Overview

L1: Machine learning and probability theory
   Introduction to pattern recognition, classification, regression, novelty detection, probability theory, Bayes rule, inference

L2: Density estimation and Parzen windows
   Nearest Neighbor, Kernels density estimation, Silverman’s rule, Watson Nadaraya estimator, crossvalidation

L3: Perceptron and Kernels
   Hebb’s rule, perceptron algorithm, convergence, kernels

L4: Support Vector estimation
   Geometrical view, dual problem, convex optimization, kernels

L5: Support Vector estimation
   Regression, Novelty detection

L6: Structured Estimation
   Sequence annotation, web page ranking, path planning, implementation and optimization
Novelty Detection
- Basic idea
- Optimization problem
- Stochastic Approximation
- Examples

Regression
- Additive noise
- Regularization
- Examples
- SVM Regression
- Quantile Regression
Data
Observations \((x_i)\) generated from some \(P(x)\), e.g.,
- network usage patterns
- handwritten digits
- alarm sensors
- factory status

Task
Find unusual events, clean database, distinguish typical examples.
Applications

Network Intrusion Detection
Detect whether someone is trying to hack the network, downloading tons of MP3s, or doing anything else unusual on the network.

Jet Engine Failure Detection
You can’t destroy jet engines just to see how they fail.

Database Cleaning
We want to find out whether someone stored bogus information in a database (typos, etc.), mislabelled digits, ugly digits, bad photographs in an electronic album.

Fraud Detection
Credit Cards, Telephone Bills, Medical Records

Self calibrating alarm devices
Car alarms (adjusts itself to where the car is parked), home alarm (furniture, temperature, windows, etc.)
Novelty Detection via Densities

Key Idea
- Novel data is one that we don’t see frequently.
- It must lie in low density regions.

Step 1: Estimate density
- Observations $x_1, \ldots, x_m$
- Density estimate via Parzen windows

Step 2: Thresholding the density
- Sort data according to density and use it for rejection
- Practical implementation: compute

\[ p(x_i) = \frac{1}{m} \sum_j k(x_i, x_j) \text{ for all } i \]

and sort according to magnitude.
- Pick smallest $p(x_i)$ as novel points.
Typical Data
Outliers

20578
75387
42970
70426
58408
A better way . . .
A better way . . .

Problems

- We do not care about estimating the density properly in regions of high density (waste of capacity).
- We only care about the relative density for thresholding purposes.
- We want to eliminate a certain fraction of observations and tune our estimator specifically for this fraction.

Solution

- Areas of low density can be approximated as the level set of an auxiliary function. No need to estimate $p(x)$ directly — use proxy of $p(x)$.
- Specifically: find $f(x)$ such that $x$ is novel if $f(x) \leq c$ where $c$ is some constant, i.e. $f(x)$ describes the amount of novelty.
Maximum Distance Hyperplane

Idea  Find hyperplane, given by $f(x) = \langle w, x \rangle + b = 0$ that has maximum distance from origin yet is still closer to the origin than the observations.

**Hard Margin**

\[
\text{minimize } \frac{1}{2} \| w \|^2 \\
\text{subject to } \langle w, x_i \rangle \geq 1
\]

**Soft Margin**

\[
\text{minimize } \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{m} \xi_i \\
\text{subject to } \langle w, x_i \rangle \geq 1 - \xi_i , \quad \xi_i \geq 0
\]
The $\nu$-Trick

**Problem**
- Depending on $C$, the number of novel points will vary.
- We would like to **specify the fraction** $\nu$ beforehand.

**Solution**
Use hyperplane separating data from the origin

\[
H := \{ x | \langle w, x \rangle = \rho \}
\]

where the threshold $\rho$ is **adaptive**.

