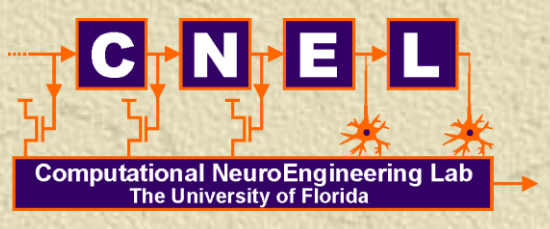
Machine Learning in Signal Processing

Jose C. Principe

Computational NeuroEngineering Laboratory (CNEL)
University of Florida
principe@cnel.ufl.edu
www.cnel.ufl.edu



Acknowledgments

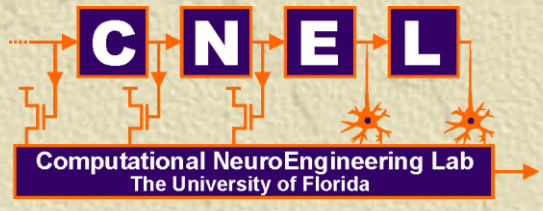
I would like to thank my 65 Ph.D. students and postdocs who helped create and sustain the Computational NeuroEngineering Laboratory research path.

And to the

Federal funding agencies for continued support

NSF ECS-0856441, IIS-0964197
ONR N00014-10-1-0375
DARPA N66001-10-C-2008

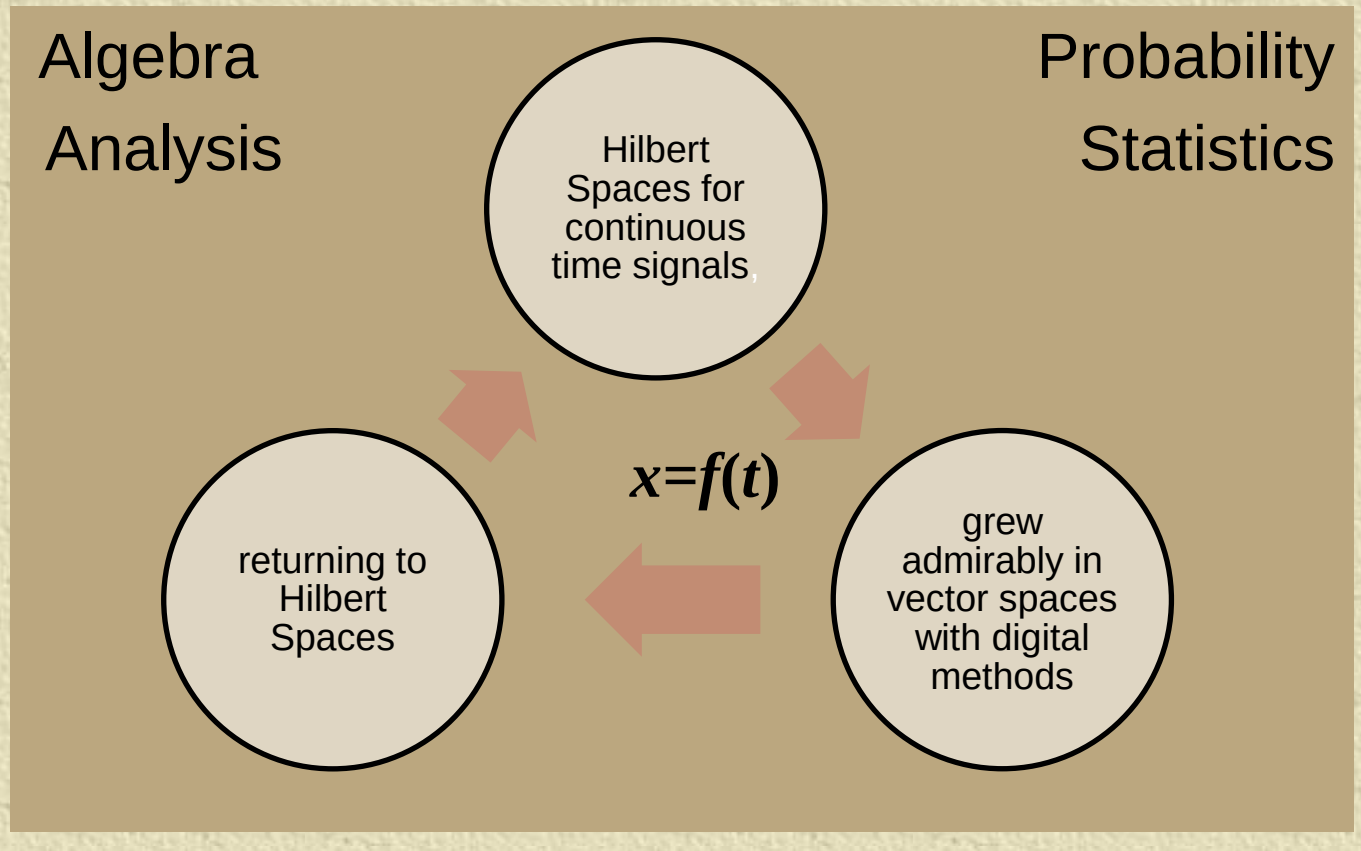
Outline



1. The Essence of the Signal Processing Framework
2. Synergisms with Machine Learning
3. Case Studies
    - Adaptive Filtering in RKHS
    - Information Theoretic Costs
    - Cognitive Memories

Signal Processing Field

- SP is applied mathematics and it helps information technologists **invent new realities**.



- SP foot print is enormous and with a remarkable balance between theory and applications.

Signal Processing Field
Tools

- SP tools are mostly based on the **linear model**, **Gaussian statistics** and **stationary assumptions**,

  While the world is nonstationary, non Gaussian and nonlinear

  Why have the linear model and the Gaussian assumptions been so effective approximations?

- "Why" questions require some philosophical answers!

Signal Processing Field
Tools

- Wigner, 1960 “Unreasonable Effectiveness of Mathematics in the Natural Sciences”

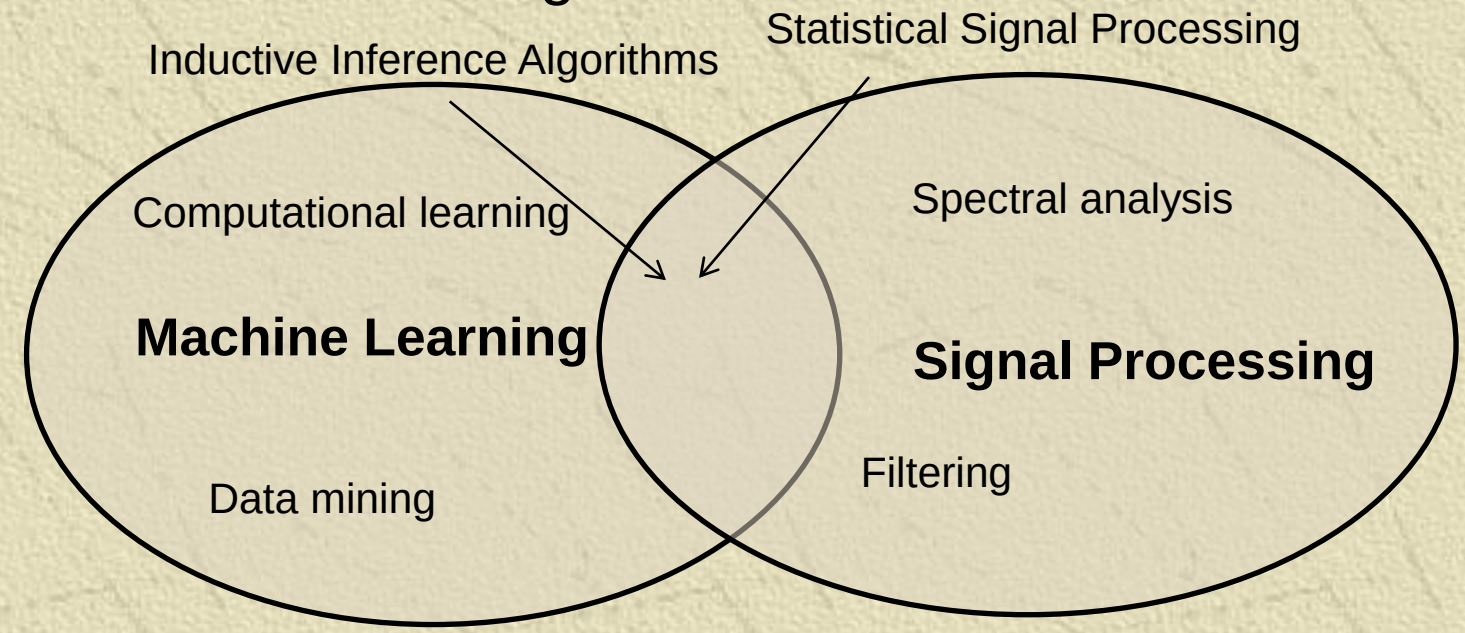
- Richard Hamming, 1980 (Naval Postgraduate School, Monterey) “Unreasonable Effectiveness of Mathematics” states
  - We see what we look for
  - We pick the mathematics we want
  - Science answers very few questions
  - The evolution of man provides the model

- I will add: SP had the luxury of building communications systems under the assumptions it mastered. But more recently SP has attacked the problem of speech and video that takes us away from our comfort zone.
- The complexities of the world demand new computational paradigms

Signal Processing Field

Statistical Signal Processing

- There is an obvious overlap between Signal Processing and Machine Learning



- *Tom Michell*: A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Signal Processing Field

Statistical Signal Processing

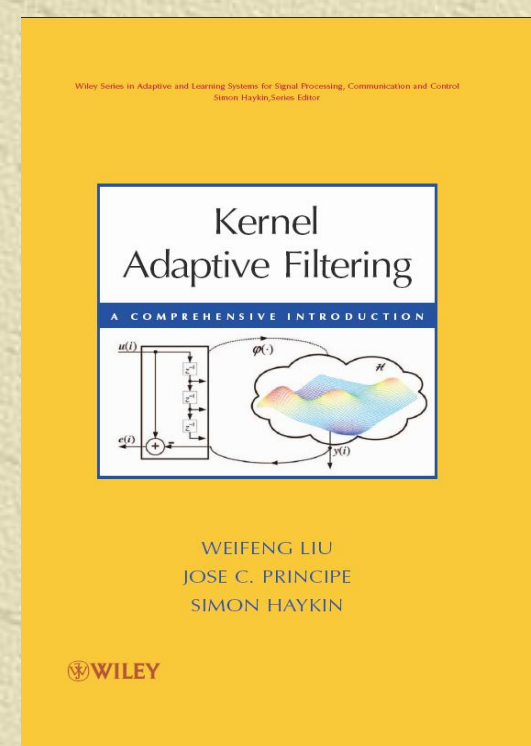
- Statistical Signal Processing (SSP) and Machine Learning (ML) share the need for another **unreasonable effectiveness: data** (Halevy et al, 2009). This makes them synergistically intertwined.

- Tools are the same (statistics either Bayesian or frequentist).
- SSP tends to address learning in time (non IID assumptions)
- Optimality conditions tend to be different in SSP and ML

- Major difference is how methods are applied.
- ML prefers to create **generative models** for the problem under study. Inference models and parameters are determined by data and their environments.
- Due to this ML often finds new solutions to complex problems (because of the heavy reliance on Bayesian Inference (BI)).

