Kernel Characterization

A function $K: X \times X \Rightarrow \mathbb{R}$ are called kernel functions

Which is either continuous or has a countable domain, can be decomposed $K(x,z) = \langle \phi(x), \phi(y) \rangle$ and $K(x,z) = K(z,x)$ is symmetric and positive semi-definite

**Input space**

$X$

**Feature space**

$\Phi$

$\phi(x)$

Direct

**Inner products**

**Kernels**

$K(x,y) = \langle \phi(x), \phi(y) \rangle$
Examples of Kernel Functions

- Polynomial Kernel of order \( p \): \( K(x,z) = (x^Tz)^p \)
- Polynomial Kernel of order \( \leq p \): \( K(x,z) = (x^Tz+1)^p \)
- Gaussian Kernel: \( K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right) \)
- Sigmoidal Kernel: \( K(x,z) = \tanh(ax^Tz - b) \) for some values of \( a \) and \( b \)
- Bilinear transformation: \( K(x,z) = f(x)f(z) \)
- Other kernels: splines, strings, \( K(x,z) = \min(x,z) \)
- Can have nonnumeric kernels (e.g. bioinformatics)
Support Vector Machine

Optimal margin classifier with slack variables and kernel functions described by Support Vector Machine (SVM).

\[
\min_{(w, \xi)} \frac{1}{2}||w||^2 + C \sum \xi_i \\
\text{subject to } \xi_i \geq 0 \forall i, d(i) (w^T \phi(x(i)) + b) \geq 1 - \xi_i, \forall i, \text{ and } C>0.
\]

(Hinge loss function)

In dual space

\[
\max W(\alpha) = \sum \alpha(i) - \frac{1}{2} \sum \alpha(i)\alpha(j) d(i)d(j) K(x(i),x(j)) \\
\text{subject to } C \geq \alpha(i) \geq 0, \text{ and } \sum \alpha(i)d(i)= 0.
\]

Weights can be found by \( w = \sum \alpha(i) d(i) \phi(x(i)) \).
Representation of decision surface

- In primal space decision surface is a linear hyperplane in feature space and can be represented as
  \[ f(x) = \text{sgn} \left( \mathbf{w}^T \phi(x(i)) + b \right) \]

- In dual space decision surface can be represented via kernels and Lagrange multipliers as
  \[ f(x) = \text{sgn} \left( \sum \alpha_i d(i) K(x, x(i)) + b \right) \]
Least Squares SVM Regression

Consider changing SVM to LS SVM by making following modifications:

\[ \min_{(w,e)} \frac{1}{2} \|w\|^2 + \frac{1}{2} C \sum e(i)^2 \]

subject to \(d(i) - (w^T \Phi(x(i)) + b) = e(i), \forall i, \) and \(C > 0.\) Note that \(e(i)\) is error term.

Key differences with between SVM and LS SVM:

- \(\varepsilon\)-insensitive cost replaced by quadratic error cost.
- Inequality constraint replaced by equality constraint.
Primal Solution

Substitute for $e_i$ and take partial derivatives of objective function with respect to $w$ and $b$ and set to 0. This yields least square solution given by

$$(R(m) + I/mlC))w = P(m)$$

$$m_X (m)^T w + b = m_Y (m)$$

where $m_X (m)$ and $m_Y (m)$ are sample means of $\Phi(X)$ and $Y$ respectively. $R(m)$ and $P(m)$ are sample autocorrelation of $\Phi(X)$ and crosscorrelation of $\Phi(X)$ and $Y$ respectively.

