Support Vector Machines
EECE 580B

Lecture 26

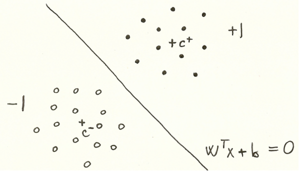
May 4, 2010

Jan Kodovský, Jessica Fridrich

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Nearest-mean Classifier

- Simple and intuitive classifier
- Decision function $\hat{y}(x) = \text{sign}\left(w^{\mathsf{T}} x + b\right)$



$$w = c^{+} - c^{-} = \frac{1}{t^{+}} \sum_{i \in D^{+}} x_i - \frac{1}{t^{-}} \sum_{i \in D^{-}} x_i$$

$$b = \frac{1}{2}\left(c^{+} - c^{-}\right)^{\mathsf{T}}\left(c^{+} + c^{-}\right)$$

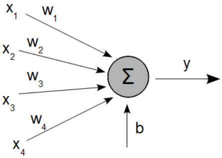
- ⊕ Efficient training
- ⊖ Poor performance
- ⊖ All the training points are equally important
- ⊖ Sensitivity to outliers

Perceptron

- Predecessor to neural networks, Rosenblatt 1950s
- Greedy search heuristics through $[w, b]$ space
- Update rules:

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b - \eta y_i r^2$$



- Dual point of view (counter of updates α_i)

$$w = \eta \sum \alpha_i y_i x_i$$

- $\alpha_i \approx 0$ easy points
- $\alpha >> 0$ difficult points

⊕ Different training points have different weights (robust to outliers)

⊖ No optimization, no relation to generalization abilities

⊖ Works only for separable data set (proven to converge)

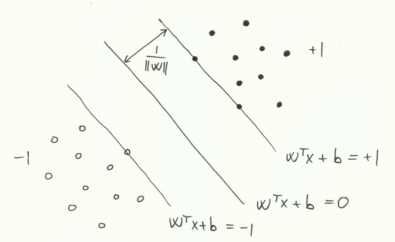
⊖ Different order of the training points $\Rightarrow$ different solution

Maximum-margin Classifier

- Optimal separating hyperplane (canonical form)
- Margin $\gamma = \frac{1}{||w||}$
- Optimization problem

$$\boxed{\begin{array}{ll} \underset{w,b}{\text{minimize}} & \frac{1}{2} w^{\mathsf{T}} w \\ \text{subject to} & y_i \left(w^{\mathsf{T}} x_i + b \right) \geq 1 \end{array}}$$

- Need for optimization background!



Optimization Theory

- Optimization basics
- QP, convexity $\Rightarrow$ no local optima
- Duality (Lagrangian theory)
  - Incorporating the constraints into the objective function
  - Lagrange multipliers, Lagrangian, Lagrange dual function $g(\lambda, \nu)$
  - Lower bound property
- Lagrange dual problem

$$\boxed{\begin{array}{ll} \underset{\lambda,\nu}{\text{maximize}} & g(\lambda,\nu) \\ \text{subject to} & \lambda \geq 0 \end{array}}$$

- Weak and strong duality $p^* = d^*$
  - Saddle point interpretation
  - The order of optimization does not matter

$$\sup_{\lambda \geq 0} \inf_{x} L(x,\lambda) = \inf_{x} \sup_{\lambda \geq 0} L(x,\lambda)$$

Optimization Theory

- KKT conditions (complementary slackness)

  Strong duality, (x, λ, ν) optimal $\Rightarrow$ KKT holds
  $(\bar{x}, \bar{\lambda}, \bar{\nu})$ satisfies KKT & convex problem $\Rightarrow$ $(\bar{x}, \bar{\lambda}, \bar{\nu})$ optimal

- Sensitivity of the solution to the constraint perturbations

$$\boxed{\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \end{array}} \quad \Rightarrow \quad \boxed{\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & g_i(x) \leq \alpha_i \end{array}}$$

- Global result

$$p^*(\alpha) \geq p^* - \lambda^{*\mathsf{T}} \alpha$$

- To remember

  λ_i^* large $\Rightarrow$ important constraint $\Rightarrow$ don't tighten it
  λ_i^* small $\Rightarrow$ less important $\Rightarrow$ relaxing won't help much

