CSCE 633: Machine Learning

Lecture 9

Overview

- Lagrange Optimization
- Support Vector Machines
  - Linearly separable case
  - Non-separable case
  - Hinge Loss
  - Multi-class SVMs

Overview

- Lagrange Optimization
- Support Vector Machines
  - Linearly separable case
  - Non-separable case
  - Hinge Loss
  - Multi-class SVMs

Lagrange Optimization

Maximize with constraints

$$\max_{w} f(w)$$
$$\text{s.t. } g_i(w) \le 0, \quad i = 1, \dots, k$$
$$h_i(w) = 0, \quad i = 1, \dots, l$$

1) Formulate Lagrangian function (primal problem)
$L_p(w, \alpha, \beta) = f(w) - \sum_{i=1}^{k} \alpha_i g_i(w) - \sum_{i=1}^{l} \beta_i h_i(w)$
2) Maximize wrt primal variable w: $\frac{\vartheta L_p(w, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\vartheta w} = 0$
3) Substitute the primal variable w and express Lagrangian wrt dual variables α_i, β_i: $L_d(\boldsymbol{\alpha}, \boldsymbol{\beta})$
4) Minimize wrt dual variables and solve for dual variables (dual problem): $\frac{\vartheta L_d(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\vartheta \alpha_i} = 0$, $\frac{\vartheta L_d(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\vartheta \beta_i} = 0$
5) Recover the solution (for the primal variables) from the dual variables

Lagrange Optimization

Minimize with constraints

$$\min_{w} f(w)$$
$$\text{s.t.} \;\; g_i(w) \leq 0, \quad i = 1, \ldots, k$$
$$h_i(w) = 0, \quad i = 1, \ldots, l$$

1) Formulate Lagrangian function (primal problem)
$L_p(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$
2) Minimize wrt primal variable w: $\frac{\vartheta L_p(w,\boldsymbol{\alpha},\boldsymbol{\beta})}{\vartheta w} = 0$
3) Substitute the primal variable w and express Lagrangian wrt dual variables α_i, β_i: $L_d(\boldsymbol{\alpha}, \boldsymbol{\beta})$
4) Maximize wrt to dual variables and solve for dual variables (dual problem): $\frac{\vartheta L_d(\boldsymbol{\alpha},\boldsymbol{\beta})}{\vartheta \alpha_i} = 0$, $\frac{\vartheta L_d(\boldsymbol{\alpha},\boldsymbol{\beta})}{\vartheta \beta_i} = 0$
5) Recover the solution (for the primal variables) from the dual variables

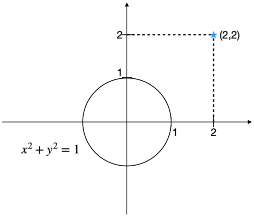
Lagrange Multipliers

Turn a constrained optimization problem into an unconstrained optimization problem by absorbing the constraints into the cost function, weighted by the *Lagrange multipliers*

Example: Find point on the circle $x^2 + y^2 = 1$ closest to the point $(2, 2)$

- Minimize $F(x, y) = (x - 2)^2 + (y - 2)^2$ subject to the constraint $x^2 + y^2 - 1 = 0$.
- Absorb the constraint into the cost function, after multiplying the *Lagrange multiplier* $\alpha > 0$:

$$F(x, y, \alpha) = (x - 2)^2 + (y - 2)^2 + \alpha(x^2 + y^2 - 1).$$



Lagrange Multipliers

Formulate Lagrangian (primal problem):
$F(x, y, \alpha) = (x-2)^2 + (y-2)^2 + \alpha(x^2 + y^2 - 1)$, with $\alpha > 0$
The optimization problem becomes $\max_\alpha \min_{x,y} F(x, y, \alpha)$
Minimize $\min_{x,y} F(x, y, \alpha)$

$$\frac{\partial F}{\partial x} = 2(x-2) + 2\alpha x = 0 \Rightarrow x = \frac{2}{1+\alpha}$$

$$\frac{\partial F}{\partial y} = 2(y-2) + 2\alpha y = 0 \Rightarrow y = \frac{2}{1+\alpha}$$

