CAP 5610: Machine Learning

Lecture 7: Support Vector Machines II

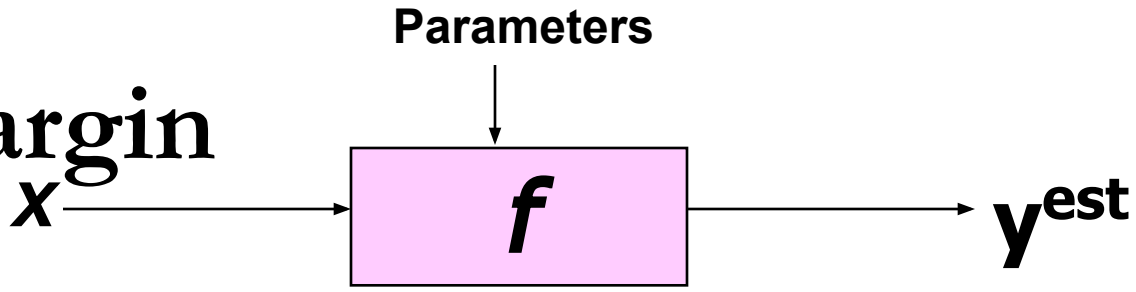
Instructor: Dr. Gita Sukthankar
Email: gitars@eecs.ucf.edu

Reference Materials From: GuoJun Qi, Ray Mooney, Geoff Hinton, Jean-Michael Renders

Timeline

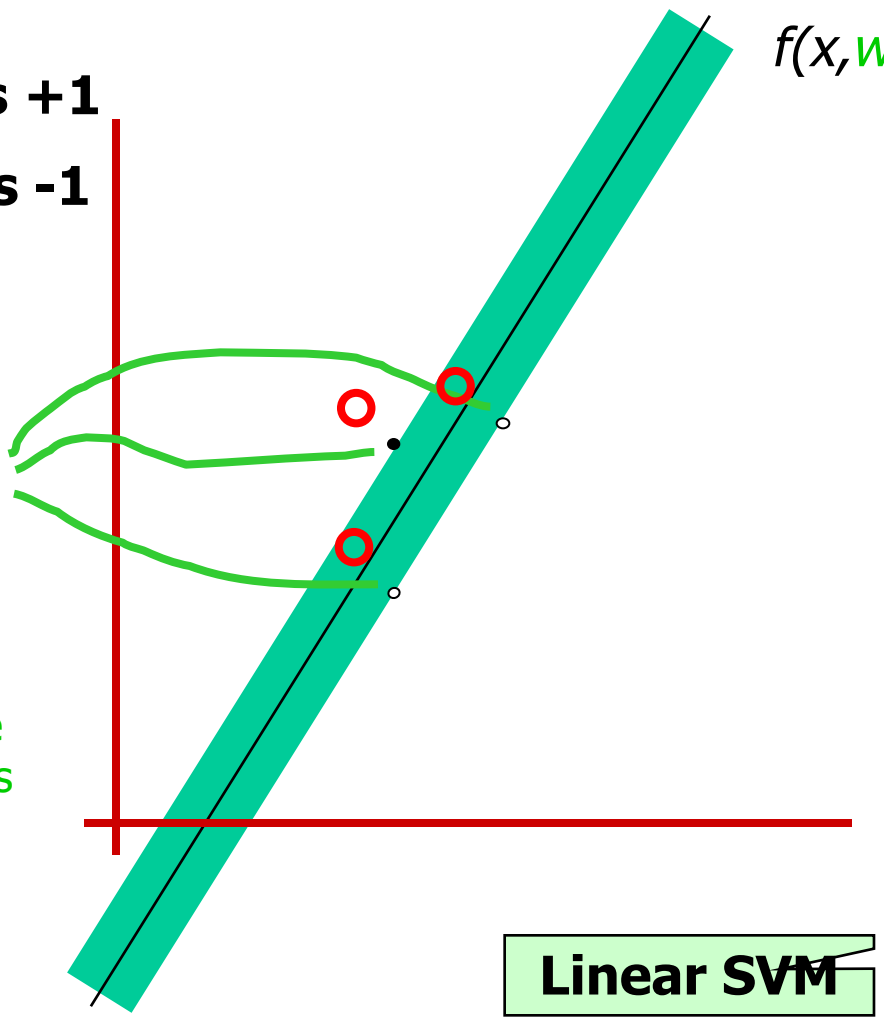
- Homework due today---no extensions
- Next homework on SVMs: due on Oct 15
  - No class on Oct 15th
- Midterm exam on Oct 22nd
  - Topics covered thus far plus:
    - Neural networks + CNNs
    - Dimensionality reduction
    - Decision trees and ensembles
  - Review session on Oct 17
- Final project proposal due on Oct 24
- Topics transition into deep learning
- Deep RL assignment (due early November)
- Final project (due during exam period)

Maximum Margin



$f(x,w,b) = sign(w\,x + b)$

- denotes +1
- denotes -1



Keeping only support vectors will not change the maximum margin classifier.

