

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)

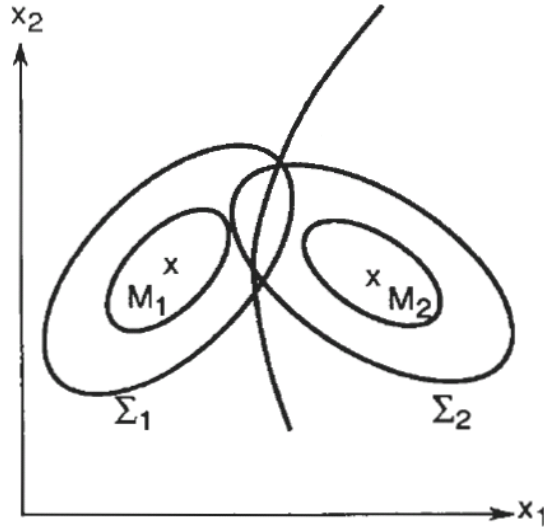
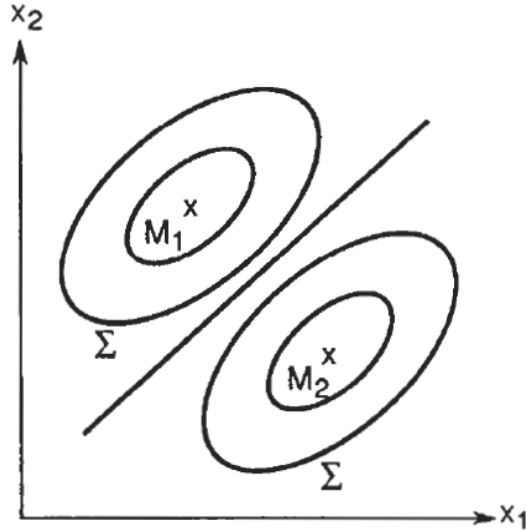


Main approaches to design classifiers

- Bayes rule + 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 $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$
 - 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

$$\theta^T x = 0$$

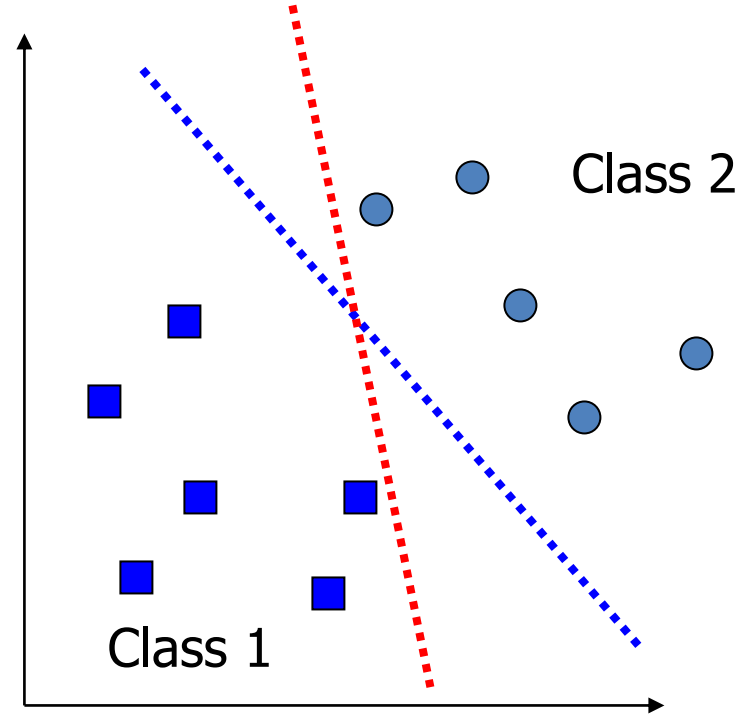
Which decision boundary is better?