We substitute x, y in the Lagrangian and express it in terms of its dual form wrt α and maximize it

$$\frac{\partial F}{\partial \alpha} = x^2 + y^2 - 1 = 0 \Rightarrow \left(\frac{2}{1+\alpha}\right)^2 + \left(\frac{2}{1+\alpha}\right)^2 = 1 \Rightarrow \alpha = 2\sqrt{2} - 1$$

Recover the solution for the x, y $(x, y) = (1/\sqrt{2}, 1/\sqrt{2})$

Overview

- Lagrange Optimization
- Support Vector Machines
    - Linearly separable case
    - Non-separable case
    - Hinge Loss
    - Multi-class SVMs

Support Vector Machines: Linearly Separable Case

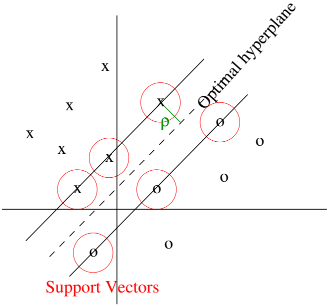
Setup for two classes

- Input: $\mathbf{x} \in \mathbb{R}^D$
- Output: $y \in \{-1, 1\}$
- Training data: $\mathcal{D}^{train} = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$
- Model:

$$f(\mathbf{x_n}) = \begin{cases} 1, & \text{if } \mathbf{w}^T\mathbf{x_n} + w_0 \geq 1 \\ -1, & \text{if } \mathbf{w}^T\mathbf{x_n} + w_0 \leq -1 \end{cases}$$

We do not only require instances to be on the right side of the hyperplane (which would be $\mathbf{w}^T\mathbf{x_n} + w_0 \geq 0$)
But we also want instances some distance away $\rightarrow$ margin
We want to maximize this margin for best generalization

Optimal Hyperplane and Support Vectors



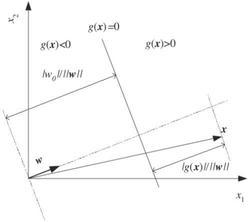
- **Margin of separation** ρ: distance between the separating hyperplane and the closest input point.
- **Support vectors**: input points closest to the separating hyperplane.

Optimal Hyperplane and Support Vectors

- The projection of a point $\mathbf{x}$ to hyperplane $\mathbf{w}$ with origin at w_0 is

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|} = \frac{\mathbf{w}^T\mathbf{x} + w_0}{\|\mathbf{w}\|}$$

(see Alpaydin 10.3)



Optimal Hyperplane and Support Vectors

- The projection of a point $\mathbf{x}$ to hyperplane $\mathbf{w}$ with origin at w_0 is

$$r = (\mathbf{w}^T \mathbf{x} + w_0)/\|\mathbf{w}\|$$

  (see Alpaydin 10.3)

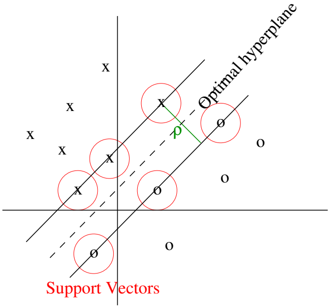
- For support vectors $\mathbf{x}_{(s)}$ on the right or left side of the hyperplane:

$$\mathbf{w}^T \mathbf{x} + w_0 = 1, \;\; \mathbf{w}^T \mathbf{x} + w_0 = -1$$

- Therefore the projection of support vectors $\mathbf{x}_{(s)}$ to hyperplane $\mathbf{w}$ is

$$r = \begin{cases} 1/\|\mathbf{w}\| & \text{if } y_s = +1 \\ -1/\|\mathbf{w}\| & \text{if } y_s = -1 \end{cases}$$

Optimal Hyperplane and Support Vectors



- Margin of separation *between two classes* is

$$\rho = 2r = \frac{2}{\|\mathbf{w}\|}.$$

- Thus, maximizing the margin of separation *between two classes* is equivalent to minimizing the Euclidean norm of the weight $\mathbf{w}$!

