Computational Data Analysis

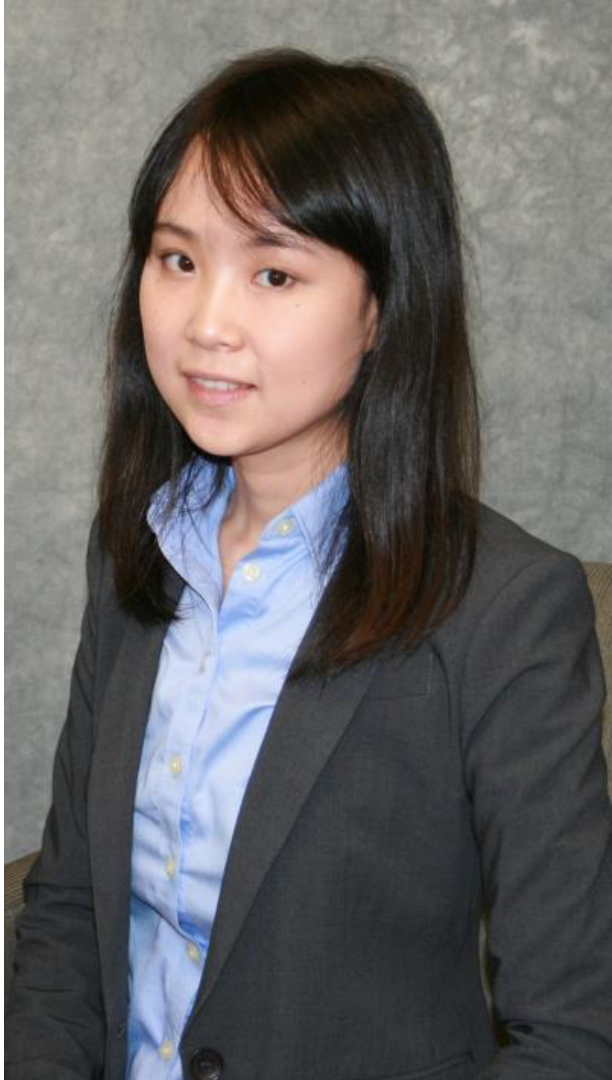
Machine Learning

Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering



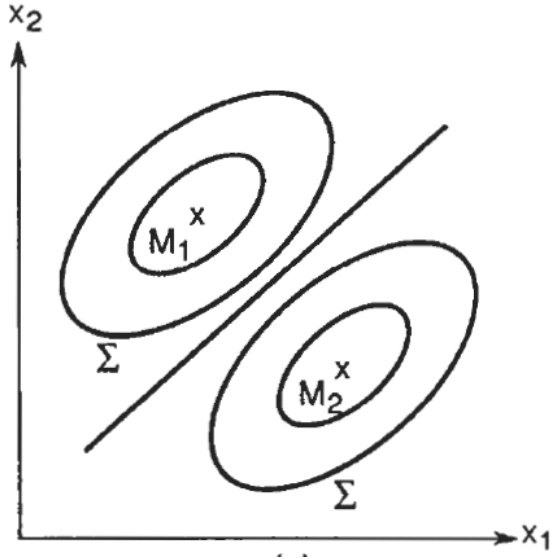
Support Vector Machine (SVM)

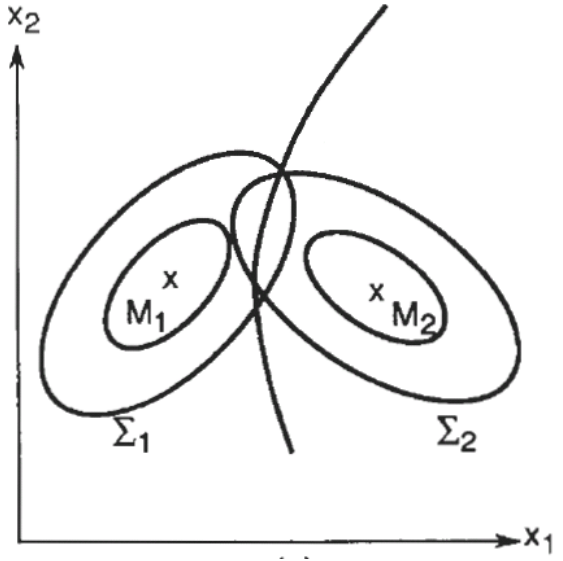
Main approaches to design classifiers

- Bayes rule, use simplifying 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