Suppose the training samples are linearly separable

We can find a decision boundary which gives zero **training** error

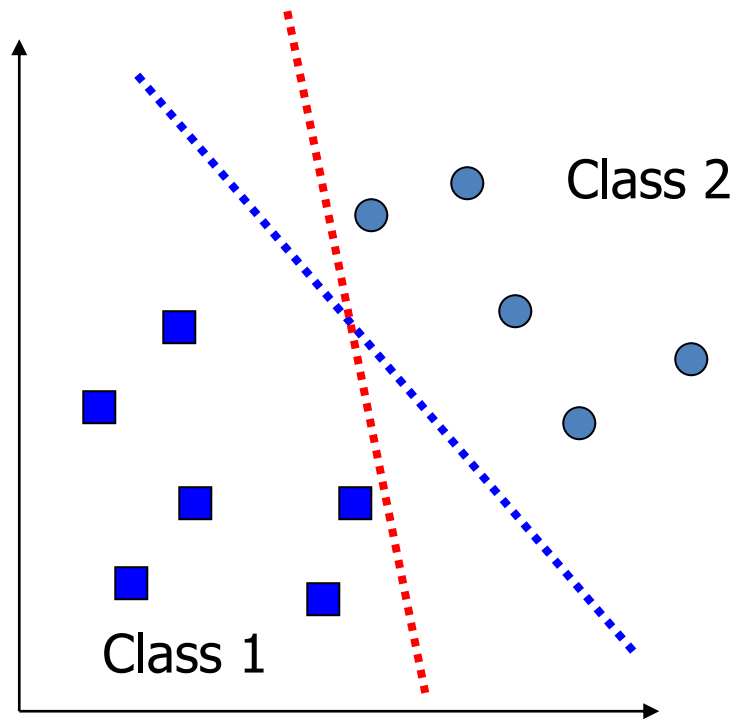
But there are many such decision boundaries

Which one is better?



Compare two decision boundaries

Suppose we perturb 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

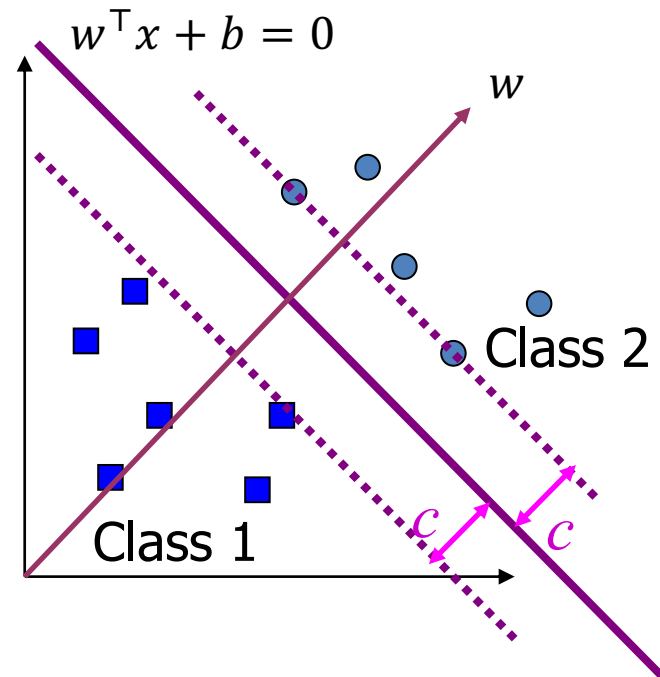
Geometric interpretation of a classifier

Parameterizing decision boundary as:

$$w^T x + b = 0$$

- 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

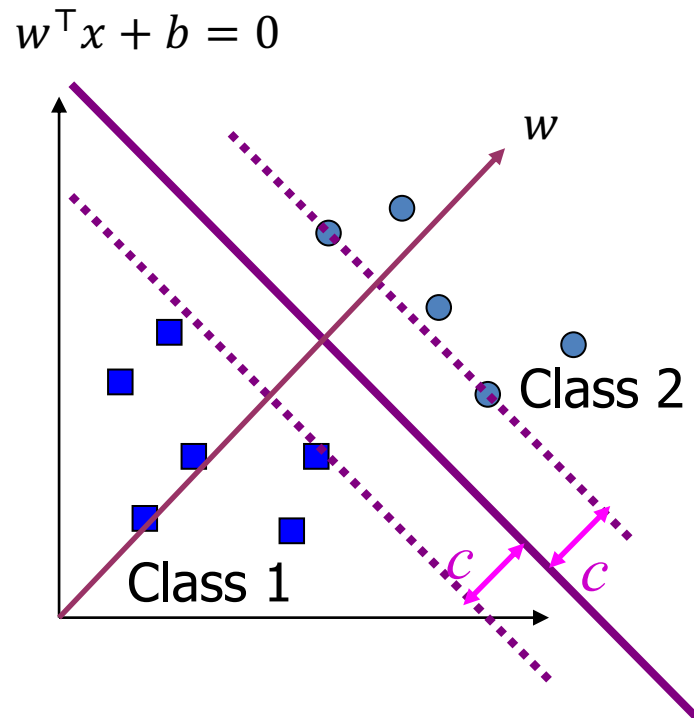


Constraints on data points

Constraints on data points

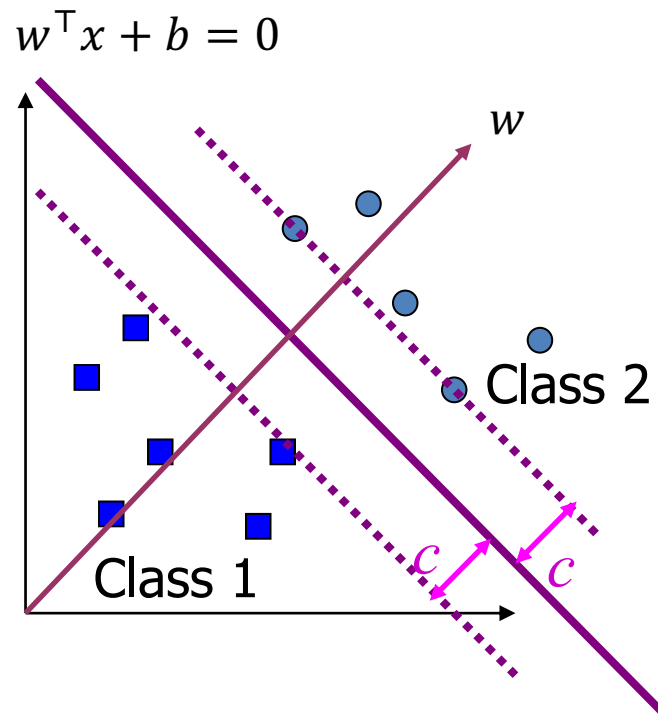
- For all x in class 2, $y = 1$ and $w^\top x + b \geq c$
- For all x in class 1, $y = -1$ and $w^\top x + b \leq -c$

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



Classifier margin

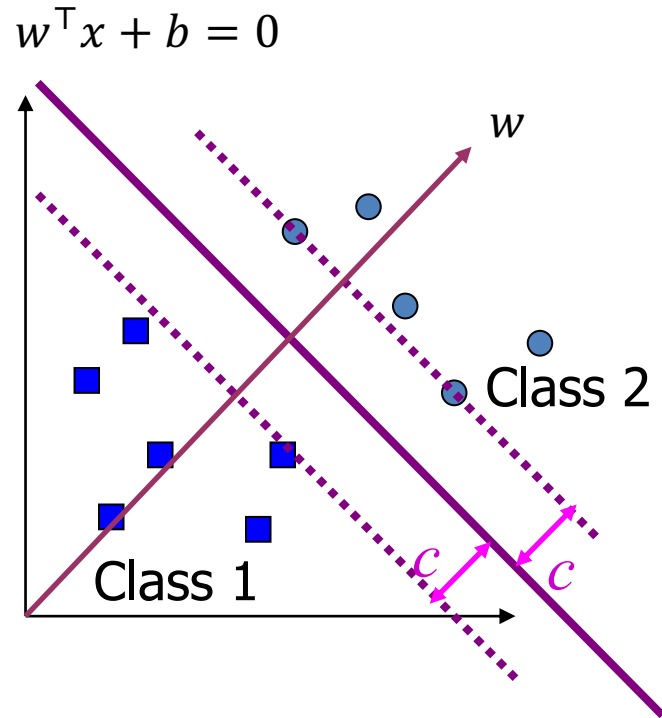
- Pick two data points x^1 and x^2 which are on each dash line respectively
- The **unnormalized** margin is $\tilde{\gamma} = w^\top(x^1 - x^2) = 2c$
- The margin is $\gamma = \frac{2c}{\|w\|}$



Maximum margin classifier

Find decision boundary w as far from data point as possible

$$\begin{aligned} \max_{w,b} \quad & \gamma = \frac{2c}{\|w\|} \\ \text{s.t.} \quad & y^i(w^\top x^i + b) \geq c, \forall i \end{aligned}$$