Primal Problem: Constrained Optimization

Linearly separable case

For the training set $\mathcal{D}^{train} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ find $\mathbf{w}$ and w_0 such that

- they minimize the *inverse* separation margin ($\frac{1}{\rho} = \frac{\|\mathbf{w}\|}{2}$) while satisfying a constraint (all examples are correctly classified):
  - Cost function: $\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$
  - Constraint: $y_n(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$ for $i = 1, 2, ..., N$

$$\boxed{\min \frac{1}{2}\|\mathbf{w}\|_2^2, \ \text{such that (s.t.)} \ y_n(\mathbf{w}^T\mathbf{x} + w_0) \geq 1 \ , \quad n = 1, \ldots, N}$$

This problem can be solved using the *method of Lagrange multipliers* (see next two slides)

Support Vector Machines: Linearly separable case

$$\min \tfrac{1}{2}\|\mathbf{w}\|_2^2, \text{ such that (s.t.) } y_n(\mathbf{w}^T\mathbf{x} + w_0) \geq 1, \quad n = 1, \ldots, N$$

1) Formulate Lagrangian function (primal problem)

$L_p = \frac{1}{2}\|\mathbf{w}\|_2^2 - \sum_{n=1}^{N} \alpha_n \left[y_n(\mathbf{w}^T\mathbf{x}_n + w_0) - 1\right]$

2) Minimize Lagrangian to solve for primal variables $\mathbf{w}, w_0$

$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x_n}$

$\frac{\partial L_p}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} \alpha_n y_n = 0$

3) Substitute the primal variables $\mathbf{w}, w_0$ into the Lagrangian and express in terms of dual variables α_n

$$\begin{aligned} L_d &= \frac{1}{2}\|\mathbf{w}\|_2^2 - \mathbf{w}^T \sum_{n=1}^{N} \alpha_n y_n \mathbf{x_n} - w_0 \sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n \\ &= -\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m \mathbf{x_n}^T\mathbf{x_m} + \sum_{n=1}^{N} \alpha_n \end{aligned}$$

Support Vector Machines: Linearly separable case

$$\min \tfrac{1}{2}\|\mathbf{w}\|_2^2, \text{ such that (s.t.) } y_n(\mathbf{w}^T\mathbf{x} + w_0) \geq 1 , \quad n = 1, \ldots, N$$

4) Maximize the Lagrangian with respect to dual variables (dual problem)

$$\max_{\alpha_n} L_d = \max_{\alpha_n} \left\{ -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{x_n}^T \mathbf{x_m} + \sum_{n=1}^{N} \alpha_n \right\}$$

s.t. $\sum_{n=1}^{N} \alpha_n y_n = 0$ and $\alpha_n \geq 0$, for $n = 1, \ldots, N$

- Solved numerically using quadratic optimization methods [see next slides]
- The dual depends on data size N, and particularly the number of support vectors R
- Most of the α_n will vanish with $\alpha_n = 0$, only a small percentage will have $\alpha_n > 0$
- The set of $\mathbf{x_n}$ whose $\alpha_n > 0$ are the support vectors

(Unconstraint) Coordinate Ascent

Unconstrained optimization problem

$$\max_{\boldsymbol{\alpha}} W(\alpha_1, \alpha_2, \ldots, \alpha_m)$$

We have already seen gradient descent (or ascent, if we negate the optimization function), now we consider another optimization method called coordinate ascent.

Loop until convergence

1 For $i = 1, \ldots, m$

1a $\alpha_i = arg \max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \hat{\alpha}_i, \ldots, \alpha_m)$

- In the innermost loop, hold all variables constant except for some fixed α_i
- Re-optimize W with respect to just the parameter α_i
- When *arg* max of the inner loop can be performed efficiently, coordinate ascent can be a fairly efficient algorithm

Sequential Minimal Optimization (SMO)

$$W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{x_n}^T\mathbf{x_m} + \sum_{n=1}^{N}\alpha_n$$

s.t. $\sum_{n=1}^{N}\alpha_n y_n = 0$ and $\alpha_n \geq 0$, for $n = 1, \ldots, N$

Loop until convergence

1 For $i = 1, \ldots, m$

- 1a Select some α_i and α_j to update
- 1b Re-optimize W wrt α_i and α_j while holding all other α_k's fixed

Let's assume that we optimize wrt α_1, α_2, while $\alpha_3, \ldots, \alpha_N$ are constant