- Robust to the small changes (noises) in non-support vectors

The maximum margin linear classifier is the linear classifier with the maximum margin.

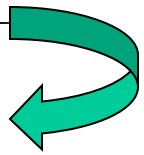
This is the simplest kind of SVM (Called an LSVM)

Linear SVM

Linear SVM Mathematically

- **Goal:** 1) Correctly classify all training data

$$wx_i + b \geq 1 \quad \text{if } y_i = +1$$
$$wx_i + b \leq -1 \quad \text{if } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$



2) Maximize the Margin $M = \frac{2}{|w|}$

same as minimize $\frac{1}{2} w^t w$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**

- **Minimize** $\Phi(w) = \frac{1}{2} w^t w$

  subject to $y_i(wx_i + b) \geq 1 \qquad \forall i$

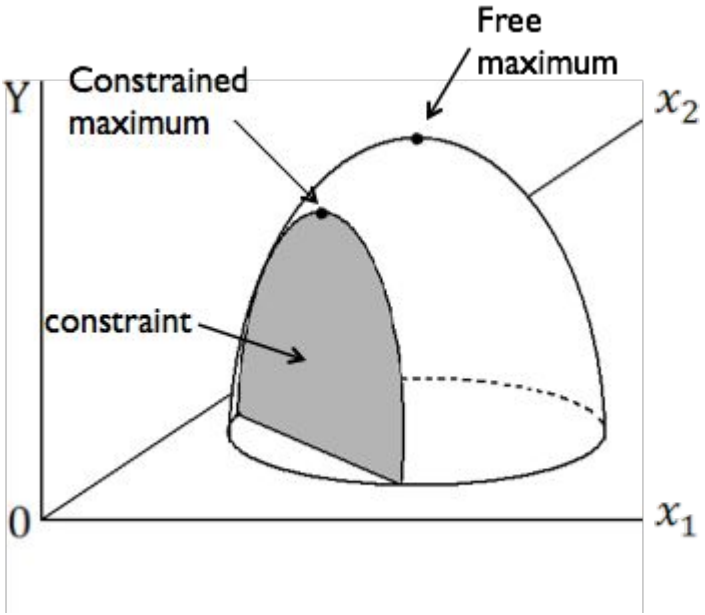
Solving the Optimization Problem

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Use *Lagrange multiplier* α_i is associated with every constraint : $y_i(wx_i + b) \geq 1$, dual problem

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

Refer: Christopher J. C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Lagrange Multipliers



$$y = f(x_1, x_2)$$
$$g(x_1, x_2) = 0$$
$$y_\lambda = f(x_1, x_2) + \lambda\big(g(x_1, x_2)\big)$$

$$\left.\begin{aligned} \frac{\partial y_\lambda}{\partial x_1} &= f_1 + \lambda g_1 = 0 \\ \frac{\partial y_\lambda}{\partial x_2} &= f_2 + \lambda g_2 = 0 \\ \frac{\partial y_\lambda}{\partial \lambda} &= g(x_1, x_2) = 0 \end{aligned}\right\}$$ Solve simultaneously for critical values

KKT conditions extend Lagrange conditions to handle inequalities.

