

# Data Modeling using Kernels and Information Theoretic Learning

**Jose C. Principe**

**Computational NeuroEngineering Laboratory  
Electrical and Computer Engineering Department  
University of Florida**

**[www.cnel.ufl.edu](http://www.cnel.ufl.edu)**

**principe@cnel.ufl.edu**





# Acknowledgments

Dr. John Fisher

Dr. Dongxin Xu

Dr. Ken Hild

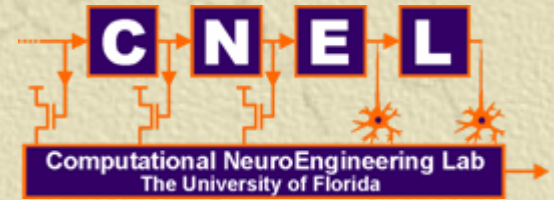
Dr. Deniz Erdogmus

My students: Puskal Pokharel  
Weifeng Liu  
Jianwu Xu  
Kyu-Hwa Jeong  
Sudhir Rao  
Seungju Han

NSF ECS – 0300340 and 0601271 (Neuroengineering program) and DARPA



# Outline



- Information Theoretic Learning
- Kernel Methods
- Correntropy as Generalized Correlation
- Applications
  - ◆ Matched filtering
  - ◆ Wiener filtering and Regression
  - ◆ Nonlinear PCA

# Twenty Years ago

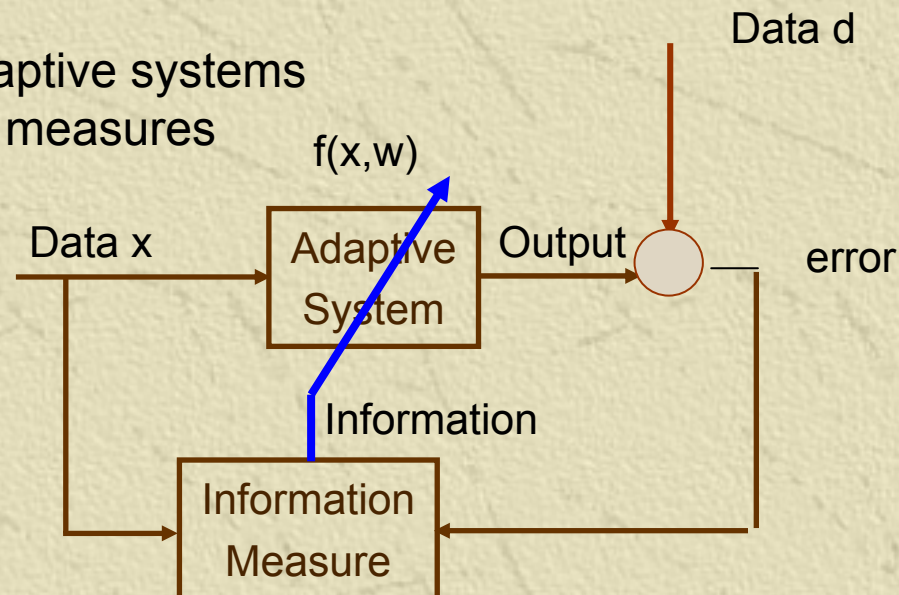
- Neural networks were a paradigm shift in data analysis because:
  - ◆ They handled **higher dimensional problems** than it was considered possible
  - ◆ They took advantage of **nonlinear multi-layer topologies** that implemented very efficient, universal approximators.
  - ◆ They could be trained with a new learning algorithm called **backpropagation**
  - ◆ They were in many respects a metaphor for brain processing due to their distributed, nonlinear, dynamical nature.
- However, they were mostly capturing the second order statistics of the error because of the Mean Square Error cost function...



# Information Filtering: From Data to Information

✧ Information Filters: Given data pairs  $\{x_i, d_i\}$

- ✧ Optimal Adaptive systems
- ✧ Information measures



- ✧ Embed information in the weights of the adaptive system
- ✧ More formally, use optimization to perform Bayesian estimation



# Information Filtering

Deniz Erdogmus and Jose Principe

**From Linear  
Adaptive Filtering  
to Nonlinear  
Information Processing**

IEEE  
SP MAGAZINE

November 2006



# What is Information Theoretic Learning?

ITL is a methodology to adapt linear or nonlinear systems using criteria based on the information descriptors of entropy and divergence.

Center piece is a non-parametric estimator for entropy that:

- ✦ Does not require an explicit estimation of pdf
- ✦ Uses the Parzen window method which is known to be consistent and efficient
- ✦ Estimator is smooth
- ✦ Readily integrated in conventional gradient descent learning
- ✦ Provides a link to Kernel learning and SVMs.
- ✦ Allows an extension to random processes

# ITL is a different way of thinking about data quantification

Moment expansions, in particular **Second Order moments** are still today the workhorse of statistics. We automatically translate deep concepts (e.g. similarity, Hebb's postulate of learning ) in 2<sup>nd</sup> order statistical equivalents.

ITL replaces 2<sup>nd</sup> order moments with a geometric statistical interpretation of data in probability spaces.

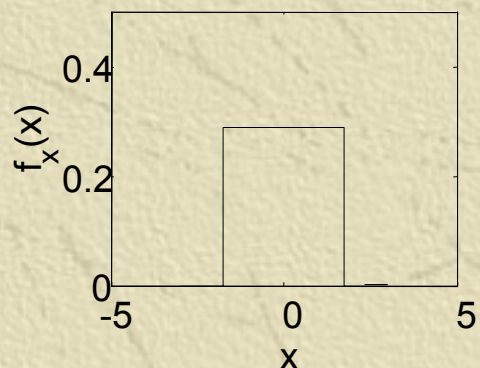
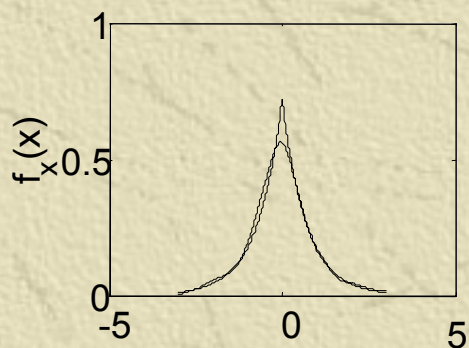
- ✧ Variance by **Entropy**
- ✧ Correlation by **Correntopy**
- ✧ Mean square error (MSE) by **Minimum error entropy** (MEE)
- ✧ Distances in data space by distances in probability spaces



# Information Theoretic Learning

## Entropy

Not all random variables (r.v.) are equally random!



✦ Entropy quantifies the degree of uncertainty in a r.v. Claude Shannon defined entropy as

$$H_S(X) = -\sum p_X(x) \log p_X(x)$$

$$H_S(X) = -\int f_X(x) \log(f_X(x)) dx$$

# Information Theoretic Learning

## Renyi's Entropy

$$H_{\alpha}(X) = \frac{1}{1-\alpha} \log \sum p_X^{\alpha}(x)$$

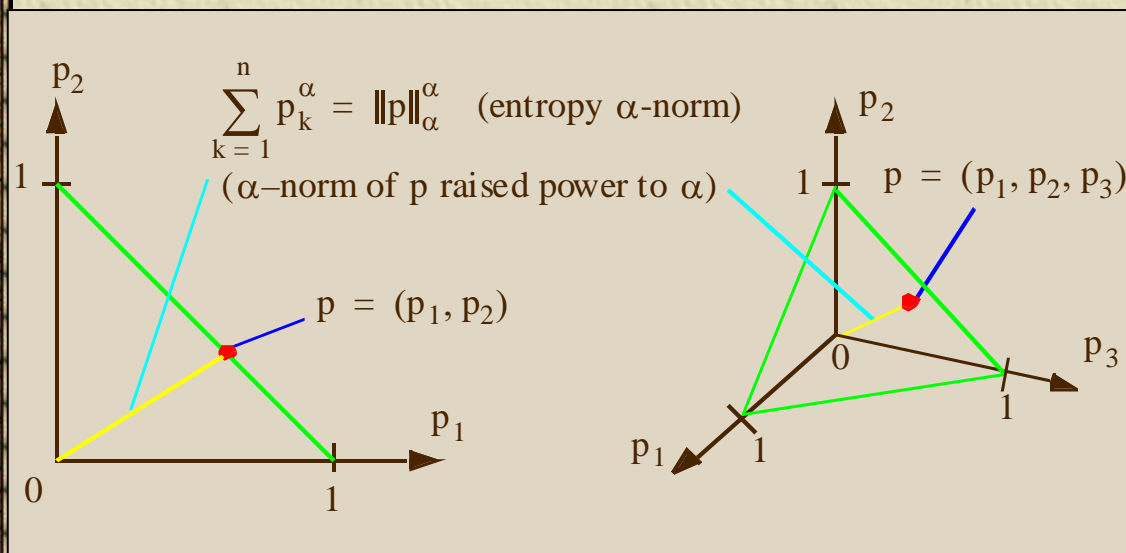
$$H_{\alpha}(X) = \frac{1}{1-\alpha} \log \int f_X^{\alpha}(x) dx$$

Renyi's entropy equals Shannon's as  $\alpha \rightarrow 1$

✧ Norm of the pdf:

$$\alpha - \text{norm} = (V_{\alpha})^{1/\alpha}$$

$$V_{\alpha} = \int f(y)^{\alpha} dy$$





# Information Theoretic Learning

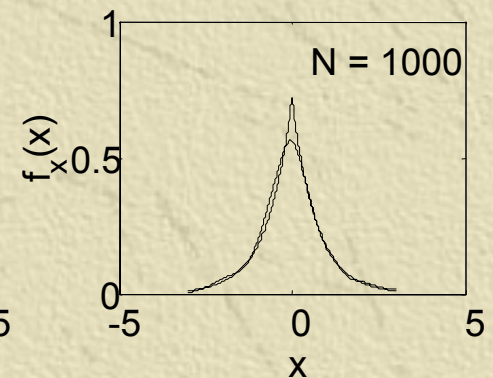
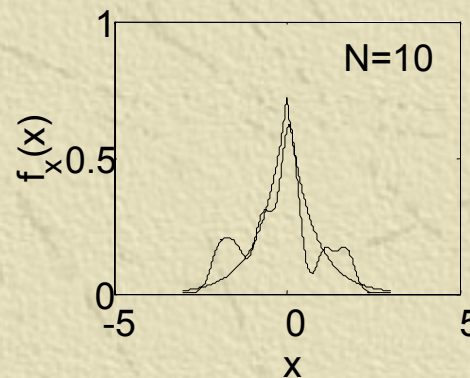
## Parzen windowing

Given only samples drawn from a distribution:

$$\{x_1, \dots, x_N\} \sim p(x)$$

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^N G_{\sigma}(x - x_i)$$

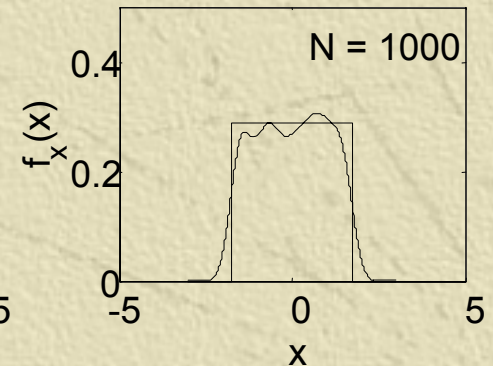
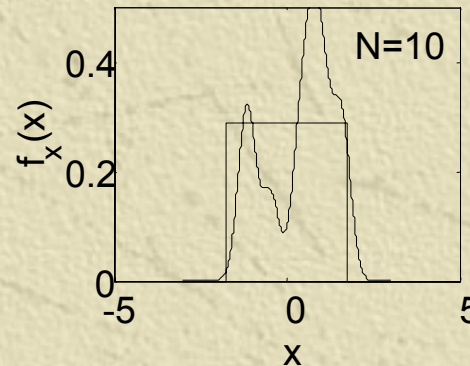
Kernel function



Convergence:

$$\lim_{N \rightarrow \infty} \hat{p}(x) = p(x) * G_{\sigma(N)}(x)$$

*provided that  $N\sigma(N) \rightarrow \infty$*



# Information Theoretic Learning

## Renyi's Quadratic Entropy

Order-2 entropy & Gaussian kernels:

$$\begin{aligned} H_2(X) &= -\log \int p^2(x) dx = -\log \int \left( \frac{1}{N} \sum_{i=1}^N G_\sigma(x - x_i) \right)^2 dx \\ &= -\log \left( \frac{1}{N^2} \sum_j \sum_i \int G_\sigma(x - x_j) G_\sigma(x - x_i) dx \right) \\ &= -\log \underbrace{\left( \frac{1}{N^2} \sum_j \sum_i G_{\sigma\sqrt{2}}(x_j - x_i) \right)}_{\text{Information potential, } V_2(X)} \end{aligned}$$

Pairwise interactions  
between samples  
 $O(N^2)$

Information potential,  $V_2(X)$

$\hat{p}(x)$  provides a potential field over the space of the samples  
parameterized by the kernel size  $\sigma$



# Information Theoretic Learning

## Information Force

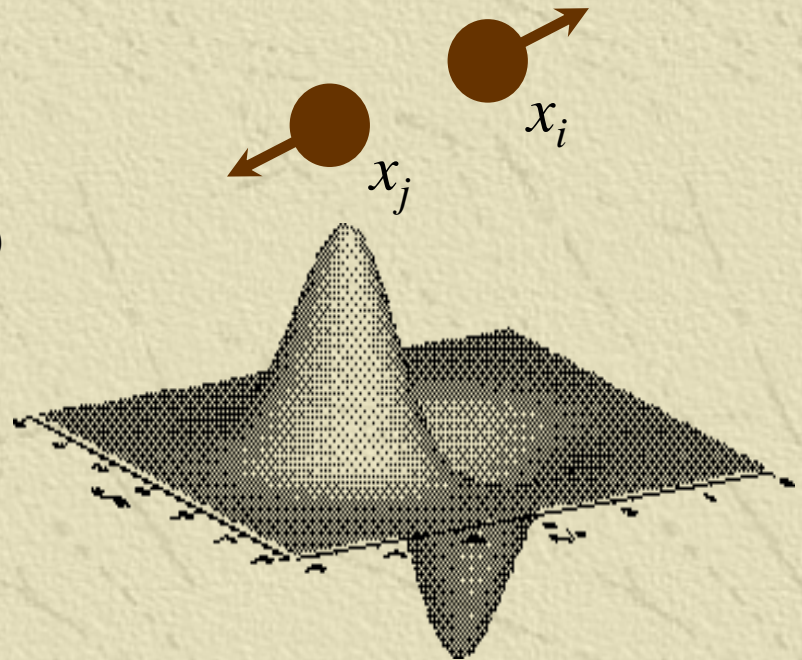
✧ In adaptation, samples become *information particles* that interact through information forces.

