Other Applications of Kernel Methods

- Optical character recognition
- Text categorization
- Image Recognition
- Bioinformatics
- Communications
Optical Character Recognition

- Application: Zipcode recognition (Digits 0-9)
- Data Sets: NIST (7291 train, 2007 test, 16 x 16, 256 levels), USPS (60000 train, 10000 test, 20 x 20, gray)
- Use pixels as vectors and use linear, polynomial, and Gaussian kernels
- Extensively studied using handcrafted features for BP algorithm, RBF networks, machine learning algorithms
- SVM performance good without using handcrafted features (test error about 4%, with rejection error rate below 1%, comparable to other more complicated approaches)
Text Categorization

- Application: email filtering, web searching, sorting documents, office automation, news classification
- Data Sets: Reuters-21578 dataset (average 200 words, remove stop words containing no information, 9603 train, 3299 test, 9947 distinct terms found in at least 3 documents)
- Kernel used: $\phi_i(x) = tf_i \log (idf_i)/Z$ where $tf_i$ is number occurrences of term $i$ in document $x$, $idf_i$ is ratio between total number of documents and number of documents containing term $i$, and $Z$ is to normalize norm of $\phi_i(x)$.
Text Categorization continued

- Kernels are sparse vectors representing information criterion
- Category examples: corporate acquisitions, earnings, money market, corn, wheat, ship
- Maximal margin classifier used
- SVM performs much better than other machine learning methods
Image Recognition

- Applications: Face recognition, image retrieval from databases, filtering internet data, medical applications, object detection in visual scenes
- Data Sets: COIL (7200 images, 72 different views of 100 different 3D images), Corel Stock Photo (200 categories, 100 images in each category)
- Use average pixels as vectors, histogram of colors, then use Gaussian or linear kernels
- Hard margin hyperplanes sufficient to separate data.
- SVM outperformed other methods
Bioinformatics

- Applications: Protein homology detection (predict structural and functional features of a protein based on amino acid sequence), gene expression (categorization of gene expression data from DNA microarrays)
- Many different data sets
- Use information about proteins and genes to construct kernels. (Hidden Markov Models (HMM), Translation Initiation Sites (TIS))
- SVM outperformed many existing systems
Communications

- Applications: Channel equalization, Recovering multiuser signals
- Data sets: simulation models, wireless testbed data
- Design adaptive SVM algorithms using linear and Gaussian kernels
- Performance can approach optimal equalization or detection systems
Multiaccess Communications

Several users share the same channel: cellular phones, satellite communications, optical communications.

CDMA schemes are becoming more prevalent: better performance and use of powerful signal processing algorithms.
Discrete Synchronous Model

In the synchronous model, chip-matched filtering followed by chip rate sampling yields:

\[ r = \sum_{k=1}^{K} A_k b_k s_k + \sigma n = SAb + \sigma n \]

where

- \( A_k \) is amplitude of \( k \)th spreading signal
- \( b_k \in \{-1,1\} \)
- \( s_k \) is spreading vector of \( N \) vector of unit norm.
- \( n \) is an \( N \)-vector white Gaussian noise with unit power.
CDMA Receivers

- Matched Filter (MF) receivers are not optimal unless signature sequences are uncorrelated. MF receivers have problems with multiple access interference (MAI).
- With MAI, optimal receivers are nonlinear and have high complexity.
- Linear receivers such as the linear minimum mean squared error (MMSE) receiver have good performance (when MAI not severe) and can be implemented adaptively.
Adaptive MMSE Methods

- Training data
  - Linear MMSE: LMS, RLS algorithms

- Blind algorithms
  - Minimum Output Energy Methods
  - Reduced order approximations: PCA, multistage Wiener Filter
  - Blind Source Separation Methods: Higher order statistics

- Nonlinear MMSE
  - Decision feedback equalizers, PIC, SIC.
Nonlinear adaptive receivers

- Consider nonlinear receivers that
  - can be implemented adaptively using on-line algorithms.
  - can be implemented at mobile station and require only training sequence data.
  - have moderate complexity.
  - can approach performance of optimal nonlinear receivers (e.g. ML receiver)
  - can adapt to time varying and fading channels.

