Support Vector Machines

Le Song

Machine Learning
CSE/ISYE 6740, Fall 2019

Ways to design classifier

- 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

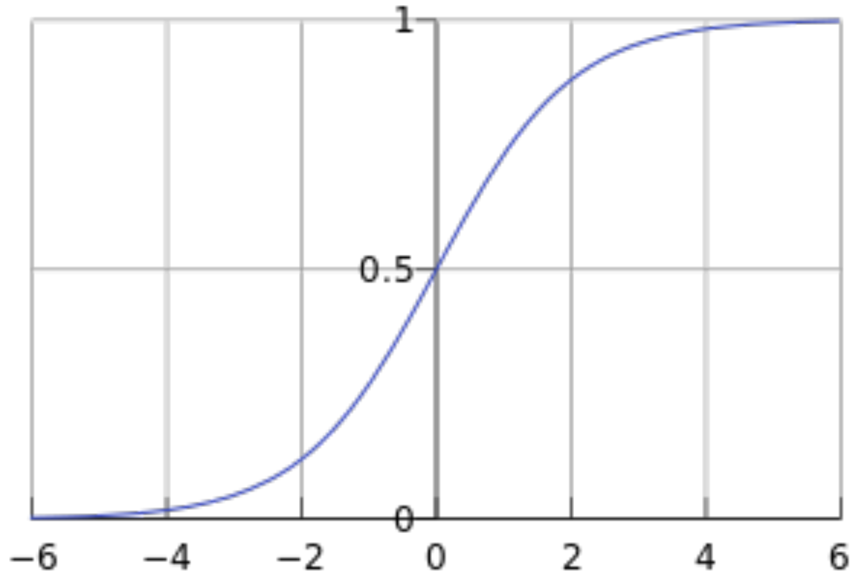
- Directly go for the decision boundary $\mathrm{h(x)} = -\ln \frac{q_i(x)}{q_j(x)}$
  - Logistic regression
  - Neural networks

What is logistic regression model

- Assume that the posterior distribution $p(y=1|x)$ take a particular form

$$p(y=1|x,\theta)=\frac{1}{1+\exp(-\theta^\top x)}$$

- Logistic function $f(u)=\frac{1}{1+\exp(-u)}$

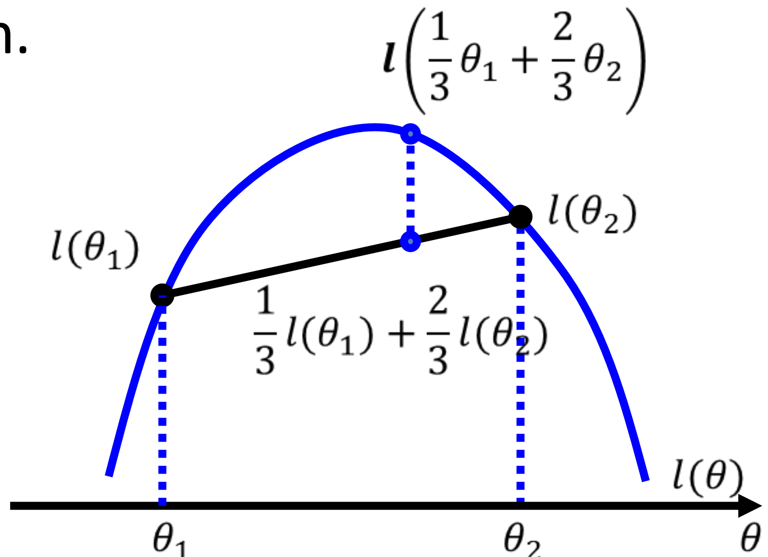


Learning parameters in logistic regression

- Find θ, such that the conditional likelihood of the labels is maximized

$$\max_{\theta} l(\theta) := \log \prod_{i=1}^{m} P\left(y^i \middle| x^i, \theta\right)$$

- Good news: $l(\theta)$ is concave function of θ, and there is a single global optimum.



