Iterative Algorithm Comments

- Algorithms based on descending energy surface by examining first and second derivatives.
- LMS (stochastic gradient descent), tradeoffs between algorithm complexity and convergence speed.
- Can use other cost functions besides quadratic cost functions: Absolute error, Minkowski error, entropy function.
- Can apply to nonlinear activation units or multi-layer networks.
- Levenberg-Marquardt algorithm: another approximation of energy function using Taylor series. Uses pseudo inverse and can approximate Newton’s method or gradient descent.
Least Squares Algorithm

- Let \((x(k),d(k)), 1 \leq k \leq m\) then LS algorithm finds weight \(w\) such that squared error is minimized. Let \(e(k) = d(k) - w^T x(k)\), then cost function for LS algorithm given by \(J(w) = 0.5 \sum_k e(k)^2\)

- In matrix form can represent
  \[
  J(w) = 0.5 \|d - Xw\|^2 = 0.5\|d\|^2 - d^TXw + 0.5w^TX^TXw
  \]
  where \(d\) is vector of desired outputs and \(X\) contains inputs arranged in rows.
Least Squares Solution

- Let $X$ be the data matrix, $d$ the desired output, and $w$ the weight vector.
- Previously we showed that
  \[ J(w) = 0.5 \|d-Xw\|^2 = 0.5\|d\|^2 - d^T X w + 0.5 w^T X^T X w \]
  where $d$ is vector of desired outputs and $X$ contains inputs arranged in rows.
- LS solution given by $X^T X w^* = X^T d$ (normal equation) with $w^* = X^\dagger d$. If $X^T X$ is of full rank then $X^\dagger = (X^T X)^{-1} X^T$.
- Output $y = X w^*$ and error $e = d - y$.
- Desired output often of form $d = X w^* + v$. 
Adaptive Filter

\[ y(n) = w_0 u(n) + w_1 u(n-1) + w_2 u(n-2) \]

\[ e(n) = d(n) - y(n) \]
Data Presentation

Windowed data

\[ X = \begin{pmatrix}
  u(n) & u(n-1) & \ldots & u(n-m+1) \\
  u(n-1) & u(n-2) & \ldots & u(n-m) \\
  \vdots & & \ddots & \vdots \\
  u(k) & u(k-1) & \ldots & u(k-m+1)
\end{pmatrix} \]

\[ d = \begin{pmatrix}
  d(n) \\
  d(n-1) \\
  \vdots \\
  d(k)
\end{pmatrix} \]

Fixed window, growing window, exponential weighted window
Least Squares Solution Comments

- Note LS solution approximates Wiener solution as window size gets large: \( R \approx (1/L) X^T X \), \( P \approx (1/L) X^T d \)
- Principle of orthogonality (Projection theorem): error orthogonal to data \( e^T X = 0 \) which results in
  \[ J(w^*) = 0.5 ||d||^2 - d^T X w^* = 0.5 ||d||^2 - 0.5 d^T X X^+ d \]
- Normal equations are derived from principle of orthogonality (scalar representation):
  \[ \sum w_j^* \sum (i-k) u(i-j) = \sum (i-k) d(i) \quad k=0\ldots m-1 \]
- Ridge regression: add regularization term to get
  \[ w^* = (\lambda I + X^T X)^{-1} X^T d \]
Time Correlations

- Let $\Phi = X^TX$, or $\Phi(k,j) = \Sigma u(i-k)u(i-j)$ represent time correlation data matrix
  - Symmetric, positive semi-definite, eigenvalues are nonnegative real numbers
- Let $z = X^Td$, or $z(k) = \Sigma u(i-k)d(i)$ represent time cross-correlation
- LS solution given by $\Phi w^* = z$
Least Squares statistical properties

- Estimate of weight $w^*$ is unbiased
  \[ d = X w^* + v \]
- When measurement error process is white with zero mean and variance $\sigma^2$, the covariance matrix of the LS estimate $w^*$ equals $\sigma^2 \Phi^{-1}$
- When measurement error process is white with zero mean, the LS estimate $w^*$ is the best linear unbiased estimate
- In addition when the measurement process is Gaussian, the LS estimate is the same as the maximum likelihood estimate and achieves the Cramer-Rao lower bound
Singular Value Decomposition

- LS solution given by $w^* = X^+d$ involves computing the pseudoinverse of the Moore-Penrose generalized inverse of the matrix $X$. (matlab $w = X/d$ or $w = \text{pinv}(X)*d$)
- Any matrix $X$ can be decomposed into a general eigenvector / eigenvalue decomposition.
  - If $X$ is symmetric we have that $X = Q\Lambda Q^T$
  - For arbitrary $X$ ($n$ by $m$ with $m \leq n$) we have that $X = U S V^T$ where $S$ contains singular values and $U$ and $V$ are unitary matrices
SVD Continued

- X (n by m) can be decomposed using SVD to get that $X = U S V^T$ where
- $S$ (n by m) with first m rows being a diagonal matrix containing square root of eigenvalues of $XX^T$ and $X^TX$. Rest of rows are zeros
- $U$ (n by n) is unitary and contains eigenvectors of $XX^T$
- $V$ (m by m) is unitary and contains eigenvectors of $X^TX$
- $S = U^TXV$ (matlab $[U,S,V] = svd(X)$)
SVD Continued

- Note $X = USV^T$ (with $n > m$) and correlation data matrix $\Phi = X^TX = VSTSV^T$.
- As number of observations in window grows large we have $R \approx (1/n) \Phi$ and $R = QA\Lambda Q^T$. Therefore $Q \approx V$ and $\lambda_i \approx (1/n) s_i^2$.
- Methods of obtaining SVD. Matrix operations such as Givens rotations and Householder transformations.
Singular Values and Pseudo-Inverse

- One useful benefit of SVD is that it is easy to express pseudoinverse in terms of SVD terms,
  \[ X^\dagger = V(S_i)^T U^T \]

where \( S_i = \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} \)

- For LS problem assuming \( n > m \) (over determined system and assume \( X^T X \) is of full rank \( m \), then can easily show by substituting \( X=USV^T \) that pseudoinverse is given by above.
Recursive Least Square Algorithm

- Can develop an on-line version of LS algorithm called Recursive LS (RLS) algorithm
- Algorithm based on using Sherman-Morrison-Woodbury formula:

\[(A+vv^T)^{-1} = A^{-1} - A^{-1} v(1+v^T A^{-1} v) v^T A^{-1}\]

where \(A = X^T X\) contains old data and \(v = x(m+1)\) contains new data at time \(m+1\)

- Similar to Kalman filter equations where we update estimate recursively adding new information or innovations.
- Update is \(O(m^2)\) operations
RLS Algorithm Comments

- Often exponentially weighted algorithm implemented. Update correlation matrix, gain factor, and weights.
- Parameters of RLS algorithm: Initial correlation matrix and weight decay factor.
- Convergence is typically an order of magnitude faster than LMS algorithm. Algorithm theoretically converges to zero excess mean squared error and convergence does not depend on eigenvalues.
- Many variations to account for more stable matrix computations: QR and Cholesky factorizations.
Linear Filter Applications

- Inverse Modeling: Channel Equalization
- Adaptive Beamforming
  - Radar
  - Sonar
  - Speech enhancement
- System Identification: Plant modeling
- Prediction
- Adaptive Interference Cancellation: Echo Cancellation
Nonlinear Methods

- Multilayer feedforward networks: error back propagation algorithm
- Kernel methods:
  - Support Vector Machines (SVM)
  - Least squares methods
  - Radial Basis Functions (RBF)