Equivalent form

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

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

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

Support vector machines

- A constrained convex quadratic programming problem

$$\begin{aligned} \min_{w,b} \quad & \|w\|^2 \\ \text{s.t.} \quad & y^i(w^\top x^i + b) \geq 1, \forall i \end{aligned}$$

- 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

$$\begin{aligned} & \min_w f(w) \\ & \text{st. } g_i(w) \leq 0, i = 1, \dots, k \\ & \quad h_i(w) = 0, i = 1, \dots, l \end{aligned}$$

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

$$\begin{aligned} \min_{w,b} \quad & ||w||^2 \\ \text{s.t.} \quad & y^i(w^\top x^i + b) \geq 1, \forall i \end{aligned}$$

Convert to standard form

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^\top w \\ \text{s.t.} \quad & 1 - y^i(w^\top x^i + b) \leq 0, \forall i \end{aligned}$$

The Lagrangian function

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

Deriving the dual problem

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

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^*):

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

The dual problem of SVM

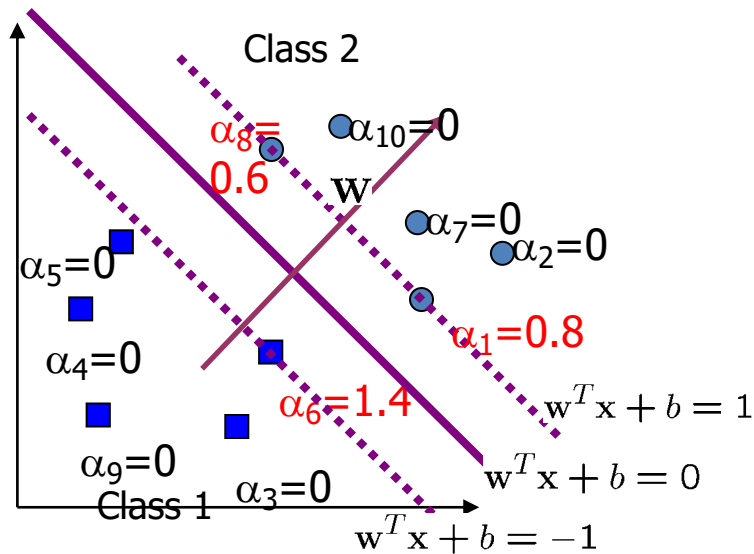
$$\begin{aligned}\max_{\alpha} L(w^*, b^*, \alpha) &= \sum_i^m \alpha_i - \frac{1}{2} \sum_{i,j}^m \alpha_i \alpha_j y^i y^j (x^{i^\top} x^j) \\ s. t. \alpha_i &\geq 0, i = 1, \dots, m \\ \sum_i^m \alpha_i y^i &= 0\end{aligned}$$