If $(R(m) + I/(mC))$ is nonsingular we have that

$$w = (R(m) + I/(mC))^{-1} P(l)$$

giving MSE solution plus regularization term.
Finding Dual Solution

Introduce Lagrange multipliers

\[ L(w,b,e,\alpha) = \frac{1}{2}||w||^2 + \frac{1}{2}C \sum e(i)^2 \]
\[ - \sum \alpha(i) (d(i) - (w^T \Phi(x(i)) + b) - e(i)) \]

where \( \alpha(i) \geq 0 \).
KKT Conditions

Again take partial derivatives and set to 0.
\[ \frac{\partial L(w,b,e,\alpha)}{\partial w} = 0, \frac{\partial L(w,b,e,\alpha)}{\partial b} = 0, \]
\[ \frac{\partial L(w,b,e,\alpha)}{\partial \alpha} = \frac{\partial L(w,b,e,\alpha)}{\partial e} = 0. \]

We therefore have that
\[ w = \Sigma \alpha(i)\Phi(x(i)) \]
\[ \Sigma \alpha(i) = 0 \]
\[ \alpha(i) = C e(i), \quad 1 \leq i \leq m \]
\[ d(i) - (w^T \Phi(x(i)) + b) - e(i)) = 0, \quad 1 \leq i \leq m \]
Dual Solution to LS SVM

Let $\alpha$ be vector of Lagrange multipliers and $d$ be vector of outputs then solution has following form:

$$
\begin{bmatrix}
0 & 1^T \\
1 & K+I/C
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
d
\end{bmatrix}
$$

where $K(x,z) = \Phi(x)^T \Phi(z)$ and denotes l vector of 1s.

$$f(x) = \sum \alpha(i) K(x,x(i)) + b$$
Comments about LS SVM

- Solution to LS SVM depends on $d$, dimensionality of feature space $\Phi(x)$ in primal space and $m$, number of training samples in dual space.
- Both solutions involve solving a set of linear equations. Work in space that has lower dimension.
- Adaptive on-line solutions can now be implemented.
- Algorithm easily constructed for pattern classification problems.
- In dual space, practically all input training examples are support vectors as Lagrange multipliers, $\alpha$ are proportional to error, $e$. 
LS Kernel method solution

• Solution in primal or dual space involves a solution to a set of respectively \((m+1, d+1)\) linear equations.
• Dual space solution: unlike SVM solution all input training examples are support vectors
• Objectives: want good performance with low to moderate computational complexity
  – Sparseness: reduced system, subspace method
  – On-line versus batch
  – Criteria for choosing SV
  – Approximate methods: kernel LMS
  – Distributed learning, complex vectors
  – Numerical stability: matrix computations
Kernel or Gram Matrix

Let $K$ be kernel matrix of all training data. Reduce computations by considering a subset of $K$.
• $K$ is symmetric and in many cases has eigenvalues that decay exponentially.
• Here $K_{SS}$ is $m_s \times m_s$ where $m_s << m$.
• Let $K_S = [K_{SS} \ K_{SN}]$.
  • Reduced system methods work with $K_{SS}$
  • Subspace methods work with $K_S$

\[
K = \begin{pmatrix}
K_{SS} & K_{SN} \\
K_{NS} & K_{NN}
\end{pmatrix}
\]
Subspace Methods

- Reduced system method use $m_S \ll m$ training examples $\Phi(X_S)$ resulting in kernel matrix $K_{SS} = \Phi(X_S) \Phi(X_S)^T$, but algorithm only uses a subset of information from kernel matrix $K$
- Subspace methods restrict weight to lie in subspace of $m_S$ training examples, $w = \Phi(X_S) \alpha$
  - Information matrix $A = (K_{SS}/C + K_S K_S^T)$ where $K_S = \Phi(X_S) \Phi(X)^T$ contains much more information than just $K_{SS}$
  - Information matrix dimensionality is still $m_S$
  - Higher complexity, but improved performance
Subspace LS regression equations

Optimization Problem:

$$\min_{(w,e)} \frac{1}{2} ||w||^2 + \frac{1}{2} C \sum e(i)^2$$

subject to

$$y(i) - (w^T \Phi(x(i)) + b) = e(i), \forall i$$

$$w = \Phi(X_S) \alpha.$$ 

Solution: 

$$A = (K_{SS}/C + K_S K_S^T) (m_S \text{ by } m_S)$$ 

$$A\alpha + K_S 1b = K_S y$$

$$1^T K_S^T \alpha + mb = 1^T y$$
Estimated Outputs

• Solve linear equations to find $\alpha$ and $b$. If $A$ is invertible

Estimated output in primal space is

$$\hat{y}(x) = \phi(x)^T w + b,$$

where $w = \Phi(X_S) \alpha$, or in dual observation space

$$\alpha = A^{-1} K_S (y - 1b) \quad \text{and} \quad b = \frac{1^T y - 1^T K_S^T A^{-1} K_S y}{m - 1^T K_S^T A^{-1} K_S 1}$$