The Optimization Problem Solution

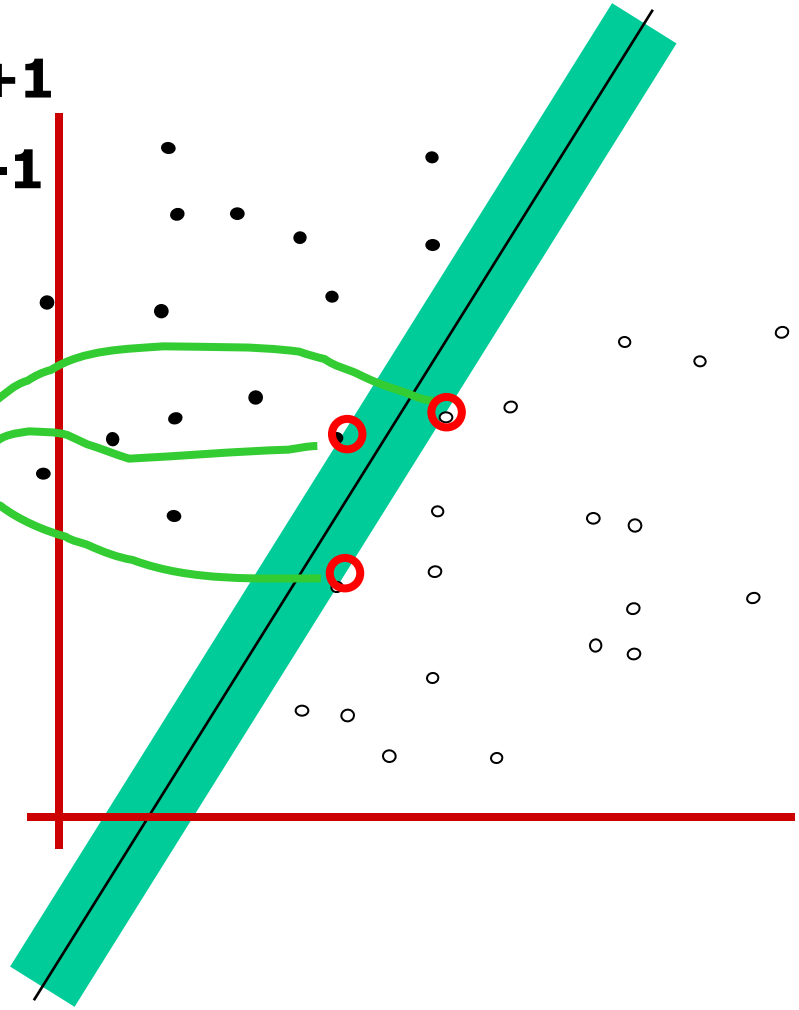
- The solution has the form:

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x}_i \qquad b = y_k - \mathbf{w}^T \mathbf{x}_k \qquad \text{for any } \mathbf{x}_k \text{ such that } \alpha_k \neq 0$$

- α_i must satisfy Karush-Kuhn-Tucker conditions:
  $\alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] = 0$, for any i
  - If $\alpha_i > 0$, $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0$, $\mathbf{x}_i$ is on the margin
  - If $y_i(\mathbf{w}^T x_i + b) > 1$, $\alpha_i = 0$
- Each non-zero α_i indicates that corresponding $\mathbf{x}_i$ is a **support vector**.

Maximum Margin

- denotes +1
- denotes -1



w, b depends only on Support Vectors via active constraints

$y_i(w^T x_k + b) - 1 = 0$

The Optimization Problem Solution

- To classify the new test point x, we use

$$f(\mathbf{x}) = \mathbf{w}\mathbf{x} + b = \Sigma \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

Find $\alpha_1 \ldots \alpha_N$ such that

$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and

(1) $\Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all α_i

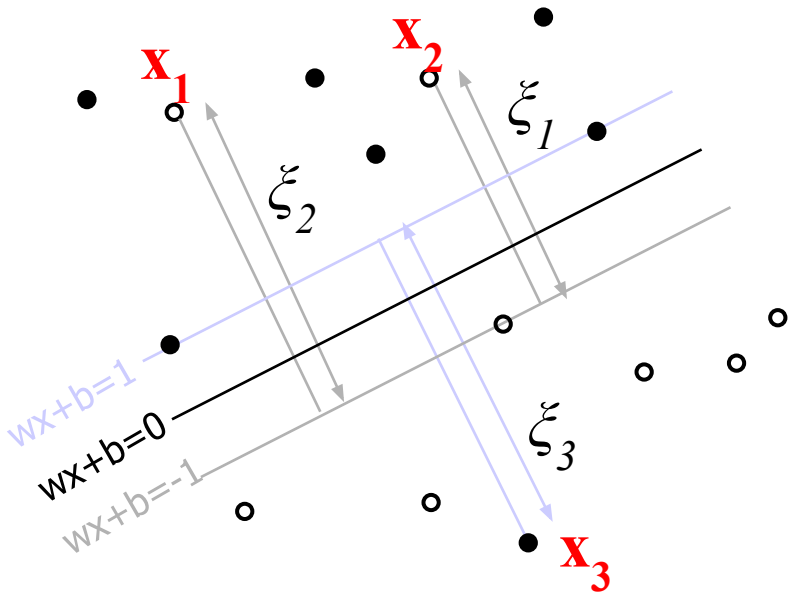
Soft Margin Classification

Previous constraints

$$y_i\,(\mathrm{w}^{\mathrm{T}}\mathrm{x}_{\mathrm{i}} + b) \geq 1$$

Slack variables ξ_i to allow misclassification:

$$y_i\,(\mathrm{w}^{\mathrm{T}}\mathrm{x}_{\mathrm{i}} + b) \geq 1 - \xi_i \qquad \xi_i \geq 0$$



What should our quadratic optimization criterion be?

We expect ξ_i to be small.

$$\Phi(\mathrm{w}) = \tfrac{1}{2}\,\mathrm{w}^{\mathrm{T}}\mathrm{w} + C\Sigma\xi_i$$

Hard Margin vs. Soft Margin

- The old formulation:

Find w and b such that

$\Phi(\mathrm{w}) = \frac{1}{2}\,\mathrm{w}^{\mathrm{T}}\mathrm{w}$ is minimized and for all $\{(\mathrm{x}_i, y_i)\}$

$y_i\,(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i + \mathrm{b}) \geq 1$

- The new formulation incorporating slack variables:

Find w and b such that

$\Phi(\mathrm{w}) = \frac{1}{2}\,\mathrm{w}^{\mathrm{T}}\mathrm{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathrm{x}_i, y_i)\}$

$y_i\,(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

- Similar solution can be obtained to that of hard margin
- Parameter C can be viewed as a way to control overfitting.

Non-linear SVMs

- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \varphi(x)$, optimization problem is similar:

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \varphi(x_i)^T\varphi(x_j)$ is maximized
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- Classifying function is:

$$f(x) = \Sigma\alpha_i y_i \varphi(x_i)^T\varphi(x) + b$$

- But relies on inner product $\varphi(x_i)^T\varphi(x)$

The "Kernel Trick"

- SVM relies on
  - *Linear:* $K(x_i,x_j)=x_i^T x_j$
  - *Non-linear:* $K(x_i,x_j)= \varphi(x_i)^T \varphi(x_j)$
- Feature mapping is time-consuming.
- Use a kernel function that directly obtains the value of inner product
- Feature mapping φ is not necessary in this case.

- Example:

  2-dimensional vectors $x=[x_1 \ x_2]$; let $K(x_i,x_j)=(1 + x_i^T x_j)^2$,

  It is inner product of $\varphi(x) = [1 \ x_1^2 \ \sqrt{2}\, x_1 x_2 \ x_2^2 \ \sqrt{2}x_1 \ \sqrt{2}x_2]$

Verify: $K(x_i,x_j)=(1+x_i^T x_j)^2$

$$= 1+ x_{i1}^2 x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \ x_{i1}^2 \ \sqrt{2}\, x_{i1}x_{i2} \ x_{i2}^2 \ \sqrt{2}x_{i1} \ \sqrt{2}x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2}\, x_{j1}x_{j2} \ x_{j2}^2 \ \sqrt{2}x_{j1} \ \sqrt{2}x_{j2}]$$

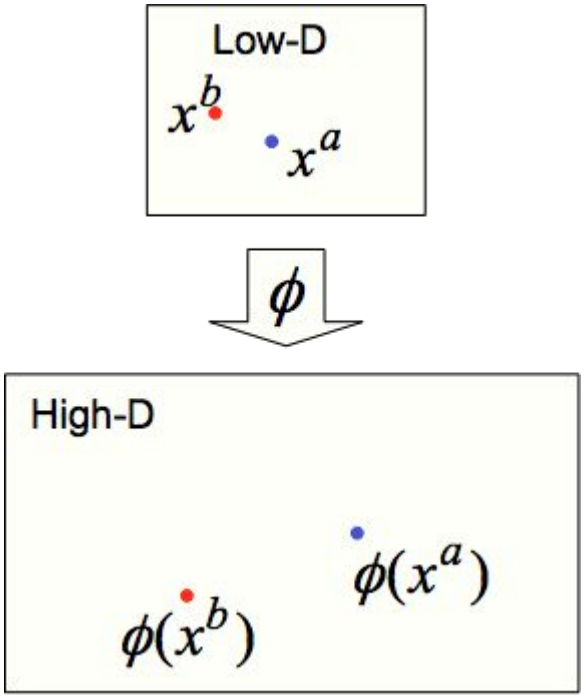
$$= \varphi(x_i)^T \varphi(x_j),$$

The "Kernel Trick"