**Intuition**
- Let the hyperplane shift by shifting $\rho$
- Adjust it such that the 'right' number of observations is considered novel.
- Do this automatically
The $\nu$-Trick

Primal Problem

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \xi_i - m\nu \rho \\
\text{where} & \quad \langle w, x_i \rangle - \rho + \xi_i \geq 0 \\
& \quad \xi_i \geq 0
\end{align*}
\]

Dual Problem

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j \langle x_i, x_j \rangle \\
\text{where} & \quad \alpha_i \in [0, 1] \text{ and } \sum_{i=1}^{m} \alpha_i = \nu m.
\end{align*}
\]

Similar to SV classification problem, use standard optimizer for it.
- Better estimates since we only optimize in low density regions.
- Specifically tuned for small number of outliers.
- Only estimates of a level-set.
- For $\nu = 1$ we get the Parzen-windows estimator back.
A Simple Online Algorithm

Objective Function

\[ \frac{1}{2} \| w \|^2 + \frac{1}{m} \sum_{i=1}^{m} \max(0, \rho - \langle w, \phi(x_i) \rangle) - \nu \rho \]

Stochastic Approximation

\[ \frac{1}{2} \| w \|^2 \max(0, \rho - \langle w, \phi(x_i) \rangle) - \nu \rho \]

Gradient

\[ \partial_w[\ldots] = \begin{cases} w - \phi(x_i) & \text{if } \langle w, \phi(x_i) \rangle < \rho \\ w & \text{otherwise} \end{cases} \]

\[ \partial_{\rho}[\ldots] = \begin{cases} (1 - \nu) & \text{if } \langle w, \phi(x_i) \rangle < \rho \\ -\nu & \text{otherwise} \end{cases} \]
Update in coefficients

\[
\alpha_j \leftarrow (1 - \eta) \alpha_j \quad \text{for } j \neq i
\]

\[
\alpha_i \leftarrow \begin{cases} 
\eta_i & \text{if } \sum_{j=1}^{i-1} \alpha_i k(x_i, x_j) < \rho \\
0 & \text{otherwise}
\end{cases}
\]

\[
\rho = \begin{cases} 
\rho + \eta (\nu - 1) & \text{if } \sum_{j=1}^{i-1} \alpha_i k(x_i, x_j) < \rho \\
\rho + \eta \nu & \text{otherwise}
\end{cases}
\]

Using learning rate \( \eta \).
Online Training Run

5 5 4 3 4 2 2 0 7 4
2 4 7 0 3 4 2 4 3
7 6 7 0 0 0 7 2 0
0 0 4 0 0 4 5 2
8 4 3 6 2 2 0 5 8
Worst Training Examples
Worst Test Examples
Mini Summary

Novelty Detection via Density Estimation
- Estimate density e.g. via Parzen windows
- Threshold it at level and pick low-density regions as novel

Novelty Detection via SVM
- Find halfspace bounding data
- Quadratic programming solution
- Use existing tools

Online Version
- Stochastic gradient descent
- Simple update rule: keep data if novel, but only with fraction $\nu$ and adjust threshold.
- Easy to implement
A simple problem
\[ p(\text{weight}|\text{height}) = \frac{p(\text{height}, \text{weight})}{p(\text{height})} \propto p(\text{height}, \text{weight}) \]
Joint Probability
We have distribution over $y$ and $y'$, given training and test data $x, x'$.

Bayes Rule
This gives us the conditional probability via

$$p(y, y'|x, x') = p(y'|y, x, x')p(y|x)$$
and hence

$$p(y'|y) \propto p(y, y'|x, x')$$
for fixed $y$. 
Normal Distribution in $\mathbb{R}^n$

Normal Distribution in $\mathbb{R}$

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{1}{2\sigma^2} (x - \mu)^2 \right)$$

Normal Distribution in $\mathbb{R}^n$

$$p(x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp \left( -\frac{1}{2} (x - \mu)^\top \Sigma^{-1} (x - \mu) \right)$$

Parameters

- $\mu \in \mathbb{R}^n$ is the mean.
- $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance matrix.
- $\Sigma$ has only nonnegative eigenvalues:
  The variance is of a random variable is never negative.
Our Model
We assume that all $y_i$ are related, as given by some covariance matrix $K$. More specifically, we assume that $\text{Cov}(y_i, y_j)$ is given by two terms:
- A general correlation term, parameterized by $k(x_i, x_j)$
- An additive noise term, parameterized by $\delta_{ij}\sigma^2$.