Case Study 1:

Integration of ML and SSP Extends the Linear Model for Adaptive Nonlinear Filters

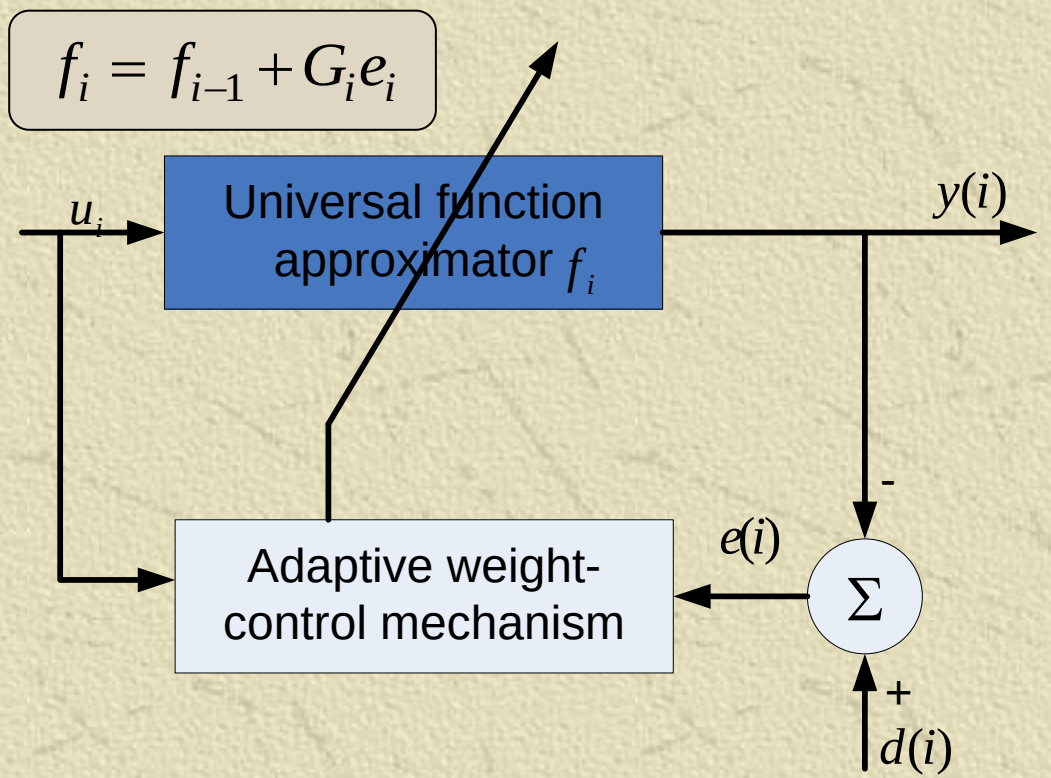


On-Line Learning for Non-Linear Filters?

- Can we generalize the LMS algorithm $w_i = w_{i-1} + \eta u_i e(i)$ to *nonlinear* models?

$$y = w^T u \longrightarrow y = f(u)$$

and create incrementally the nonlinear mapping?



Reproducing Kernel Hilbert Spaces

- The simplest kernel is a function of two arguments $\kappa : T \times T \longrightarrow R.$ symmetric $\kappa(x, y) = \kappa(y, x)$ positive definite $\sum_{m=1}^{N} \sum_{n=1}^{N} a_n a_m \kappa(x_n, y_m) \geq 0$ and shift invariant $\kappa(x, y) = \kappa(x + \theta, y + \theta) = \kappa(x - y)$ as the Gaussian or Laplacian

$$\kappa(x, y) = \exp(-h\|x - y\|^2)$$

- Positive definite kernel functions define a Reproducing Kernel Hilbert Space (RKHS). A RKHS is a special type of Hilbert Space with the reproducing property (kernel trick)

$$f(x) = < f, \kappa(., x) >_{H_\kappa}$$

- Operating with functions seems complicated and it is! But it becomes much easier in RKHS if we restrict the computation to inner products, as the most conventional operators in DSP.

$$y(n) = \sum_{i=0}^{L-1} w_i x(n - i) = \left\langle \mathbf{w}^{\mathrm{T}} \mathbf{x}(n) \right\rangle$$

Kernel Least-Mean-Square (KLMS)

- LMS algorithm $w_0 \quad e(i) = d(i) - w_{i-1}^T u_i \qquad w_i = w_{i-1} + \eta u_i e(i)$
- Select our kernel $\kappa(x, y) = <\varphi(x), \varphi(y)>_F$, e.g. a Gaussian
- Transform data into a high dimensional feature space *F* $\varphi_i := \varphi(u_i)$ and build a linear model $y = \langle \Omega, \varphi(u) \rangle_F$ in the following way:

$$\Omega_0 = 0$$

$$e(i) = d(i) - \langle \Omega_{i-1}, \varphi(u_i) \rangle_F$$

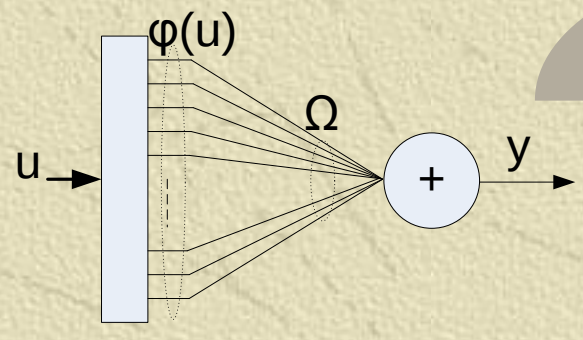
$$\Omega_i = \Omega_{i-1} + \eta \varphi(u_i) e(i)$$

$$\Omega_i = \sum_{j=1}^{i} \eta e(j) \varphi(u_j)$$

$$f_i(u) = \langle \Omega_i, \varphi(u) \rangle_F = \sum_{j=1}^{i} \eta e(j) \kappa(u, u_j)$$

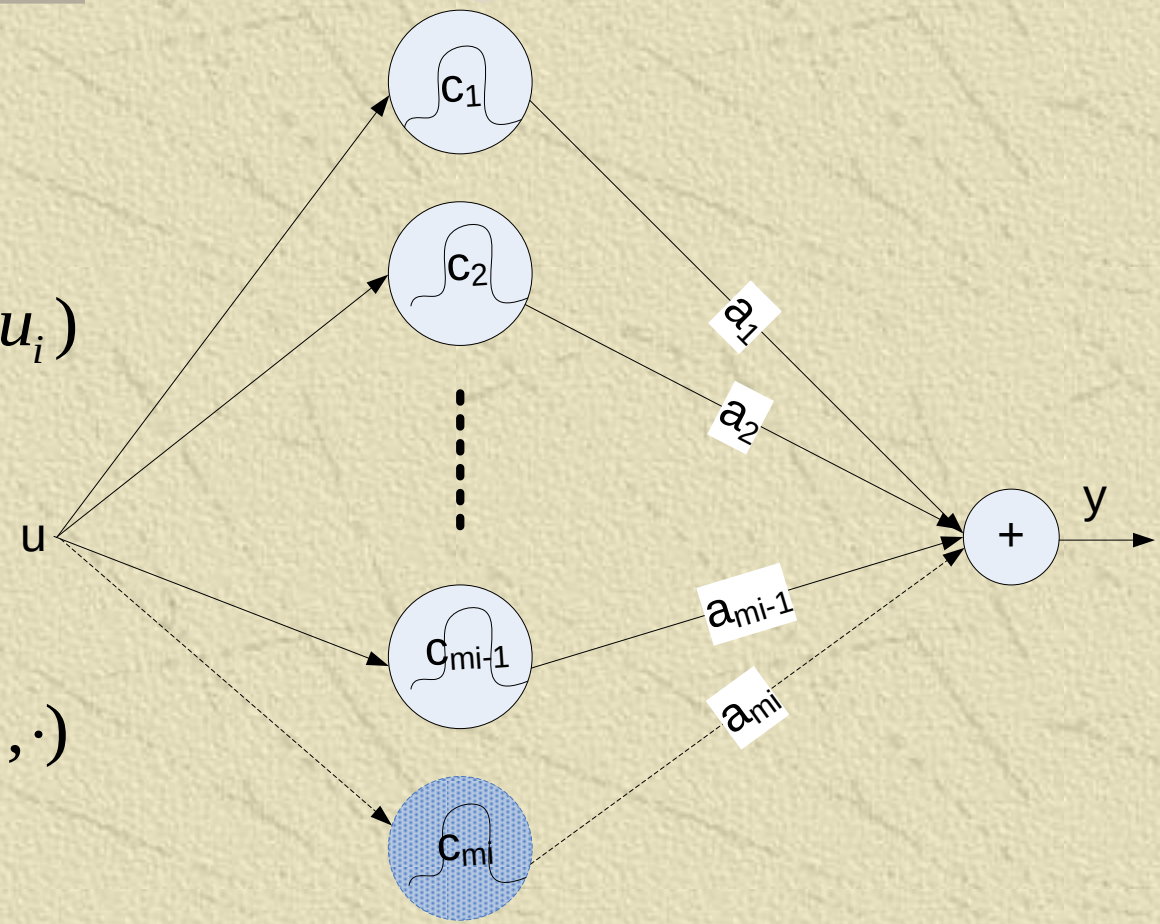
- RBF Centers are the samples, and weights are the errors!
- KLMS is model free (uses the basis of the space)

Growing Network Structure



$$\Omega_i = \Omega_{i-1} + \eta e(i)\varphi(u_i)$$

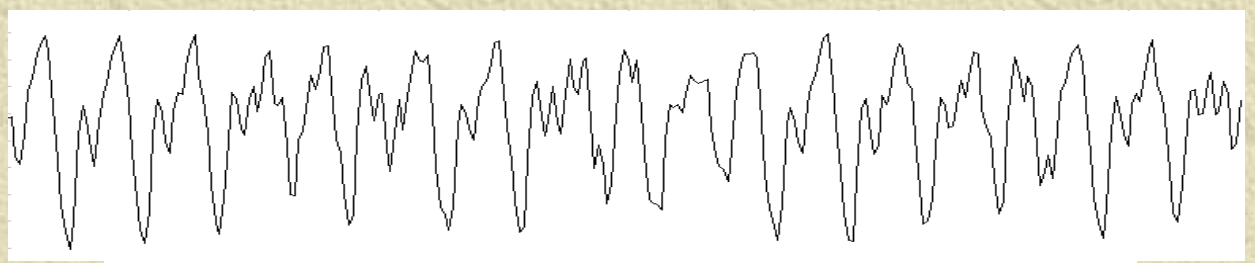
$$f_i = f_{i-1} + \eta e(i)\kappa(u_i, \cdot)$$

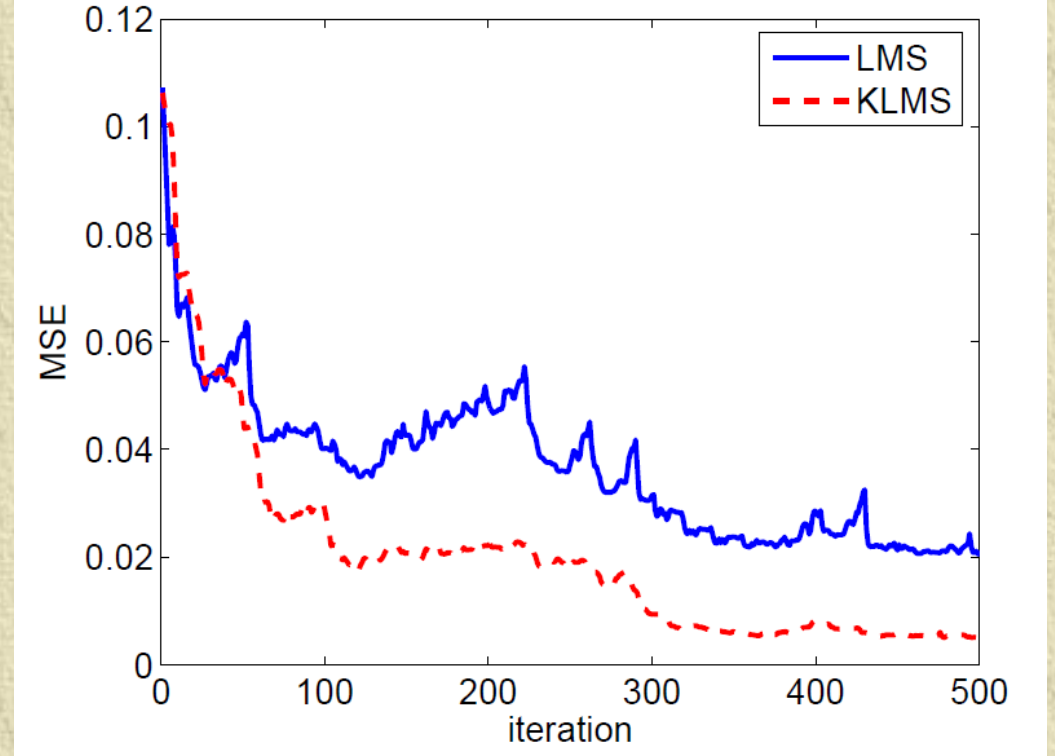


Weight function is incrementally implementing the representer in RKHS of the desired input output mapping.