$$K(x^a, x^b) = \phi(x^a).\phi(x^b)$$

Letting the kernel do the work

doing the scalar product in the obvious way



Examples of Kernel Functions

- **Linear: $K(x_i,x_j)= x_i^T x_j$**

- **Polynomial of power *p*: $K(x_i,x_j)= (1+ x_i^T x_j)^p$**

- **Gaussian (radial-basis function network):**

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

- **Sigmoid: $K(x_i,x_j)= \tanh(\beta_0 x_i^T x_j + \beta_1)$**

Kernel Requirements

The function *k*(x,y) is a valid kernel, if there exists a mapping f into a vector space (with a dot-product) such that *k* can be expressed as *k*(x,y)=f(x)•f(y)

Theorem: *k*(x,y) is a valid kernel if *k* is positive definite and symmetric (Mercer Kernel)

A function is P.D. if $\int K(\mathbf{x},\mathbf{y})f(\mathbf{x})f(\mathbf{y})d\mathbf{x}d\mathbf{y} \geq 0 \quad \forall f \in L_2$

In other words, the Gram matrix **K** (whose elements are $k(x_i,x_j)$)

must be positive definite for all x_i, x_j of the input space

Gram Matrix

- With KM-based learning, the **sole** information used from the training data set is the Kernel Gram Matrix

$$K_{training} = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & \dots & k(\mathbf{x}_1,\mathbf{x}_m) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & \dots & k(\mathbf{x}_2,\mathbf{x}_m) \\ \dots & \dots & \dots & \dots \\ k(\mathbf{x}_m,\mathbf{x}_1) & k(\mathbf{x}_m,\mathbf{x}_2) & \dots & k(\mathbf{x}_m,\mathbf{x}_m) \end{bmatrix}$$

- If the kernel is valid, K is symmetric definite-positive .

Choosing a Kernel

- Assume a categorization task: the ideal Kernel matrix is
  - $k(\mathbf{x}_i,\mathbf{x}_j)$=+1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class
  - $k(\mathbf{x}_i,\mathbf{x}_j)$=-1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different classes
  - → concept of target alignment (adapt the kernel to the labels), where alignment is the similarity between the current gram matrix and the ideal one ["two clusters" kernel]
- A certainly bad kernel is the diagonal kernel
  - $k(\mathbf{x}_i,\mathbf{x}_j)$=+1 if $\mathbf{x}_i = \mathbf{x}_j$
  - $k(\mathbf{x}_i,\mathbf{x}_j)$=0 elsewhere
  - All points are orthogonal : no more cluster, no more structure

Creating Kernels

$$K(\mathbf{x},\mathbf{y}) = \lambda K_1(\mathbf{x},\mathbf{y}) + (1-\lambda)K_2(\mathbf{x},\mathbf{y}) \quad 0 \le \lambda \le 1$$

$$K(\mathbf{x},\mathbf{y}) = a.K_1(\mathbf{x},\mathbf{y}) \quad a > 0$$

$$K(\mathbf{x},\mathbf{y}) = K_1(\mathbf{x},\mathbf{y}).K_2(\mathbf{x},\mathbf{y})$$

$$K(\mathbf{x},\mathbf{y}) = f(x).f(y) \quad f \text{ is real-valued function}$$

$$K(\mathbf{x},\mathbf{y}) = K_3(\boldsymbol{\varphi}(\mathbf{x}),\boldsymbol{\varphi}(\mathbf{y}))$$

$$K(\mathbf{x},\mathbf{y}) = \mathbf{x}'P\mathbf{y} \quad P \text{ symmetric definite positive}$$

$$K(\mathbf{x},\mathbf{y}) = \frac{K_1(\mathbf{x},\mathbf{y})}{\sqrt{K_1(\mathbf{x},\mathbf{x})}\sqrt{K_1(\mathbf{y},\mathbf{y})}}$$