Information potential:

$$V_2(X) = \frac{1}{N^2} \sum_j \sum_i G_{\sigma\sqrt{2}}(x_j - x_i)$$

Information force:

$$\frac{\partial V_2}{\partial x_j} = \frac{1}{N^2} \sum_i G'_{\sigma\sqrt{2}}(x_j - x_i)$$

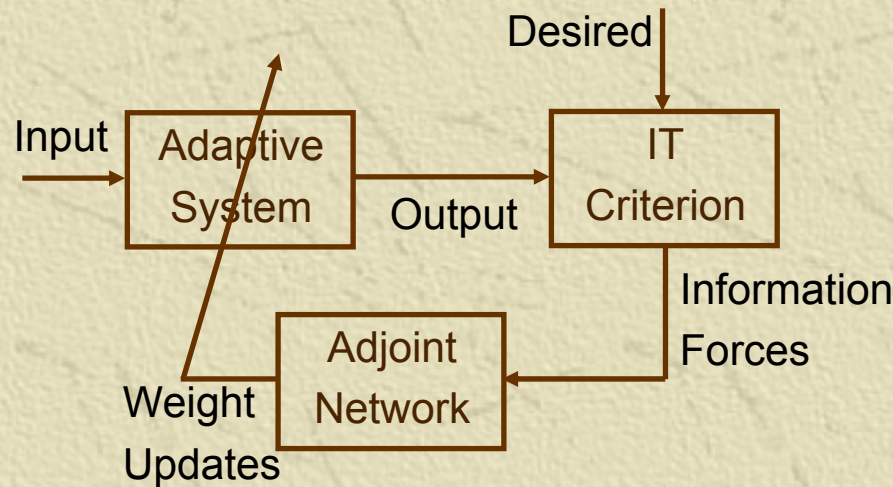


Principe, Fisher, Xu, *Unsupervised Adaptive Filtering*, (S. Haykin), Wiley, 2000.

Erdogmus, Principe, Hild, *Natural Computing*, 2002.

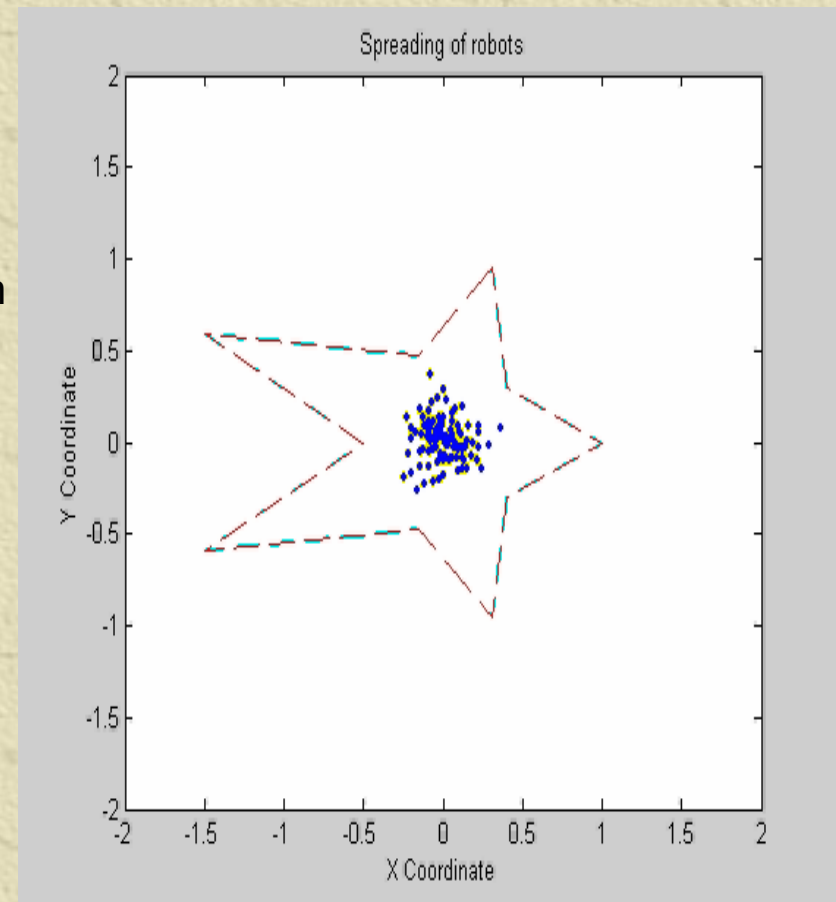
# Information Theoretic Learning

## Backpropagation of Information Forces



$$\frac{\partial J}{\partial w_{ij}} = \sum_{p=1}^k \sum_{n=1}^N \frac{\partial J}{\partial e_p(n)} \frac{\partial e_p(n)}{\partial w_{ij}}$$

Information forces become the **injected error** to the dual or adjoint network that determines the weight updates for adaptation.





# Information Theoretic Learning

## Quadratic divergence measures

Kulback-Liebler  
Divergence:

$$D_{KL}(X;Y) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

Renyi's Divergence:

$$D_{\alpha}(X;Y) = \frac{1}{\alpha-1} \log \int p(x) \left( \frac{p(x)}{q(x)} \right)^{\alpha-1} dx$$

Euclidean Distance:

$$D_E(X;Y) = \int (p(x) - q(x))^2 dx$$

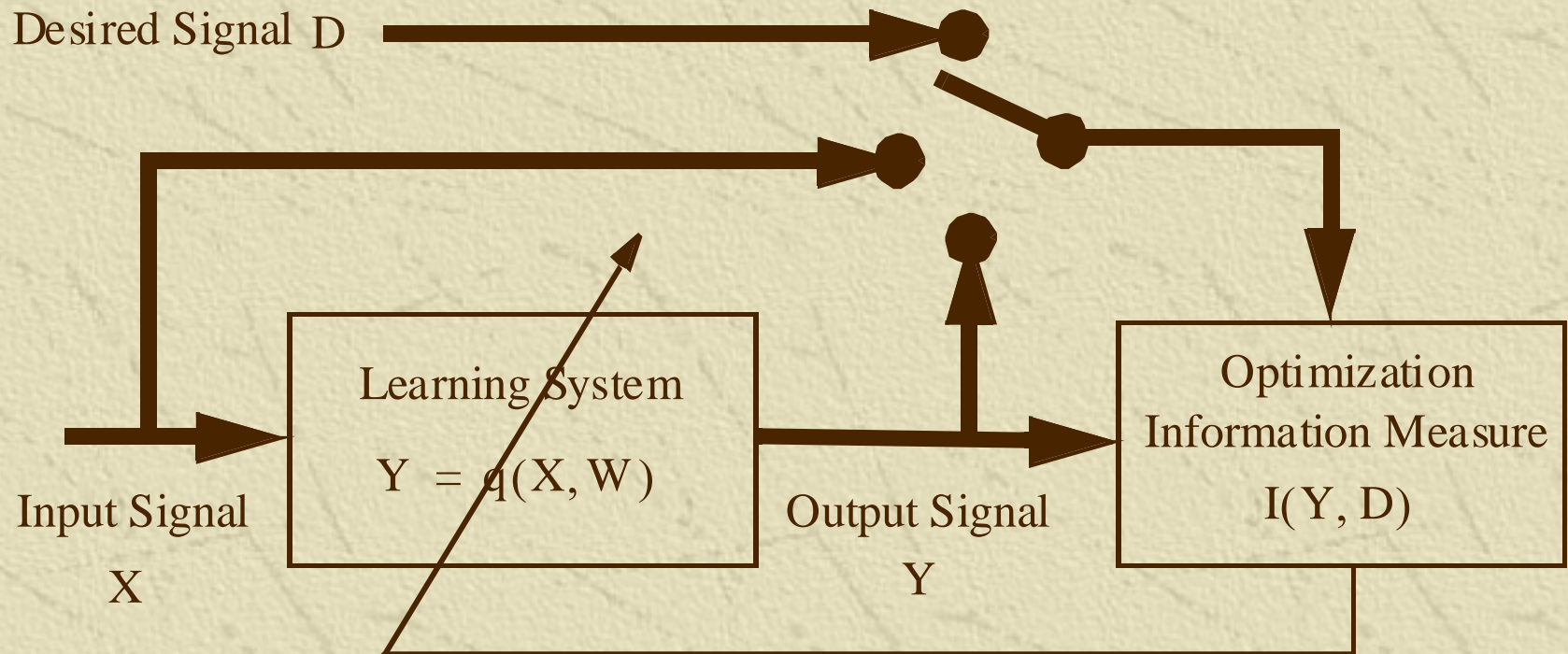
Cauchy- Schwartz  
Distance :

$$D_C(X;Y) = -\log \left( \frac{\int p(x)q(x)dx}{\sqrt{\int p^2(x)dx \int q^2(x)dx}} \right)$$

Mutual Information is a special case (divergence  
between the joint and the product of marginals)

# Information Theoretic Learning

## Unifying criterion for learning from samples





# Training ADALINE sample by sample

## Stochastic information gradient (SIG)

Batch gradient for any  $\alpha$  and any kernel:

$$\frac{\partial \hat{V}_\alpha(e)}{\partial w} = -\frac{(\alpha-1)}{N^\alpha} \sum_j \left( \sum_i \kappa_\sigma(e_j - e_i) \right)^{\alpha-2} \left[ \sum_i \kappa'_\sigma(e_j - e_i) \left( \frac{\partial y_j}{\partial w} - \frac{\partial y_i}{\partial w} \right) \right]$$

Stochastic approximation (SIG) with M samples:

$$\frac{\partial \hat{V}_\alpha(e(n))}{\partial w_k} = -\frac{(\alpha-1)}{M^\alpha} \left( \sum_{i=n-M}^{n-1} \kappa_\sigma(e_n - e_i) \right)^{\alpha-2} \left[ \sum_{i=n-M}^{n-1} \kappa'_\sigma(e_n - e_i) (x_k(i) - x_k(n)) \right]$$

For  $\alpha = 2$  and Gaussian kernel

$$\frac{\partial \hat{V}_\alpha(e(n))}{\partial w_k} = -\frac{1}{M^2} \left[ \sum_{i=n-M}^{n-1} G_\sigma(e_n - e_i) (e_n - e_i) (x_k(i) - x_k(n)) \right]$$

For  $M=1$ , the **SIG becomes LMS equivalent**

$$\frac{\partial \hat{V}_\alpha(e(n))}{\partial w_k} = -G_\sigma(e(n) - e(n-1)) (e(n) - e(n-1)) (x_k(i) - x_k(n))$$

# Training ADALINE sample by sample

## Stochastic information gradient (SIG)

**Theorem:** The expected value of the stochastic information gradient (SIG), is the gradient of Shannon's entropy estimated from the samples using Parzen windowing.

$$\hat{H}_{S,n}(e) \approx -E \left[ \log \left( \frac{1}{L} \sum_{i=n-L}^{n-1} \kappa_{\sigma}(e_n - e_i) \right) \right] \quad \frac{\partial \hat{H}_{S,n}}{\partial w_k} = E \left[ \frac{\sum_{i=n-L}^{n-1} \kappa'_{\sigma}(e_n - e_i)(x_k(n) - x_k(i))}{\sum_{i=n-L}^{n-1} \kappa_{\sigma}(e_n - e_i)} \right]$$

For the Gaussian kernel and  $M=1$

$$\frac{\partial \hat{H}_{S,n}}{\partial w_k} = -\frac{1}{\sigma^2} (e_n - e_{n-1})(x_k(n) - x_k(n-1))$$

The form is the same as for LMS except that **entropy learning works with differences in samples.**

The SIG works implicitly with the **L1 norm of the error.**



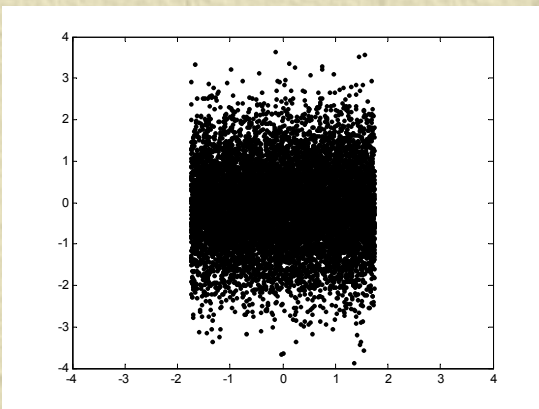
# SIG Hebbian updates

In a linear network the Hebbian update is  $\Delta w_k = \eta y_k x_k$

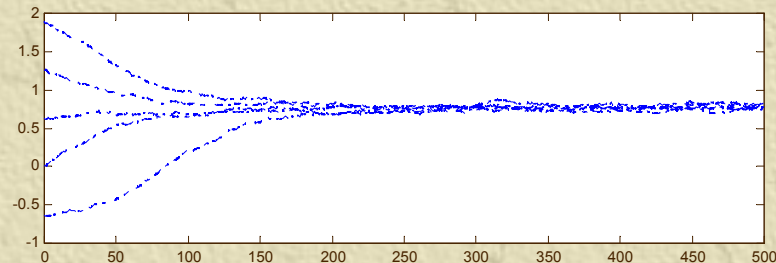
The update maximizing Shannon output entropy with the SIG becomes

$$\frac{\partial \bar{H}_\alpha(y_k)}{\partial w} = \frac{1}{\sigma^2} (y_k - y_{k-1}) \cdot (x_k - x_{k-1})$$

**Which is more powerful and biologically plausible?**



Generated 50 samples of a 2D distribution where the x axis is uniform and the y axis is Gaussian and the sample covariance matrix is 1

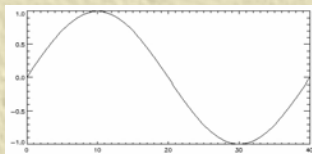
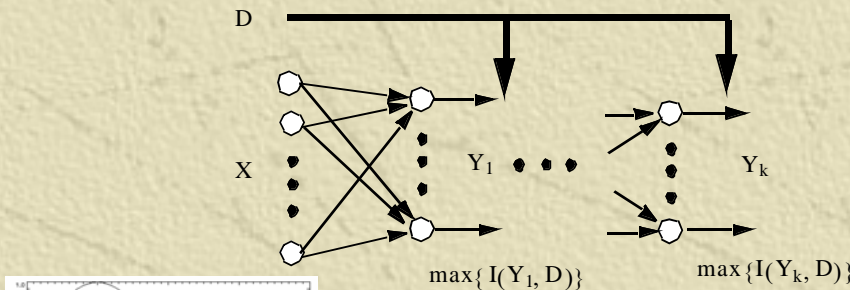
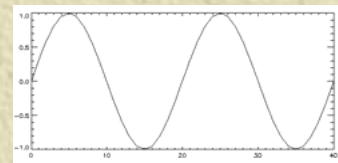


Hebbian updates would converge to any direction but SIG found consistently the 90 degree direction!

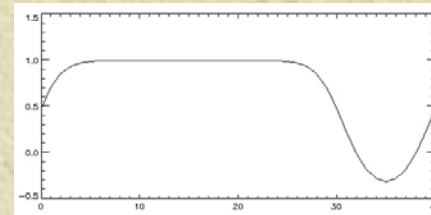
# Training ADALINE sample by sample

- ✪ An MLP can be trained layer by layer, without backpropagating the errors by maximizing the mutual information (QMI CS) between the output of each layer and the desired response.
- ✪ This is similar to Linsker's InfoMax (but nonparametric and with a desired signal). Weights are trained with the delta rule.