KLMS- Mackey-Glass Prediction

$$\dot{x}(t) = -0.1x(t) + \frac{0.2x(t-\tau)}{1+x(t-\tau)^{10}} \qquad \tau = 30$$





LMS

η=0.2

KLMS

h=1, η=0.2

Nonlinear Channel Equalization

Algorithms	Linear LMS (η=0.005)	KLMS (η=0.1) (NO REGULARIZATION)	RN (REGULARIZED λ=1)
BER (σ = .1)	0.162±0.014	**0.020±0.012**	0.008±0.001
BER (σ = .4)	0.177±0.012	**0.058±0.008**	0.046±0.003
BER (σ = .8)	0.218±0.012	**0.130±0.010**	0.118±0.004

$s_t \rightarrow z_t = s_t + 0.5 s_{t-1} \rightarrow r_t = z_t - 0.9 z_t^2 + n_\sigma \rightarrow r_t$

$$\kappa(u_i, u_j) = \exp(-0.1 \| u_i - u_j \|^2)$$

Algorithms	Linear LMS	KLMS	RN
Computation (training)	O(l)	O(i)	$O(i^3)$
Memory (training)	O(l)	O(i)	$O(i^2)$
Computation (test)	O(l)	O(i)	O(i)
Memory (test)	O(l)	O(i)	O(i)

SSP

ML

The KLMS solution is well posed in the sense of Hadamard, and does not need to be explicitly regularized. So robustness of LMS carries over.

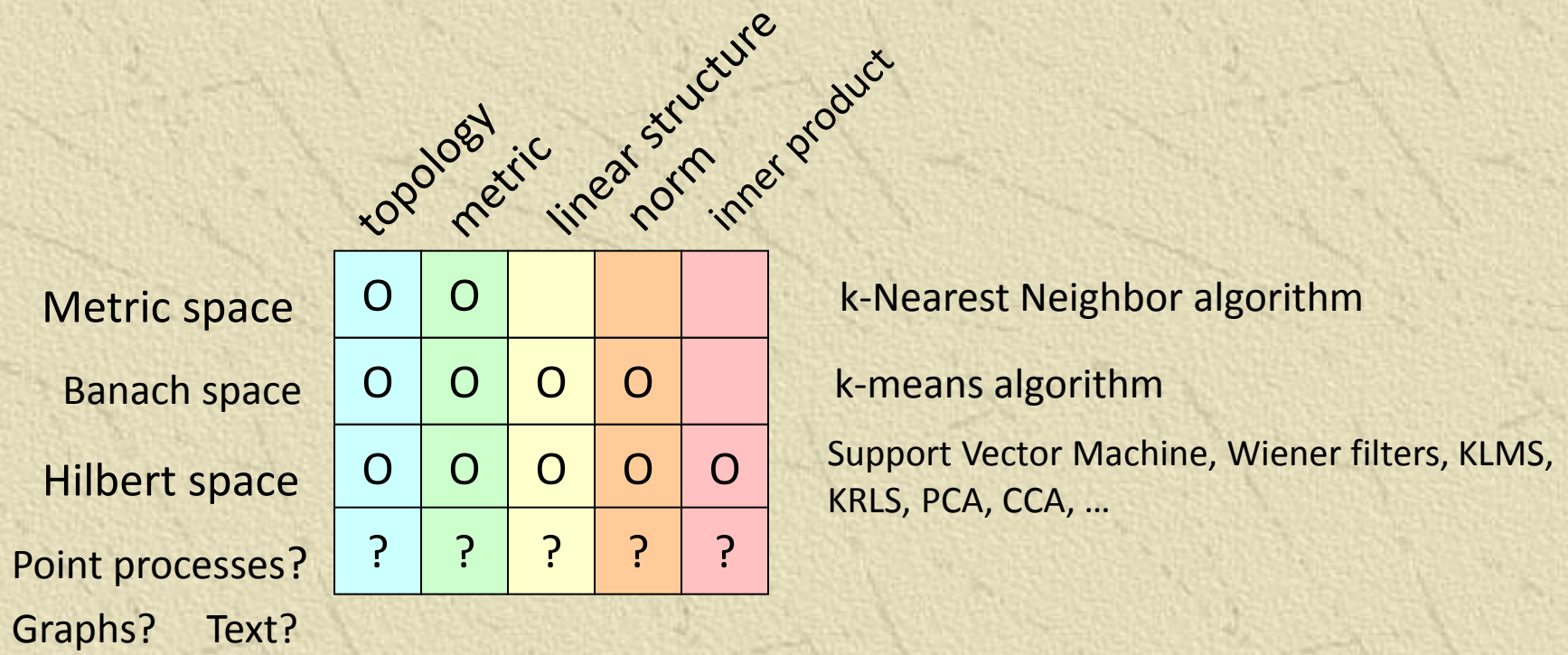
Neural Networks versus Kernel Filters

	ANNs	Kernel filters
Universal Approximators	YES	YES
Convex Optimization	NO	**YES**
Model Topology grows with data	NO	**YES**
Require Explicit Regularization	NO	YES/**NO** (KLMS)
Online Learning	YES	YES
Computational Complexity	LOW	MEDIUM

ANNs are semi-parametric, nonlinear approximators

Kernel filters are non-parametric, nonlinear approximators

Kernels Extend Signal Processing to Non Numeric Data



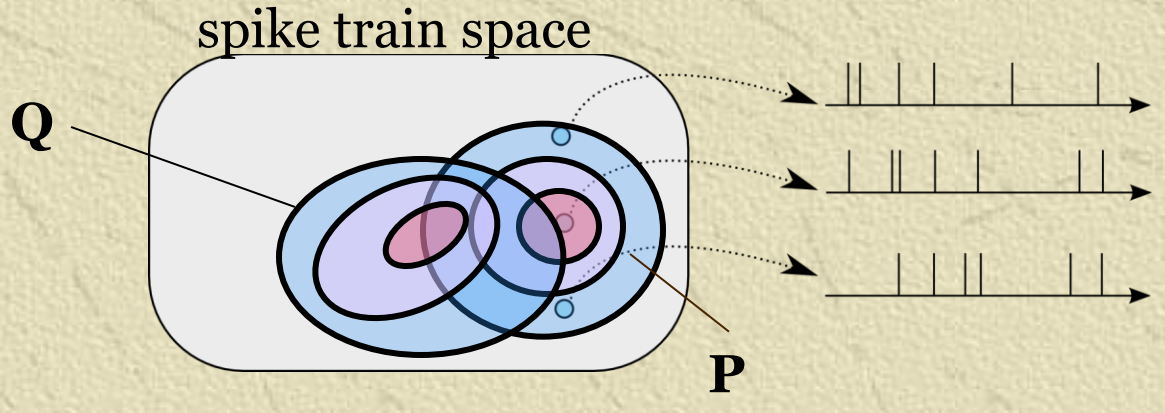
The idea is to find (injective) mappings to the Hilbert space using kernelization.

We will exemplify the technique for neural spike trains, which are point processes.

Definition of Neural Spike Trains

Point process describes stochastically a sequence of events occurring in time

A neural spike train is **a realization** of a point process



- The probability measure over the spike train space defines a point process
- The conditional intensity function fully describes the PP

$$\lambda(t \mid H_t) = \lim_{\Delta t \to 0} \frac{\Pr\{\text{event in}[t, t+\Delta t) \mid H_t\}}{\Delta t}$$

- The Poisson point process is memoryless $\lambda(t|H_t)=\lambda(t)$.

Neural Spike Trains

RKHS for spikes with cross-intensity kernels

- Given two point processes p_i, p_j, define the inner product between their intensity functions

$$I(p_i, p_j) = \left\langle \lambda_{p_i}(t \mid H_t^i), \lambda_{p_j}(t \mid H_t^j) \right\rangle_{L_2(T)} == E[\int_T \lambda_{p_i}(t \mid H_t^i)\lambda_{p_j}(t \mid H_t^j)dt]$$

- This yields a family of cross-intensity (CI) kernels, in terms of the model imposed on the point process history, H_t. For Poisson Point Processes
  - Memoryless CI (mCI) kernel:

$$I(p_i, p_j) = \int_T \lambda_{p_i}(t)\lambda_{p_j}(t)dt$$

$$\hat{I}(p_i, p_j) = \sum_{m=1}^{N_i}\sum_{n=1}^{N_j} \kappa(t_m^i - t_n^j)$$

(κ Laplacian)