- Bad new: no closed form solution (resort to numerical method)

The gradient of $l(\theta)$

$$l(\theta) := \log \prod_{i=1}^{m} P\left(y^i \middle| x^i, \theta\right)$$

$$= \sum_i (y^i - 1)\, \theta^\top x^i - \log(1 + \exp\left(-\theta^\top x^i\right))$$

- Gradient

$$\frac{\partial l(\theta)}{\partial \theta} = \sum_i (y^i - 1)\, x^i + \frac{\exp\left(-\theta^\top x^i\right) x^i}{1 + \exp(-\theta^\top x)}$$

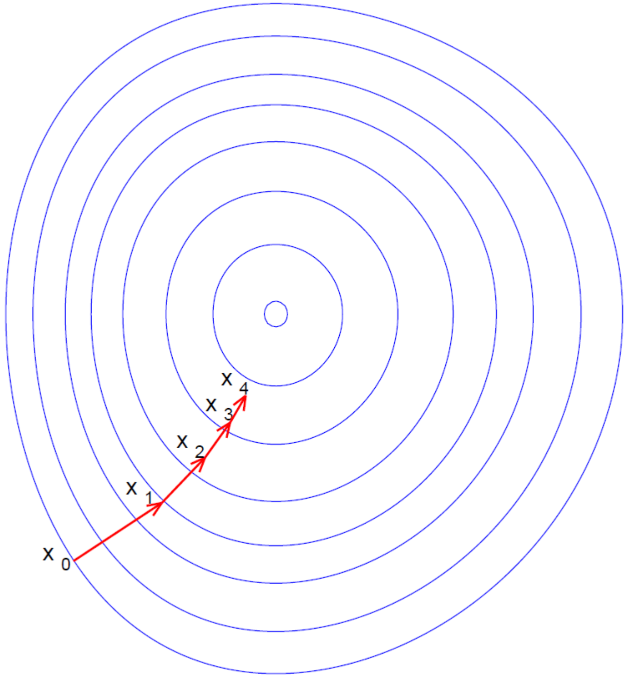
- Setting it to 0 does not lead to closed form solution

Gradient descent

- One way to solve an *unconstrained* optimization problem is gradient descent

- Given an initial guess, we *iteratively* refine the guess by taking the direction of the negative gradient

- Think about going down a hill by taking the steepest direction at each step



- Update rule

$$\theta_{k+1} = \theta_k - \gamma_k \nabla f(\theta_k)$$

γ_k is called the step size or learning rate

Gradient Ascent/Descent algorithm

- Initialize parameter θ^0

- Do

$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i (y^i - 1)\, x^i + \frac{\exp(-\theta^\top x^i) x^i}{1 + \exp(-\theta^\top x)}$$

- While the $||\theta^{t+1} - \theta^t|| > \epsilon$

Batch gradient vs stochastic gradient

- The gradient involves all data points

$$\nabla f(\theta) = \sum_i (y^i - 1)\, x^i + \frac{\exp(-\theta^\top x^i) x^i}{1 + \exp(-\theta^\top x)}$$

- To compute the gradient at *each* iteration, we need to sum over *all* data points in the dataset

- What if we have a huge dataset? For example, 1 Million data points?

- We can take one data point and compute a stochastic gradient

$$\nabla \hat{f}(\theta) = (y^i - 1) x^i + \frac{\exp(-\theta^\top x^i) x^i}{1 + \exp(-\theta^\top x)}$$

Multiclass logistic regression

- Assign input vector $x^i, i = 1, \ldots, m$ into one of classes $c, c = 1, \ldots, C$

- Assume that the posterior distribution take a particular form:

范围 0 ~1
且对c求和为1

$$P(y^i = c | x^i, \theta_1, \ldots, \theta_C) = \frac{\exp\left(\theta_c{}^\top x^i\right)}{\sum_{c'} \exp\left(\theta_{c'}{}^\top x^i\right)}$$