Non-linear SVMs Mathematically

- **Dual problem formulation:**

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(x_i, x_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- **The solution is:**

$$f(x) = \Sigma\alpha_i y_i K(x_i, x_j) + b$$

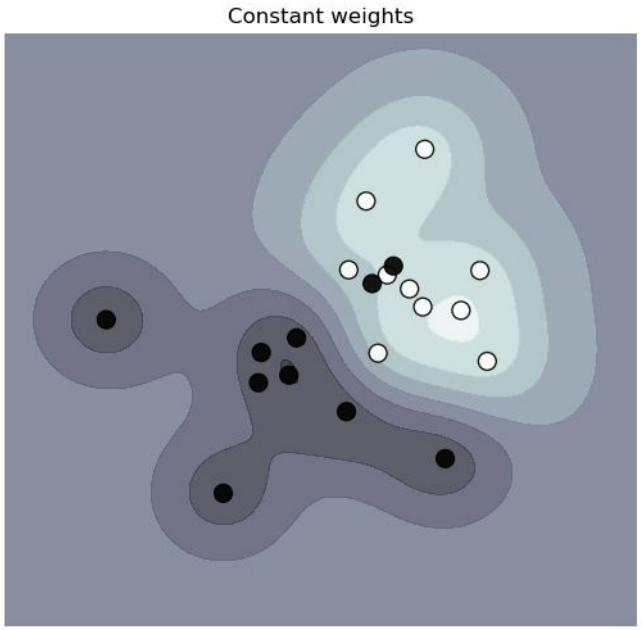
- **Optimization techniques for finding α_i's remain the same!**

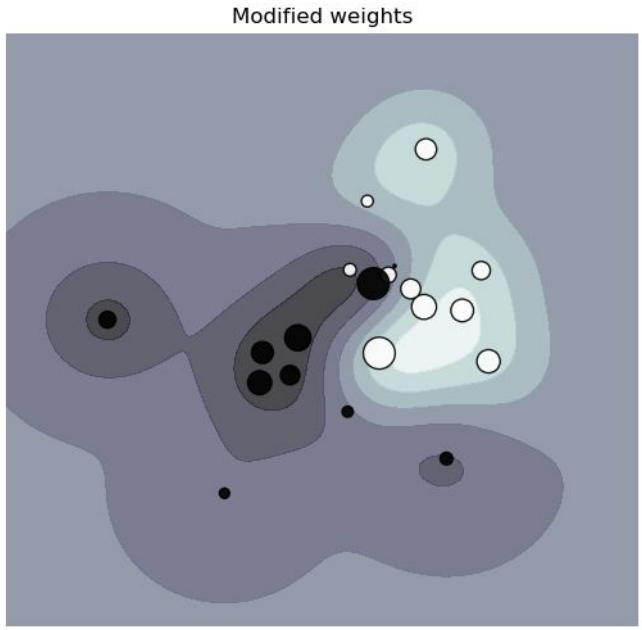
Weighted SVMs

To handle unbalanced training datasets, you can weight the datasets to apply more weight to the class that has fewer training instances.

The sample weighting rescales the C parameter.

Result: classifier puts more emphasis on getting these points right.





scikit-learn

SVM Multiclass

- SVMs only work for 2 classes!
- How to overcome this limitation:
- Train N SVMs for each class
  - One vs. all
  - Pick the classification that has the strongest positive prediction (furthest from the margin)
- Train N(N-1)/2 classifiers for every pair of classes
  - One vs. one
  - Choose the class with the most votes
  - More classifiers but training dataset for each of classifier is smaller
  - Preferred method

SVM Regression

SVM can be used as a replacement for linear regression (outputting continuous values rather than classes) by replacing LSE with epsilon insensitive error.

Least Squares Error (used in linear regression)