- LS SVM receiver satisfies above conditions.
CDMA Receiver Simulations

Tested variety of receivers including SVM receivers and LS SVM receivers. For simplicity assumed a discrete time model with ideal distortionless channel.

\[ X = UB + \sigma N \]

Interested in downlink receivers. Considered five users and high correlation between spreading sequences (2/7). Used 400 training samples for SVM and LS SVM receiver. Also compared to ML receiver and linear MMSE receiver.
CDMA simulations

Desired user with weak energy, $\rho=0.429$, 5 user synchronous system, 400 training examples
Mitra and Poor (1994) showed that optimal maximum likelihood solution can be implemented exactly using a weighted sum of $2^k$ Gaussian basis functions.

Keys (performing tasks separately or together):
- Choosing centers
- Choosing widths
- Number of basis functions
- Learning output weight layer
LS SVM Solutions

- Mitra and Poor (1994): used Radial Basis function to approximate optimal solution. Used K means algorithm to determine location of Gaussian basis function centers.
- We have conducted simulations on different systems with high MAI and different power constraints showing time update LS SVM with Gaussian kernels can do better than linear receivers and get close to optimal solution.
- By using subspace methods and selectively choosing data we can approach optimal solutions.
Learning Algorithm Behavior

How can we measure how good learning algorithms such as the SVM and error back propagation do?

- Performance
- Complexity
- Implementation
  - Software
  - Hardware
Generalization Error

- Data can be divided into training data, test data, and validation data.
- Learning algorithms use training data to learn parameters of the network. Given \( m \) training data, the training error also known as empirical error can be expressed as

\[
J_{\text{emp}}(w) = \frac{1}{m} \sum (d_i - f(x_i,w))^2
\]

- Training error is different from test error also known as generalization error which is given by

\[
J(w) = \int (d-f(x,w))^2 p(x,d) \, dx \, dd
\]

where \( p(x,d) \) is the joint probability density function of \( x \) and \( d \).
Generalization Error Continued

There are relationships between empirical error and generalization error, but it is helpful to have third set of data (validation set) to train network. Low training error does not necessarily mean low test error.

\[ J(w) \]

![Graph showing training and validation error over iterations]

- Training error
- Validation error
- Iterations
When learning a task such as a pattern recognition problem there are a variety of functions or hypotheses that we could choose. The class of all functions that we can choose is called a function or concept class. Examples:
- Linear threshold functions
- Boolean functions
- Feedforward neural networks with one hidden layer

The function class complexity is determined by the Vapnik-Chervonenkis (VC) dimension.
Definitions

- $x \in X$ (input or instance)
- $d = c(x)$, $c \in C$ (label or concept) (consider binary labels)
- $S = ((x(1),d(1), \ldots, x(m),d(m))$ (sample drawn iid from some unknown distribution $D$)
- $h = L(S,C)$ takes a sample $S$ and chooses a hypothesis concept class consistent with $S$ (i.e. $h(x(i)) = d(i)$)
- $P(c \neq h) = D(c \Delta h)$
PAC Learning

Probably Approximately Correctly: Let $C$ be a concept class over $\mathcal{X}$. $C$ is PAC learnable if there exists $\mathcal{L}$ such that for $c \in C$, for every distribution $D$ on $\mathcal{X}$, for all $0 < \varepsilon < 1/2$, $0 < \delta < 1/2$, then with probability of at least $1 - \delta$, $\mathcal{L}$ outputs hypothesis $h \in C$ such that $P(c \neq h) < \varepsilon$.

If $\mathcal{L}$ runs in time polynomial in $1/\varepsilon$, $1/\delta$ and $m$ we say $C$ is efficiently PAC learnable. $\varepsilon$ is error parameter and $\delta$ is confidence parameter.
VC dimension

- Consider function classes where each function labels each input as 1 or –1.
- A set of \( m \) points is shattered by function class if the function class represents all \( 2^m \) possible labelings of the points.
- The VC dimension of a function class is the largest cardinality of points that is shattered by the function class. Example: linear threshold functions in Euclidean \( n \) space has VC dimension of \( n+1 \).
- The VC dimension measures the complexity of the function class.