- Now, let's introduce some notation:

$$u_c^i := P\left(y^i = \mathrm{c} \mid x^i, \theta_1, \ldots, \theta_C \right)$$
$$y_c^i = I\ (y^i = c)$$

Learning parameters in multiclass logistic regression

- Given all the input data

$$(x^1, y^1), (x^2, y^2), \dots, (x^m, y^m)$$

- The log-likelihood can be written as:

$$l(\theta) := \log \prod_{i=1}^{m} \prod_{c=1}^{C} (u_c^i)^{y_c^i}$$

$$= \sum_{i=1}^{m} \sum_{c=1}^{C} y_c^i \log u_c^i$$

$$= \sum_{i=1}^{m} \sum_{c=1}^{C} y_c^i \theta_c^T x^i - \log \sum_{i=1}^{m} \sum_{c'=1}^{C} \exp(\theta_{c'}^T x^i)$$

Learning parameters in multiclass logistic regression

- Find θ such that the conditional likelihood of the labels is maximized

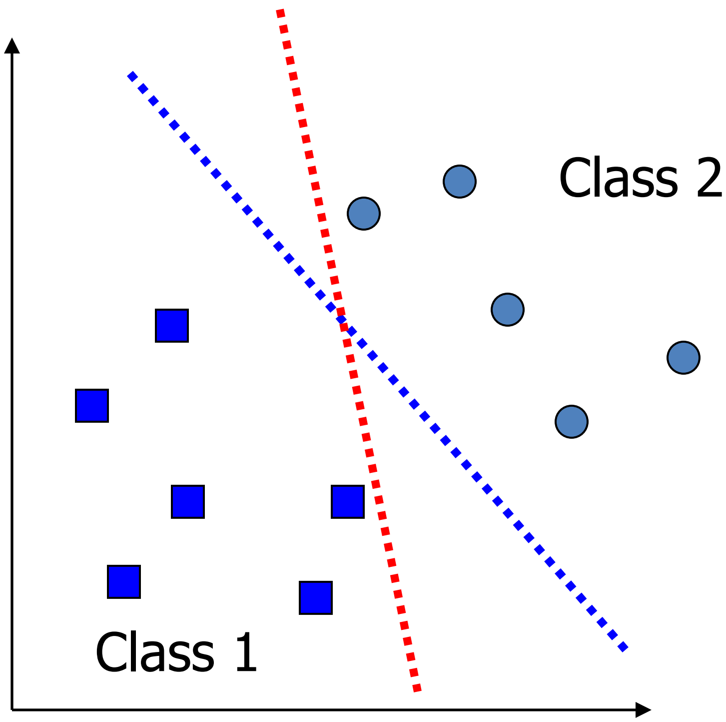
- $-l(\theta)$ also known as cross-entropy error function for multiclass

- Compute the gradient of $f(\theta)$ with respect to one parameter vector θ_c :

$$\frac{\partial f}{\partial \theta_c} = -\sum_{i}^{m} \left(u_c^i - y_c^i\right) x^i$$