Example: Gaussian class conditional distribution

- Depending on the Gaussian distributions, the decision boundary can be very different



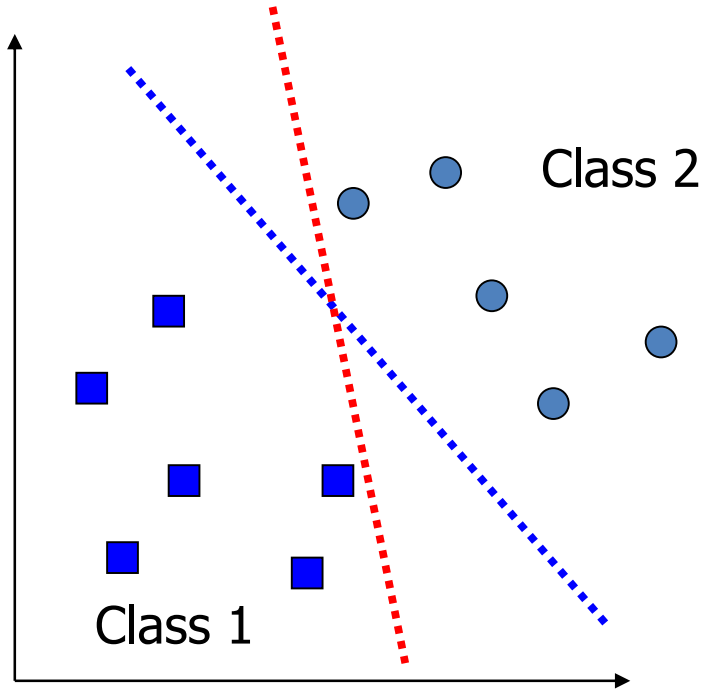


- Decision boundary of logistic regression

$$w^T x = 0$$

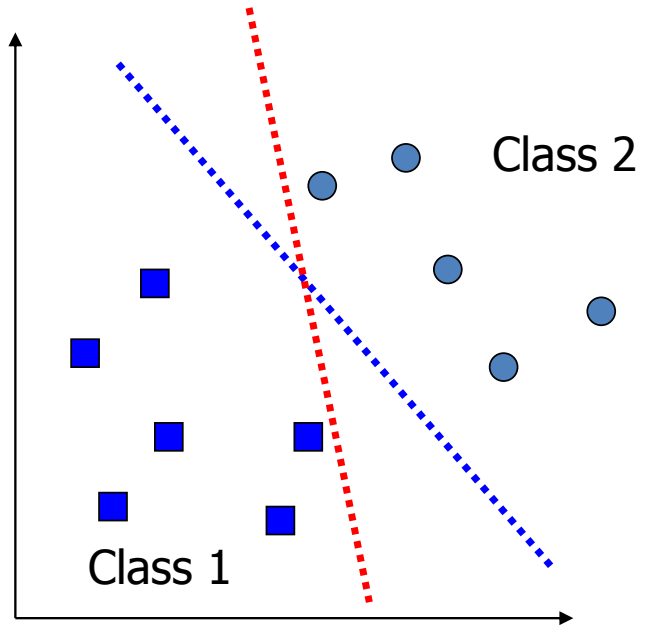
Which decision boundary is better?

- 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

- Suppose we perturb the 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.

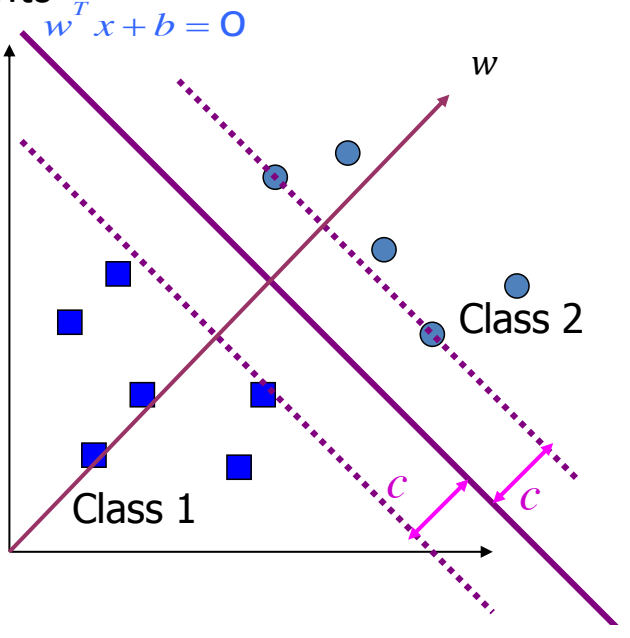
Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297.