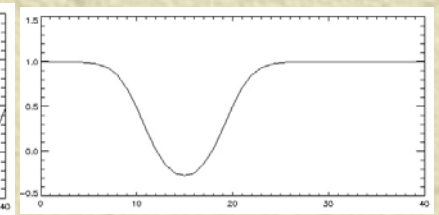
## Frequency doubler network



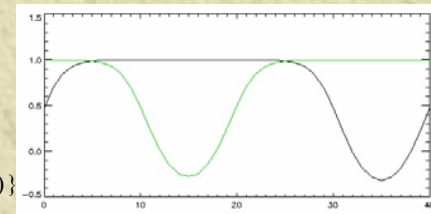
tdnn 5-2-1



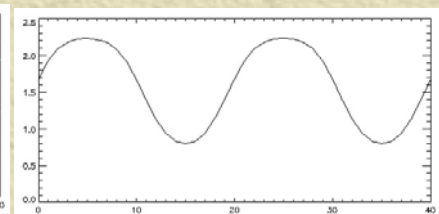
First Hidden Node



Second Hidden Node



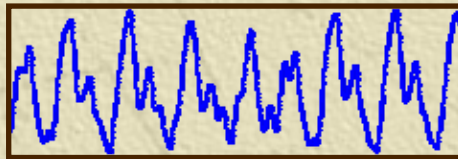
Plot the output of two hidden nodes together



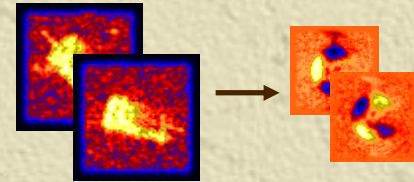
The output of the network



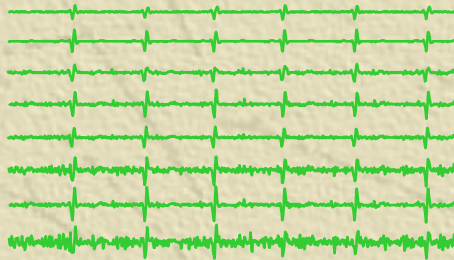
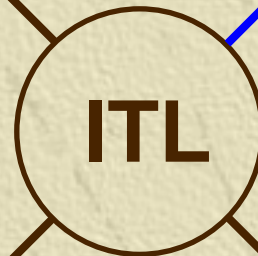
# ITL - Applications



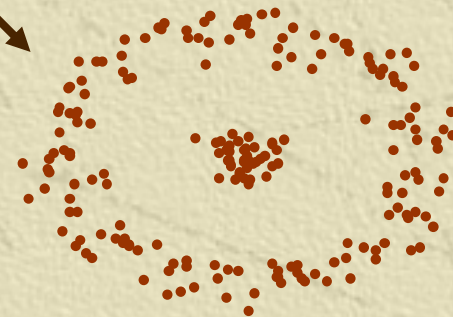
System identification



Feature extraction



Blind source separation



Clustering

[www.cnel.ufl.edu](http://www.cnel.ufl.edu) → ITL has examples and Matlab code



# **Reproducing Kernel Hilbert Spaces as a Tool for Nonlinear System Analysis**



# Fundamentals of Kernel Methods

Kernel methods are a very important class of algorithms for nonlinear optimal signal processing and machine learning. Effectively they are shallow (one layer) neural networks (RBFs).

- ✦ They exploit the linear structure of Reproducing Kernel Hilbert Spaces (RKHS) with very efficient computation.
- ✦ ANY (!) SP algorithm expressed in terms of inner products has in principle an equivalent representation in a RKHS, and may correspond to a nonlinear operation in the input space.
- ✦ Solutions may be analytic instead of adaptive, when the linear structure is used.

# Fundamentals of Kernel Methods

## Definition

- ✧ An Hilbert space is a space of functions  $f(\cdot)$
- ✧ Given a continuous, symmetric, positive-definite kernel  $\kappa: U \times U \rightarrow R$ , a mapping  $\Phi$ , and an inner product  $\langle \cdot, \cdot \rangle_H$
- ✧ A RKHS  $H$  is the closure of the span of all  $\Phi(u)$ .

- ✧ Reproducing

$$\langle f, \Phi(u) \rangle_H = f(u)$$

- ✧ Kernel trick

$$\langle \Phi(u_1), \Phi(u_2) \rangle_H = \kappa(u_1, u_2)$$

- ✧ The induced norm

$$\|f\|_H^2 = \langle f, f \rangle_H$$



# Fundamentals of Kernel Methods

## RKHS proposed by Parzen

For a stochastic process  $\{X_t, t \in T\}$  with  $T$  being an index set, the auto-correlation function  $R_X : T \times T \rightarrow \mathbb{R}$  defined as

$$R_X(t, s) = E[X_t X_s]$$

is symmetric and positive-definite, thus induces a RKHS by the Moore-Aronszajn theorem, denoted as PRKHS (PH).

Further there is a kernel mapping  $\Theta$  such that

$$R_X(t, s) = \langle \Theta(t), \Theta(s) \rangle_{PH}$$

Second order statistics on r. p. are equivalent to algebraic operations in the PRKHS.

PRKHS brought understanding but no new results.

# Fundamentals of Kernel Methods

## RKHS induced by the Gaussian kernel

The Gaussian kernel is symmetric and positive definite

$$k_{\sigma}(x, x') = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - x')^2}{2\sigma^2}\right).$$

thus induces a RKHS on a sample set  $\{x_1, \dots, x_N\}$  of reals, denoted as GRKHS or GH.

Further, by Mercer's theorem, a kernel mapping  $\Phi$  can be constructed which transforms data from the input space to GRKHS where:

$$k_{\sigma}(x - x_i) = \langle \Phi(x), \Phi(x_i) \rangle_{GH}$$

where  $\langle, \rangle$  denotes inner product in GRKHS.



# ITL and Kernel Methods:

## Central moments in feature space

ITL is also a kernel method.

Information potential:

$$V_2(X) = \frac{1}{N^2} \sum_j \sum_i G_\sigma(x_j - x_i)$$

$$G_\sigma(x_j - x_i) = \Phi^T(x_j) \Lambda \Phi(x_i) = \langle \Phi(x_j), \Phi(x_i) \rangle$$

Substituting in the information potential

$$V_2(X) = \frac{1}{N^2} \sum_j \sum_i \Phi^T(x_j) \Lambda \Phi(x_i) = \left\| \hat{\boldsymbol{\mu}}_{\Phi(x)} \right\|^2$$

So Renyi's quadratic entropy is the log of the norm square of the projected data mean, i.e. a central moment in feature space

# ITL and Kernel Methods:

## Central Moments Estimators

**Mean:**

$$E[X] \cong \frac{1}{N} \sum_{i=1}^N x_i$$

**Variance**

$$E[(X - E[X])^2] \cong \frac{1}{N} \sum_{i=1}^N (x_i - \frac{1}{N} \sum_{i=1}^N x_i)^2$$

**Information Potential**

$$E[f(x)] \cong \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N G_{\sigma\sqrt{2}}(x_i - x_j) \quad \sigma \rightarrow \text{scale}$$

**Renyi's Quadratic Entropy**

$$-\log E[f(x)] \cong -\log\left(\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N G_{\sigma\sqrt{2}}(x_i - x_j)\right)$$



# ITL and Kernel Methods:

## SVM as a ITL cost function

The SVM classifier maximizes

$$L = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N,N} \alpha_i \alpha_j d_i d_j k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{subject} \quad \sum_{i=1}^N \alpha_i d_i = 1, \quad \alpha_i \geq 0, \quad \forall i$$
$$\sum_{i=1}^{N_1} \alpha_i = \sum_{j=1}^{N_2} \alpha_j^* = A$$

Rewrite as

$$L = 2A - \frac{A^2}{2} \left\{ \frac{1}{A^2} \sum_{i,i'=1}^{N_1,N_1} \alpha_i \alpha_{i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) + \frac{1}{A^2} \sum_{i,i'=1}^{N_2,N_2} \alpha_i \alpha_{i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) - \frac{2}{A^2} \sum_{i,i'=1}^{N_1,N_2} \alpha_i \alpha_{i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) \right\}$$
$$= 2A - \frac{A^2}{2} D_{ED}(\hat{p}, \hat{q})$$

Where

$$\hat{p}(\mathbf{x}) = \frac{1}{A} \sum_{i=1}^{N_1} \alpha_i k(\mathbf{x}, \mathbf{x}_i) \quad \hat{q}(\mathbf{x}) = \frac{1}{A} \sum_{j=1}^{N_2} \alpha_j^* k(\mathbf{x}, \mathbf{x}_j)$$

# Adaptive Filtering in RKHS

## The Kernel LMS algorithm

Widrow's famed LMS algorithm can be written easily in RKHS

$$\min_w R_{emp} = \sum_{i=1}^N (d_i - w_i x_i)^2$$

$$w_0 = 0$$

$$e_n^a = d_n - w_{n-1} x_n$$

$$w_n = w_{n-1} + \eta e_n^a x_n$$

$$w_n = \eta \sum_{i=1}^n e_i^a x_i$$

$$y_t = w_n x_t = \eta \sum_{i=1}^n e_i^a \langle x_i, x_t \rangle_{L2}$$

$$\min_{\Omega} R_{emp} = \sum_{i=1}^N (d_i - \Omega_i \Phi(x_i))^2$$

$$\Omega_0 = 0$$

$$e_n^a = d_n - \Omega_{n-1} \Phi(x_n)$$

$$\Omega_n = \Omega_{n-1} + \eta e_n^a \Phi(x_n)$$

$$\Omega_n = \eta \sum_{i=1}^n e_i^a \Phi(x_i)$$

$$y_t = \Omega_n \Phi(x_t) = \eta \sum_{i=1}^n e_i^a \langle \Phi(x_i), \Phi(x_t) \rangle_f$$

$$y_t = \eta \sum_{i=1}^n e_i^a G(x_i - x_t)$$



# Adaptive Filtering in RKHS

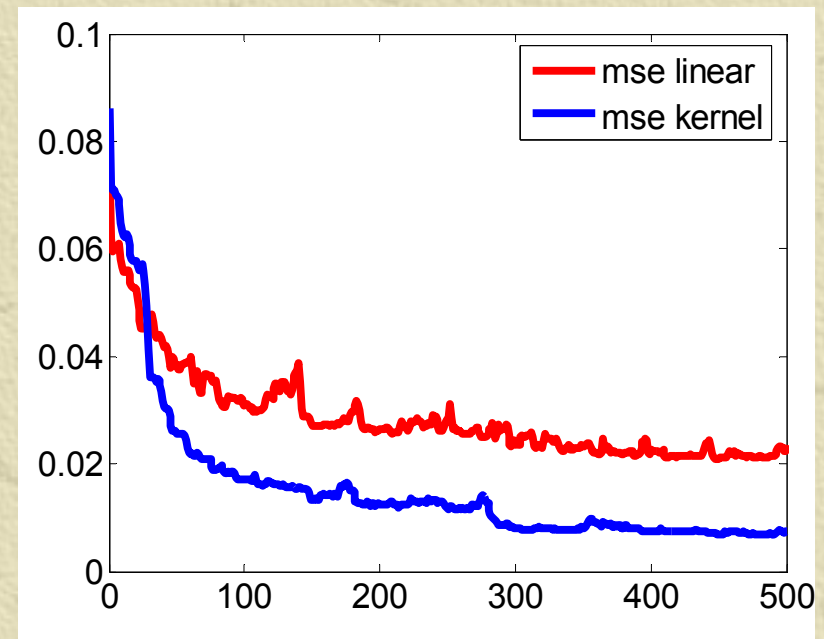
## The Kernel LMS algorithm

The KLMS gives rise to a RBF network that can be **trained on-line**, where the **weights are the errors** and it is **constantly growing** with each sample.

We have recently proved that the KLMS is well posed in the sense of Hadamar, so it does not need regularization as most kernel algorithms in the literature.

Prediction of Mackey-Glass  
Time series (t=30)

$\eta=0.2$





# Correntropy:

## A new generalized similarity measure

Correlation is one of the most widely used functions in signal processing.

But, correlation only quantifies similarity fully if the random variables are Gaussian distributed.

Can we define a new function that measures similarity but it is not restricted to second order statistics?

Use the ITL framework.



# Correntropy:

## A new generalized similarity measure

Define correntropy of a random process  $\{x_t\}$  as

$$V_x(t, s) = E(\delta(x_t - x_s)) = \int \delta(x_t - x_s) p(x) dx$$

We can easily estimate correntropy using kernels

$$\hat{V}_x(t, s) = E(k(x_t - x_s))$$

The name correntropy comes from the fact that the average over the lags (or the dimensions) is the information potential (the argument of Renyi's entropy)

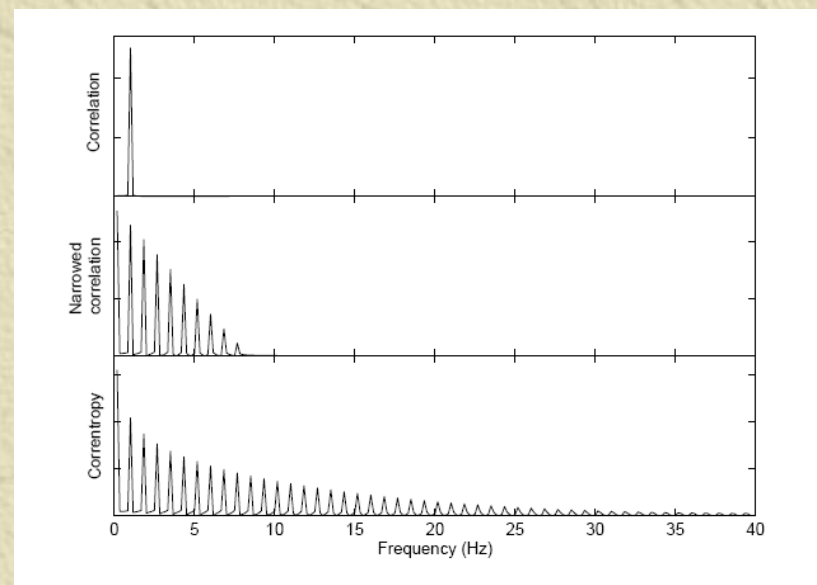
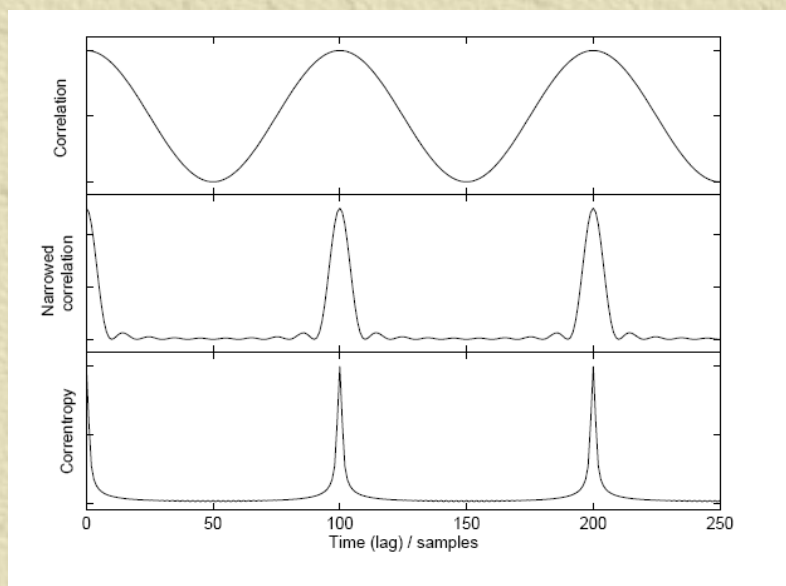
For strictly stationary and ergodic r. p.

$$\hat{V}_m = \frac{1}{N} \sum_{n=1}^N k(x_n - x_{n-m})$$

# Correntropy:

## A new generalized similarity measure

How does it look like? The sinewave





# Correntropy:

## A new generalized similarity measure

Properties of Correntropy:

- ✧ It has a maximum at the origin ( $1/\sqrt{2\pi\sigma}$  )
- ✧ It is a symmetric positive function
- ✧ Its mean value is the information potential
- ✧ Correntropy includes higher order moments of data

$$V(s, t) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n \sigma^{2n} n!} E \|x_s - x_t\|^{2n}$$

- ✧ The matrix whose elements are the correntropy at different lags is Toeplitz

# Correntropy:

## A new generalized similarity measure

✧ Correntropy as a cost function versus MSE.

$$MSE(X, Y) = E[(X - Y)^2]$$

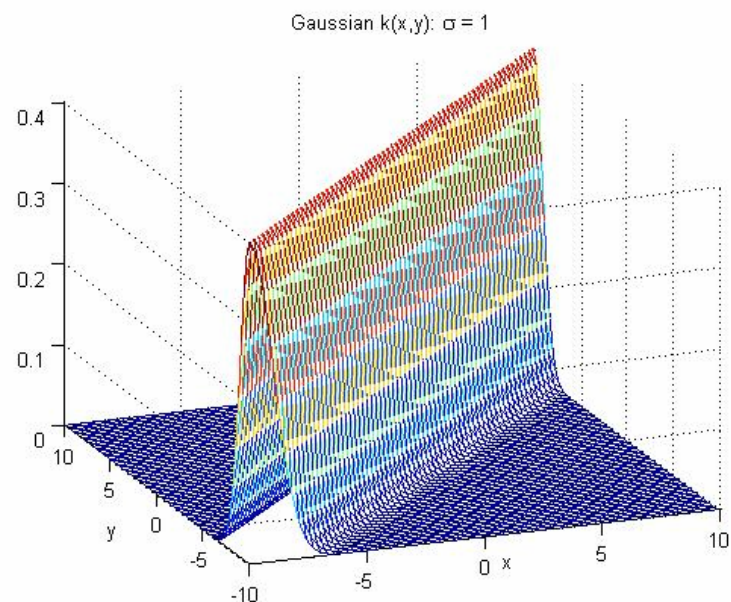
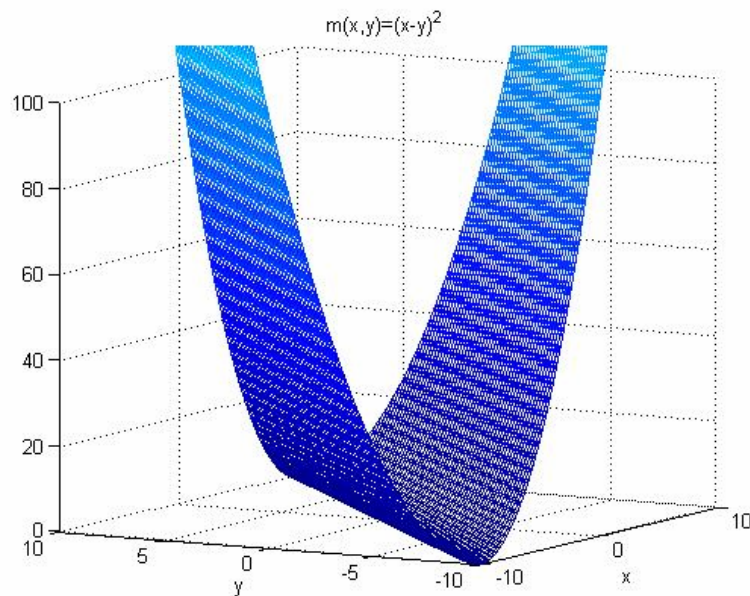
$$= \iint_{x, y} (x - y)^2 f_{XY}(x, y) dx dy$$

$$= \int_e e^2 f_E(e) de$$

$$V(X, Y) = E[k(X - Y)]$$

$$= \iint_{x, y} k(x - y) f_{XY}(x, y) dx dy$$

$$= \int_e k(e) f_E(e) de$$





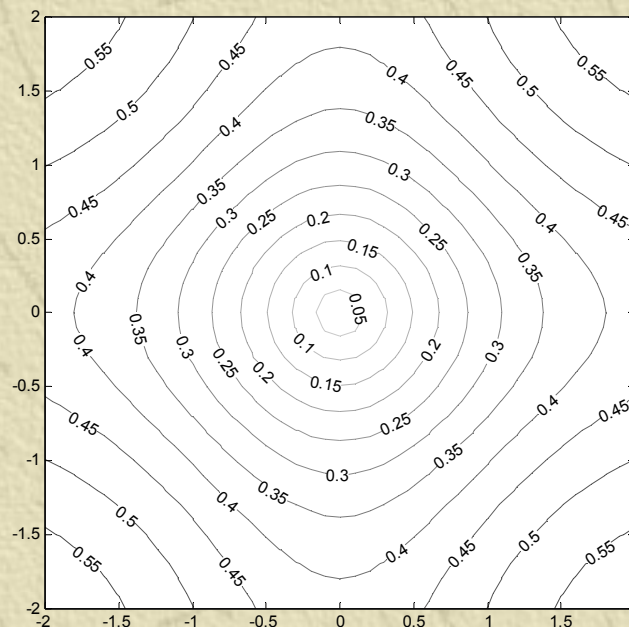
# Correntropy:

## A new generalized similarity measure

- ✧ Correntropy induces a metric (CIM) in the sample space defined by

$$CIM(X, Y) = (V(0, 0) - V(X, Y))^{1/2}$$

- ✧ Therefore correntropy can be used as an alternative similarity criterion in the space of samples.



# Correntropy:

## A new generalized similarity measure

- ✧ Correntropy criterion implements M estimation of robust statistics. M estimation is a generalized maximum likelihood method.

$$\arg_{\theta} \min \sum_{i=1}^N \rho(x, \theta) \quad \sum_{i=1}^N \psi(x_i, \hat{\theta}_M) = 0 \quad \psi = \rho'$$

In adaptation the weighted square problem is defined as

$$\lim_{\theta} \sum_{i=1}^N w(e_i) e_i^2 \quad w(e) = \rho'(e) / e$$

When

$$\rho(e) = (1 - \exp(-e^2 / 2\sigma^2)) / \sqrt{2\pi}\sigma$$

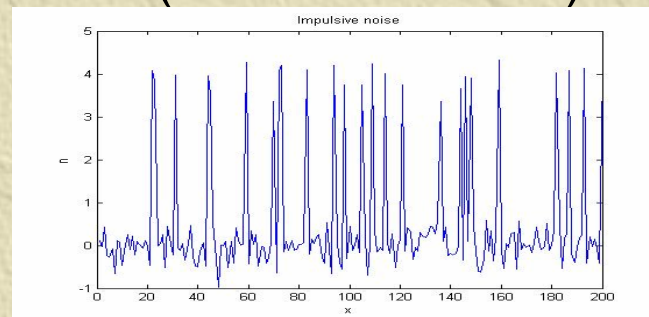
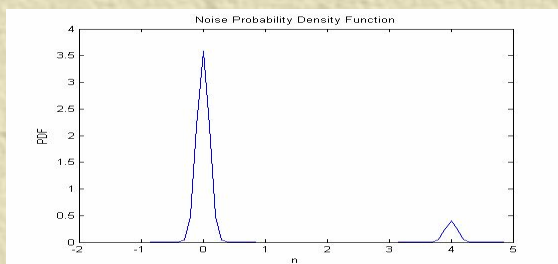
this leads to maximizing the correntropy of the error at the origin.

$$\begin{aligned} \min_{\theta} \sum_{i=1}^N \rho(e_i) &= \min_{\theta} \sum_{i=1}^N (1 - \exp(-e_i^2 / 2\sigma^2)) / \sqrt{2\pi}\sigma \\ \Leftrightarrow \max_{\theta} \sum_{i=1}^N \exp(-e_i^2 / 2\sigma^2) / \sqrt{2\pi}\sigma &= \max_{\theta} \sum_{i=1}^N \kappa_{\sigma}(e_i) \end{aligned}$$

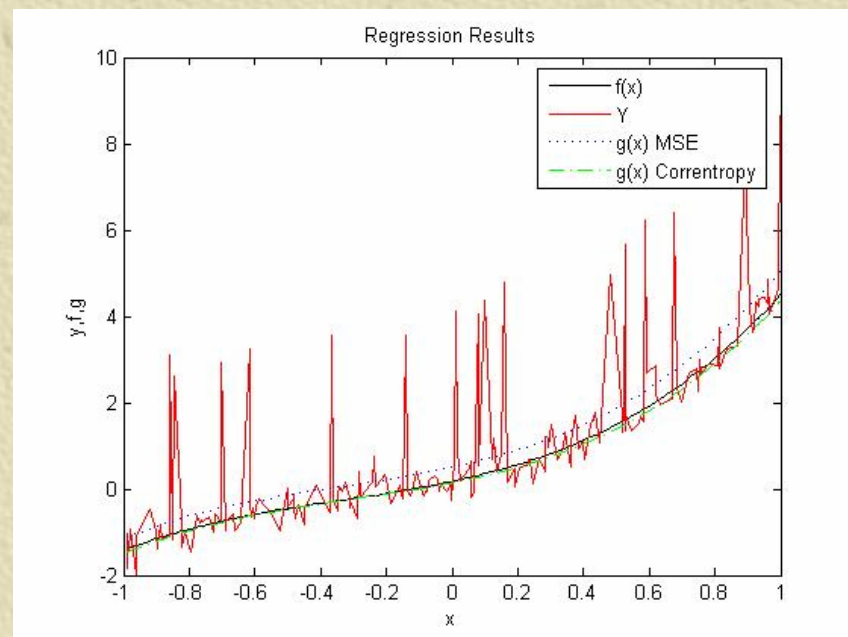
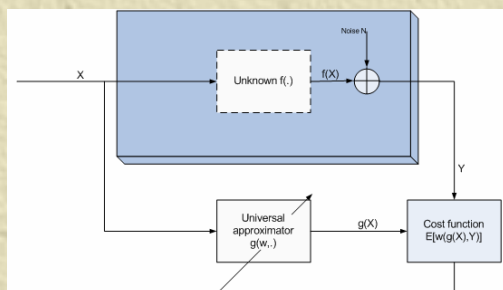


# Correntropy: A new generalized similarity measure

✧ Nonlinear regression with outliers (Middleton model)



✧ Polynomial approximator



# RKHS induced by Correntropy

## Definition

For a stochastic process  $\{X_t, t \in T\}$  with  $T$  being an index set, correntropy defined as

$$V_X(t,s) = E[K(X_t X_s)]$$

is symmetric and positive definite.

Thus it induces a new RKHS denoted as VRKHS ( $V_H$ ). There is a kernel mapping  $\Psi$  such that

$$V_X(t,s) = \langle \Psi(t), \Psi(s) \rangle_{V_H}$$

Any symmetric non-negative kernel is the covariance kernel of a random function and vice versa (Parzen).

Therefore, given a random function  $\{X_t, t \in T\}$  there exists another random function  $\{f_t, t \in T\}$  such that

$$E[f_t f_s] = V_X(t,s)$$



# RKHS induced by Correntropy

## Relation with PRKHS

- ✧ Correntropy estimates similarity in feature space. It transforms the component wise data to feature space by  $f(\cdot)$  and then takes the correlation

$$V(x, y) = E[k(x, y)] = E[f(x).f(y)]$$

$f(\cdot)$  is a function also implicitly defined (as  $\phi$  was)

- ✧ VRKHS is nonlinearly related to the input space (unlike PRKHS).
- ✧ VRKHS seems very appropriate to EXTEND traditional linear signal processing methods to nonlinear system identification (there is no need for regularization and time consuming quadratic programming).

# RKHS induced by Correntropy

## Relation with GRKHS.

- ✧ Gram matrix estimates pairwise evaluations in feature space. It is a sample by sample evaluation of functions
- ✧ Gram matrix is  $N \times N$  (data size)
- ✧ Correntropy matrix is  $L \times L$  (space size)

This means huge savings:

If we have a million samples in a  $5 \times 5$  input space, Gram matrix is  $10^6 \times 10^6$

The correntropy matrix is  $5 \times 5$ , but each element is computed from a million pairs.

(very much like a MA model of order 5 estimated with one million samples)



# Applications of VRKHS

## Matched Filtering

Matched filter computes the inner product between the received signal  $r(n)$  and the template  $s(n)$  ( $R_{sr}(0)$ ).

The Correntropy MF computes

$$V_{rs}(0) = \frac{1}{N} \sum_{i=1}^N k(r_i - s_i)$$

Hypothesis	received signal	Similarity value
$H_0$	$r_k = n_k$	$V_0 = \frac{1}{N\sqrt{2\pi\sigma^2}} \sum_{i=1}^N e^{-(s_i - n_i)^2 / 2\sigma^2}$
$H_1$	$r_k = s_k + n_k$	$V_1 = \frac{1}{N\sqrt{2\pi\sigma^2}} \sum_{i=1}^N e^{-n_i^2 / 2\sigma^2}$