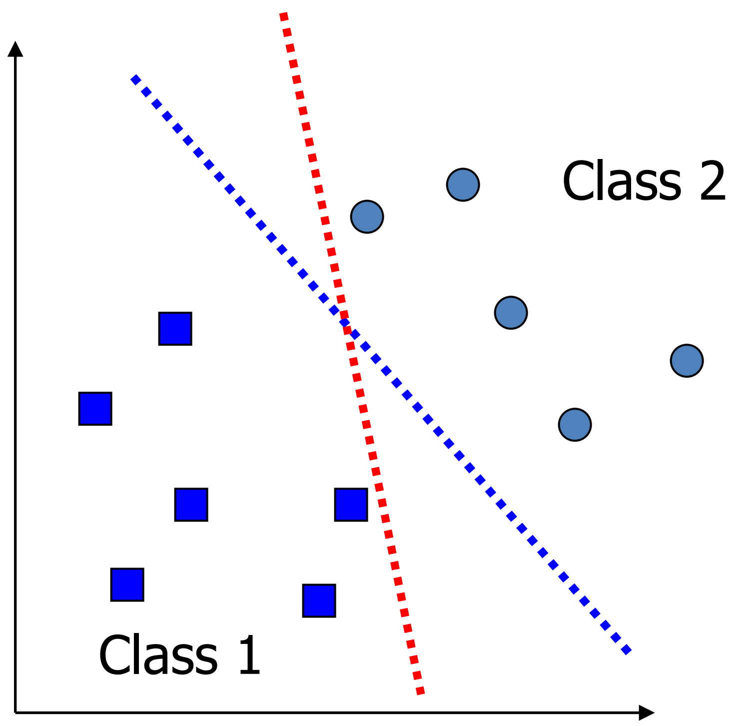
Which decision boundary is better?

- 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?



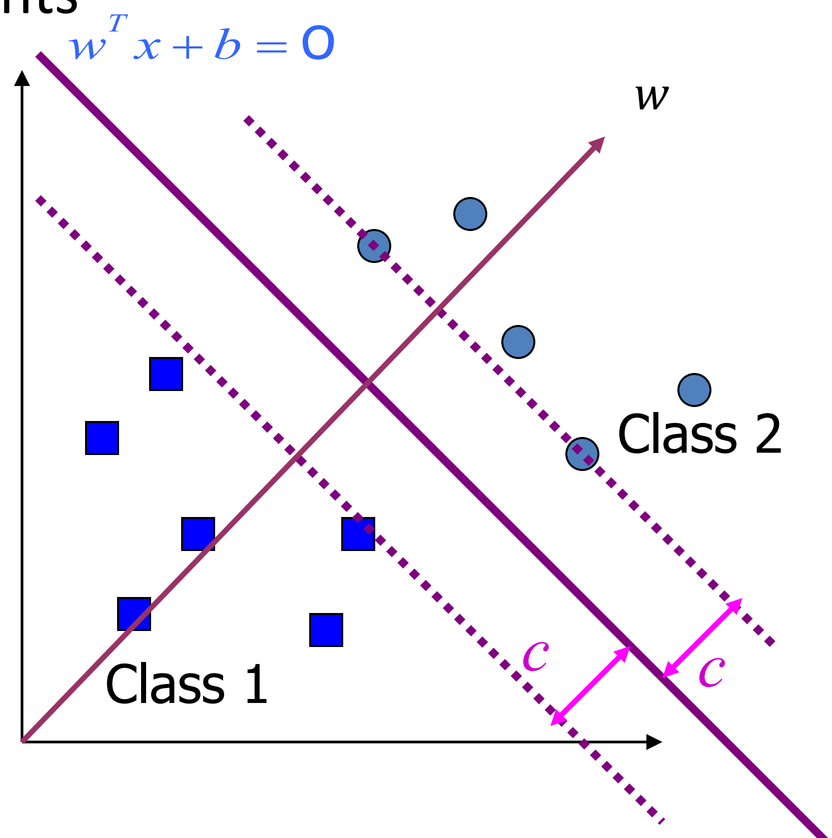
Compare two decision boundaries

- Suppose we perturb the data, which boundary is more susceptible to error?