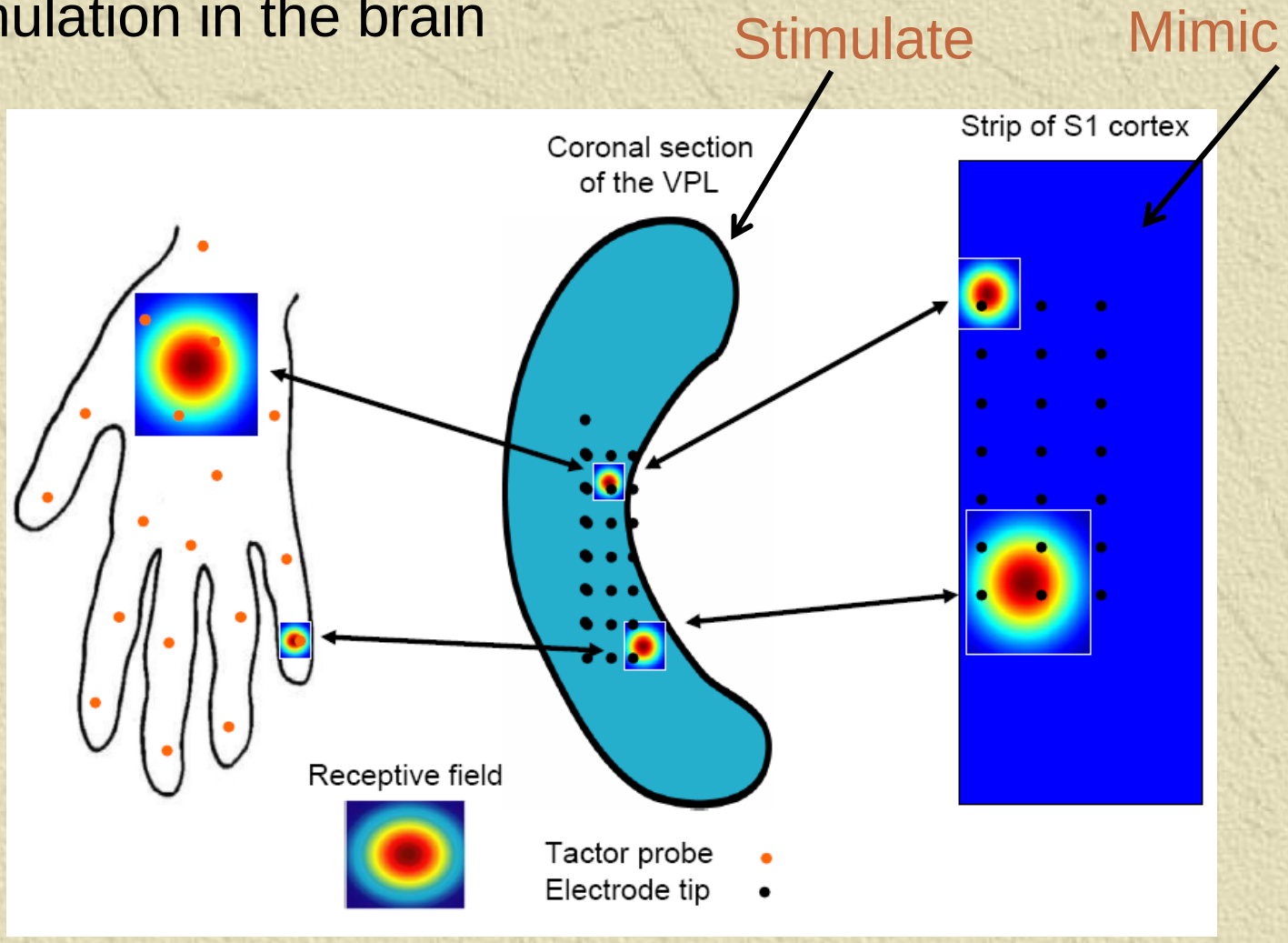
  - Nonlinear cross-intensity (nCI) kernel:

$$I_\sigma^*(p_i, p_j) = \int_T \kappa_\sigma(\lambda_{p_i}(t), \lambda_{p_j}(t))dt$$

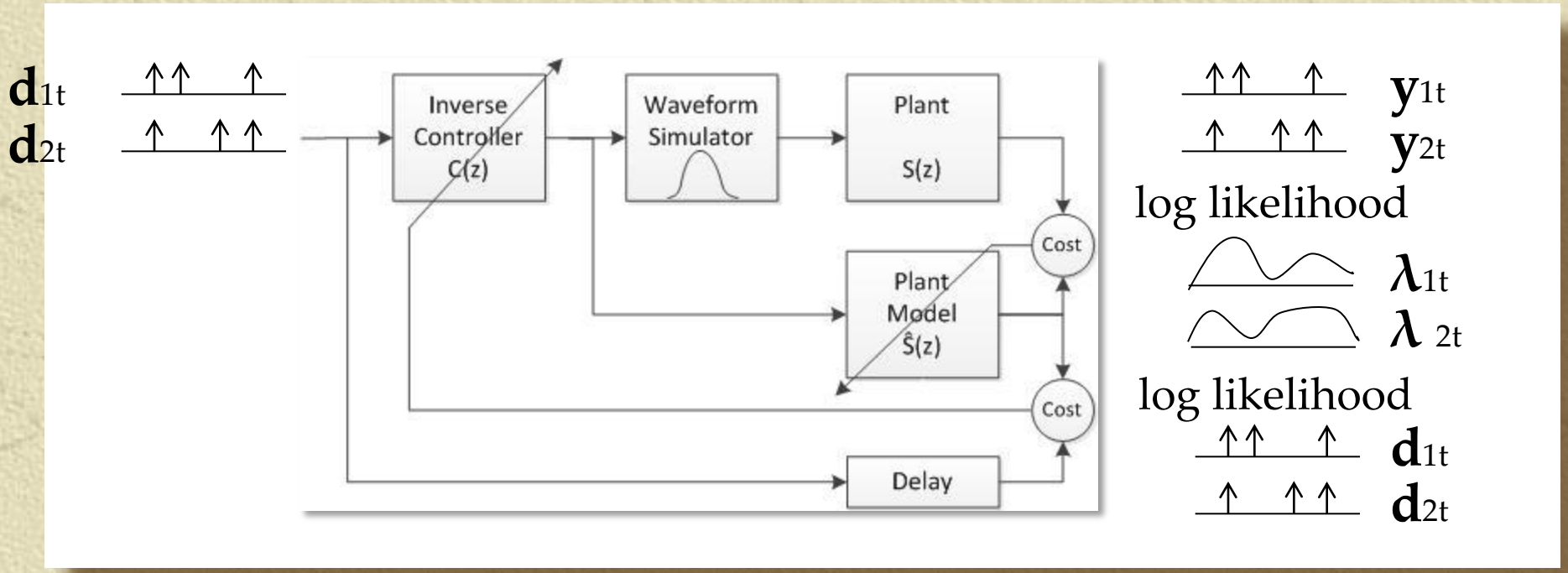
Paiva, et al Neural Computation, 2008

Somatosensory Motor Stimulation

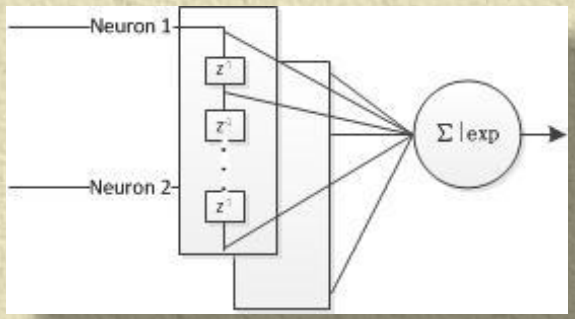
How to mimic the perception of "touch" by electrical stimulation in the brain



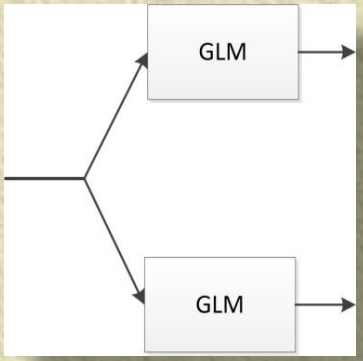
Inverse Adaptive Control for Sensory Motor Stimulation



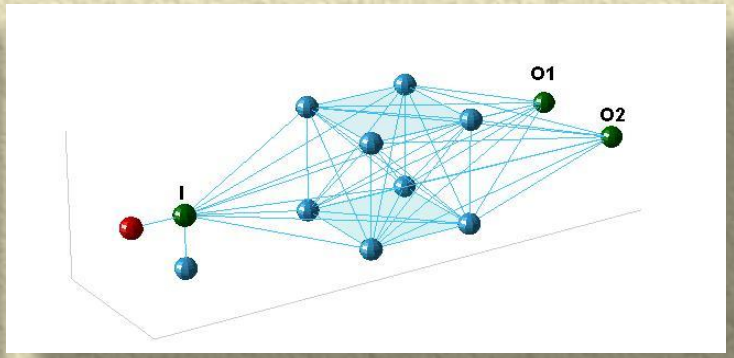
Inverse controller C(z)



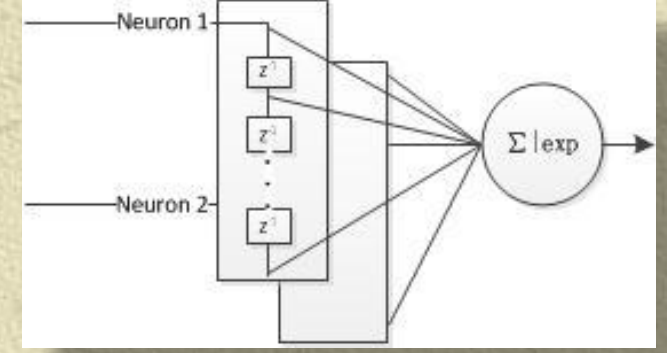
Plant model $\hat{S}(z)$



Plant S(z)



Inverse Controller



- Receives spikes as inputs and produces a continuous output.
- Create a linear controller in a RKHS adapted by the Kernel LMS algorithm.
- But now the kernel is the nCI kernel (strictly positive definite).

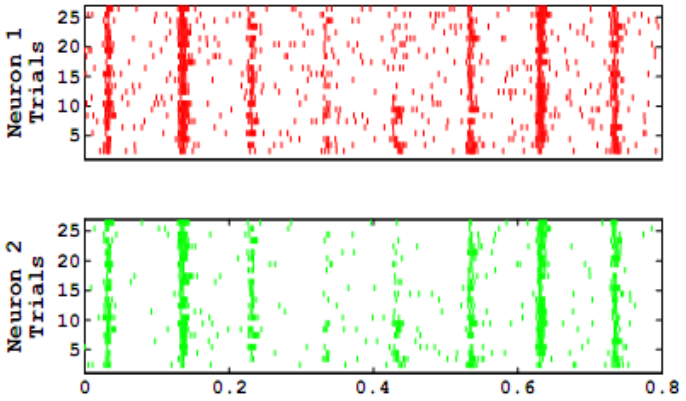
$$\kappa_\sigma(s_i, s_j) = \exp\left(\frac{< \lambda_{s_i} - \lambda_{s_j} >^2}{\sigma^2}\right) \quad \kappa_\sigma(\mathbf{s}_i, \mathbf{s}_j) = \prod_{k=1}^{K} \exp\left(\frac{< \lambda_{s_i^k} - \lambda_{s_j^k} >^2}{\sigma^2}\right)$$

- Back propagate the cost through the plant model.

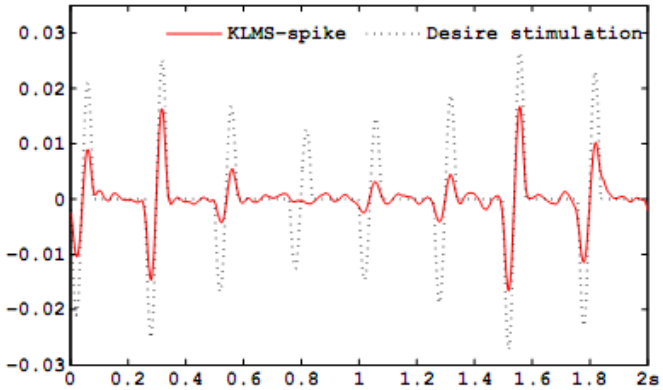
Sensory Motor Stimulation

Results (synthetic data)

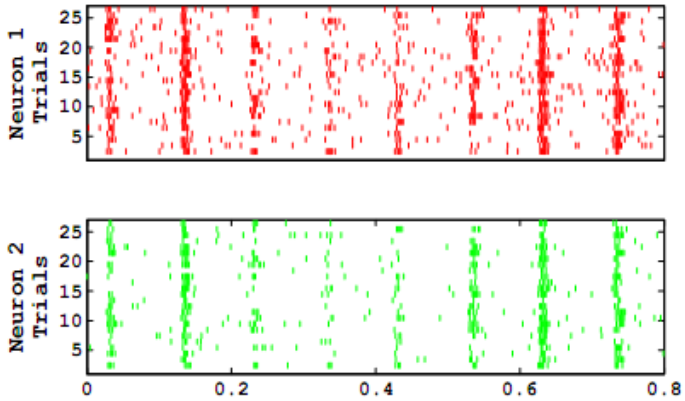
Microstimulation produced by the model



(a) The raster plot of the neural response recorded from two output neurons, which is produced by the desired stimulation.