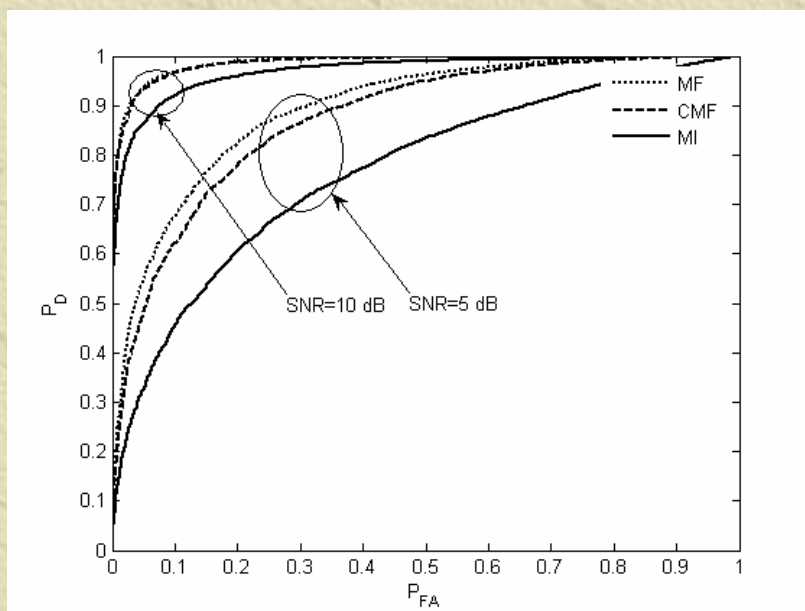
*(Patent pending)*

# Applications of VRKHS

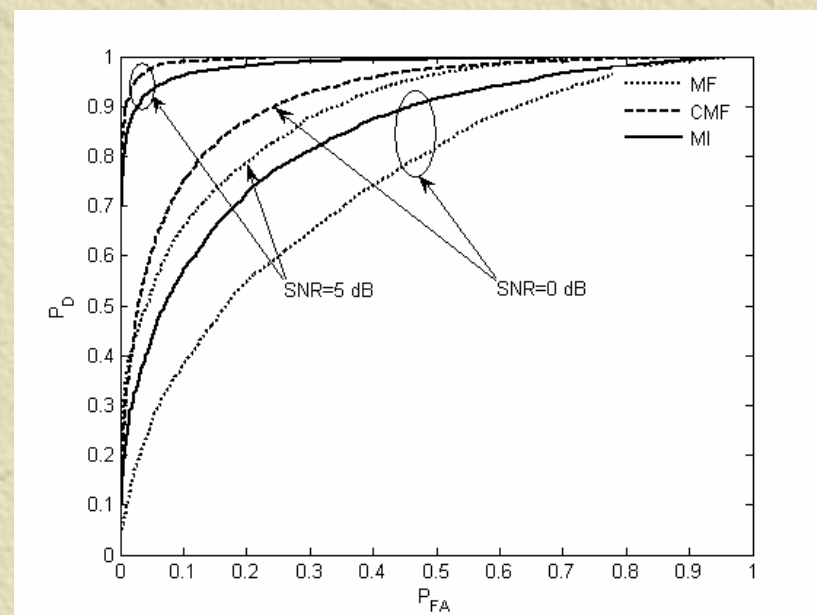
## Matched Filtering

Linear Channels

White Gaussian noise



Impulsive noise



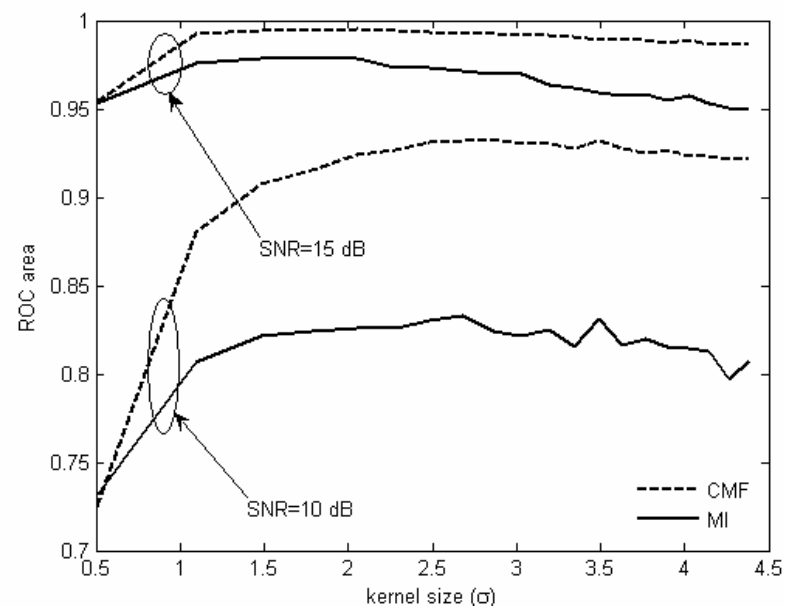
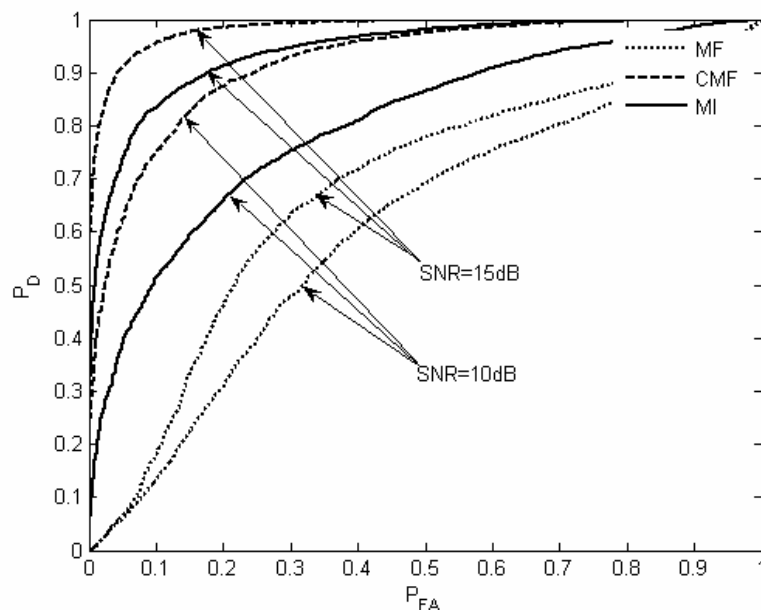
Template: binary sequence of length 20. kernel size using Silverman's rule.



# Applications of VRKHS

## Matched Filtering

Alpha stable noise ( $\alpha=1.1$ ), and the effect of kernel size



Template: binary sequence of length 20. kernel size using Silverman's rule.

# Applications of VRKHS

## Minimum Average Correlation Energy filters

- ✠ We consider a 2-dimensional image data as a  $d \times 1$  column vector.
- ✠ The conventional MACE filter is formulated in the freq. domain.

➤ Training Image matrix :  $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N],$

➤ Correlation energy of the  $i$ -th image  $E_i = \mathbf{H}^H \mathbf{D}_i \mathbf{H},$

$\mathbf{D}_i$  is a diagonal matrix whose elements are the magnitude squared of the associated element of,  $\mathbf{X}_i$  that is, the power spectrum of  $x_i(n)$

➤ Correlation value at the origin  $g_i(0) = \mathbf{X}_i^H \mathbf{H} = c_i,$

➤ Average energy over all training images  $E_{avg} = \mathbf{H}^H \mathbf{D} \mathbf{H},$

➤ MACE filter minimizes the average energy while satisfying the constraint

$$\mathbf{H} = \mathbf{D}^{-1} \mathbf{X} (\mathbf{X}^H \mathbf{D}^{-1} \mathbf{X})^{-1} \mathbf{c}.$$



# Application of VRKS

## The Correntropy MACE filter

- Exploiting the linear space properties of the VRKHS, the optimization problem in the feature space becomes

$$\min \mathbf{f}_h^T \mathbf{V}_X \mathbf{f}_h, \quad \text{subject to} \quad \mathbf{F}_X^T \mathbf{f}_h = \mathbf{c}.$$

- Solution

$$\mathbf{f}_h = \mathbf{V}_X^{-1} \mathbf{F}_X (\mathbf{F}_X^T \mathbf{V}_X^{-1} \mathbf{F}_X)^{-1} \mathbf{c}.$$

Do not know  $\mathbf{F}$  explicitly! But can compute

- Test output

$$\mathbf{y} = \mathbf{F}_Z^T \mathbf{V}_X^{-1} \mathbf{F}_X (\mathbf{F}_X^T \mathbf{V}_X^{-1} \mathbf{F}_X)^{-1} \mathbf{c}.$$

$$\mathbf{K}_{ZX} = \mathbf{F}_Z^T \mathbf{V}_X^{-1} \mathbf{F}_X$$

$$\mathbf{K}_{XX} = (\mathbf{F}_X^T \mathbf{V}_X^{-1} \mathbf{F}_X)^{-1}$$

$$(\mathbf{K}_{XX})_{ij} = \sum_{l=1}^d \sum_{k=1}^d w_{lk} f(x_i(k)) f(x_j(l)) \cong \sum_{l=1}^d \sum_{k=1}^d w_{lk} k(x_i(k) - x_j(l)),$$

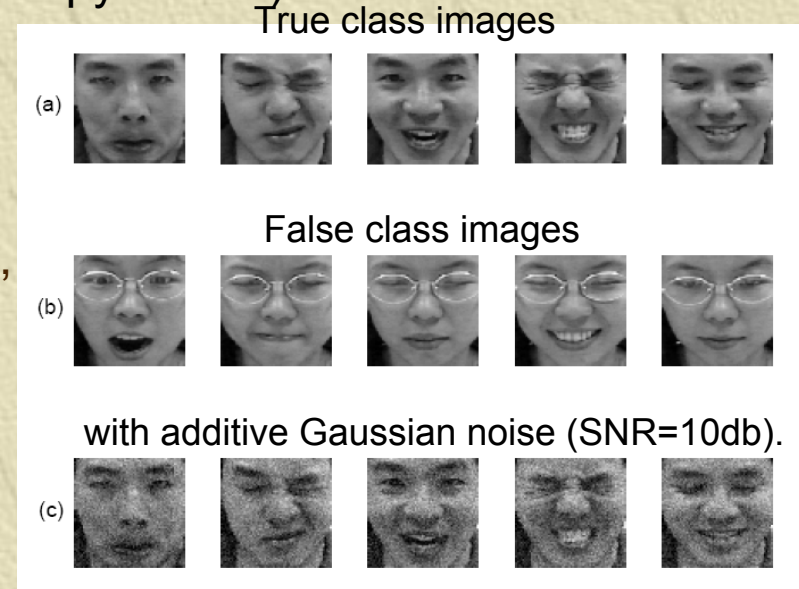
$$(\mathbf{K}_{ZX})_{ij} = \sum_{l=1}^d \sum_{k=1}^d w_{lk} f(z_i(k)) f(x_j(l)) \cong \sum_{l=1}^d \sum_{k=1}^d w_{lk} k(z_i(k) - x_j(l))$$

# Applications of VRKHS

## VMACE Simulation Results

### ❑ Face Recognition (MACE vs. Correntropy MACE)

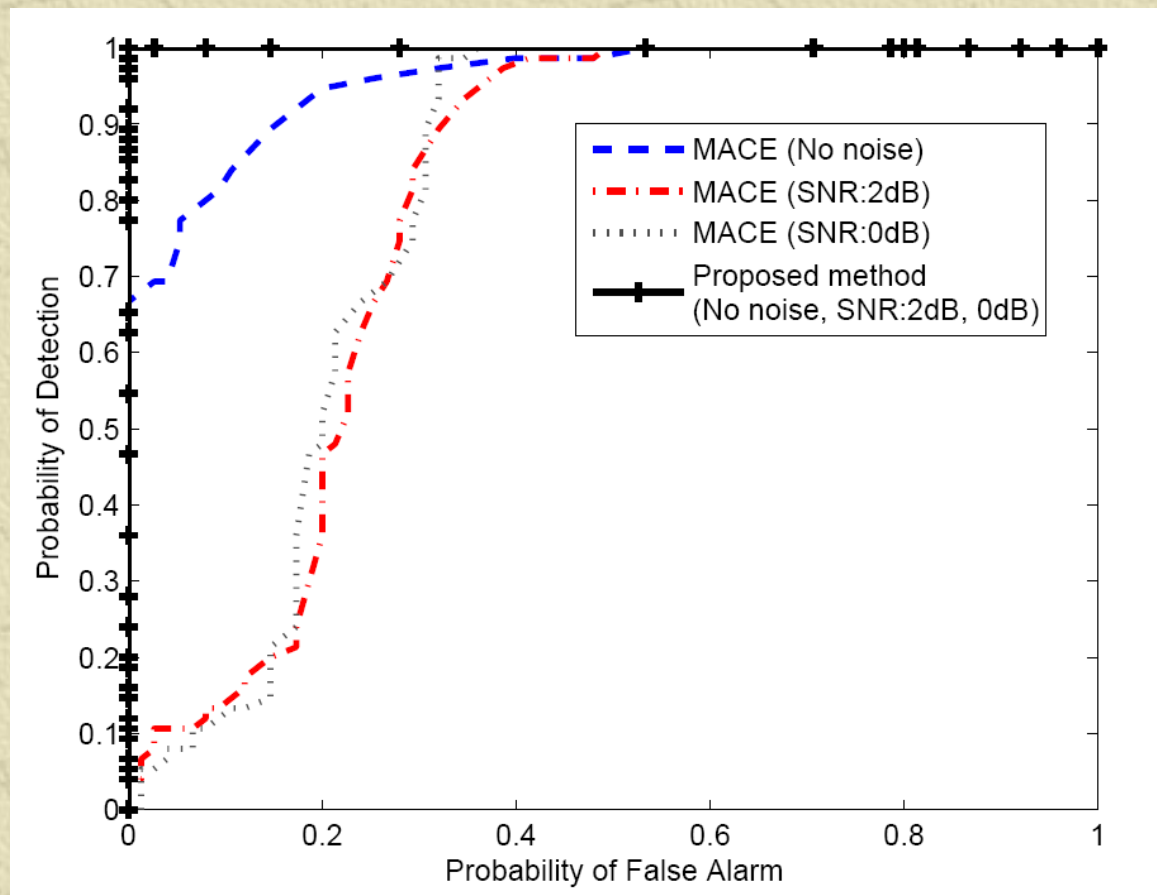
- We used the facial expression database from the Advanced Multimedia Processing Lab in CMU.
- The database consists of 13 subjects, whose facial images were captured with 75 varying expressions.
- The size of each image is 64x 64.
- ✓ Report the results of the **two most difficult cases** who produced the worst performance with the MACE method.
- ✓ The simulation results have been obtained by averaging (Monte-Carlo approach) over 100 different training sets (each training set consists of randomly chosen 5 images)
- ✓ The kernel size is  $\sim 40\%$  of the standard deviation of the input data.





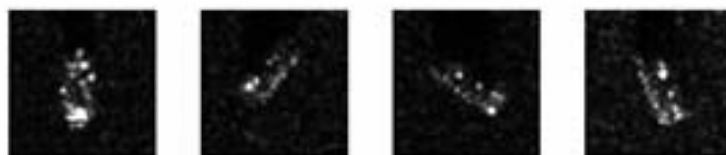
# Applications of VRKHS

## VMACE ROC curves with different SNRs

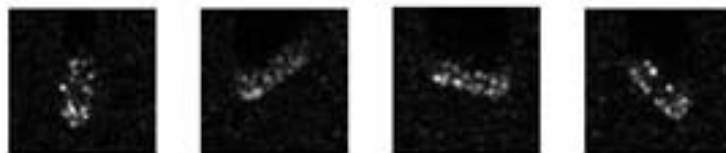


# Applications of VRKHS

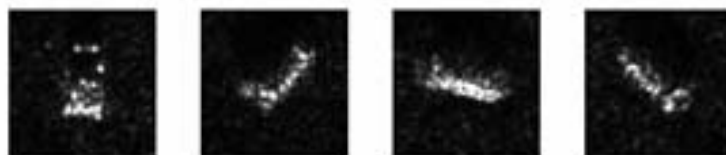
## CMACE on SAR/ATR – aspect angle mismatch



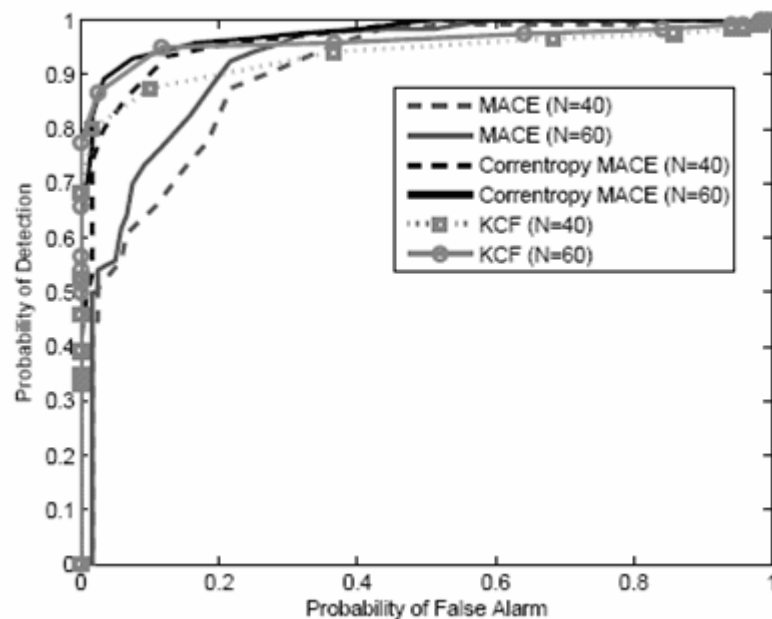
(a) Training images (BTR 60) of aspect angle 0, 35, 124, 159 degrees