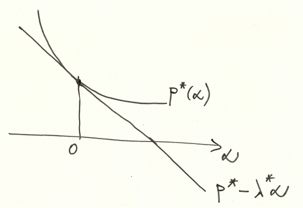
Optimization Theory

- Local perturbation analysis
  - Quantitative measure of 'how active' an active constraint is at the optimum x^*

$$\lambda_i^* = -\frac{\partial p^*(0)}{\partial \alpha_i}$$

- To remember

$$\boxed{\lambda_i^* = 0 \quad \Rightarrow \quad \text{perturbation does not affect the solution}}$$



Back to SVMs

- Maximizing the margin $\rightarrow$ the dual domain

$$\begin{array}{ll} \underset{w,b}{\text{minimize}} & \frac{1}{2} w^{\mathsf{T}} w \\ \text{subject to} & y_i \left(w^{\mathsf{T}} x_i + b \right) \geq 1 \end{array} \Rightarrow \begin{array}{ll} \underset{\alpha}{\text{maximize}} & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^{\mathsf{T}} x_j \\ \text{subject to} & \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \end{array}$$

- Nice QP convex problems
- Dual problem is always feasible ($\alpha_i = 0 \ \forall i$)
- Connection to the primal variables:

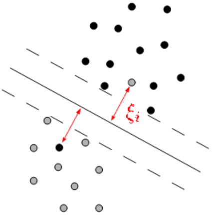
$$w^* = \sum_i \alpha_i^* y_i x_i, \ b^* = \frac{1}{|S|} \sum_i \left(y_i - {w^*}^{\mathsf{T}} x_i \right)$$

- Complementary slackness $\Rightarrow$ sparseness of the solution
- Everything in terms of the dot-products

Soft-margin SVM

- Introduction of slack variables

$$\begin{array}{ll} \underset{w,b}{\text{minimize}} & \frac{1}{2} w^{\mathsf{T}} w \\ \text{subject to} & y_i \left(w^{\mathsf{T}} x_i + b \right) \geq 1 \end{array}$$



$\Rightarrow$

$$\begin{array}{ll} \underset{w,b,\xi}{\text{minimize}} & \frac{1}{2} w^{\mathsf{T}} w + C \sum_i \xi_i^k \\ \text{subject to} & y_i \left(w^{\mathsf{T}} x_i + b \right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

L1-SVM

- Dual domain

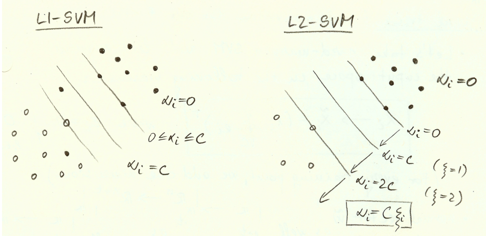
$$\boxed{\begin{array}{ll} \underset{\alpha}{\text{maximize}} & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i^\mathsf{T} x_j \\ \text{subject to} & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \end{array}}$$

- Box-constraint interpretation
- Importance of the outliers is limited to C
- Complementary slackness $\rightarrow$ bounded / unbounded SVs
- b^* to be calculated only over unbounded SVs
- How to choose C $\rightarrow$ cross-validation, grid search

L2-SVM

$$
\begin{array}{ll}
\underset{\alpha}{\text{maximize}} & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \left(x_i^T x_j + \frac{1}{C}\delta_{i,j}\right) \\
\text{subject to} & \sum_i \alpha_i y_i = 0 \\
& 0 \le \alpha_i
\end{array}
$$

- Kernel matrix is PD $\Rightarrow$ unique solution
- b^* to be calculated over all SVs again



Non-linear Classification

- Idea: perform classification in a feature space $\mathscr{F}$

$$\boxed{\mathscr{X}} \xrightarrow{\Phi} \boxed{\mathscr{F}}$$

$$\begin{array}{ll} \underset{\alpha}{\text{maximize}} & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle \\ \text{subject to} & \sum_i \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \end{array}$$

- Decision function $\hat{y}(x) = \text{sign}\left\{\sum_i \alpha_i^* y_i \langle \Phi(x_i), \Phi(x) \rangle + b^*\right\}$
- Non-linear 'preprocessing' Φ should be part of the SVM
- Curse of dimensionality?