Cortes, Corinna

Vladimir Naumovich Vapnik

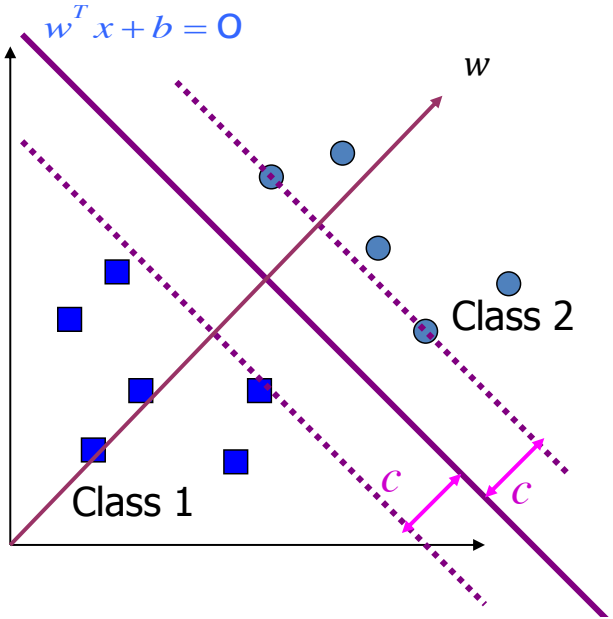
Geometric interpretation of a classifier

- Parameterizing decision boundary as: $w^{\mathsf{T}}x + b = 0$
  - 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

- Constraints on data points
  - For all x in class 2, $y = 1$ and $w^{\mathsf{T}}x + b \geq c$
  - For all x in class 1, $y = -1$ and $w^{\mathsf{T}}x + b \leq -c$
- Or more compactly, $(w^{\mathsf{T}}x + b)y \geq c$



Classifier margin

- Pick two data points x^1 and x^2 which are on each dash line respectively
- The unnormalized margin is $\tilde{\gamma} = w^\top (x^1 - x^2) = 2c$
- The margin is $\gamma = \frac{2c}{\|w\|}$

Given 2 parallel lines with equations

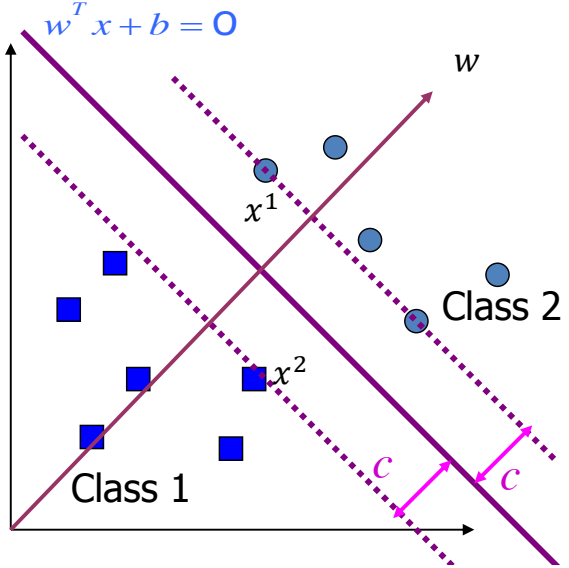
$$(1) \quad ax + by + c_1 = 0$$

and

$$(2) \quad ax + by + c_2 = 0$$

the distance between them is given by:

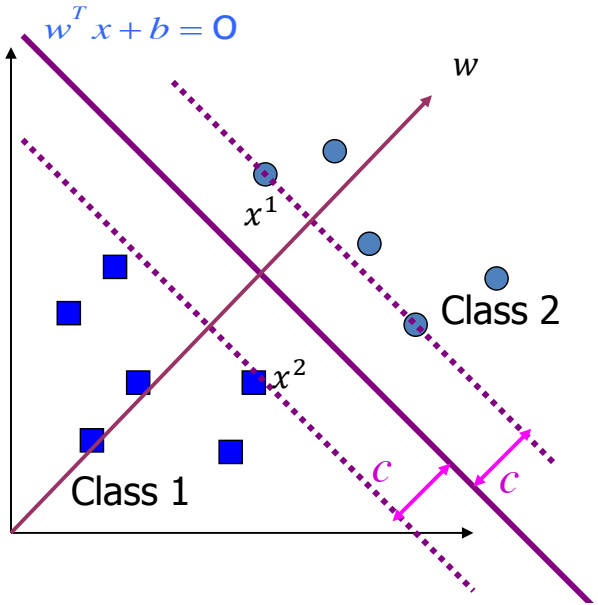
$$d = \frac{|c_2 - c_1|}{\sqrt{a^2 + b^2}}$$



Maximum margin classifier

- Find decision boundary w as far from data point as possible

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



instead of having two formula (one for positive and one for negative), we can use Yi in which it can take -1 or +1

When the sign of the equation inside the parenthesis is negative, Yi is -1

When the sign of the equation inside the parenthesis is positive, Yi is +1

Equivalent form

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

- 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

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

- After optimization, the margin is given by $\frac{2}{\|w\|}$
- Only a few of the constraints are relevant → support vectors
- Kernel methods are introduced for nonlinear classification problem

Lagrangian Duality

- The primal problem

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

- 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:

$$\frac{\partial L}{\partial w} = 0$$
$$\frac{\partial L}{\partial b} = 0$$
$$\frac{\partial L}{\partial \alpha} = 0$$
$$\frac{\partial L}{\partial \beta} = 0$$
$$g_i(w) \leq 0$$
$$h_i(w) = 0$$
$$\alpha_i \geq 0$$
$$\alpha_i g_i(w) = 0$$

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

converting it to non-positive inequality

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

alpha is lagrangian multiplier

- The lagrangian function

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

Deriving the dual problem

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

- 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

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

- After simplification

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

The dual problem of SVM

$$\max g(\alpha) := \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y^i y^j \left(x^{i\top} x^j \right)$$

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

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

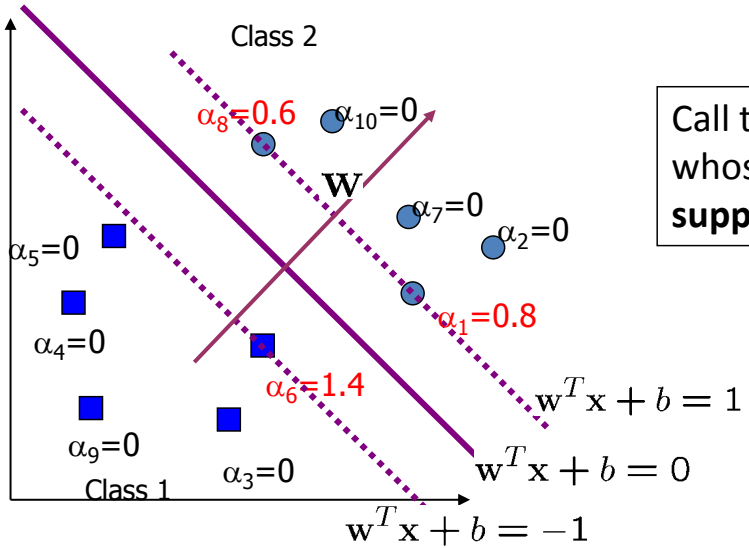
- This is a constrained quadratic programming
- Nice and convex, and global maximum can be found

