Projection of Data

- Principal Component Analysis (PCA) (maximize variance)
- Fisher Discriminant Analysis (FDA) (minimize in class scatter while maximizing distance between means)
- Canonical Correlation Analysis (CCA) (maximize covariance)
- Kernelize PCA, FDA, CCA
- Factor analysis, partial regression analysis
- Projection pursuit
- Independent Component Analysis
Given a signal $S_1$ mixed with additive impairment $S_2$ we can use an adaptive filter (using algorithms such as LMS) to cancel out the effects of the noise.
Give X1 and X2 with Adaptive Filter 1 having access to S2 and Adaptive Filter 2 having access to S1 we can recover S1 and S2.
Independent Component Analysis

- Let $X=AU$ where $A$ is a square mixing matrix, $U$ is a random $m$ vector, and $X$ is the observed random $m$ vector.
- Can we recover $U$ from $X$ if $A$ and $U$ are unknown?
- Under assumptions that components of $U$ are independent random variables we can recover $U$ from $X$ under certain assumptions. We need to establish optimization criteria to recover $U$ from $X$.
- Closely related to projection pursuit and factor analysis
- Applications: Blind Source Separation, Blind Deconvolution, Feature Extraction
Independent Component Analysis

PCA decorrelates inputs. However in many instances we may want to make outputs independent.

Inputs U assumed independent and user sees X. Goal is to find W so that Y is independent.
Applications of ICA

- Speech Separation: several speech signals are mixed together (cocktail problem)
- Array antenna processing: several narrowband signals mixed together from unknown directions
- Hyperspectral Images: images at multiple wavelengths
- Biomedical information: Brain signals, EEG data, FMRI data
- Financial market data analysis: extract dominant signals
ICA Solution

- $Y = DPU$ where $D$ is a diagonal matrix and $P$ is a permutation matrix.
- Algorithm is unsupervised. What are assumptions where learning is possible? All components of $U$ except possibly one are nongaussian.
- Establish criterion to learn from (use higher order statistics): information based criteria, kurtosis function.
- Kullback Leibler Divergence:
  \[ D(f, g) = \int f(x) \log \frac{f(x)}{g(x)} \, dx \]
ICA Information Criterion

- Kullback Leibler Divergence nonnegative
- Mutual Information $I(X:Y) = H(X) - H(X|Y)$ nonnegative
- Set $f$ to joint density of $Y$ and $g$ to products of marginals of $Y$ then

$$D(f,g) = -H(Y) + \sum H(Y_i)$$

which is minimized when components of $Y$ are independent.

- When outputs are independent they will be a permutation and scaled version of $U$. 

ICA Objective Functions

- Kurtosis
- Kullback Leibler Divergence nonnegative
- Mutual Information $I(X;Y) = H(X) - H(X|Y)$ nonnegative. Set $f$ to joint density of $Y$ and $g$ to products of marginals of $Y$ then
  \[ D(f,g) = -H(Y) + \sum H(Y_i) \]
  which is minimized when components of $Y$ are independent.
- Negenentropy
- Contrast functions
ICA Preprocessing

- Signal processing and filtering
- Center data (remove means)
- Decorrelate data (apply PCA). If data is jointly Gaussian cannot do any more
Learning Algorithms

- Can learn weights by approximating divergence cost function established using contrast functions.
- Iterative gradient estimate algorithms can be used.
- Faster convergence can be achieved with fixed point algorithms that approximate Newton’s methods.
- Algorithms have been shown to converge.
ICA Example

- Three signals are linearly mixed

FIGURE 10.13 Waveforms on left-hand side: original source signals. Waveforms on right-hand side: separated source signals.
Clustering

- Group input training data into clusters where training data are likely to be grouped into the same cluster if they are similar to one another
- Mixture densities: inputs are drawn from multiple random variables, semi-parametric models
- K-means, Expectation-Maximization Algorithm, Self-Organization Maps (topological ordering)
- Source coding and data compression
k-Means Algorithm

- Given m inputs, \( x(i) \) \( 1 \leq i \leq m \) we want to encode these inputs into k outputs, \( c(j) \) \( 1 \leq j \leq k \) where \( k \ll m \).
- k-means algorithm is an iterative update algorithm consisting of:
  - Initialization (set initial codewords \( c(j) \))
  - Iterate
    - Compute closest codeword for each \( x(i) \)
      \[ d(i) = \text{argmin}_j \| x(i) - c(j) \| \]
    - Recompute codewords
      \[ c(j) = \frac{\Sigma x(i)I(d(i) = c(j))}{\Sigma I(d(i) = c(j))} \]
k-Means Algorithm Comments

- Distortion measure: $\text{MSE} = \frac{1}{m} \sum (x(i)-d(i))^2$
- k-Means algorithm: MSE at each iteration either improves or remains the same and converges to local minimum
- Algorithm is a method for performing vector quantization and source coding
- Initialization: randomly, pick first k inputs, intelligently select initial codewords
- Variations: on-line, adding and deleting codewords
Parametric vs Semiparametric Models

- Parametric model: $f_X(x)$ pdf is known
- Semiparametric model (mixture model):
  \[ f_X(x) = \sum_k f_{X|D}(x|k)P(D=k) \]

Conditional densities may be known, but parameters of density may be unknown (e.g. mean and variance of Gaussian random variable), mixture pmf $D$ unknown
Points drawn from a mixture of Gaussians
Learning mixtures from supervised data

- Given $m$ training data, $(x(i), d(i))$, $1 \leq i \leq m$ learn the unknown parameters of mixture density. Assume each mixture is Gaussian with unknown mean and variance.
- Mixture class probabilities: $P(D=j) = \sum I(d(i)=j)/m$
- Mean of each Gaussian mixture: $m(j) = \sum I(d(i)=j)x(i)/\sum I(d(i)=j)$
- Covariance matrix of each Gaussian mixture: $\Lambda(j) = \sum I(d(i)=j)(x(i)-m(j))(x(i)-m(j)^T)/\sum I(d(i)=j)$
Expectation-Maximization (EM) Algorithm

- If we have only inputs $x(i), \ 1 \leq i \leq m$ it is still possible to learn Gaussian mixtures
- Here we have parameters that are observable, $\mathcal{X}$ and other parameters that are not observable, $\mathcal{Z}$. $w$ contains information about priors and sufficient statistics
- Would like to maximize $J(w|\mathcal{X})$, but there are hidden variables and must work with $J(w|\mathcal{X},\mathcal{Z})$ where we estimate $\mathcal{Z}$ with $w$
- The algorithm is an iterative algorithm with two steps:
  - E step: $Q(w|w(n)) = E(J(w|\mathcal{X},\mathcal{Z})|\mathcal{X}, w(n))$
  - M step: $w(n+1) = \operatorname{argmax}_w Q(w|w(n))$
EM Algorithm Comments

- Energy function increases at each update and converges
- With the two iterative steps, EM algorithm estimates unknown parameters from training inputs
- EM algorithm can be considered as a soft decision rule as it works with probabilities
- EM algorithm has been applied wide range of applications
  - Mixture of densities: (k-Means hard decision form of EM)
  - Hidden Markov models
  - Bayesian learning models