  Computational degradation $\rightarrow$ kernel trick

  Degradation of generalization $\rightarrow$ margin maximization

Non-linear Classification

- What if there exists a mapping $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ such that

$$k(x,z) = \langle \Phi(x), \Phi(z) \rangle \quad \forall x, z \in \mathscr{X}$$

$$\boxed{\begin{array}{ll} \underset{\alpha}{\text{maximize}} & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} & \sum_i \alpha_i y_i = 0 \\ & 0 \le \alpha_i \end{array}}$$

- Decision function $\hat{y}(x) = \text{sign}\left\{\sum_i \alpha_i^* y_i k(x_i, x) + b^*\right\}$
- Implications:
  1. No need to explicitely map everything to $\mathscr{F}$
  2. We don't even have to know Φ
  3. Dimensionality of $\mathscr{F}$ is not necessarily important

Functional Analysis

- Question: How to obtain such a mapping k?
- We don't want to construct it from Φ

  $\rightarrow$ Find conditions on k that would guarantee the existence of Φ and $\mathscr{F}$

- Vector spaces (space of functions), dot-product, Hilbert spaces
- Cauchy-Schwarz inequality
- Kernel matrix, PSD kernel
- Theorem:

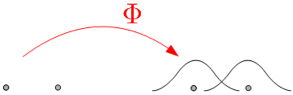
$$\text{PSD kernel } k \quad \Leftrightarrow \quad \mathscr{F}, \Phi$$

$$\text{such that } k(x,z) = \langle \Phi(x), \Phi(z) \rangle \quad \forall x,z \in \mathscr{X}$$

Proof of the Theorem

$$\boxed{\Phi(z) = k(\cdot, z)}$$

1. Define Φ
   - Partially evaluated kernel
   - $\Phi : \mathscr{X} \to \mathbb{R}^{\mathscr{X}}$
   - $\Phi(x) = k(x,z) \equiv \Phi^x : \mathscr{X} \to \mathbb{R}$



2. Turn $\Phi(\mathscr{X})$ into a vector space
   - $\mathscr{F} = \text{span}\{k(\cdot,z) | z \in \mathscr{X}\}$
3. Define $\langle \cdot, \cdot \rangle$ on $\mathscr{F}$ $\to$ turn it into a Hilbert space
   - $f = \sum_{i=1}^{m} \alpha_i k(\cdot, z_i)$, $g = \sum_{j=1}^{m'} \beta_j k(\cdot, z'_j)$
   - $\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k\left(z_i, z'_j\right)$
   - Reproducing property of kernels

$$\begin{aligned} \langle f, k(\cdot,z) \rangle &= f(z) \\ \langle k(\cdot,z), k(\cdot,\bar{z}) \rangle &= k(z,\bar{z}) \\ \langle \Phi(z), \Phi(\bar{z}) \rangle &= k(z,\bar{z}) \end{aligned}$$

Constructing New Kernels

- Application of operations preserving PSD properties of matrices
- Rules for constructing new kernels

R1. $k(x,z) = k_1(x,z) + k_2(x,z)$

R2. $k(x,z) = C \cdot k_1(x,z), \quad C \geq 0$

R3. $k(x,z) = C, \quad C \geq 0$

R4. $k(x,z) = k_1(x,z) \cdot k_2(x,z)$

R5. $k(x,z) = p(k_1(x,z)), \quad p \ldots$ polynomial with positive coeffs.

R6. $k(x,z) = f(x) \cdot f(z), \quad \forall f : \mathscr{X} \to \mathbb{R}$

R7. $k(x,z) = k_1(\Phi(x), \Phi(z)), \quad \forall \Phi : \mathscr{X} \to \mathbb{R}^m$

R8. $k(x,z) = \exp\{k_1(x,z)\}$

Constructing New Kernels

- Linear kernel

$$k(x, z) = x^{\mathsf{T}} z$$

- Polynomial kernel

$$k(x, z) = \left(x^{\mathsf{T}} z + 1\right)^{d}$$

- Gaussian kernel

$$k(x, z) = \exp\left\{-\gamma ||x - z||^{2}\right\}$$

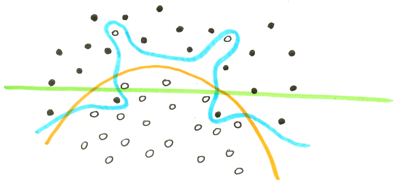
⊖ Additional parameter to be optimized through grid search