(b) The reconstructed stimulation



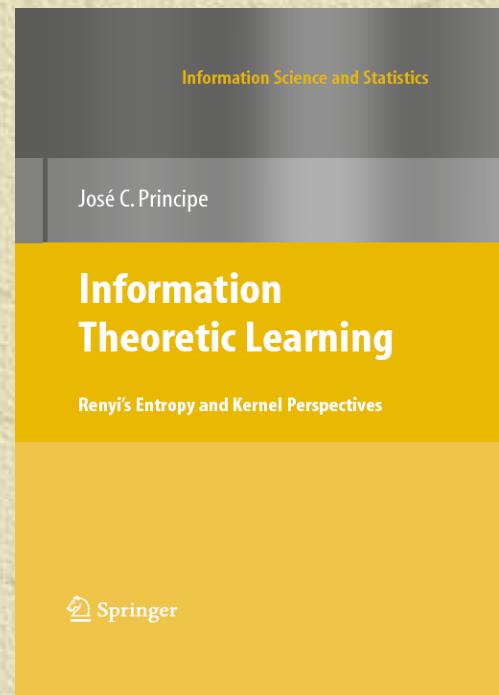
(c) The raster plot of the neural response recorded from two output neurons, which is produced by the reconstructed stimulation.

What have we accomplished?

- We showed the power of kernel methods: define a kernel for your purpose (Gaussian or mCI) and keep the SSP model the same!
- Showed how RKHS can extend DSP tools for more abstract objects like point processes.
- Used KLMS to implement an universal mapper, online, sample by sample, with complexity O(n).
- Optimization is convex on the parameters
- It does not need explicit regularization
- Solution grows with the number of data samples, so needs to be pruned in practice

Case Study 2:

Integration of Information Theory and SSP goes Beyond 2nd Order Statistics: Robust Similarity Functions



Correntropy:
A new generalized similarity measure

Definition: Cross-correntropy between two arbitrary scalar random variables X and Y is defined by

$$v(X,Y) = E_{XY}[\kappa(X,Y)] = \int\int \kappa(x,y)\, p_{X,Y}(x,y)\, dxdy$$

where $\kappa(.,.)$ is any continuous positive definite kernel.

When $\kappa(.,.) = xy$, we obtain cross-correlation.

When $\kappa(.,.)$ is a shift invariant kernel (e.g. Gaussian) we have

$$v_\sigma(X,Y) = E_{XY}[G_\sigma(X-Y)] = \int\int G_\sigma(x-y)\, p_{X,Y}(x,y)\, dxdy$$

That can be estimated as

$$\hat{v}_{\sigma,N}(X,Y) = \frac{1}{N}\sum_{i=1}^{N} G_\sigma(x_i - y_i)$$

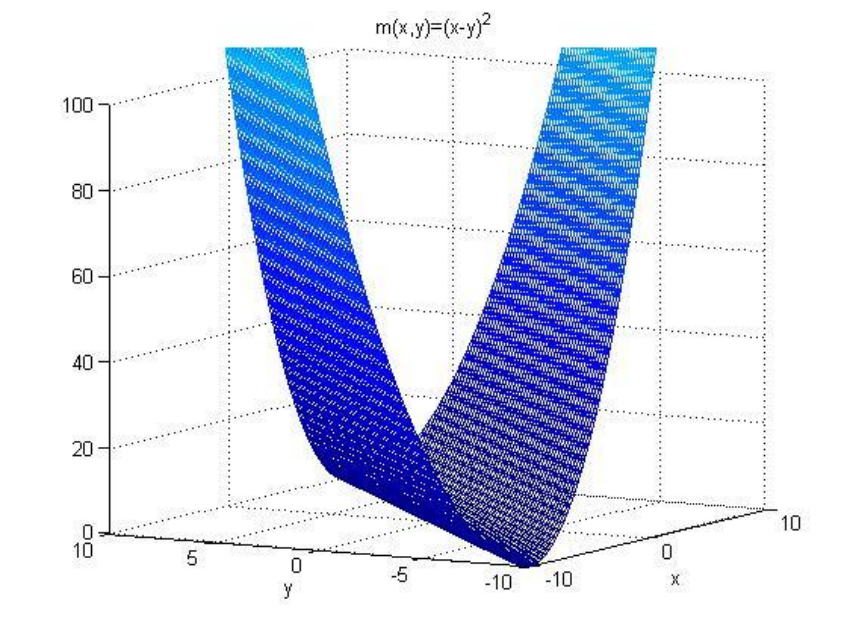
What are the properties of this new similarity function?

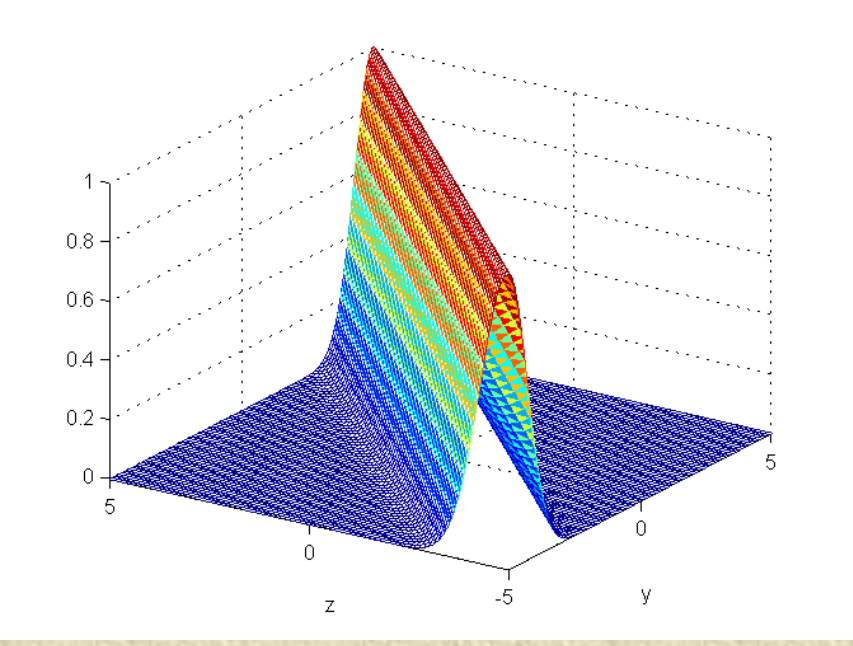
Correntropy:
A new generalized similarity measure

- Correntropy as a cost function versus MSE.

$$MSE(X,Y) = E[(X-Y)^2]$$
$$= \iint_{x,y} (x-y)^2 f_{XY}(x,y)dxdy$$
$$= \int_{e} e^2 f_E(e)de$$

$$V(X,Y) = E[k(X-Y)]$$
$$= \iint_{x,y} k(x-y) f_{XY}(x,y)dxdy$$
$$= \int_{e} k(e) f_E(e)de$$



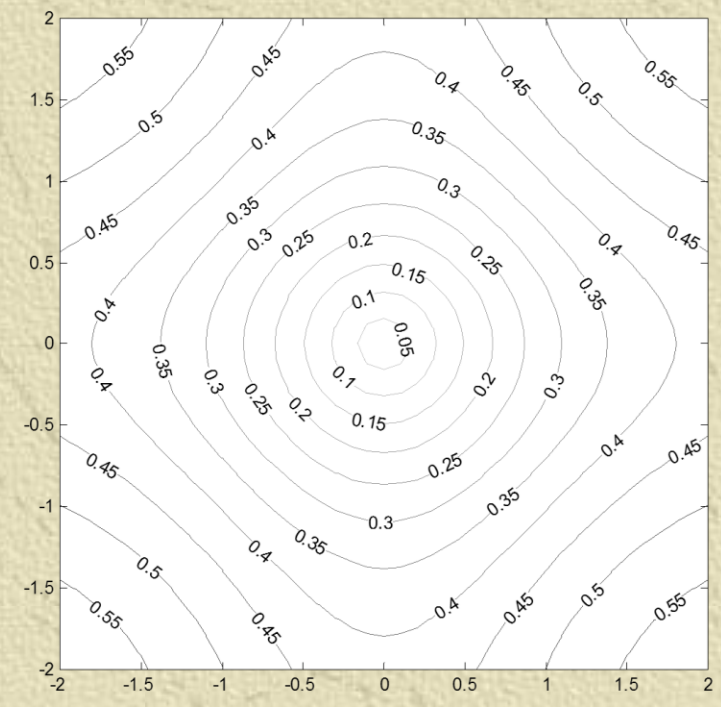


Correntropy:
A new generalized similarity measure

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

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

- Kernel size controls the metric scale
- Therefore correntropy can be used as an alternative similarity criterion in the space of samples.



Liu W., Pokharel P., Principe J., "Correntropy: Properties and Applications in Non Gaussian Signal Processing", IEEE Trans. Sig. Proc, vol 55; # 11, pages 5286-5298

Correntropy:
A new generalized similarity measure

- When we use correntropy criterion for regression we are implementing Huber's **M estimation** (robust statistics). When applied to the weighted least square problem gives

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

when

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

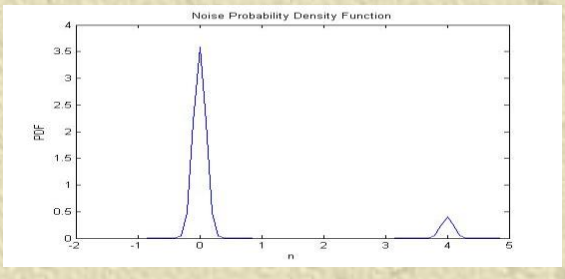
this leads to maximizing the correntropy of the error at the origin.

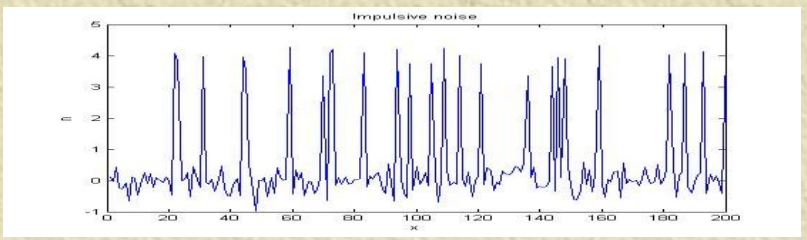
$$\min_{\theta} \sum_{i=1}^{N} \rho(e_i) = \min_{\theta} \sum_{i=1}^{N} (1 - \exp(-e_i^{\,2}/2\sigma^2))/\sqrt{2\pi}\sigma$$

$$\Leftrightarrow \max_{\theta} \sum_{i=1}^{N} \exp(-e_i^{\,2}/2\sigma^2)/\sqrt{2\pi}\sigma = \max_{\theta} \sum_{i=1}^{N} \kappa_{\sigma}(e_i)$$

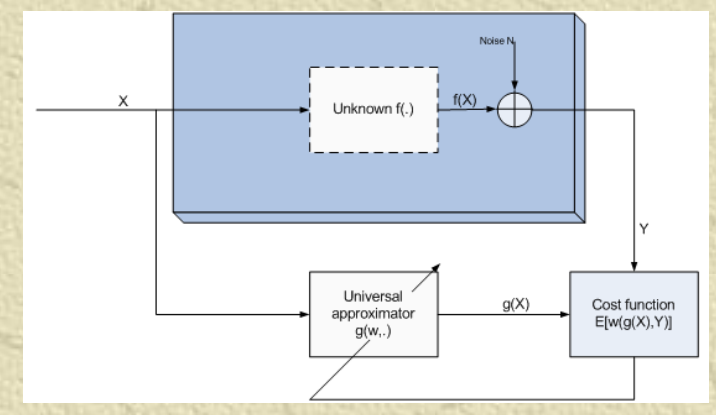
Correntropy:
Nonlinear Filtering with Outliers