- From the constraint:
  $\alpha_1 y_1 + \alpha_2 y_2 = \sum_{n=3}^{N}\alpha_n y_n = \zeta \rightarrow \alpha_1 = (\zeta - \alpha_2 y_2)/y_1$
- The objective is $W(\boldsymbol{\alpha}) = W\left(\left(\zeta - \alpha_2 y_2\right)/y_1, \alpha_2, \ldots, \alpha_N\right)$ (some quadratic function of $\alpha_2 \rightarrow$ easily solved by setting its derivative to zero)

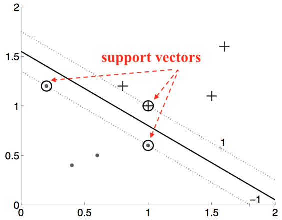
Support Vector Machines: Linearly separable case

$$\min \frac{1}{2}\|\mathbf{w}\|_2^2, \text{ such that (s.t.) } y_n(\mathbf{w}^T\mathbf{x} + w_0) \geq 1 \ , \quad n = 1, \ldots, N$$

5) Recover the solution (for the primal variables) from the dual variables

- Find $\mathbf{w}$: Substitute α_n from (4) to $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x_n}$
- Find w_0:
  - From $\mathbf{w}^T\mathbf{x_n} + w_0 = y_n$, where $\mathbf{x_n}$ is a support vector, calculate $w_0 = y_n - \mathbf{w}^T\mathbf{x_n}$.
  - For numerical stability average w_0 values estimated from all support vectors.

Support Vector Machines: Linearly separable case



- Samples $\mathbf{x_n}$ for which $\alpha_n = 0$
  - majority of samples
  - lie away from the hyperplane: $y_n(\mathbf{w}^T\mathbf{x_n} + w_0) > 1$
  - have no effect on the hyperplane
- Samples $\mathbf{x_n}$ for which $\alpha_n > 0$
  - support vectors
  - lie close to the hyperplane: $y_n(\mathbf{w}^T\mathbf{x_n} + w_0) = 1$
  - determine the hyperplane

Support Vector Machines: Linearly separable case

Testing

- Testing doesn't enforce a margin, i.e. $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$
- Choose C_1 if $g(\mathbf{x}) > 0$, C_2 otherwise
- Function $g(\mathbf{x_n}) = \mathbf{w}^T \mathbf{x} + w_0 = \sum_{m=1}^{N} \alpha_m y_m \mathbf{x_m}^T \mathbf{x_n} + w_0$ does not need explicit calculation of $\mathbf{w}$
  - $g(\mathbf{x_n})$ can be calculated from the dot product between the input vectors $\mathbf{x_m^T x_n}$
  - many of α_n's are zero, therefore computationally this is not very expensive

Overview

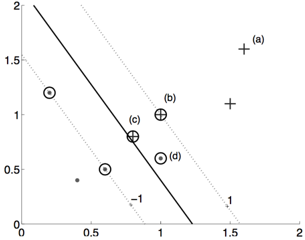
- Lagrange Optimization
- Support Vector Machines
  - Linearly separable case
  - Non-separable case
  - Hinge Loss
  - Multi-class SVMs

Support Vector Machines: Non-separable case

- If two classes are not linearly separable, we look for the hyperplane that yields the least error
- We define slack variables $\xi_n \geq 0$ which represent the deviation from the margin

$$y_n(\mathbf{w}^T \mathbf{x_n} + w_0) \geq 1 - \xi_n$$

- *Case (a):* Far away from the margin, $\xi_n = 0$
- *Case (b):* On the right side and on the margin, $\xi_n = 0$
- *Case (c):* On the right side, but in the margin, $0 \leq \xi_n \leq 1$
- *Case (d):* On the wrong side, $\xi_n \geq 1$



We incorporate the number of misclassifications $\#\{\xi_n > 1\}$ and the number of non-separable points $\#\{\xi_n > 0\}$ as a soft error $\sum_n \xi_n$.