(b) Test images from BTR 60 of aspect angle 3, 53, 104, 137 degrees



(c) Test images from confuser (T62) of aspect angle 2, 41, 103, 137 degrees





# Applications of Correntropy

## Wiener Filtering

We can also formulate the Wiener filter as a Correntropy filter instead of SVM regression by directly exploiting the linear structure of the VRKHS. However, there is still an approximation in the computation.

$$\begin{aligned}y(n) &= F^T(n)\Omega \\&= F^T(n)V^{-1}\frac{1}{N}\sum_{k=1}^Nd(k)F(k) \\&= \frac{1}{N}\sum_{k=1}^N\sum_{j=0}^{L-1}\sum_{i=0}^{L-1}f(n-i)a_{ij}f(k-j)d(k) \\&= \frac{1}{N}\sum_{k=1}^Nd(k)\sum_{j=0}^{L-1}\sum_{i=0}^{L-1}a_{ij}\{f(n-i)f(k-j)\} \\&\cong \frac{1}{N}\sum_{k=1}^N\left\{d(k)\sum_{j=0}^{L-1}\sum_{i=0}^{L-1}a_{ij}K(x(n-i),x(k-j))\right\}\end{aligned}$$

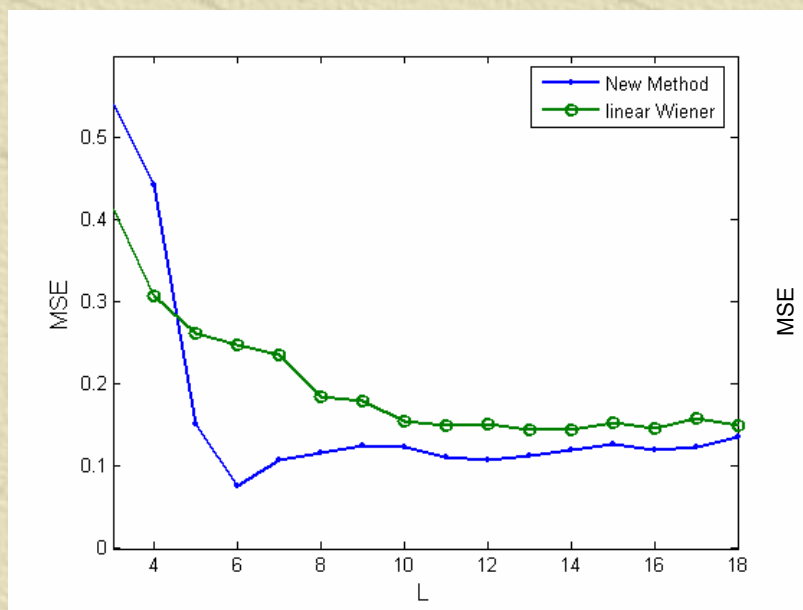
Patent Pending

# Applications of Correntropy

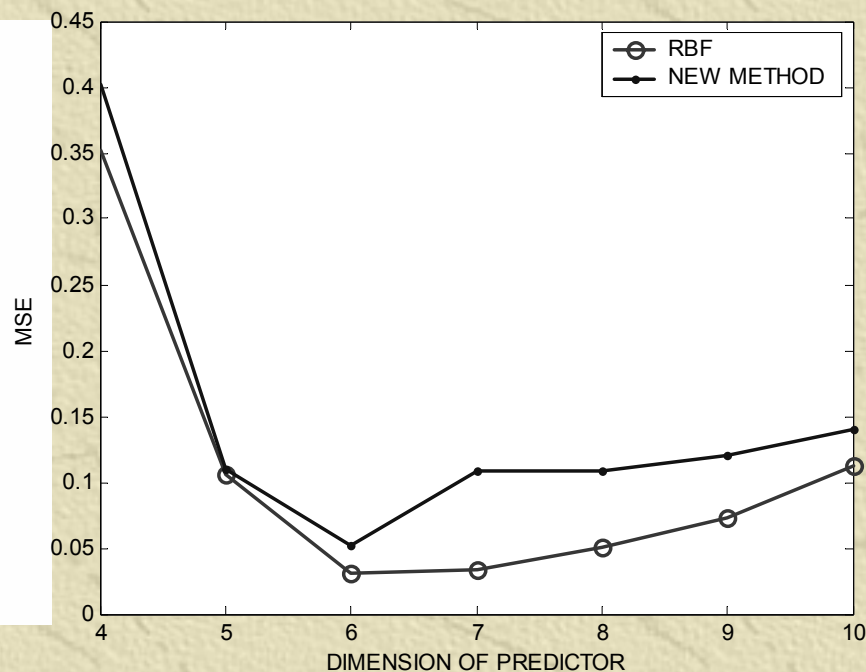
## Wiener Filtering

Prediction of the Mackey Glass equation ( a mild chaotic system)

Wiener versus correntropy



RBF (50) versus correntropy





# Applications of VRKHS

## Nonlinear temporal PCA

The Karhunen Loeve transform performs Principal Component Analysis (TPCA) of the autocorrelation of the r. p.

$$X = \begin{bmatrix} x(1) & \dots & x(N) \\ \dots & \dots & \dots \\ x(L) & \dots & x(N+L-1) \end{bmatrix}_{L \times N}$$

$$R = XX^T$$

$$\approx N \times \begin{bmatrix} r(0) & r(1) & \dots & r(L-1) \\ r(1) & O & O & M \\ M & O & r(0) & r(1) \\ r(L-1) & L & r(1) & r(0) \end{bmatrix}_{L \times L}$$

$$K = X^T X$$

$$\approx L \times \begin{bmatrix} r(0) & r(1) & \dots & r(N-1) \\ r(1) & O & O & M \\ M & O & r(0) & r(1) \\ r(N-1) & L & r(1) & r(0) \end{bmatrix}_{N \times N}$$

$$R = XX^T = UDD^T U^T$$

$$K = X^T X = VD^T DV^T$$

D is LxN diagonal

$$\{\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_L}\}$$

$$U_i^T X = \sqrt{\lambda_i} V_i^T, \quad i = 1, 2, \dots, L$$

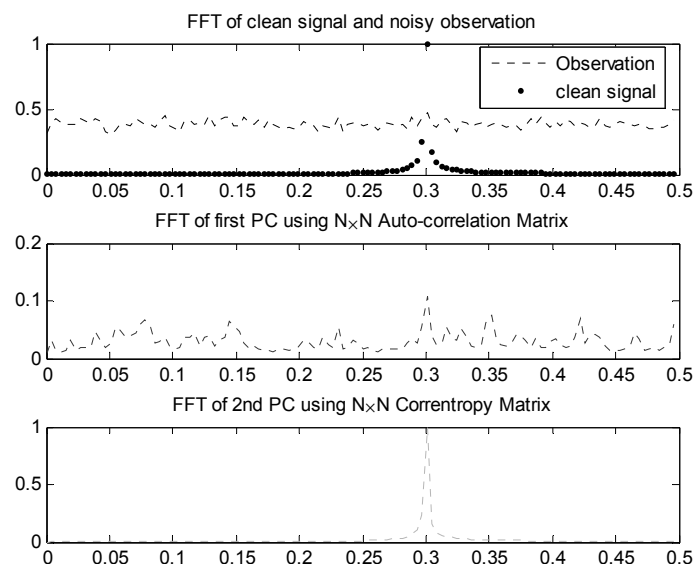
TPCA can be also done by decomposing the Gram matrix K directly.

# Applications of VRKHS

## Nonlinear temporal PCA

Since the autocorrelation function of the projected data in VRKHS is given by correntropy, we can directly construct  $K$  with the correntropy.

Example:  $x(m) = A \sin(2\pi fm) + z(m)$  where  $p_N(n) = 0.8 \times N(0, 0.1) + 0.1 \times N(4, 0.1) + 0.1 \times N(-4, 0.1)$



A	VPCA (2 <sup>nd</sup> PC)	PCA by N-by-N (N=256)	PCA by L-by-L (L=4)	PCA by L-by-L (L=100)
0.2	100%	15%	3%	8%
0.25	100%	27%	6%	17%
0.5	100%	99%	47%	90%

1,000 Monte Carlo runs.  $\sigma=1$

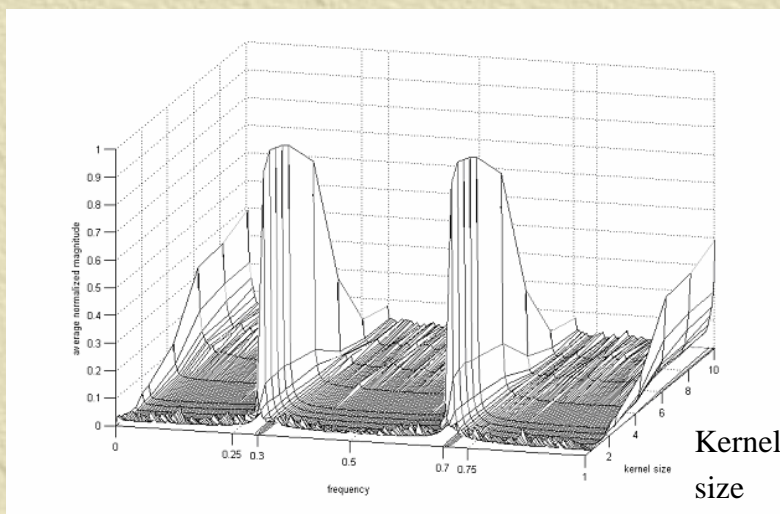


# Applications of VRKHS

## Correntropy Spectral Density

CSD is a function of the kernel size, and shows the difference between PSD ( $\sigma$  large) and the new spectral measure

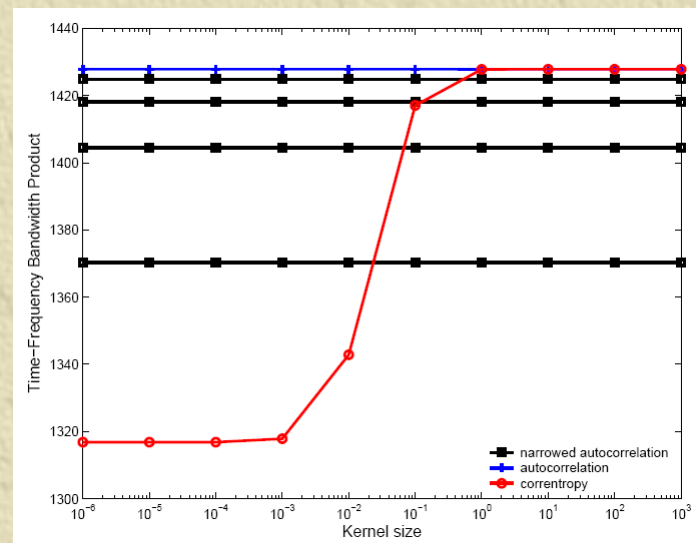
Average normalized amplitude



frequency

Kernel  
size

Time Bandwidth product



Kernel size

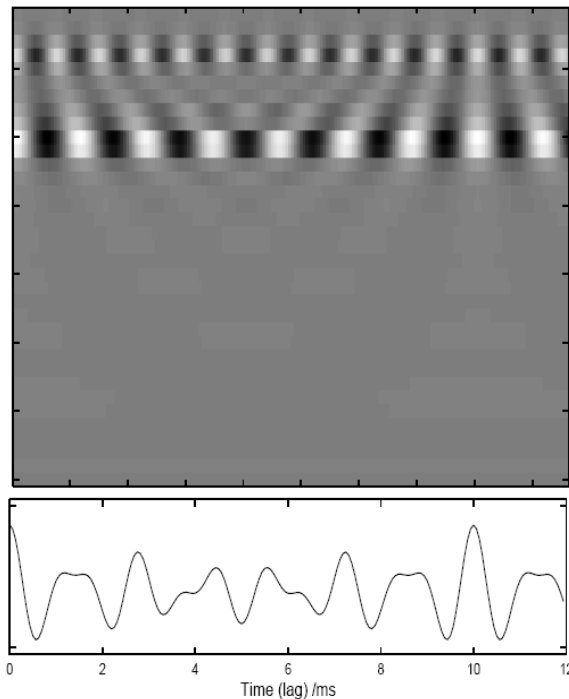
# Applications of VRKHS

## Correntropy based correlograms

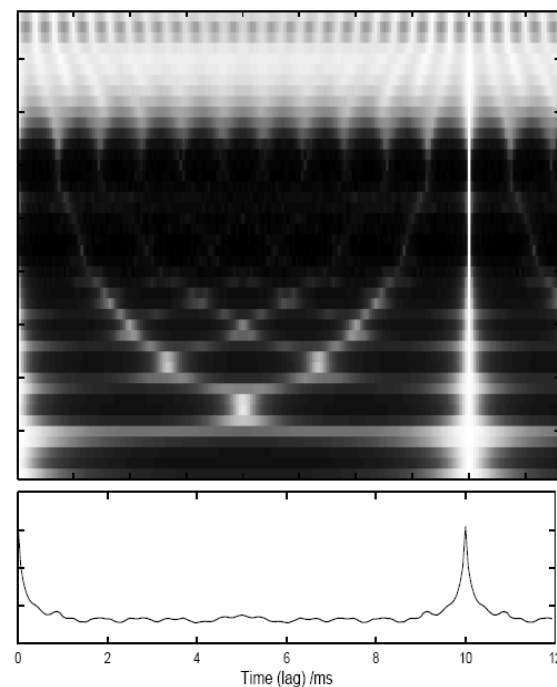
Correntropy can be used in computational auditory scene analysis (CASA), providing much better frequency resolution.

Figures show the correlogram from a 30 channel cochlea model for one (pitch=100Hz)).

Auto-correlation Function



Auto-correntropy Function





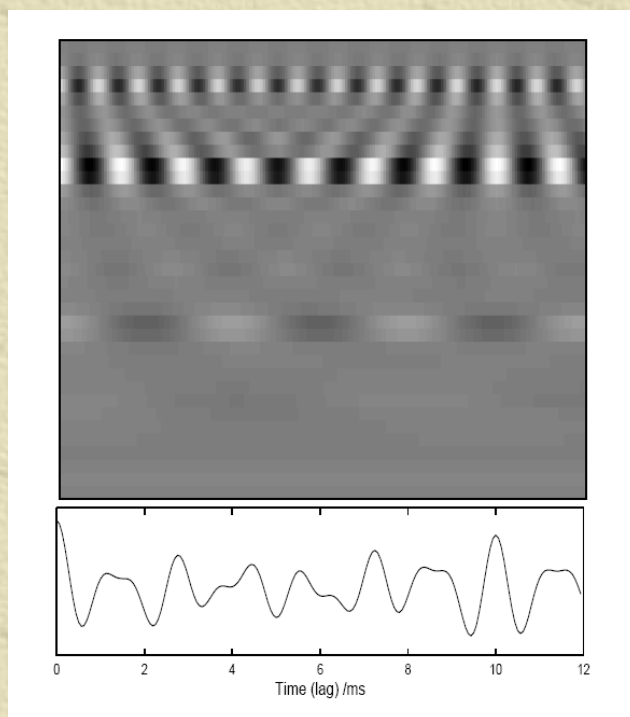
# Applications of VRKHS

## Correntropy based correlograms

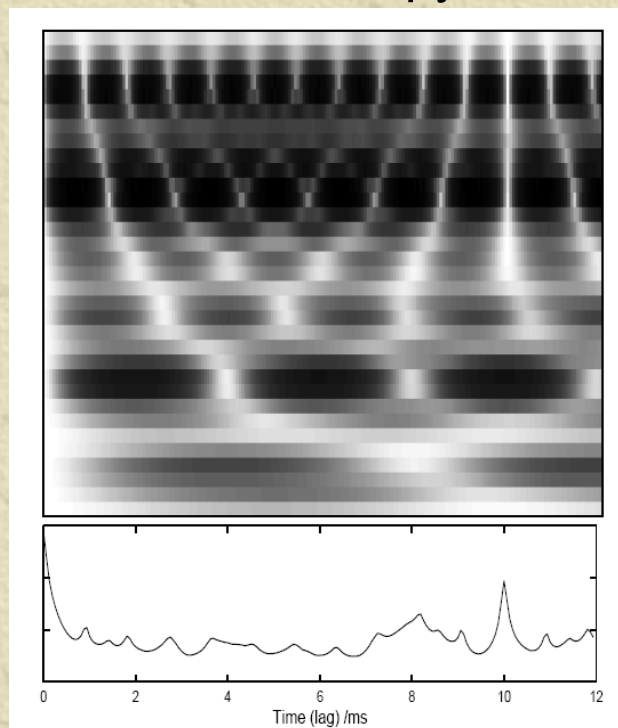
Correntropy can be used in computational auditory scene analysis (CASA), providing much better frequency resolution.

