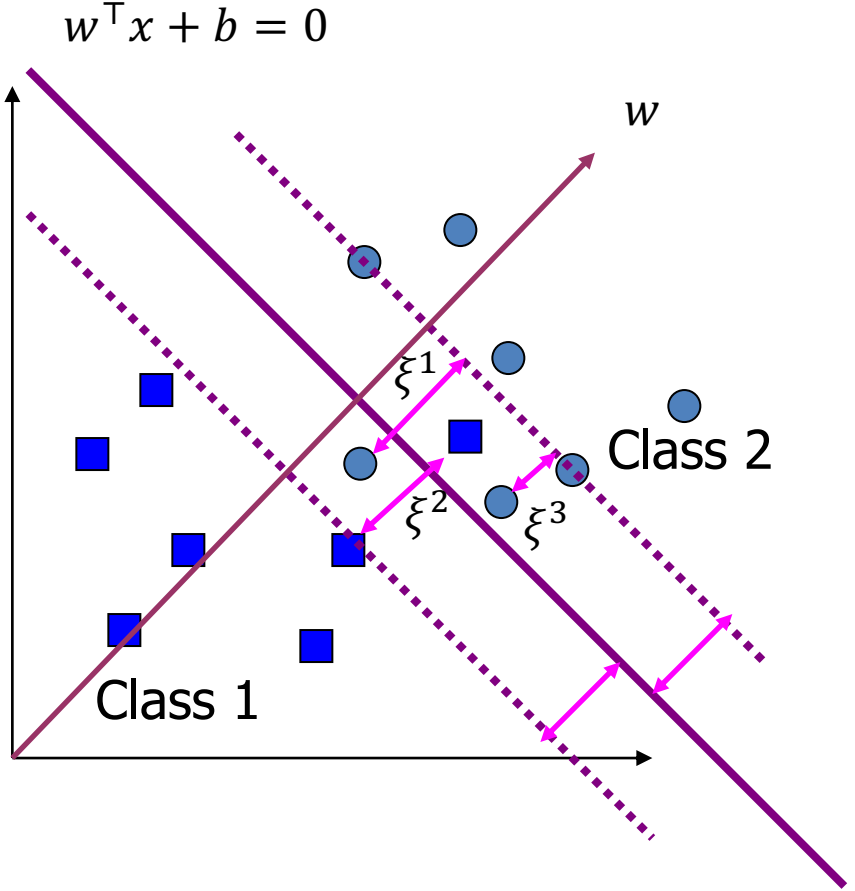


Soft-margin SVM

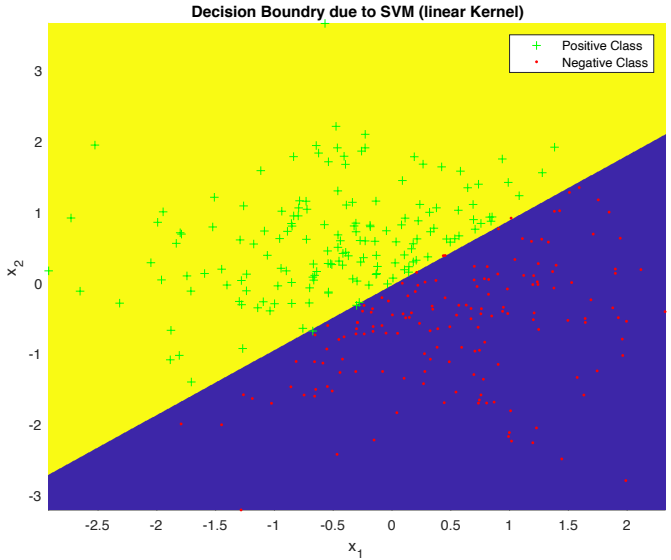
- What if the data is not linearly separable?
- We will allow a few points to violate the hard margin constraint
- $(w^\top x + b)y \geq 1 - \xi$

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

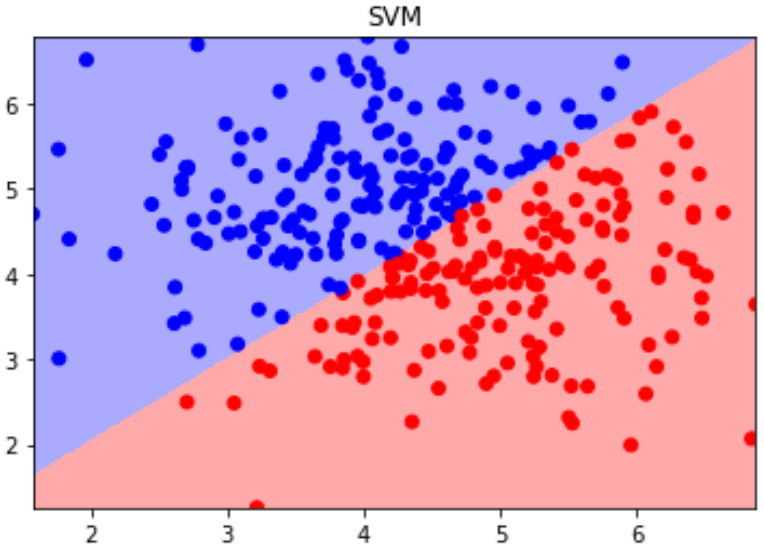
$$s.t.\ y^i\left(w^\top x^i + b\right) \geq 1 - \xi^i, \xi^i \geq 0, \forall i$$