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



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

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

  - Classify z as class 1 if the result is positive, and class 2 otherwise

Interpretation of support vector machines

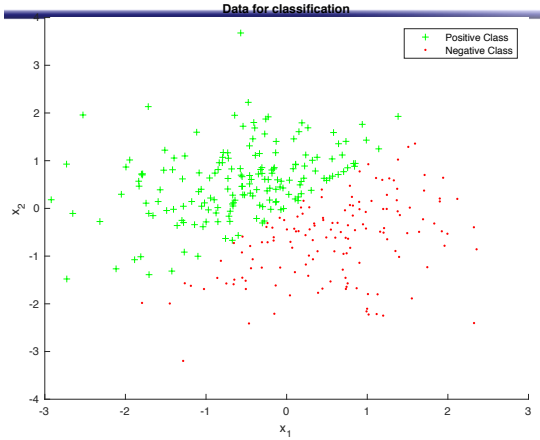
- The optimal $\boldsymbol{w}$ is a linear combination of a small number of data points. This “sparse” representation can be viewed as data compression

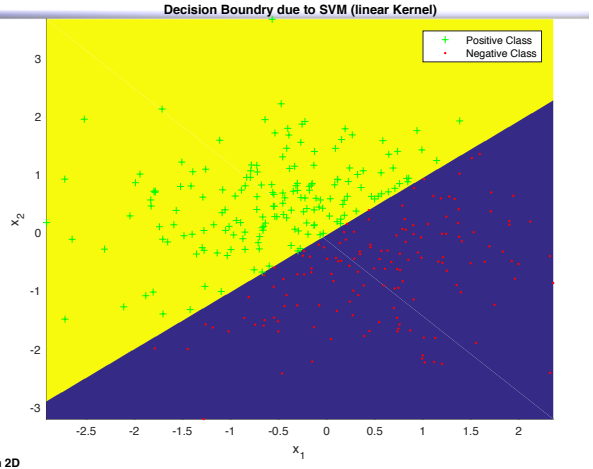
- To compute the weights α_i, and to use support vector machines we need to specify only the inner products (or kernel) between the examples $x^{i^\top} x^j$

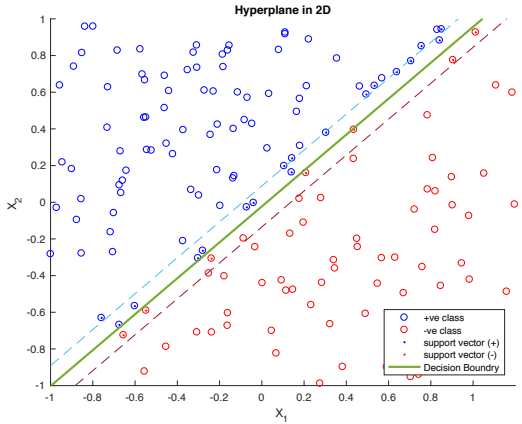
- We make decisions by comparing each new example z with only the support vectors:

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

MATLAB Demo


Data for classification
Positive Class
Negative Class
x1
x2

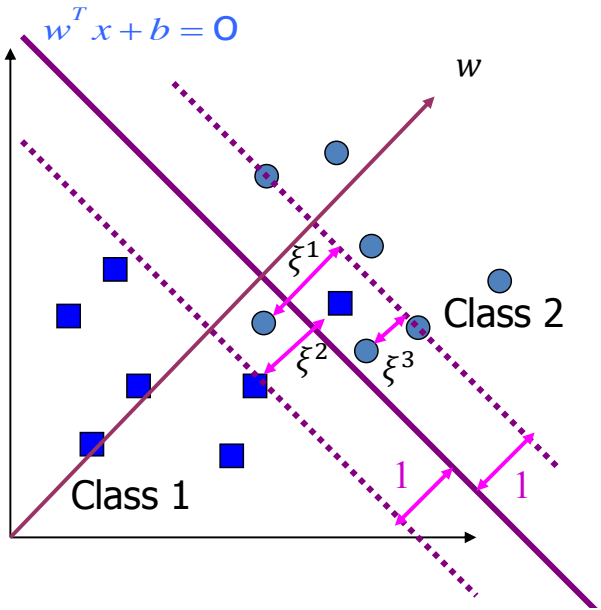

Decision Boundry due to SVM (linear Kernel)
Positive Class
Negative Class
x1
x2


Hyperplane in 2D
+ve class
-ve class
support vector (+)
support vector (-)
Decision Boundry
X1
X2