Statistical Learning Theory

- Question: So why is margin maximization a good strategy?
- V. Vapnik: The nature of statistical learning theory, 1995

$$R(\lambda) \leq R_{\text{emp}}(\lambda) + \Phi\left(\frac{h}{t}\right)$$

- Trade-off between complexity of the solution and the empirical risk
- VC dimension, SRM principle
- Large margin $\Rightarrow$ low VC dimension $h \Rightarrow$ low complexity term $\Phi\left(\frac{h}{t}\right)$



Implementation

- Implementation of SVM = implementation of the training phase
- No local optima $\Rightarrow$ iterative methods
- Stopping criteria

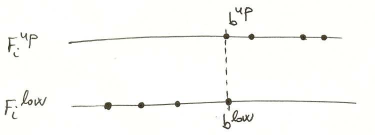
  Monitoring the feasibility gap $P(\alpha) - D(\alpha)$

  Monitoring the KKT conditions $\rightarrow$ exact form

  - $F_i(\alpha) = y_i - \sum_j \alpha_j y_j k(x_i, x_j)$

    $\rightarrow$ obtain either lower (F_i^{low}) or upper (F_i^{up}) bound on b
  - $b^{low} = \max_i F_i^{low}, \quad b^{up} = \min_i F_i^{up}$

$$\boxed{b^{\mathrm{up}} \geq b^{\mathrm{low}} - \tau}$$

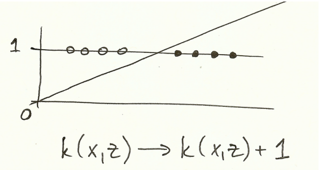


Implementation

- Stochastic gradient ascent

$$\alpha_i^{t+1} = \alpha_i^t + \eta_i \frac{\partial D(\alpha^t)}{\partial \alpha_i}$$

- Problem: constraint violation
  1. $\sum_i \alpha_i y_i = 0 \Rightarrow k(x,z) = k(x,z) + 1$
  2. $0 \leq \alpha_i \leq C \Rightarrow$ truncating
- Resulting algorithm: kernel-adatron



⊕ Never leaves feasible region

⊕ Shown to converge

⊕ Simple, works for small problems

⊖ Smaller margin in the augmented space

⊖ Can be slow or oscilate before converging

Subset Selection Methods

- Idea: work only with the subset of the training points (repeatedly)
- Chunking – working set W, adding M points after every run
- Decomposition – W has constant size, freezing other variables

---

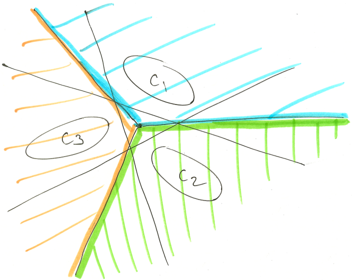
- Sequential Minimal Optimization (SMO)
  = decomposition with $|W| = 2$
  - John Platt, 1998
  - $\sum_i \alpha_i y_i = 0$ can be easily mantained
  - SVM(W) has analytical solution $\Rightarrow$ no QP solver needed!
  - Smart selection heuristics may speed up the algorithm

Multi-class SVM

- One-against-all SVM
  - n classes $\Rightarrow$ n binary subproblems (i vs. all remaining)
  - Unclassifiable regions

    Membership functions (fuzzy approach)

    Decision-tree based SVM



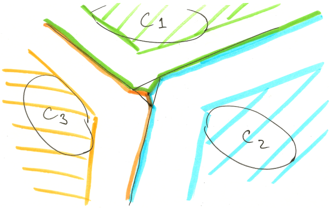
⊕ Small number of subproblems

⊖ Imbalanced data

⊖ All subproblems are large

Multi-class SVM

- Pairwise SVM
  - n classes $\Rightarrow \binom{n}{2} = \frac{n(n-1)}{2}$ binary subproblems
  - Unclassifiable regions are smaller



⊕ Balanced data

⊕ Smaller subproblems

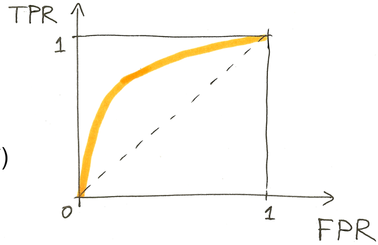
⊕ Fewer SVs, easier decision boundaries

⊖ For large n large number of subproblems

Warning: Performance is highly problem dependent!