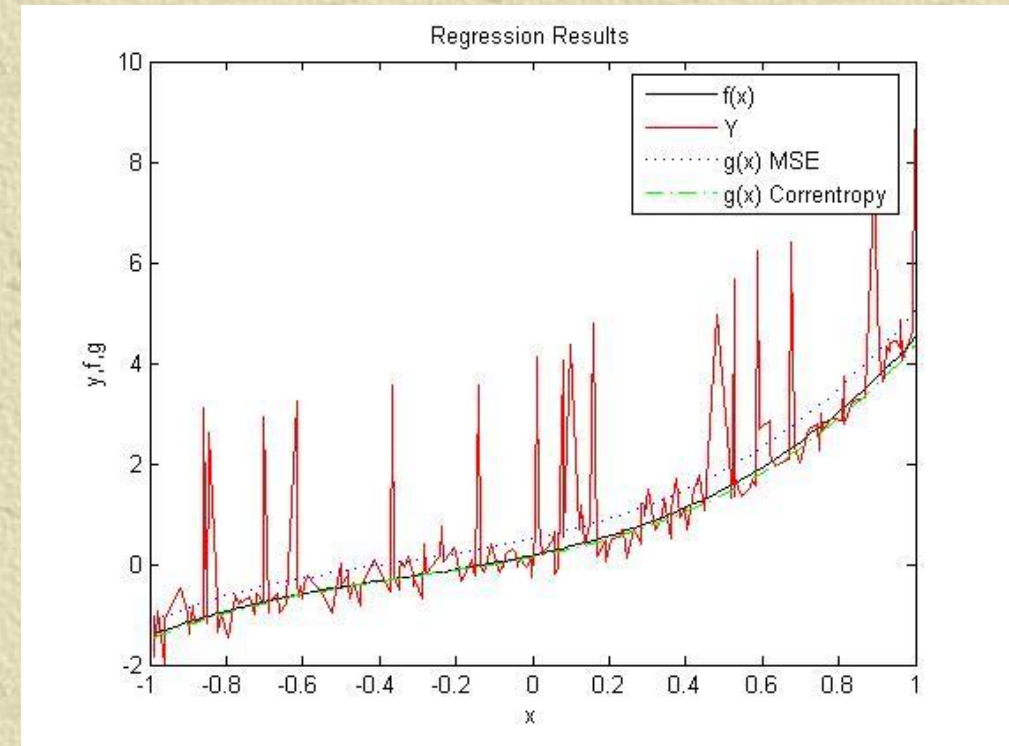
- Middleton noise model





Neural Network approximator





Auto-Correntropy Function

Definition: The auto-correntropy of a stationary random process $\{x_t\}$ is

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

The name correntropy stems from the fact that the mean of V(.) over the lags (or the dimensions) is the argument of the log of Renyi's quadratic entropy.

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int_X f^\alpha(x) dx \qquad \alpha = 2$$

For strictly stationary and ergodic r. p.

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

Santamaria I., Pokharel P., Principe J., "Generalized Correlation Function: Definition, Properties and Application to Blind Equalization", IEEE Trans. Signal Proc. vol 54, no 6, pp 2187- 2186, 2006.

Auto-Correntropy Function
Properties

- It has a maximum at the origin ($1/\sqrt{2\pi}\sigma$)
- It is a symmetric positive function (hence defines RKHS)
- Its mean value is the argument of the log of Renyi's quadratic entropy

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\int_X f^\alpha(x)dx \qquad \alpha = 2$$

- For the Gaussian kernel, correntropy includes second and higher order moments of the r.v.

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

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

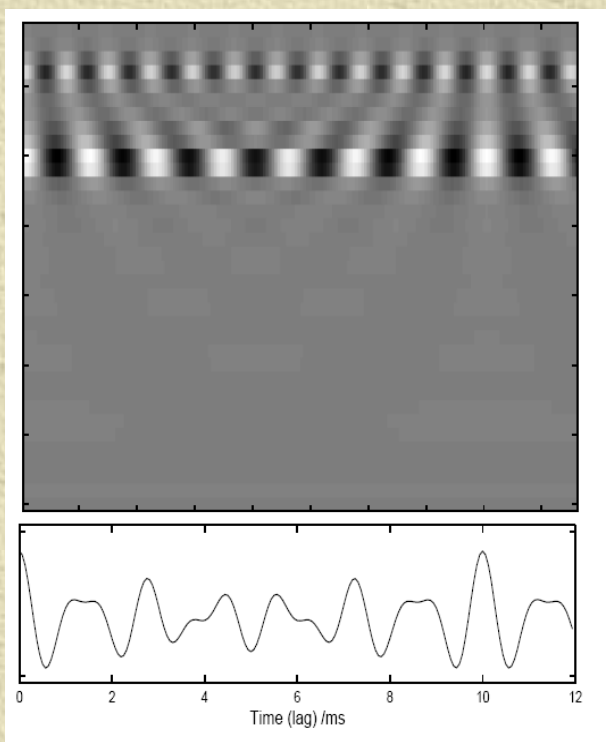
Applications of Auto-Correntropy

Correntropy based correlograms

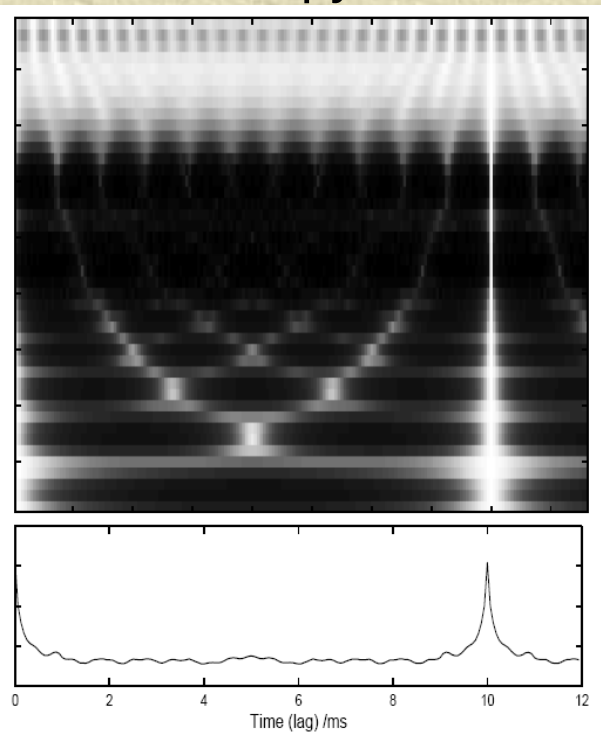
Correntropy can be used in Computational Auditory Scene Analysis (CASA), providing high resolution for pitch estimation.

Figures show the correlogram from a 64 channel cochlea model for one synthetic vowel “a” (pitch=100Hz).

Auto-correlation Function



Auto-correntropy Function



Beyond 2nd Moment Spectral Representations
Correntropy Spectral Density

Definition: The Fourier transform of the centered auto-correntropy function of a strictly stationary process x(t) is called the correntropy spectral density (CSD) and is defined by

$$P_\sigma(\omega) = \int_{-\infty}^{\infty} u(\tau) e^{-j\omega\tau} d\tau.$$

u(τ) is the centered auto-correntropy and ω is the frequency in radians. u(0) is the generalized variance (projected power in the RKHS).

$$u(0) = \mathrm{E}_{x_t x_t}[G_\sigma(x_t - x_t)] - \mathrm{E}_{x_t}\mathrm{E}_{x_t}[G_\sigma(x_t - x_t)] = \mathrm{E}[\| \Phi(x_t) \|^2] - \| \mathrm{E}[\Phi(x_t)] \|^2$$

The CSD **contains a sum of higher even moments of the r.v.**

We have implemented a taper approach to eliminate the kernel size and defined the weighted taper CSD as the

$$\hat{P}_{taper}(\omega) = \sum_{l=1}^{L-1} \frac{1}{\sqrt{2\pi\sigma_l}} \frac{\sum_{k=0}^{K-1} \lambda_k | \hat{P}_{\sigma_l}(\omega) |^2}{\sum_{k=0}^{K-1} \lambda_k}$$