where

$$\hat{y}(x) = K(x, x_S) \alpha + b,$$

$$K(x, x_S) = \phi(x)^T \Phi(x_S).$$
LS Kernel Related Research

- Kernel Ridge Regression
- Kernel Fisher Discriminant Analysis (pattern recognition problems)
- Radial Basis Functions
- Gaussian Processes
- Kriging
Data Processing $K_S$

- Support vectors
- Information vectors
- Kernel data
- New data
- Delete SV
- Add SV
- Delete data
Information vectors (finding $A^{-1}$)

- Effect of adding and deleting information vectors
- Computing $A^{-1}$ based on new information relies on using Sherman Morrison Woodbury formula given by

$$ (A + uv^T)^{-1} = A^{-1} - A^{-1}u (I + v^T A^{-1}u)^{-1} v^T A^{-1} $$

where $A = (K_{SS}/C + K_S K_S^T)$.

Here $hn$ is new kernel vector to be added (compute kernels between new input and SVs) and $ho$ is old kernel vector to be deleted

$u = [hn \ ho]$ and $v = [hn \ -ho]$
Support Vectors (finding $A^{-1}$)

- Effect of adding a SV. This increases dimensionality of matrix $A$ by 1.
- Computing $A^{-1}$ based on new information relies on inverting block matrices given by

$$
\begin{pmatrix}
A & c \\
c & d
\end{pmatrix}^{-1} = \begin{pmatrix}
A^{-1} & 0 \\
0 & 0
\end{pmatrix} + v v^T r
$$

where $v = [A^{-1} c ; -1]$ and $r = d - c^T A^{-1} c$. Here $A$ is matrix with information vectors updated, $c$ and $d$ are new kernel information, and $r$ gives information about degree of independence.
Criteria for choosing SV

- Random updates
- Time updates
- Update based only on inputs
- Training error based criteria: update based on inputs and outputs
- Training based on information about problem
Algorithms for choosing SV

Choose training examples that give the most information.

• Training based error: choose SV that can reduce training error
• Information based criteria: Renyi Information
• Approximate Linear Dependence (ALD): depends only on inputs, choose inputs that are almost linearly independent from other inputs
• Surprise criteria: Information based criterion based on modeling training examples as Gaussian processes
• Coherence criteria: low cost criteria examining maximum absolute kernel value between new and old inputs
Number of SVs

- For criteria (e.g. coherence and ALD) the number of SVs chosen remains finite as number of training examples grows large.
- Must also control amount of information vectors to control complexity
- Deletion of SVs can be helpful to improve performance and when functions to be learned change with time
- Increasing number of SVs by changing selection criteria parameters can lead to improved performance, but with higher computational costs
Simpler Approximate Algorithms

- RLS kernel algorithms have good performance, but complexity is not low, $O(m^2)$ operations per update
- Recently class of kernel LMS algorithms have appeared where complexity is $O(m)$ operations per update
  - Primal kernel LMS
  - Dual kernel LMS
  - Kernel Affine Projection algorithm
Primal Kernel LMS Algorithm

- Work by Liu, Pokharel, Principe
- Estimate: \( f(x) = w^T \Phi(x) \)
- Error: \( e = y - f(x) \)
- Algorithm
  - Initialization: \( w(0) = 0 \);
  - Iterative update: if example satisfies criteria update
    \[
    w(k+1) = w(k) + \eta e(k) \Phi(x(k))
    \]
    \[
    e(k) = y(k) - w(k)^T \Phi(x(k)) = y(k) - \sum_{1 \leq i \leq k-1} \eta e(i) K(x,x(i))
    \]
- For support vectors \( (x(i), 1 \leq i \leq m) \) estimate given by
  \[
  f(x) = \sum_{1 \leq i \leq m} \eta e(i) K(x,x(i))
  \]
Comments about algorithm