Other Topics

- Data preprocessing is important!
- Receiver Operating Characteristic (ROC curve)



- Novelty detection – one-class SVM
  - Separate data from the origin (in $\mathscr{F}$)
  - Useful also for outlier detection
- Virtual SVM
  - Use the problem invariants for generating new points

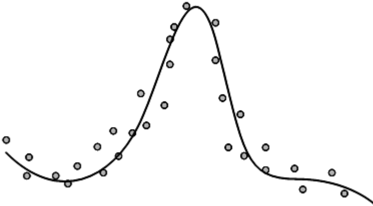
Support Vector Regression

- $(x, y) \in \mathscr{X} \times \mathbb{R}$
- ε-insensitive loss function

$$L^{\varepsilon}(y, w^{\mathsf{T}} x + b) = \max\left\{0, \left|y - w^{\mathsf{T}} x - b\right| - \varepsilon\right\}$$

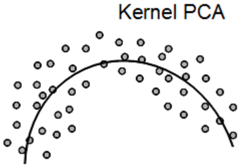
- Larger margin = flatter function
- Optimization problem:

$$\boxed{\begin{array}{ll} \underset{w,b,\xi,\xi'}{\text{minimize}} & \frac{1}{2} w^{\mathsf{T}} w + C \sum_i \left(\xi_i + \xi_i'\right) \\ \text{subject to} & y_i - w^{\mathsf{T}} x_i - b \leq \varepsilon + \xi_i \\ & w^{\mathsf{T}} x_i + b - y_i \leq \varepsilon + \xi_i' \\ & \xi_i \geq 0 \\ & \xi_i' \geq 0 \end{array}}$$



Kernel PCA

- Standard PCA
  - Orthogonal Linear transformation
  - After PCA, data are uncorrelated and sorted by variance
  - Non-parametric dimensionality reduction method
  - Principal components = projections into the eigenvectors of the covariance matrix
- Kernel PCA = standard PCA in $\mathscr{F}$
- Kernel trick $\Rightarrow$ need for dot-products



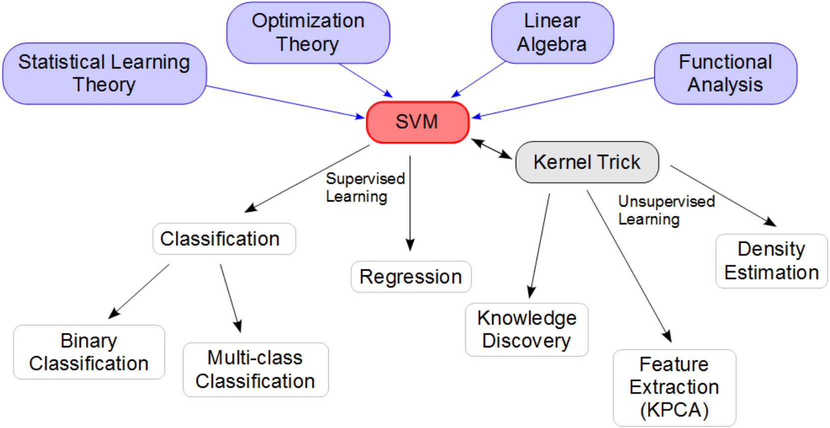
⊕ No non-linear optimization needed

⊖ Difficulties with data reconstruction

Course Objectives

- Understand the core concepts SVMs are built on
- Gain practical experience with using SVM for classification problems
- Implement your own SVM machine (in Matlab)
- Be aware of potential issues when using SVMs
- Be able to use publicly available SVM libraries (and understand them)

Big Picture



Optimization Theory
Linear Algebra
Statistical Learning Theory
Functional Analysis
SVM
Kernel Trick
Supervised Learning
Unsupervised Learning
Classification
Regression
Knowledge Discovery
Density Estimation
Binary Classification
Multi-class Classification
Feature Extraction (KPCA)