Support Vector Machines: Non-separable case

$$\min_{\mathbf{w},\boldsymbol{\xi}} \left[\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\xi_n\right], \quad \text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{x} + w_0) \geq 1 - \xi_n \quad \text{and} \quad \xi_n \geq 0\,, \quad \forall n$$

1) Formulate Lagrangian function (primal problem)

$$L_p = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}\alpha_n\left[y_n(\mathbf{w}^T\mathbf{x}_n + w_0) - 1 + \xi_n\right] - \sum_{n=1}^{N}\mu_n\xi_n$$

2) Minimize Lagrangian to solve for primal variables $\mathbf{w}$, w_0

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N}\alpha_n y_n \mathbf{x_n}$$

$$\frac{\partial L_p}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N}\alpha_n y_n = 0$$

$$\frac{\partial L_p}{\partial \xi_n} = 0 \quad \Rightarrow \quad C - \alpha_n - \mu_n = 0$$

Support Vector Machines: Non-separable case

3) Substitute the primal variables $\mathbf{w}, w_0$ into the Lagrangian and express in terms of dual variables α_n

$$
\begin{aligned}
L_d = \frac{1}{2}\|\mathbf{w}\|_2^2 + \underbrace{C\sum_{n=1}^{N}\xi_n}_{} - \mathbf{w}^T\sum_{n=1}^{N}\alpha_n y_n \mathbf{x_n} \\
- w_0\sum_{n=1}^{N}\alpha_n y_n + \sum_{n=1}^{N}\alpha_n - \underbrace{\sum_{n=1}^{N}\alpha_n\xi_n}_{} - \underbrace{\sum_{n=1}^{N}\mu_n\xi_n}_{} \\
= -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{x_n}^T\mathbf{x_m} + \sum_{n=1}^{N}\xi_n\left(C - \alpha_n - \mu_n\right) + \sum_{n=1}^{N}\alpha_n \\
= \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{x_n}^T\mathbf{x_m}
\end{aligned}
$$

Support Vector Machines: Non-separable case

4) Maximize the Lagrangian with respect to dual variables (dual problem)

$$\max_{\alpha_n} \left\{ \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{x_n}^T \mathbf{x_m} \right\}$$

s.t. $\sum_{n=1}^{N} \alpha_n y_n = 0$ and $0 \leq \alpha_n \leq C$, for $n = 1, \ldots, N$

- Solved numerically using quadratic optimization methods
- $\alpha_n = 0$: instances at the correct side of the hyperplane with sufficient margin
- $\alpha_n > 0$: support vectors
  - $0 < \alpha_n < C$: instances lying on the margin
  - $\alpha_n = C$: instances in the margin or misclassified

Overview

- Lagrange Optimization
- Support Vector Machines
  - Linearly separable case
  - Non-separable case
  - Hinge Loss
  - Multi-class SVMs

Upper bound estimate of expected number of errors

- The number of support vectors is an upper bound estimate of the expected number of errors (Vapnik, 1995)

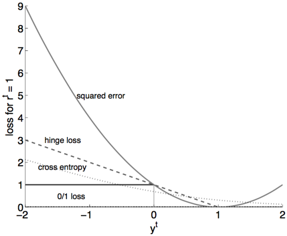
$$\mathbb{E}_N[P(error)] \leq \frac{\mathbb{E}_N[\#\text{support vectors}]}{N}$$

- The error rate depends on the number of support vectors and not the feature dimensionality

Support Vector Machines: Hinge Loss

- Decision rule: $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + w_0)$
  - $f(\mathbf{x}) = 1$, if $\mathbf{w}^T\mathbf{x} + w_0 > 0$
  - $f(\mathbf{x}) = -1$, if $\mathbf{w}^T\mathbf{x} + w_0 < 0$
- If $f(\mathbf{x})$ is the output and y_n the actual label

$$l^{\text{hinge}}(f(\mathbf{x}), y) = \begin{cases} 0 & \text{if } y(\mathbf{w}^T\mathbf{x} + w_0) \geq 1 \\ 1 - y(\mathbf{w}^T\mathbf{x} + w_0) & \text{otherwise} \end{cases}$$



- Hinge loss penalizes incorrectly classified samples more than 0/1 loss and cross-entropy loss
- Hinge loss penalizes correctly classified samples as well

Overview

- Lagrange Optimization
- Support Vector Machines
  - Linearly separable case
  - Non-separable case
  - Hinge Loss
  - Multi-class SVMs