- Algorithm performs LMS in feature space, but computations may have to be done in dual space.
- Algorithm does not need regularization and converges for proper step sizes $\eta$.
- Each $\alpha(i) = \eta \, e(i)$ is computed one time and is then fixed. Compactly we have $\alpha = \eta \, (I + \eta L)^{-1} \, y$ where $L$ is the lower triangular kernel matrix with:

$$
L = \begin{bmatrix}
0 & 0 \\
K(2,1) & 0 \\
\vdots & \vdots \\
K(m,1) & \ldots \ldots \ldots \ldots K(m,m-1) & 0
\end{bmatrix}
$$
Dual Kernel LMS

- Work by Richard, Bermudez, and Honeine
- Consider least squares subspace problem in observation space with zero bias and no regularization.
- Solution given by finding $\alpha$ to minimizing $||A\alpha - d||^2$ with $A = K_S K_S^T$ and $d = K_s y$.
- Alternative method to finding solution is applying a modified LMS algorithm directly in dual space. Here input is given by kernel vector
- $h(x) = (K(z(1), x), \ldots K(z(j), x))^T$ where $z(1), \ldots, z(j)$ are current SVs and output given by $y(x)$. 
Dual Kernel LMS Algorithm

1) Initialization of algorithm
2) Get current kernel vector \( h(x(k)) \) and test coherence
3) If \( h(x(k)) \) has small coherence current input \( x(k) \) becomes a SV and set of SVs are augmented and weight vector \( \alpha \) dimensionality is increased by one

\[
S = S + \{x(k)\}
\]
\[
\alpha(k+1) = [\alpha(k); 0]
\]
4) Update weights, \( \alpha(k+1) = \alpha(k) + \eta e(k)h(x(k)) \)
5) Increment \( k \) and go to 2)
Algorithm Comments

- Richard, Bermudez, and Honeine examine KNLMS with weight magnitudes bounded.
- They add a regularization term $\varepsilon$, creating slightly different problem than subspace optimization problem.
- Coherence criteria has $\mathcal{O}(m)$ operations, but other criteria can easily be used.
- This algorithm focuses on updating weights based on current input.
- Kernel Affine Projec(tion) (KAP) is a generalization updating weights based on $k$ most recent inputs.
Example 1: Noisy sinc function

Considered LS-SVM regression problem, formulation similar to LS-SVM classification

\[ d = \text{sinc}(t) + v \]

Using subspace methods and intelligent updating we can get roughly same performance with ten chosen SVs as 100 random points using LS SVM.

Noise has deviation \( \sigma = 0.7 \) and \( C = 1.1 \). Trained LS SVM (MSE = 0.0109) and subspace method (MSE = 0.0108). Kernel eigenvalues decrease at an exponential rate.

Subspace method has higher deviation than LS SVM.
LS SVM and Subspace MSE
Sinc approximation with 10 SV
Example 2: Time Series Prediction

- Nonlinear random time series
- Initial Conditions: $d(1)=d(2)=.1$
- Discrete time system
  \[
  d(k) = (0.8 - 0.5 \exp(-d(k-1)^2))d(k-1) \\
  - (0.3+\exp(-d(k-1)^2))d(k-2) + 0.1 \sin(\pi d(k-1)) + v(k)
  \]

$v(k)$ is AWGN deviation .1. Used Gaussian kernels, coherence criteria, and (KRLS, KLMS) algorithms
KRLS vs KNLMS

$m_s \approx 25$, 200 simulations averaged KNLMS runs three times faster than KRLS
KNLMS (different coher. thresholds)

\[ \mu_0 = 0.3, 0.5, 0.7 \] yields \( m_s \approx 16, 25, 45 \) and \( t_t = 350s, 510s, 840s \).
KRLS with different window sizes
Suite of Online Kernel Algorithms

- KRLS, KAP, KLMS
- Criteria for choosing SVs
- Choose number of information vectors
- Adjust parameters of algorithm: type of kernel used, regularization parameter, step size

Powerful nonlinear adaptive online algorithms where you can tradeoff performance for complexity