Nearest-mean Classifier

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

\[
\begin{align*}
  w &= c^+ - c^- = \frac{1}{t^+} \sum_{i \in D^+} x_i - \frac{1}{t^-} \sum_{i \in D^-} x_i \\
  b &= \frac{1}{2} (c^+ - c^-)^T (c^+ + c^-)
\end{align*}
\]

- 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 \(\alpha_i\))
  \[ w = \eta \sum \alpha_i y_i x_i \]
- \(\alpha_i \approx 0\) 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

\[
\text{minimize} \quad \frac{1}{2} w^T w \\
\text{subject to} \quad y_i (w^T x_i + b) \geq 1
\]

- 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

$$\begin{align*}
\text{maximize} & \quad g(\lambda, \nu) \\
\text{subject to} & \quad \lambda \geq 0
\end{align*}$$

- 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, \((x, \lambda, \nu)\) 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**

  \[
  \begin{align*}
  \text{minimize} & \quad f(x) \\
  \text{subject to} & \quad g_i(x) \leq 0 \\
  \end{align*}
  \]

  \[
  \begin{align*}
  \Rightarrow \quad \text{minimize} & \quad f(x) \\
  \text{subject to} & \quad g_i(x) \leq \alpha_i \\
  \end{align*}
  \]

- **Global result**

  \[p^*(\alpha) \geq p^* - \lambda^* \T \alpha\]

- **To remember**

  \[\lambda_i^* \text{ large } \Rightarrow \text{ important constraint } \Rightarrow \text{ don’t tighten it}\]

  \[\lambda_i^* \text{ small } \Rightarrow \text{ less important } \Rightarrow \text{ relaxing won’t help much}\]
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

$$\lambda_i^* = 0 \implies \text{perturbation does not affect the solution}$$
Maximizing the margin → the dual domain

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_i (w^T x_i + b) \geq 1
\end{align*}
\]

\[
\begin{align*}
\Rightarrow \quad \text{maximize} & \quad \sum_{i} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{subject to} & \quad \sum_{i} \alpha_i y_i = 0, \quad 0 \leq \alpha_i
\end{align*}
\]

Nice QP convex problems

Dual problem is always feasible (\(\alpha_i = 0 \ \forall i\))

Connection to the primal variables:

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

Complementary slackness ⇒ sparseness of the solution

Everything in terms of the dot-products
Soft-margin SVM

- Introduction of slack variables

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_i (w^T x_i + b) \geq 1
\end{align*}
\]  

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w + C \sum_i \xi_i \\
\text{subject to} & \quad y_i (w^T x_i + b) \geq 1 - \xi_i \\
& \quad \xi_i \geq 0
\end{align*}
\]
L1-SVM

Dual domain

maximize

subject to

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

\[
\begin{align*}
\text{maximize} \quad & \sum_{i} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \left( x_i^T x_j + \frac{1}{C} \delta_{i,j} \right) \\
\text{subject to} \quad & \sum_{i} \alpha_i y_i = 0 \\
& 0 \leq \alpha_i
\end{align*}
\]

- Kernel matrix is PD ⇒ unique solution
- \( b^* \) to be calculated over all SVs again
Non-linear Classification

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

maximize $\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle$
subject to $\sum_i \alpha_i y_i = 0$
$0 \leq \alpha_i$

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' $\Phi$ 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 \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \) such that

\[
k(x, z) = \langle \Phi(x), \Phi(z) \rangle \quad \forall x, z \in \mathcal{X}
\]

\[
\begin{align*}
\text{maximize} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\
\text{subject to} \quad & \sum_i \alpha_i y_i = 0 \\
& 0 \leq \alpha_i
\end{align*}
\]

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 explicitly map everything to \( \mathcal{F} \)
2. We don’t even have to know \( \Phi \)
3. Dimensionality of \( \mathcal{F} \) is not necessarily important
Question: How to obtain such a mapping $k$?
We don’t want to construct it from $\Phi$

→ Find conditions on $k$ that would guarantee the existence of $\Phi$ and $\mathcal{F}$

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

Theorem:

$$PSD \text{ kernel } k \iff \mathcal{F}, \Phi$$

such that $k(x, z) = \langle \Phi(x), \Phi(z) \rangle \quad \forall x, z \in \mathcal{X}$
Proof of the Theorem

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

2. Turn $\Phi(\mathcal{X})$ into a vector space
   - $\mathcal{F} = \text{span}\{k(\cdot, z) | z \in \mathcal{X}\}$

3. Define $\langle \cdot, \cdot \rangle$ on $\mathcal{F} \rightarrow$ turn it into a Hilbert space
   - $f = \sum_{i=1}^{m} \alpha_i k(\cdot, z_i), \ g = \sum_{j=1}^{m'} \beta_j k(\cdot, z'_j)$
   - $\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(z_i, z'_j)$
   - Reproducing property of kernels
     $\langle f, k(\cdot, z) \rangle = f(z)$
     $\langle k(\cdot, z), k(\cdot, \bar{z}) \rangle = k(z, \bar{z})$
     $\langle \Phi(z), \Phi(\bar{z}) \rangle = k(z, \bar{z})$
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. \( k(x, z) = p(k_1(x, z)), \quad p \ldots \text{polynomial with positive coeffs.} \)

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

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

R8. \( k(x, z) = \exp \{k_1(x, z)\} \)
Constructing New Kernels

- **Linear kernel**
  \[ k(x, z) = x^T z \]

- **Polynomial kernel**
  \[ k(x, z) = (x^T z + 1)^d \]

- **Gaussian kernel**
  \[ k(x, z) = \exp \left\{ -\gamma ||x - z||^2 \right\} \]

- Additional parameter to be optimized through grid search
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 ⇒ low VC dimension \( h \) ⇒ 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 ($F_i^{\text{low}}$) or upper ($F_i^{\text{up}}$) bound on $b$

  $b^{\text{low}} = \max_i F_i^{\text{low}}$, $b^{\text{up}} = \min_i F_i^{\text{up}}$

\[
b^{\text{up}} \geq b^{\text{low}} - \tau
\]
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 oscillate 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 maintained
- 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 \((\mathbb{F})\)
  - Useful also for outlier detection

- Virtual SVM
  - Use the problem invariants for generating new points
Support Vector Regression

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

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

- Larger margin = flatter function
- Optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w + C \sum_i (\xi_i + \xi'_i) \\
\text{subject to} & \quad y_i - w^T x_i - b \leq \varepsilon + \xi_i \\
& \quad w^T x_i + b - y_i \leq \varepsilon + \xi'_i \\
& \quad \xi_i \geq 0 \\
& \quad \xi'_i \geq 0
\end{align*}
\]
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 $\mathcal{F}$
- 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

- Statistical Learning Theory
- Optimization Theory
- Linear Algebra
- Functional Analysis

SVM

- Supervised Learning
- Kernel Trick
- Unsupervised Learning
- Density Estimation

- Classification
  - Binary Classification
  - Multi-class Classification
- Regression
- Knowledge Discovery
- Feature Extraction (KPCA)