Correntropy Function

Detection of sinewave in α stable noise

PSD: power spectral density

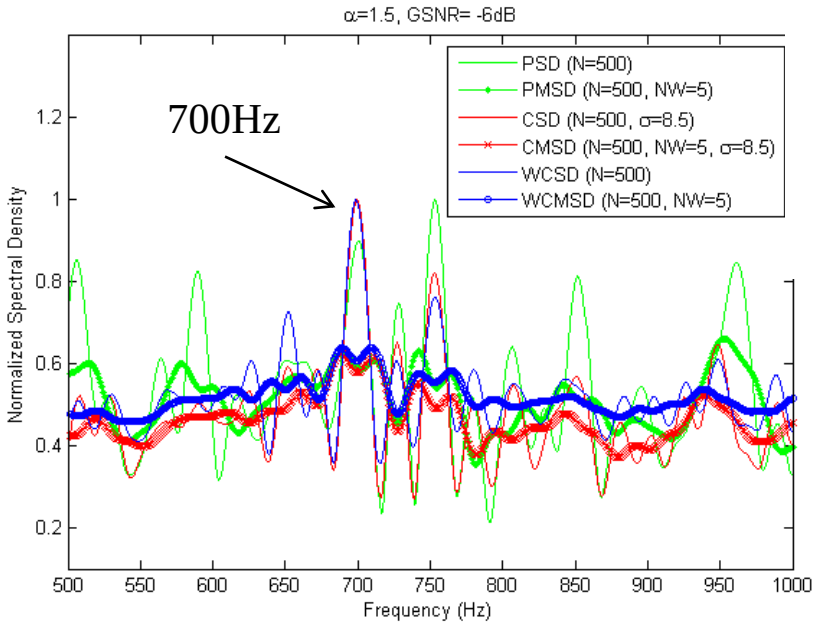
CSD: correntropy spectral density

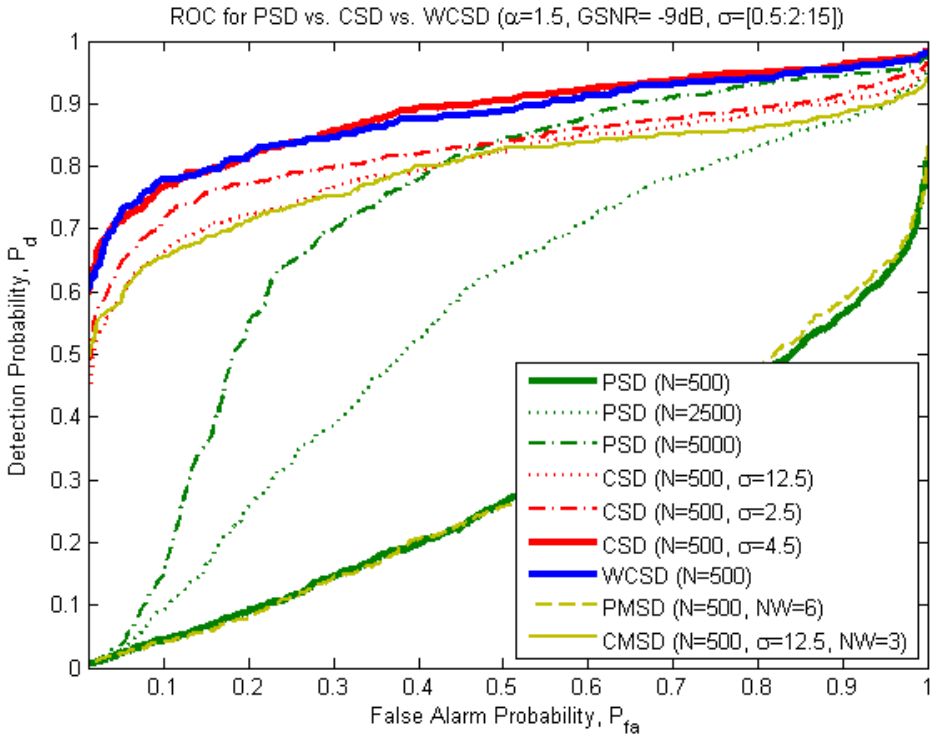
WCSD: weighted correntropy spectral d

PMSD: power multitapered spectral den

CMSD: correntropy-based multitaper sp

WCMSD: weighted correntropy-based r





Sinewave at 700 Hz

in α=1.5 noise

GSNR=-9dB

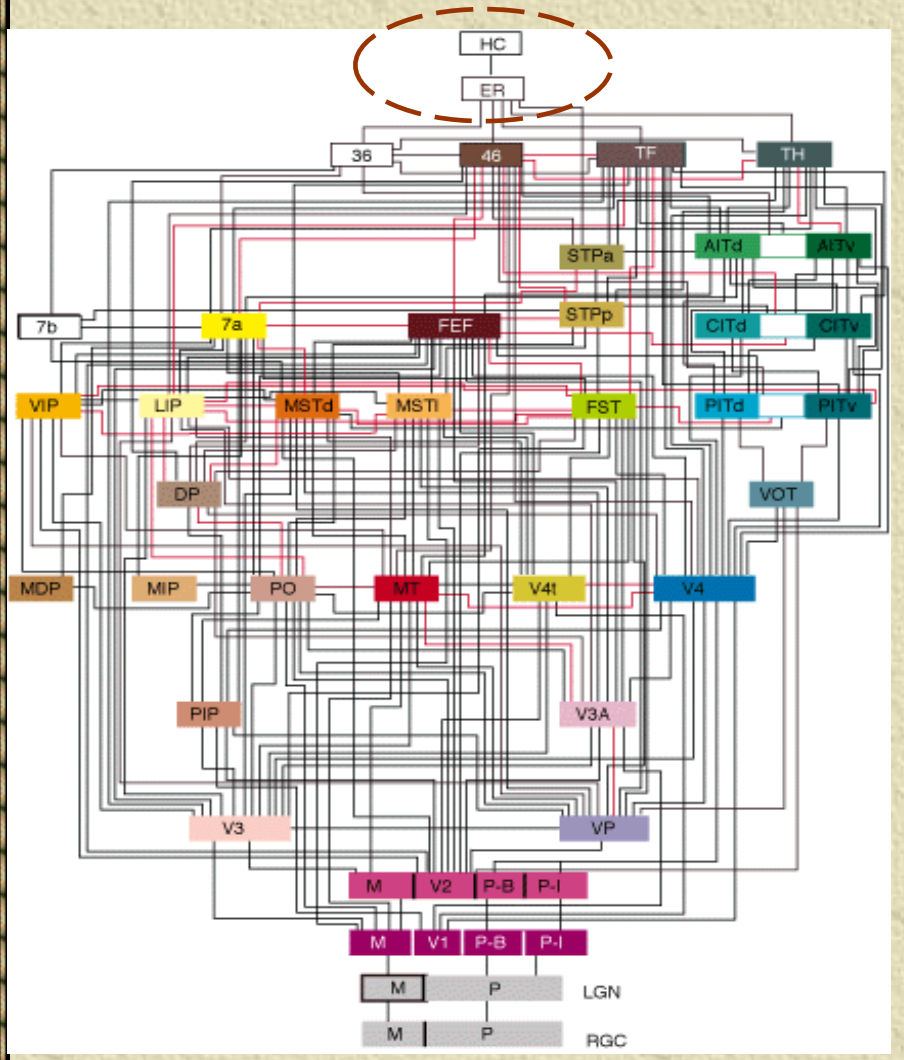
What have we accomplished?

- Correntropy creates a metric space where the conventional constant Minkowski metrics are substituted with a metric that is a function of distance and includes a sum of higher order moments of the data.
- This is very useful for outlier rejection in adaptive filtering and is equivalent to M estimation.
- Correntropy function is an alternative for correlation in statistical signal processing (also model based).
- A correntropy spectral density can be defined and goes beyond second order spectral representations and seems appropriate for non Gaussian noise backgrounds.

Case Study 3,

How Neuroscience can Inspire new Paradigms for Cognitive Information Processing

Cognitive Memory Embodiment
Hierarchical Organization and Distributed Nature

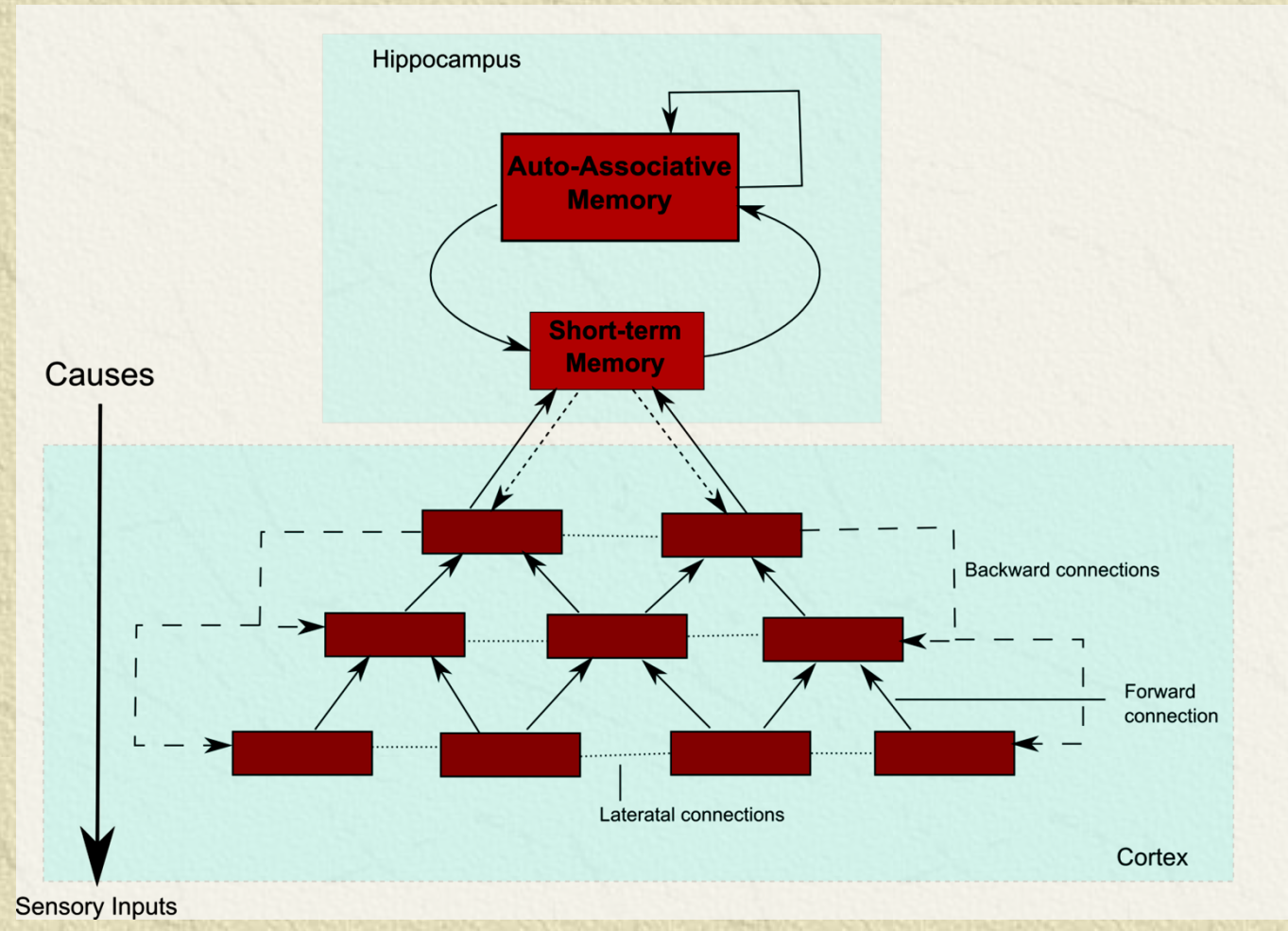


- Circuit diagram of visual system[1].
- Different layers are organized in a hierarchy with both forward and backward connections.
- Memory (Hippocampus) uses the rest of the brain for its functionality. Can not dissociate memory from visual processing.
- Storage and recall can be modeled as hierarchical brain networks.

[1] Felleman DJ, Van Essen DC. Distributed hierarchical processing in the primate cerebral cortex. Cereb Cortex. 1991 Jan-Feb;1(1):1-47.

Cognitive Memory Functional Principles
Generative Model for Learning and Memory

- Visual Memory is modeled as an inference machine where the hidden causes try to predict the sensory inputs.



Cognitive Memory Functional Principles
Hierarchical State Space Models with Unknown Inputs

HSSM represents the data and the state of the environment by considering both hidden states and causal states.

Dynamic hidden states quantify the history and the internal state of the system, while the causal states determine the “inputs” driving the system .

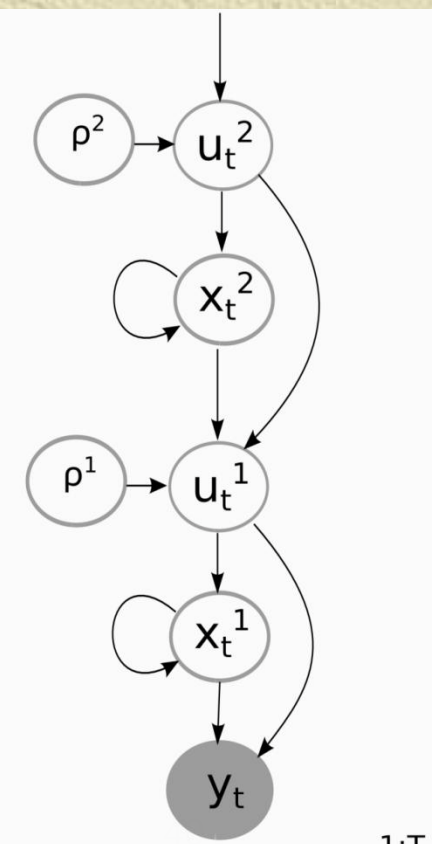
Empirical Bayesian priors create a hierarchical model, the layer on the top predict the causes for layer below.

Linear Dynamical Systems

y_t – Observations
x_t – Hidden states
u_t – Causal states

$$y_t = Cx_t + Du_t + w_t; \quad w \sim N(0, R)$$
$$x_t = Ax_{t-1} + Bu_t + v_t; \quad v \sim N(0, I)$$

But model has to be sparse



Karl Friston, “A theory of cortical responses,” Philosophical Transactions of The Royal Society B: Biological Sciences, vol. 360, pp. 815–836, 2005

1:T

Cognitive Memory Functional Principles
Optimal Dynamic Systems with Sparse Coding

Kalman filter: A popular dynamic system used for estimation and prediction. Optimal under linear and Gaussian assumptions.