Demo



MATLAB



Python

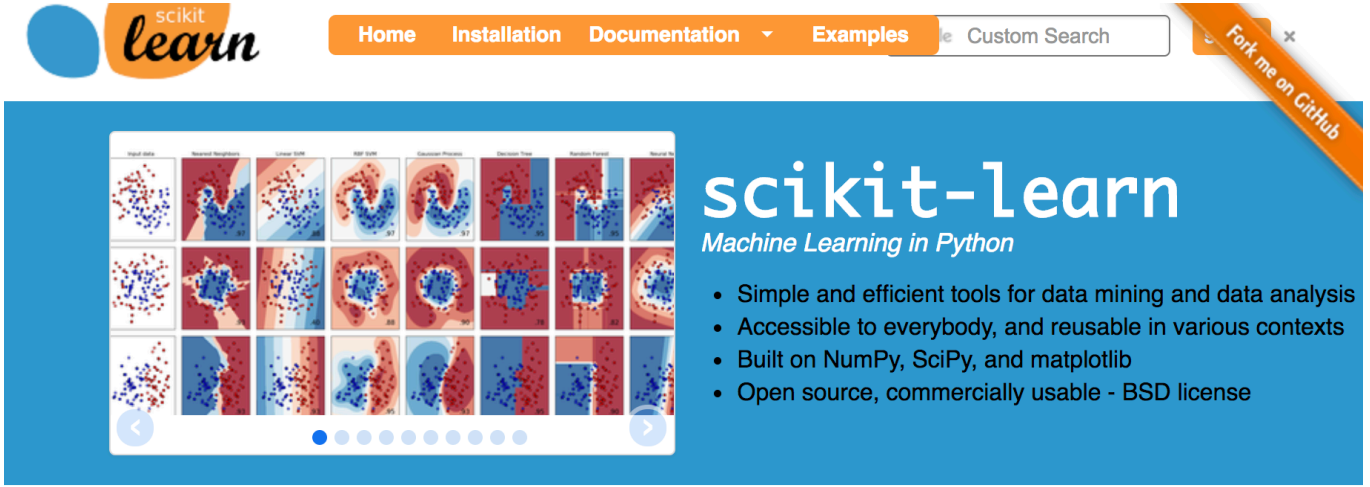
Comparison with logistic regression

- Logistic regression focuses on maximizing the probability of the data.

- SVM tries to find the separating hyperplane that maximizes the distance of the closest points to the margin

- Which one to use?
    - SVM typically works better for “clearly” linearly/nonlinearly separable classes
    - Logistic regression can work better for classes are not separable “in the middle”, due to its probabilistic formulation
- In practice, you should try both and compare.

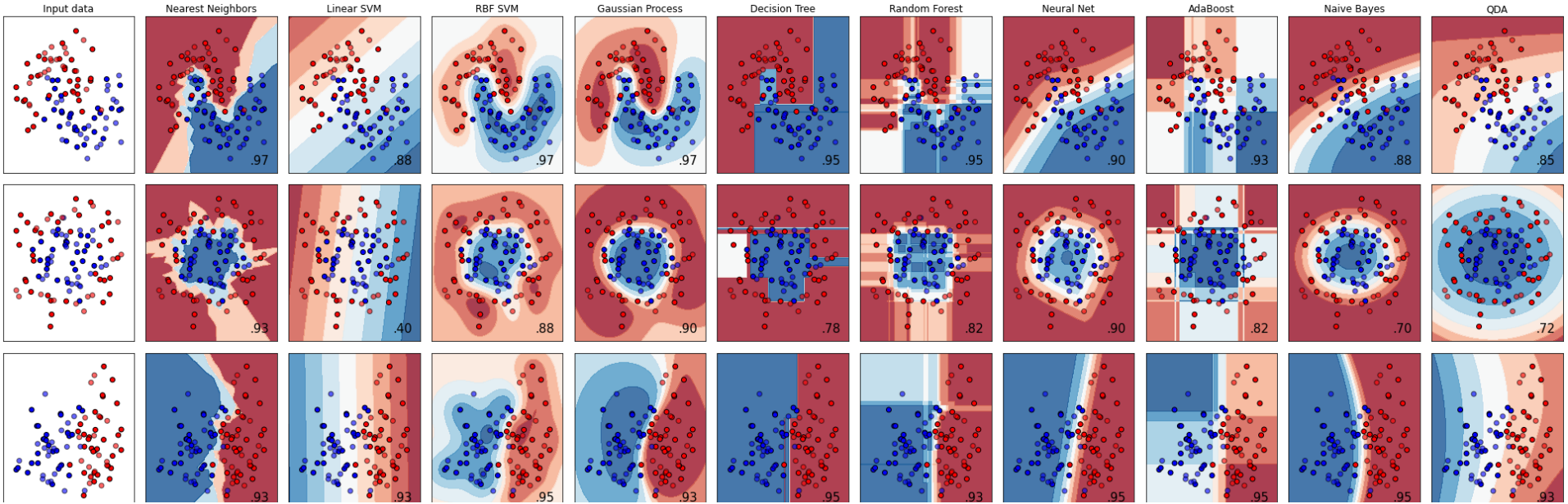
Scikit-learn – python library for machine learning



Hand-written digits classification example using SVM

http://scikit-learn.org/stable/tutorial/basic/tutorial.html#introduction
http://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

Demo: comparison of classification algorithms




SUMMARY