- 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

Support vectors

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

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



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

Computing b and obtain the classifier

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

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

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

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

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

Interpretation of support vector machines

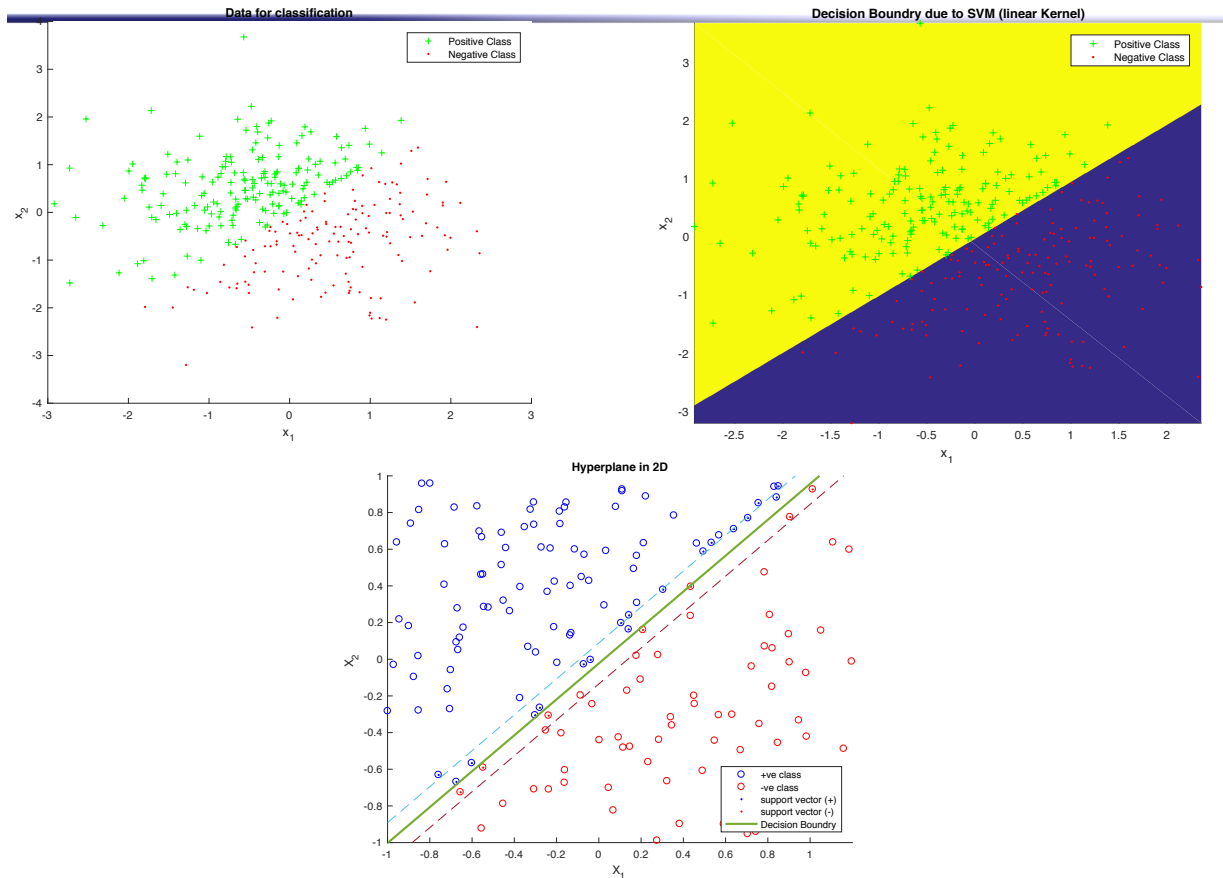
The optimal w is a linear combination of a small number of data points. This “sparse” representation can be viewed as data compression

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:

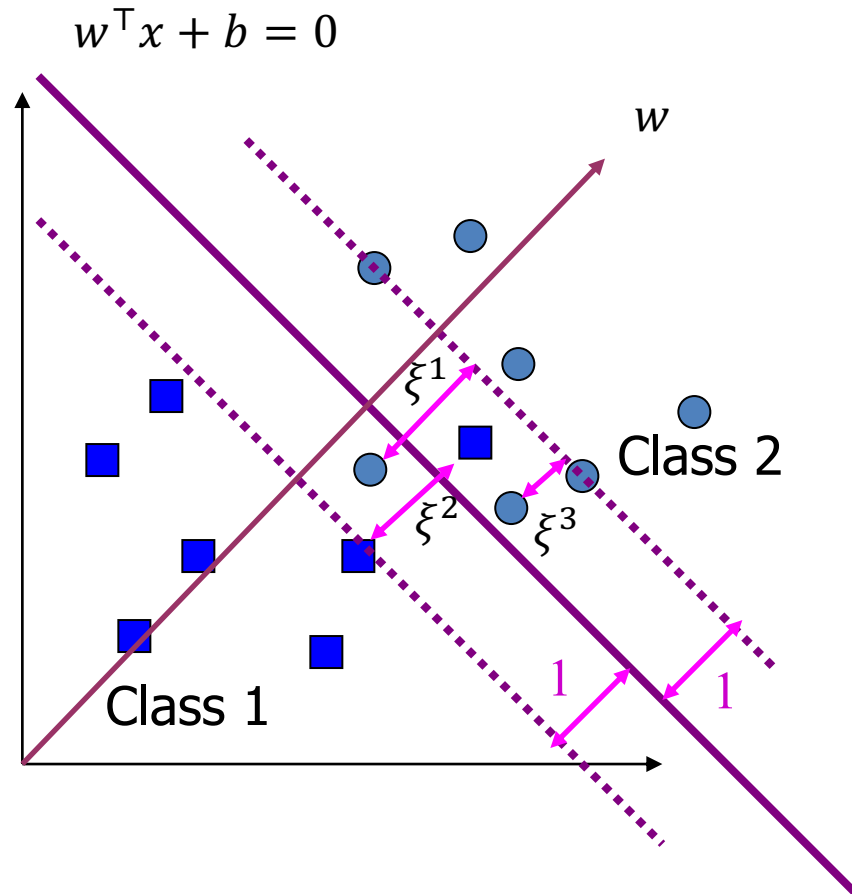
$$y^* = \text{sign} \left(\sum_{i \in \text{support vectors}} \alpha_i y^i (x^i z) + b \right)$$