$$1/2\sum_{i=1}^{N}(t_i - y_i)^2 + 1/2\lambda||w||^2$$

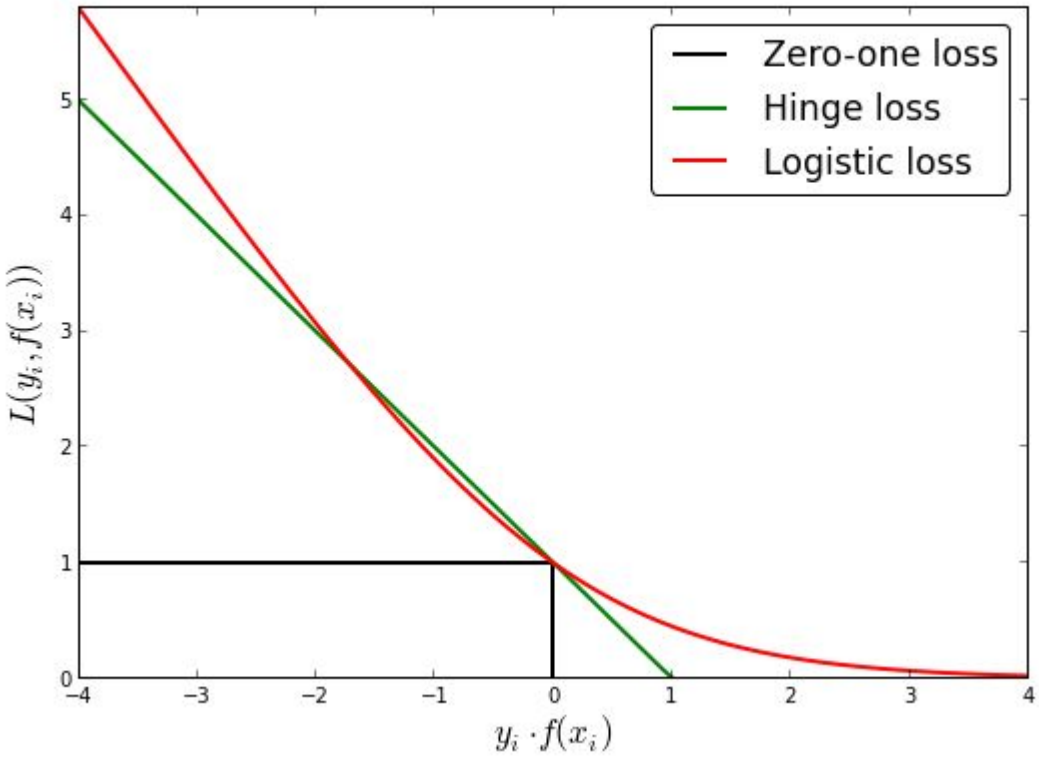
Epsilon Insensitive Error (returns 0 if the difference between the target and predicted value is less than epsilon)

$$\sum_{i=1}^{N} E_\epsilon(t_i - y_i) + 1/2\lambda||w||^2$$

Prediction (mu and alpha are constraint variables)

$$f(x) = \sum_{i=1}^{n}(\mu_i - \alpha_i K(x_i, z) + b)$$

Loss Functions



$$L(\hat{y}, y) = I(\hat{y} \neq y),$$

$$\ell(y) = \max(0, 1 - t \cdot y)$$

$$V(f(\vec{x}), y) = \frac{1}{\ln 2} \ln(1 + e^{-yf(\vec{x})})$$

Hinge loss is used by the SVM; logistic loss is used by logistic regression.

Platt Scaling

- Method of interpreting margin distances as probabilities
- A and B are estimated from data using MLE
- f(x) is the output of the SVM

$$\mathrm{P}(y = 1|x) = \frac{1}{1 + \exp(Af(x) + B)}$$

Sequential Minimal Optimization

Replacement for the quadratic programming techniques previously used that relies on selected pairs of constraints

$$0 \leq \alpha_1, \alpha_2 \leq C,$$
$$y_1\alpha_1 + y_2\alpha_2 = k,$$

1. Find a Lagrange multiplier that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
2. Pick a second multiplier and optimize the pair
3. Repeat steps 1 and 2 until convergence.
4. Process can be accelerated by selecting pairs intelligently

cvxopt

Python toolkit for convex optimization that can be used to program SVMs

1. Create H where $H_{i,j} = y^{(i)} y^{(j)} < x^{(i)}, x^{(j)} >$
2. Calculate w
3. Determine the set of support vectors
4. Calculate the intercept term
5. For each new point x' classify using w and b

Active Learning using SVM

“Incremental Relabeling for Active Learning with Noisy Crowdsourced Annotations". Liyue Zhao*, Gita Sukthankar and Rahul Sukthankar. *IEEE International Conference on Social Computing*. 2011. pp. 728--733.

Idea of active learning is reduce the amount of initial data and have the classifier “ask” for the most informative data samples

https://youtu.be/wuc8Iq9XhFE

Summary

- SVMs were developed as a way to handle high dimensional feature vectors which k-NN and many other methods performed badly on.
- Main limitation: difficult to scale to large datasets because of the reliance on quadratic programming
- Next assignment will be on SVMs