Practical Solution
Since $y'|y \sim \mathcal{N}(\tilde{\mu}, \tilde{K})$, we only need to collect all terms in $p(t, t')$ depending on $t'$ by matrix inversion, hence

$$\tilde{K} = K_{y'y'} - K_{yy'}K_{yy}^{-1}K_{yy'} \quad \text{and} \quad \tilde{\mu} = \mu' + K_{yy'} K_{yy}^{-1} (y - \mu)$$

Key Insight
We can use this for regression of $y'$ given $y$. 
Some Covariance Functions

Observation
Any function $k$ leading to a symmetric matrix with nonnegative eigenvalues is a valid covariance function.

Necessary and sufficient condition (Mercer’s Theorem)
$k$ needs to be a nonnegative integral kernel.

Examples of kernels $k(x, x')$

- **Linear**
  \[ \langle x, x' \rangle \]
- **Laplacian RBF**
  \[ \exp(-\lambda \| x - x' \|) \]
- **Gaussian RBF**
  \[ \exp(-\lambda \| x - x' \|^2) \]
- **Polynomial**
  \[ (\langle x, x' \rangle + c)^d, \ c \geq 0, \ d \in \mathbb{N} \]
- **B-Spline**
  \[ B_{2n+1}(x - x') \]
- **Cond. Expectation**
  \[ E_c[p(x|c)p(x'|c)] \]
Linear Covariance
Laplacian Covariance

$k(x, y)$ for $x = 1$
Gaussian Covariance

The graph illustrates the Gaussian covariance function for $x=1$. The function is symmetric around $y=0$ and reaches its maximum at $y=1$. The y-axis represents $k(x,y)$ for $x=1$. The function decays to zero as $y$ moves away from $x$. The covariance is defined as $k(x,y) = \exp(-\|x-y\|^2/2\sigma^2)$, where $\sigma$ is the standard deviation.
Polynomial (Order 3)
$B_3$-Spline Covariance
Gaussian Processes and Kernels

Covariance Function
- Function of two arguments
- Leads to matrix with nonnegative eigenvalues
- Describes correlation between pairs of observations

Kernel
- Function of two arguments
- Leads to matrix with nonnegative eigenvalues
- Similarity measure between pairs of observations

Lucky Guess
- We suspect that kernels and covariance functions are the same ...
Mean $\vec{k}^\top(x)(K + \sigma^2 1)^{-1} y$
Variance $k(x, x) + \sigma^2 - \vec{k}^\top(x)(K + \sigma^2\mathbf{1})^{-1}\vec{k}(x)$
Putting everything together...
The ugly details

Covariance Matrices

- Additive noise

\[ K = K_{\text{kernel}} + \sigma^2 \mathbf{1} \]

- Predictive mean and variance

\[ \tilde{K} = K_{yy'} - K_{yy} K_{yy}^{-1} K_{yy'} \quad \text{and} \quad \tilde{\mu} = K_{yy'} K_{yy}^{-1} y \]

Pointwise prediction

\[ K_{yy} = K + \sigma^2 \mathbf{1} \]
\[ K_{yy'} = k(x, x) + \sigma^2 \]
\[ K_{yy'} = (k(x_1, x), \ldots, k(x_m, x)) \]

Plug this into the mean and covariance equations.
Gaussian Process

- Like function, just random
- Mean and covariance determine the process
- Can use it for estimation

Regression

- Jointly normal model
- Additive noise to deal with error in measurements
- Estimate for mean and uncertainty
Support Vector Regression

Loss Function
Given $y$, find $f(x)$ such that the loss $l(y, f(x))$ is minimized.

- Squared loss $(y - f(x))^2$.
- Absolute loss $|y - f(x)|$.
- $\epsilon$-insensitive loss $\max(0, |y - f(x)| - \epsilon)$.
- Quantile regression loss $\max(\tau(y - f(x)), (1 - \tau)(f(x) - y))$.

Expansion
$$f(x) = \langle \phi(x), w \rangle + b$$

Optimization Problem
$$\min_w \sum_{i=1}^m l(y_i, f(x_i)) + \frac{\lambda}{2} \|w\|^2$$
Regression loss functions

Squared Loss

Absolute Loss

Huber’s Robust Loss

ε-insensitive
Summary

Novelty Detection
- Basic idea
- Optimization problem
- Stochastic Approximation
- Examples

LMS Regression
- Additive noise
- Regularization
- Examples
- SVM Regression