Soft margin constraints

- What if the data is not linearly separable?
- We will allow points to violate the hard margin constraint

$$(w^\top x + b)y \geq 1 - \xi$$



Soft margin SVM

$$\min_{w,b,\xi} \ ||w||^2 + C\sum_{i=1}^{m} \xi^i$$
$$s.t.\ y^i\left(w^\top x^i + b\right) \geq 1 - \xi^i, \xi^i \geq 0, \forall i$$

- Convert to standard form

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

- The Lagrangian function

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

Deriving the dual problem

$$L(w, \alpha, \beta)$$

$$= \frac{1}{2} w^\top w + \sum_i^m C\xi^i + \alpha_i\left(1 - y^i\left(w^\top x^i + b\right) - \xi^i\right) - \beta_i \xi^i$$

- Taking derivative and set to zero

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

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

$$\frac{\partial L}{\partial \xi^i} = C - \alpha_i - \beta_i = 0$$

Plug back relation of w, b and ξ

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

- After simplification

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

The dual problem

$$\max_{\alpha} \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y^i y^j \left({x^i}^{\mathsf{T}} x^j \right)$$

$$s.t.\ C - \alpha_i - \beta_i = 0, \alpha_i \geq 0, \beta_i \geq 0, i = 1, \ldots, m$$

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

this is the only change with adding regulization

That's the upper bound for alpha

- The constraint $C - \alpha_i - \beta_i = 0, \alpha_i \geq 0, \beta_i \geq 0$ can be simplified to $C \geq \alpha_i \geq 0$
- This is a constrained quadratic programming
- Nice and concave, global maximum can be found

Extensions of SVM

- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the “kernel trick”, implicitly mapping their inputs into high-dimensional feature spaces.

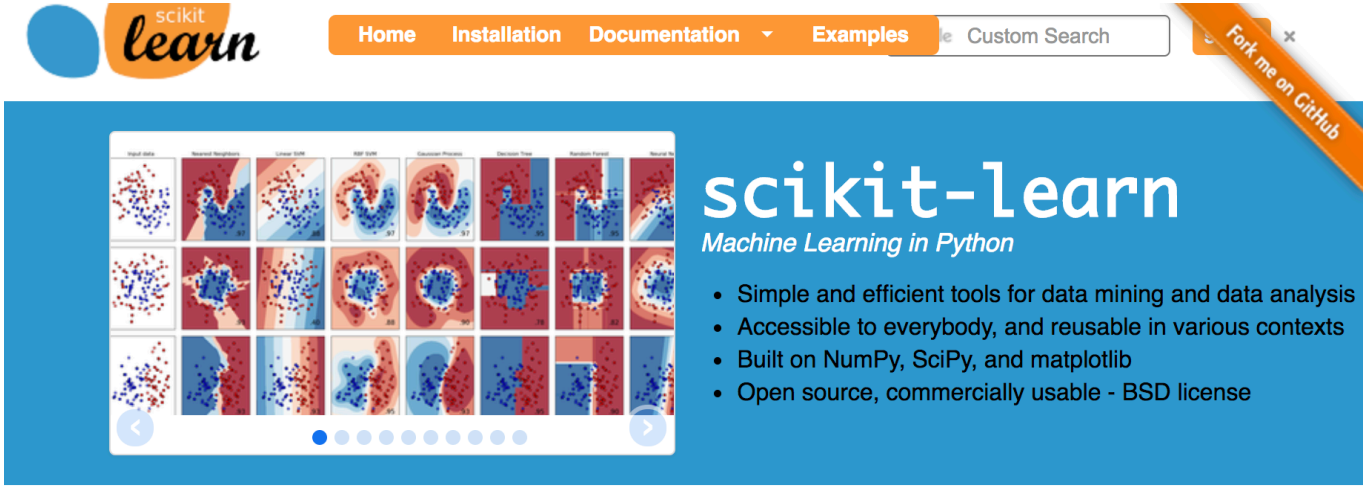
- There is also multi-class SVM.

- For unsupervised learning (without labels of data), there is also support vector clustering.

Comparison with logistic regression

- Logistic regression and SVM are closely linked
- Compared with logistic regression (which also has linear decision boundary)
    - Logistic regression focuses on maximizing the probability of the data.
    - An 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” linear separable classes
    - Logistic regression can work better for classes are not separable “in the middle”, due to its probabilistic formulation, which gives a smooth objective
    - 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


SUMMARY