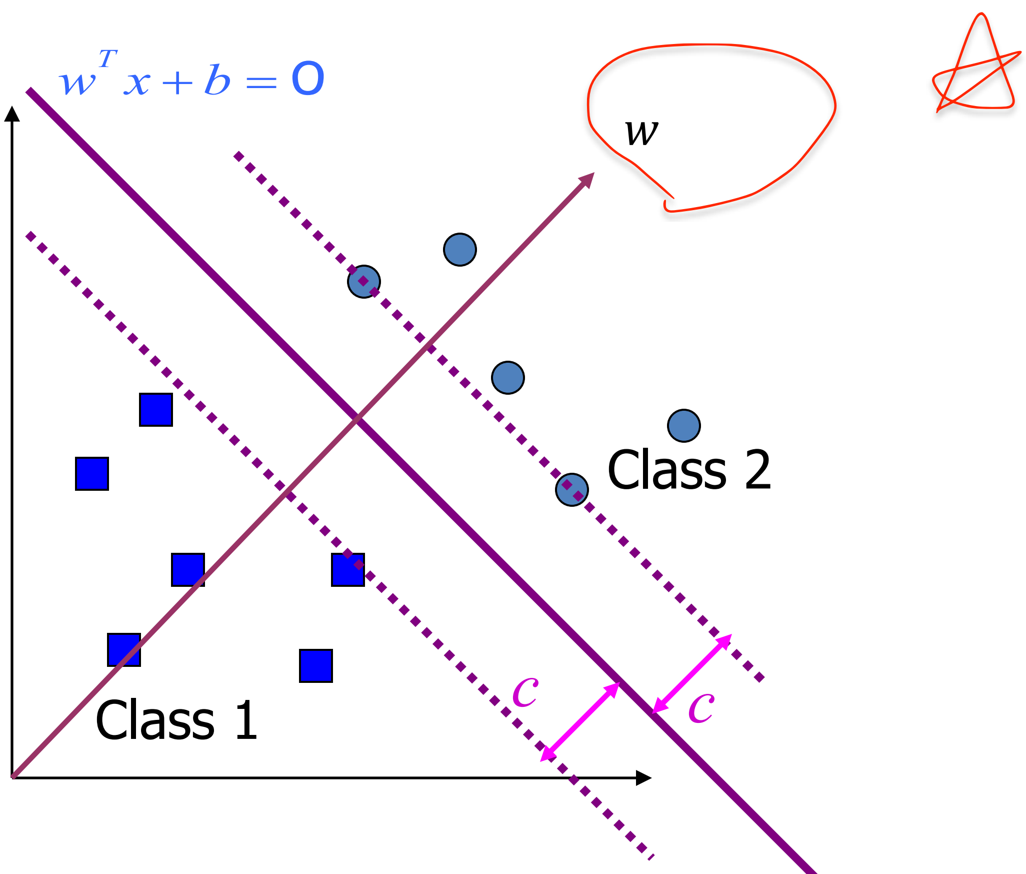
Geometric interpretation of a classifier