Figures show the correlogram from a 30 channel cochlea model for two superimposed vowels from two speakers ( $f_0=100, 126$  Hz)).

Auto-correlation Function



Auto-correntropy Function



# Conclusions

- ✧ Information Theoretic Learning took us out of the local minimum of Gaussian statistics and MSE as cost functions.
  - ◆ ITL generalizes many of the statistical concepts we take for granted.
- ✧ Kernel methods implement shallow neural networks (RBFs) and extend easily the linear algorithms we all know.
  - ◆ KLMS is a simple algorithm for on-line learning of nonlinear systems
- ✧ Correntropy defines a new RKHS that seems to be very appropriate for nonlinear system identification and robust control
  - ◆ Correntropy may take us out of the local minimum of the (adaptive) design of optimum linear systems

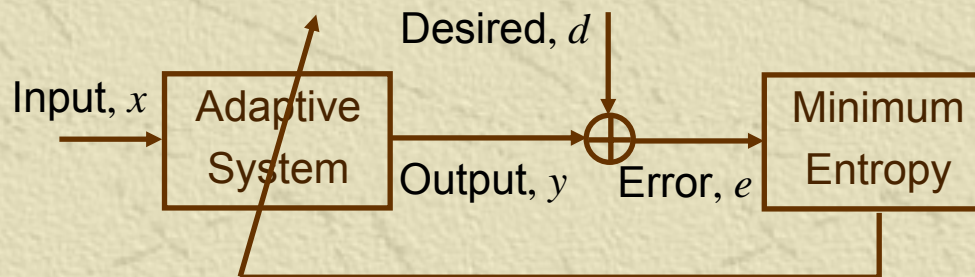
For more information go to the website [www.cnel.ufl.edu](http://www.cnel.ufl.edu) → ITL resource for tutorial, demos and downloadable MATLAB code



# ITL – Applications

## Nonlinear system identification

- ✧ Minimize information content of the residual error



- ✧ Equivalently provides the best density matching between the output and the desired signals.

$$\min_{\mathbf{w}} \frac{1}{1-\alpha} \log \int p_e^\alpha(\varepsilon; \mathbf{w}) d\varepsilon \equiv \min_{\mathbf{w}} \iint p_{xy}(\xi, \eta; \mathbf{w}) \left( \frac{p_{xy}(\xi, \eta; \mathbf{w})}{p_{xd}(\xi, \eta)} \right)^{\alpha-1} d\xi d\eta$$

# ITL – Applications

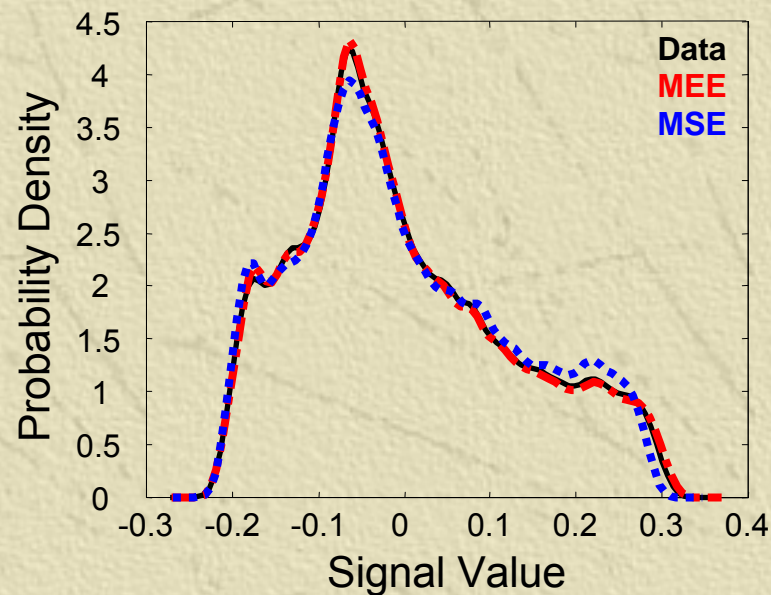
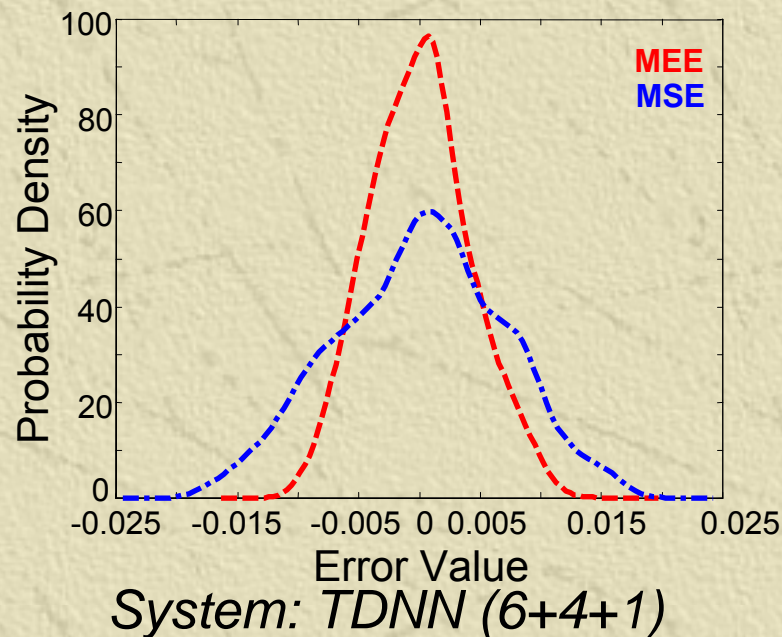
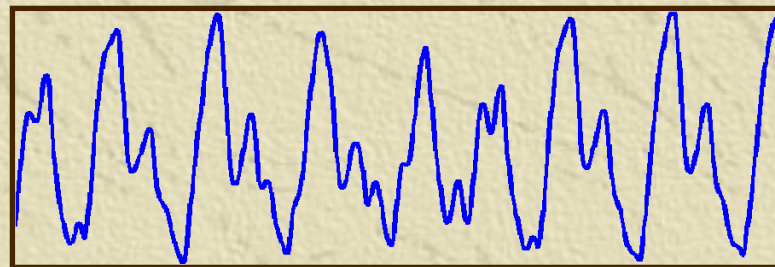
## Time-series prediction

Chaotic Mackey-Glass (MG-30) series

Compare 2 criteria:

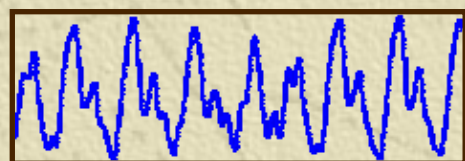
*Minimum squared-error*

*Minimum error entropy*

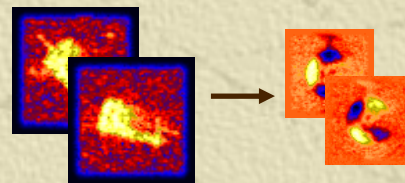




# ITL - Applications

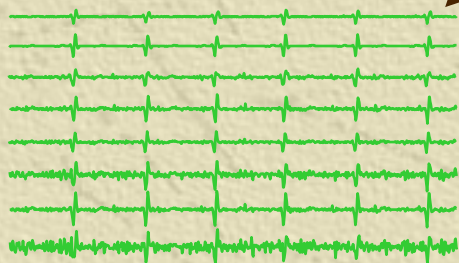


System identification

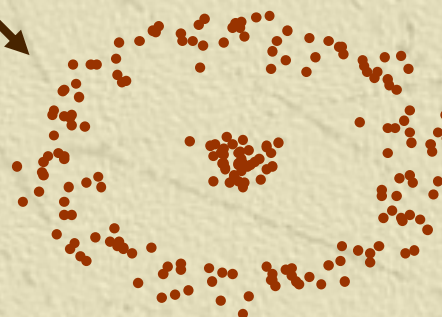


Feature extraction

ITL



Blind source separation

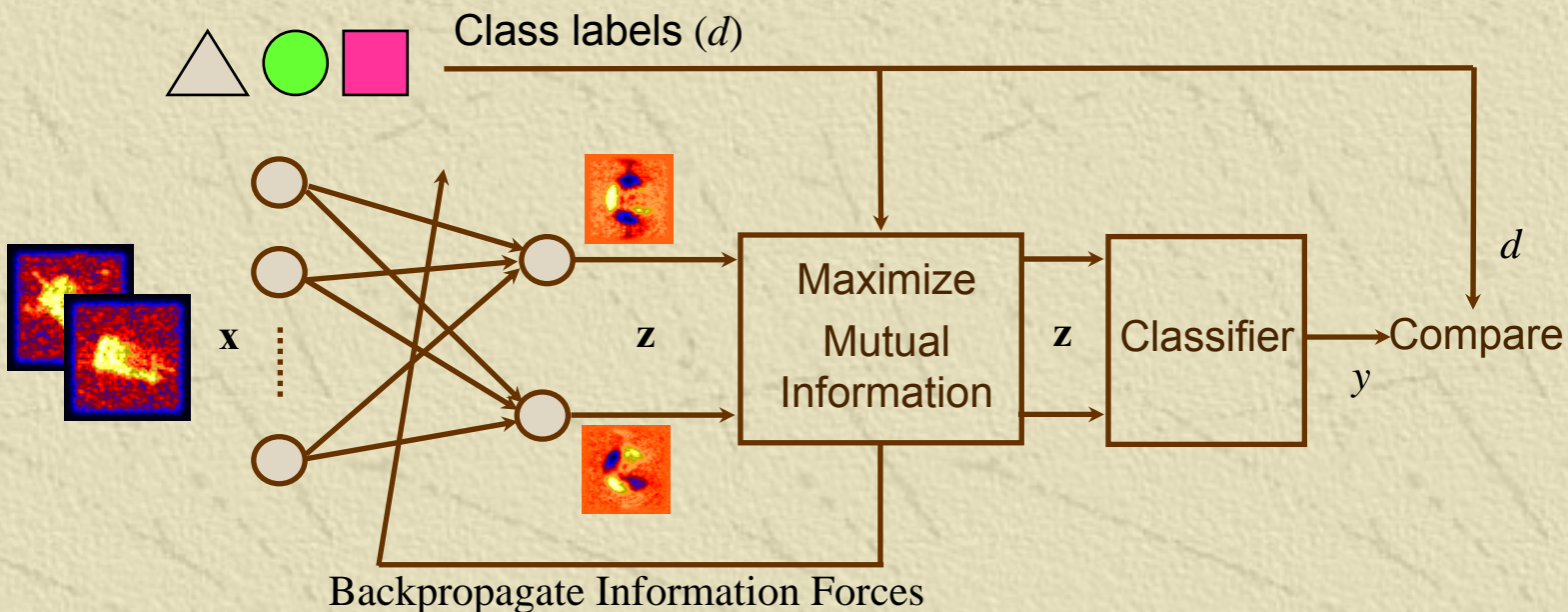


Clustering

# ITL – Applications

## Optimal feature extraction

- ✧ Data processing inequality: Mutual information is monotonically non-increasing.
- ✧ Classification error inequality: Error probability is bounded from below and above by the mutual information.