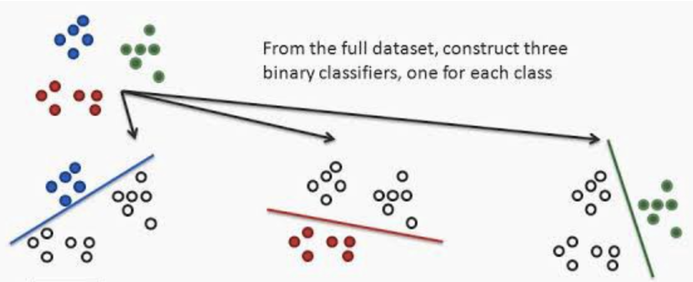
Multi-class kernel machines

One-vs-all approach

- Define K two-class problems, each separating one class from all others
- Learn K binary support vector machines $f_i(\mathbf{x})$, $i = 1, \ldots, K$
  - Class 1: Examples from class i
  - Class -1: Examples from all classes besides class i, i.e. $\{1, \ldots, K\} \setminus i$
- A sample point would be classified under a certain class if and only if that class's SVM accepted it and all other classes' SVMs rejected it

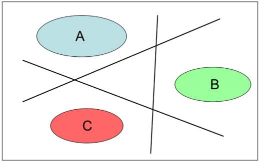
Multi-class kernel machines

One-vs-all approach



One-vs-all approach

When does one-vs-all fail?

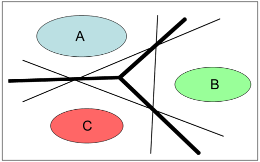


This method leaves regions of the space to be unaccounted for.

Multi-class kernel machines

One-vs-all approach

How to address regions of the space that are unaccounted for?



Decision during testing based on the class that has been selected with the highest confidence (e.g., highest distance to hyperplane).

Multi-class kernel machines

One-vs-one approach

- Define $K(K-1)$ two-class problems, each separating class i from class j
- Learn $K(K-1)$ binary support vector machines $f_{ij}(\mathbf{x})$
  - Class 1: Examples from class i
  - Class -1: Examples from class j
  - Note that $f_{ij} = -f_{ji}$
- During testing, the class chosen by the maximal number of SVMs is selected
- Alternatively (e.g., in case of tie)

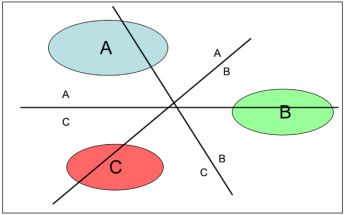
$$f(\mathbf{x}) = arg \max_i \left(\sum_j f_{ij}(\mathbf{x}) \right)$$

(selects the class based on the sum of distance from all hyperplanes)

Multi-class kernel machines

One-vs-one approach

Example of decision boundaries



Multi-class kernel machines

Comparison of one-vs-all and one-vs-one approach

- One-vs-one
  - requires $O(K^2)$ classifiers instead of $O(K)$
  - but each classifier is on average smaller $O(2\frac{N}{K})$
- One-vs.-all approach solves $O(K)$ separate problems, each of size $O(N)$

Multi-class kernel machines

Multi-class formulation

- Define K weights for each class $\mathbf{w_1}, \ldots, \mathbf{w_K}$ and K bias terms $w_{01}, \ldots, w_{0K}$
- Training data $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$, $y_n \in \{1, \ldots, K\}$
- Optimization criterion

$$\min_{\mathbf{w_1}, \ldots \mathbf{w_K}} \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{w_k}\|_2^2 + C \sum_{k=1}^{K} \sum_{n=1}^{N} \xi_{ni}$$

s.t. $\mathbf{w_{y_n}^T x_n} + w_{0y_n} \geq \mathbf{w_k^T x_n} + w_{0k} + 2 - \xi_{nk}, \forall k \neq y_n$

(i.e. so that the weight for each class yields a sufficient margin form the other classes)

What have we learnt so far

- SVM aims at finding the hyperplane from which instances have a margin of distance
- Prime and dual problem formulation (Lagrange multiplies)
- Support vectors: instances closest to separating hyperplane
- Linearly separable case: maximize margin of separation between two classes
- Non-separable case: look for the hyperplane that yields the least error (soft error)
  - Prime: minimizes Lagrangian wrt the primal variables of the problem
  - Dual: maximizes Lagrangian wrt multipliers
- Readings: Alpaydin 13.1-13.3