- 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



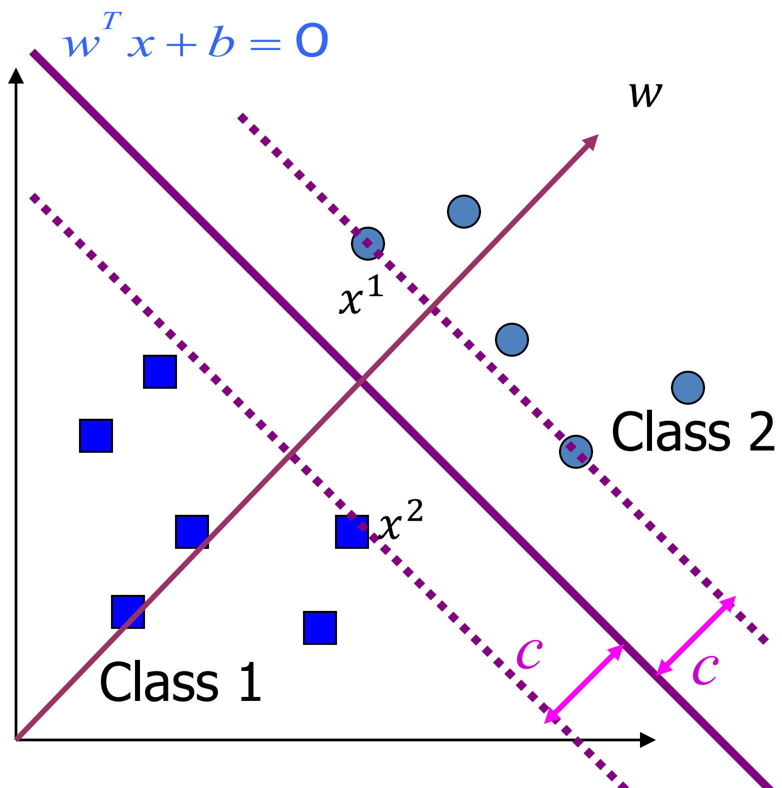
Constraints on data points

- 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

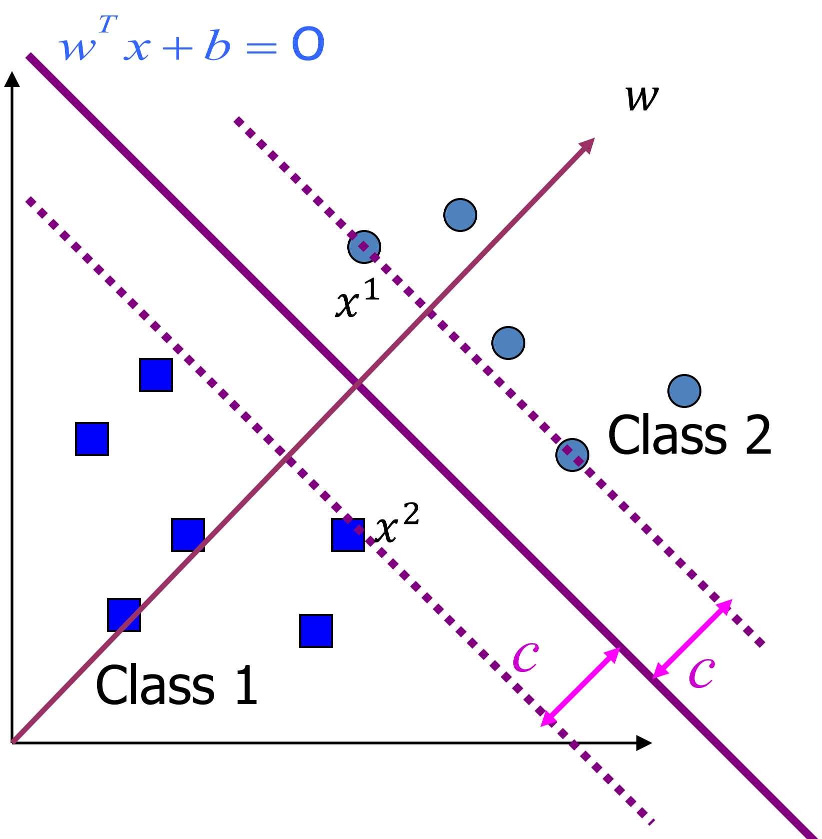
- Pick two data points x^1 and x^2 which are on each dash line respectively
- The unnormalized margin is $\tilde{\gamma} = w^\top(x^1 - x^2) = 2c$
- The margin is $\gamma = \frac{2c}{||w||}$



Maximum margin classifier

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

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



Equivalent form

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

- 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

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

- 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:

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

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

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

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

$$g_i(w) \leq 0$$

$$h_i(w) = 0$$

$$\alpha_i \geq 0$$

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

complimentary slackness

Dual problem of support vector machines

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

- Convert to standard form

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

- The lagrangian function

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

Deriving the dual problem

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

- 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

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

- After simplification

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

b 作为常数扔掉

The dual problem of SVM

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

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

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

- This is a constrained quadratic programming

?