*PhD on feature extraction for sonar target recognition (2002)*

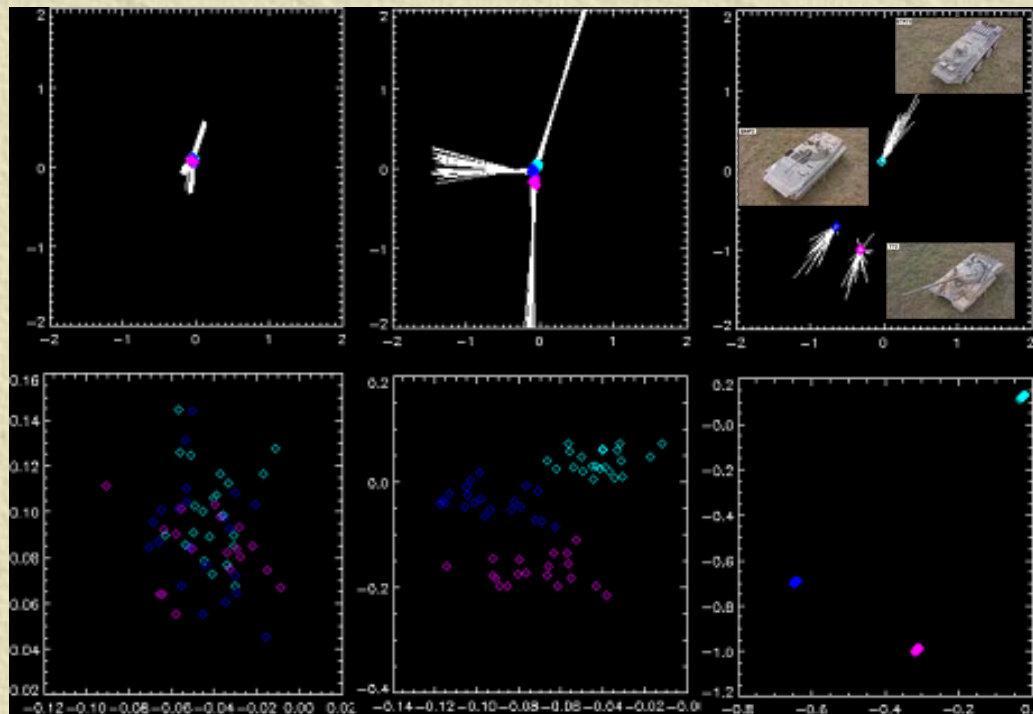


# ITL – Applications

## Extract 2 nonlinear features

64x64 SAR images of 3 vehicles: BMP2, BTR70, T72

Information forces in training



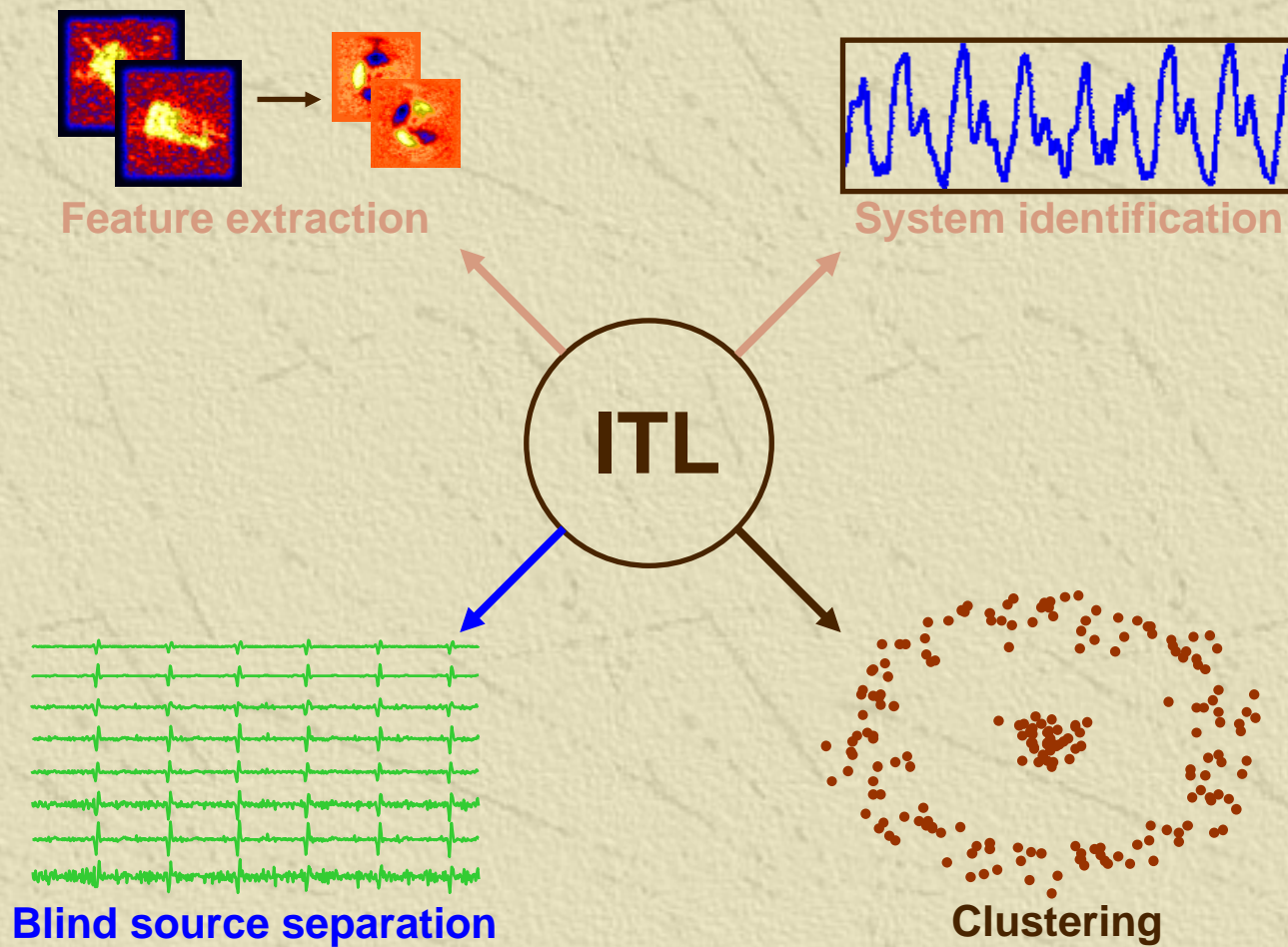
Classification results

	P(Correct)
MI+LR	94.89%
SVM	94.60%
Templates	90.40%



Zhao, Xu and Principe, SPIE Automatic Target Recognition, 1999.  
Hild, Erdogmus, Principe, IJCNN Tutorial on ITL, 2003.

# ITL - Applications





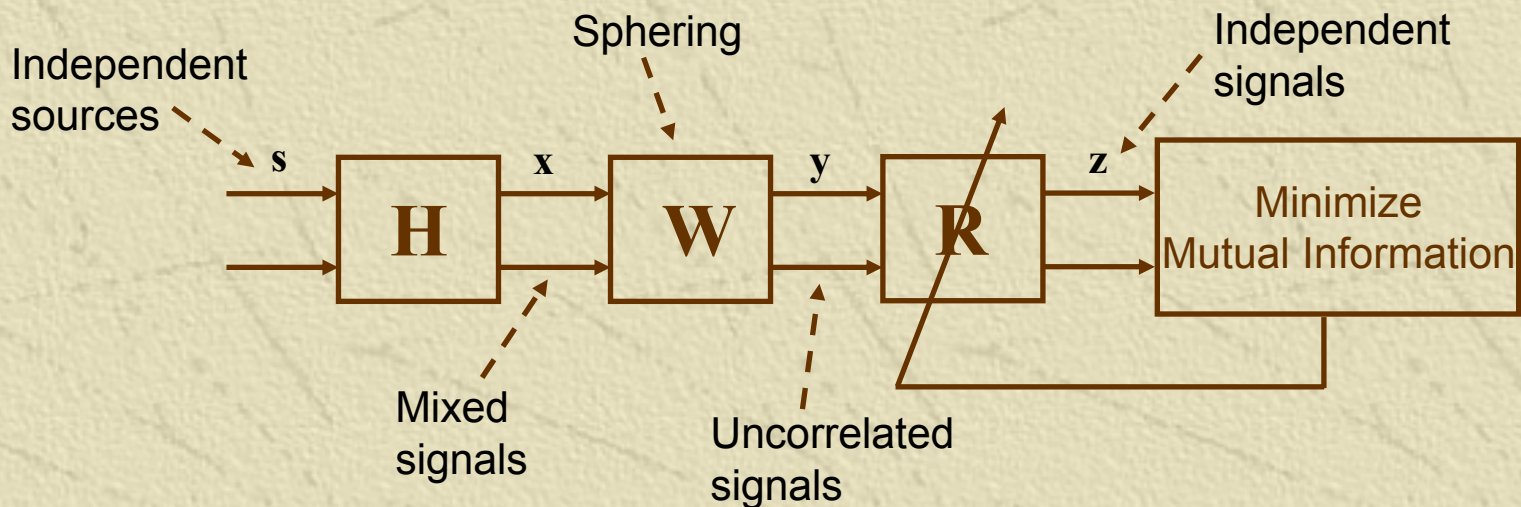
# ITL – Applications

## Independent component analysis

- ✪ Observations are generated by an unknown mixture of statistically independent unknown sources.

$$\mathbf{x}_k = \mathbf{H}\mathbf{s}_k$$

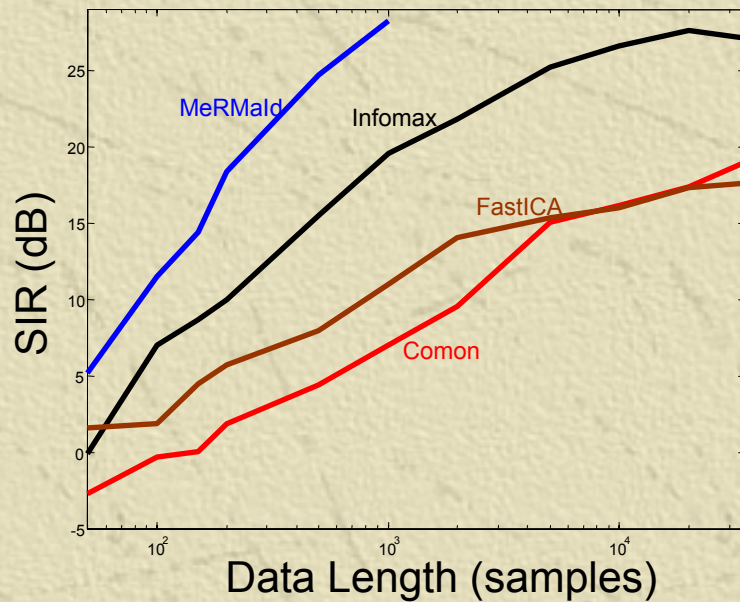
$$I(\mathbf{z}) = \sum_{c=1}^n H(z_c) - H(\mathbf{z})$$



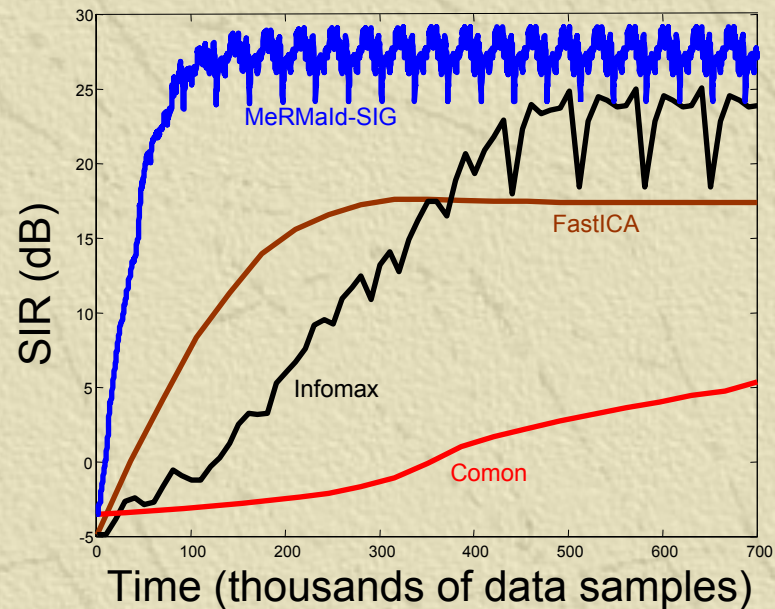
# ITL – Applications

## On-line separation of mixed sounds


Off-line separation, 10x10 mixture





On-line separation, 3x3 mixture





## Observed mixtures and separated outputs


X1: 

X2: 

X3: 

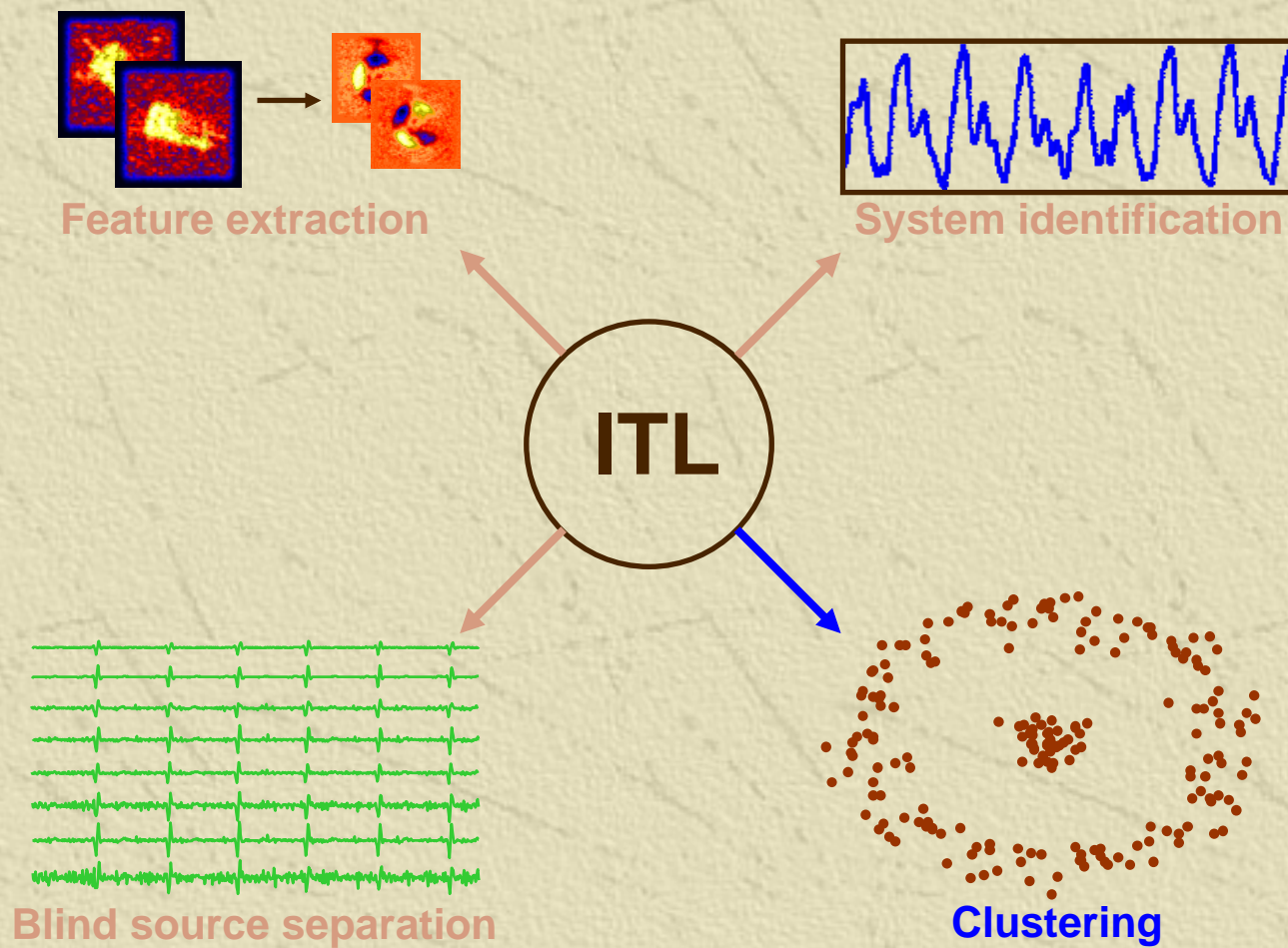
Z1: 

Z2: 

Z3: 



# ITL - Applications



# ITL – Applications

## Information theoretic clustering

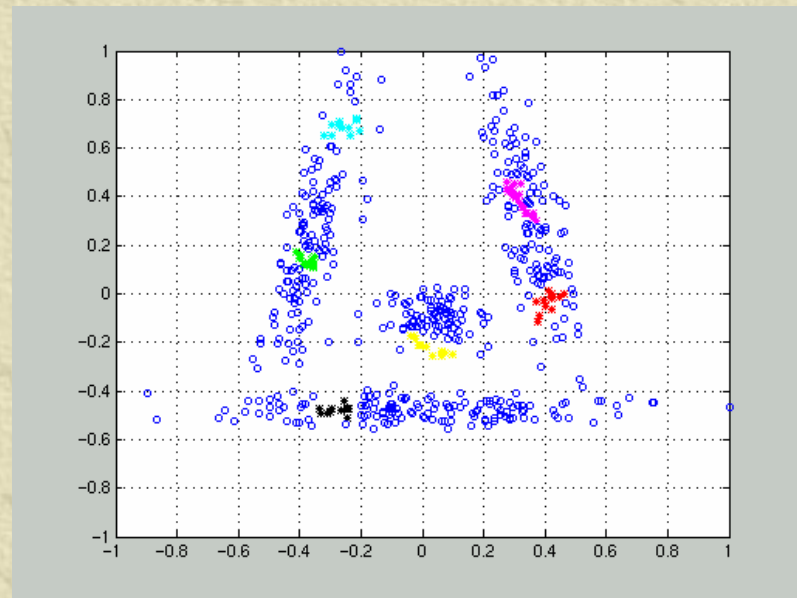
- ✦ Select clusters based on entropy and divergence:
  - ✦ Minimize within cluster entropy
  - ✦ Maximize between cluster divergence

Between cluster divergence

$$\min_{\mathbf{m}} \frac{\int p(x)q(x)dx}{\left( \int p^2(x)dx \int q^2(x)dx \right)^{1/2}}$$

Membership vector

Within cluster entropy



*Robert Jenssen PhD on information theoretic clustering*