MATLAB Demo



Soft margin constraints

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



Soft margin SVM

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi^i \\ \text{s.t.} \quad & y^i (w^\top x^i + b) \geq 1 - \xi^i, \xi^i \geq 0, \forall i \end{aligned}$$

Convert to standard form

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} w^\top w + C \sum_{i=1}^m \xi^i \\ \text{s.t.} \quad & 1 - y^i (w^\top x^i + b) - \xi^i \leq 0, \xi^i \geq 0, \forall i \end{aligned}$$

The Lagrangian function

$$\begin{aligned} & L(w, b, \alpha, \beta) \\ = & \frac{1}{2} w^\top w + \sum_i^m C \xi^i + \alpha_i (1 - y^i (w^\top x^i + b) - \xi^i) - \beta_i \xi^i \end{aligned}$$

Deriving the dual problem

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

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

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

After simplification (plug back the optimal variables)

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

The dual problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_i^m \alpha_i - \frac{1}{2} \sum_{i,j}^m \alpha_i \alpha_j y^i y^j (x^{i^\top} x^j) \\ \text{s.t.} \quad & C - \alpha_i - \beta_i = 0, \alpha_i \geq 0, \beta_i \geq 0, i = 1, \dots, m \\ & \sum_i^m \alpha_i y^i = 0 \end{aligned}$$

- 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, and global maximum can be found

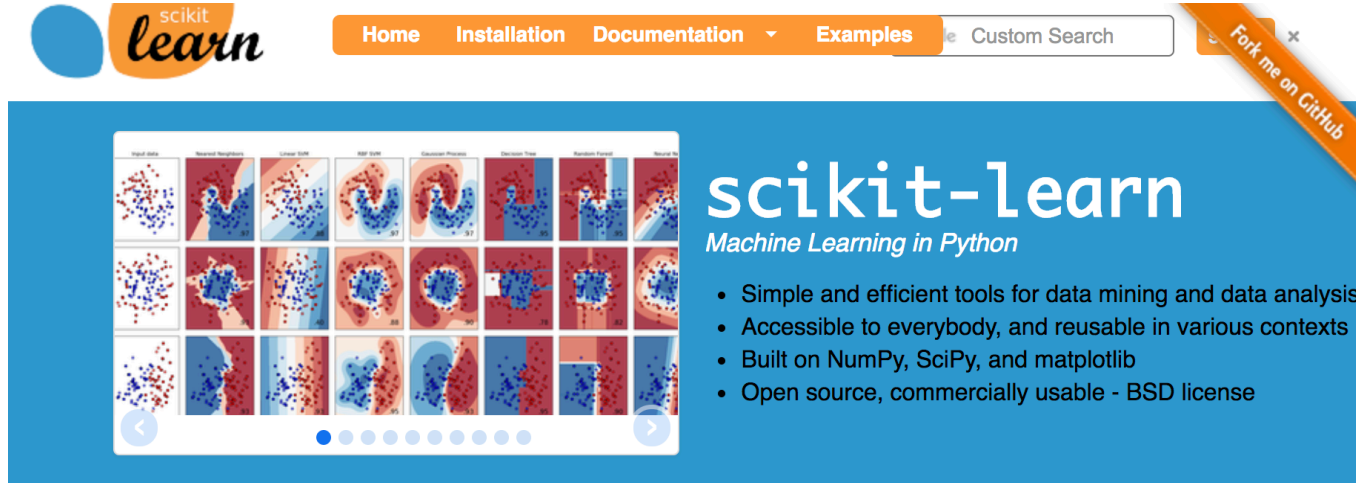
Extensions of SVM

- 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.
- 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