- Nice and convex, and global maximum can be found
- Very similar to the dual of minimum enclosing ball problem

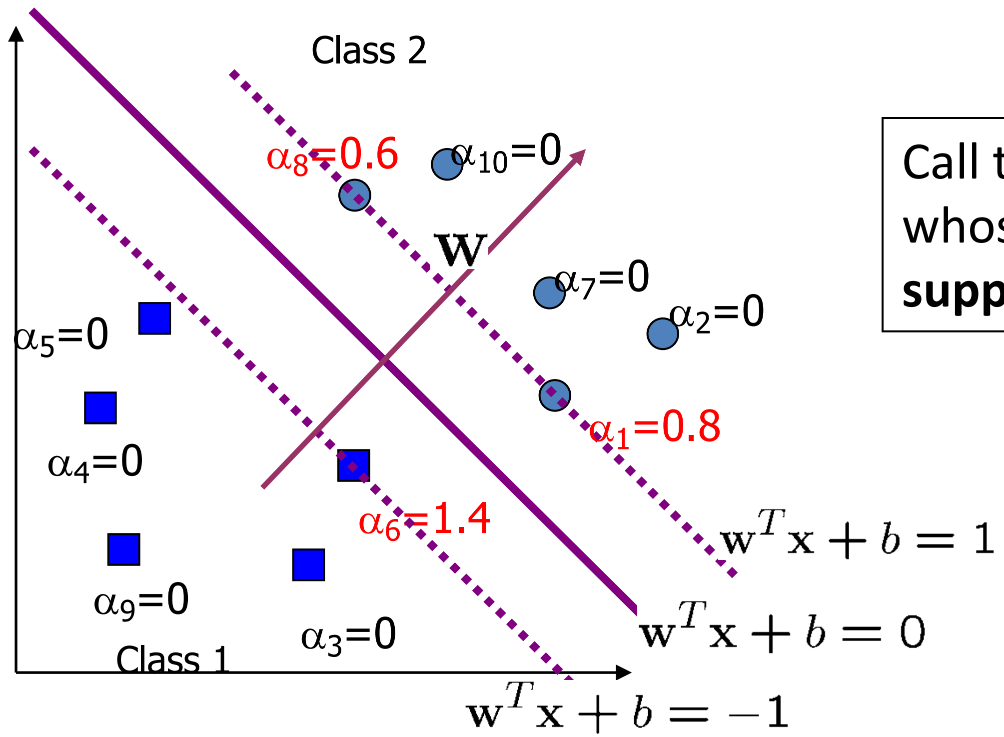
Support vectors

- Note that the KKT condition $\alpha_i g_i(w) = 0$

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

不在boundary上

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



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

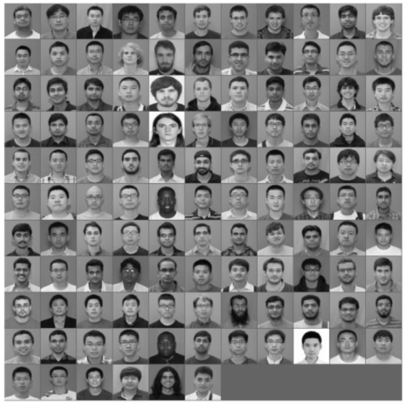
- To compute the weights α_i, and to use support vector machines we need to specify only the inner products (or kernel) between the examples ${x^i}^\top x^j$

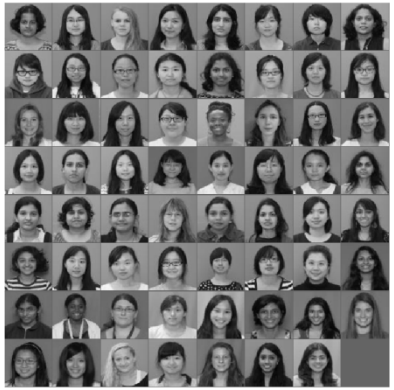
- We make decisions by comparing each new example z with only the support vectors:

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

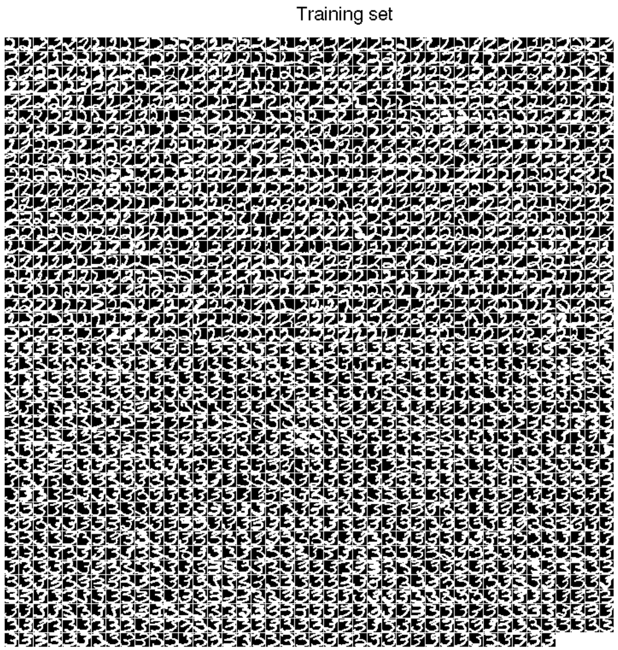
Demo

- Boys vs Girls





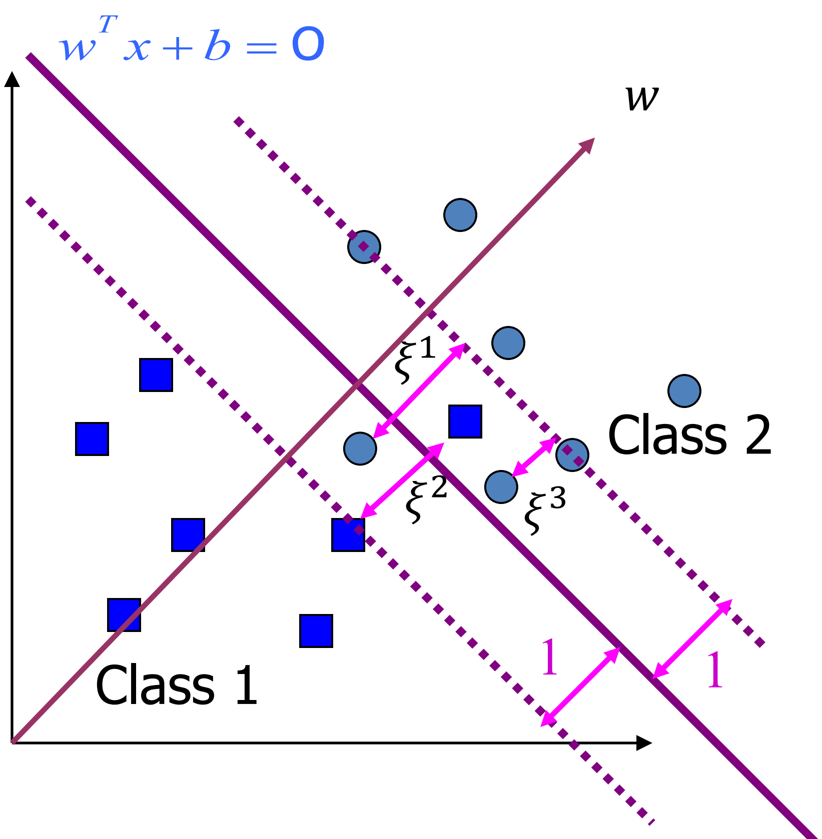
- Handwritten digits 2 vs 3



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

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

- 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

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

- After simplification

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

The dual problem

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

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

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

- 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 convex, and global maximum can be found