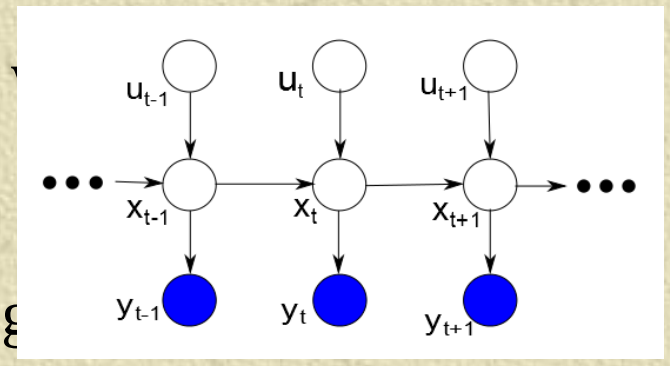
Sparse Coding: Encode the observation over a set of over-complete basis such that only a few basis are active at each instance.

?

Dynamic Model with Sparse Coding

$$x_t = Axt\_1 + But + v$$
$$y_t = Cxt + nt$$
$$s.t\ \|x_t\|_1 \leq \epsilon$$

$u_t =$ Unknown control/input sig



Sequential Dual Estimation Problem: Train the network assuming that the parameters are changing dynamically over time, while considering the transition matrix fixed. Alternate between inferring the states and learning the parameters. How to propagate the second (or higher order) statistics over time, while maintaining the constraints?

Cognitive Memory Functional Principles
Bayesian Sparse Coding

- Consider the prior on the states as

$$p(x_t;\tau) = \mathrm{N}(x_t;0,\Sigma(\tau)) \qquad p(\tau_i;\gamma) = \frac{\gamma}{2}\exp(-\frac{\gamma}{2}\tau_i)\,\forall i \quad and \quad \tau_i > 0$$

  It can be shown that it leads to an l_1-norm constraint.

- Optimization: Find MAP estimate of x_t and u_t using EM algorithm while considering τ as a hidden variable
- **E Step**: Marginalize τ to obtain its posterior

$$V = \gamma diag(|x_t^1|^{-1},...,|x_t^k|^{-1})$$

- **M Step**: Solve for x_t and u_t using

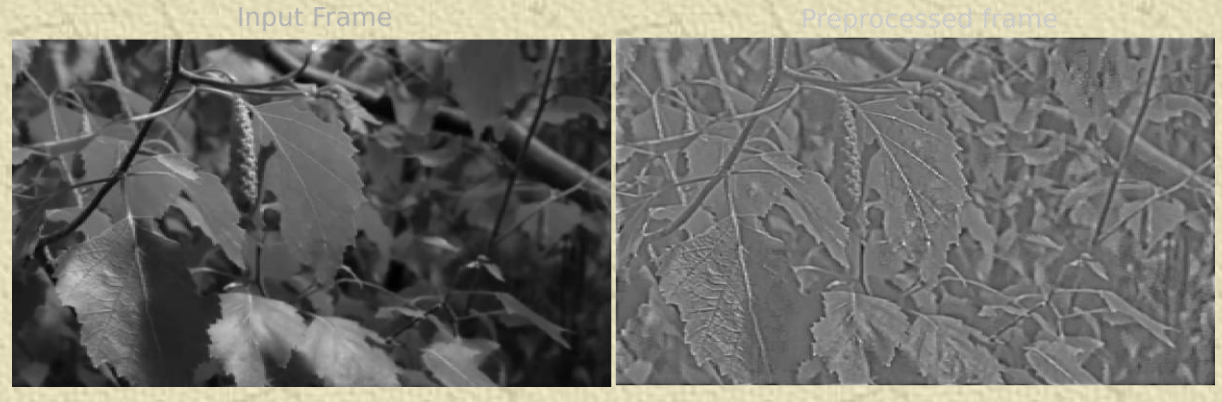
$$\hat{x}_t, \hat{u}_t = \arg\max\, p(y_t;x_t)\,p(x_t,Y_{T-1},u_t)\,p(x_t;V)\,p(u_t)$$

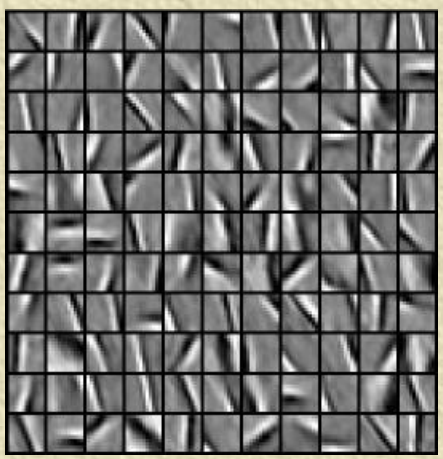
- The posterior distribution over states x_t is approximated as a Super Gaussian, and the propagation of the Covariance Matrix over time is still possible.

Alex Nelson. *Nonlinear estimation and modeling of noisy time-series by dual Kalman filtering methods*. PhD thesis, 2000.

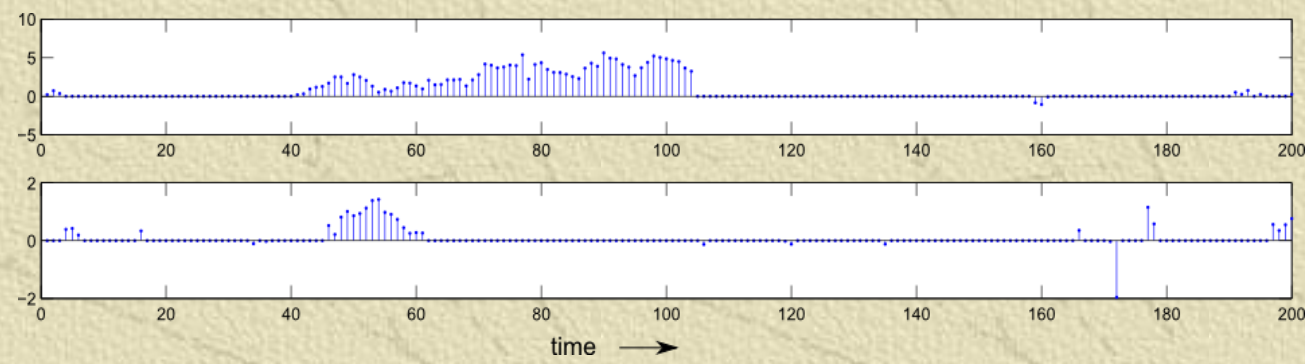
Example: Natural Time-Varying Images

11x11 video patches (observations). The dataset contains 800, 150 frames obtained from videos of natural scenes.





Basis



The states change gradually i.e., gradual change in the basis over time.

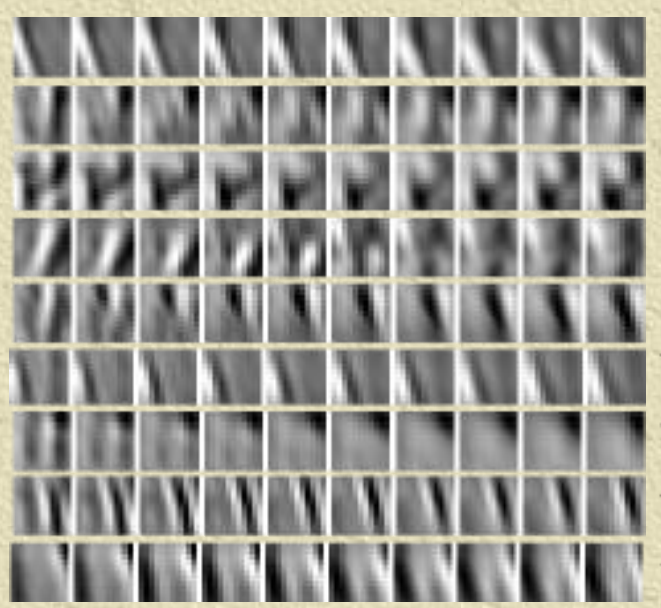
Example: Natural Time-Varying Images

Results show the "generated" data for a single layer network.

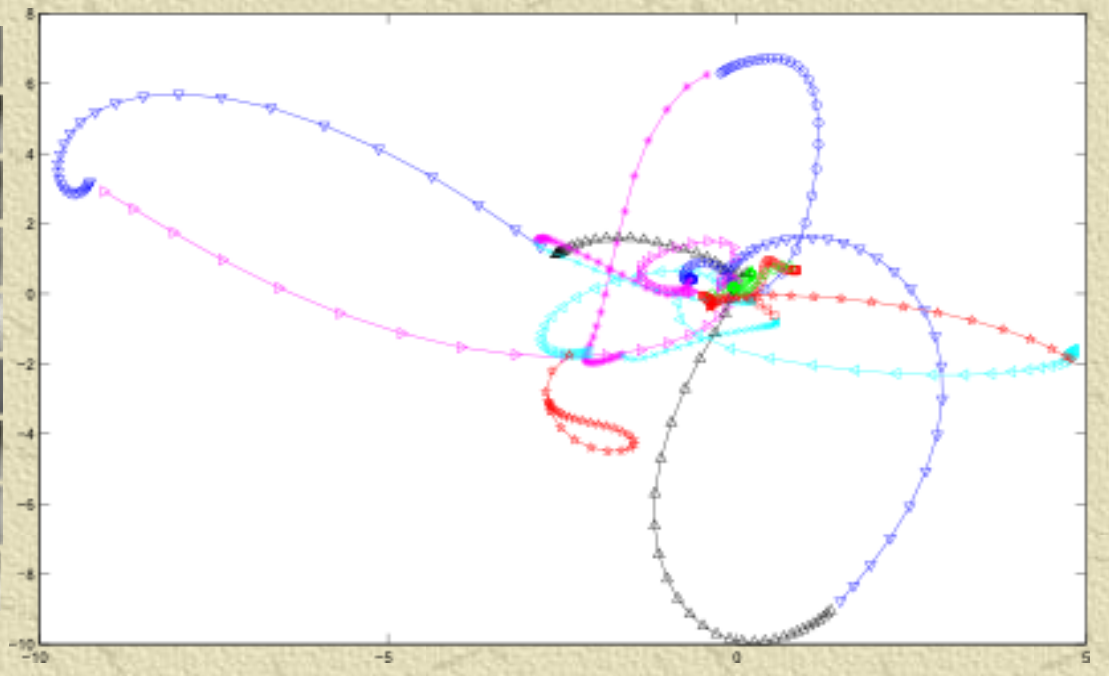
Once the system is learnt, fix the cause u(t) over time and initialize the states to zeros and let the system evolve over time (same procedure for the attractor curves as well).

Causes act as control inputs and determine the shape of the attractor.

With fixed inputs



Change in the attractors for different control signals/inputs.



What have we accomplished?

- NeuroScience tells us that neural processing is bottom up but also top down in close loop feedback.
- We are determined to understand using ML and SSP principles how distributed hierarchical dynamical models with sparse constraints can model visual memory and provide efficient CAM storage.
- Coupling causes (memory) with the processing may simplify tremendously scene labeling, and explain figure background segregation (you see what you want to see!)
- This system uses compressive sampling but learns the basis functions. This can have impact both on multimodal video processing and reduce greatly storage.

Conclusions

- We showed how Machine Learning and Information Theoretic ideas can push Statistical Signal Processing out of its mold of linear models, Gaussian, and non stationary assumptions
- Machine Learning and Information Theory provide rigorous frameworks and the skills of Signal Processing experts are badly needed to make these ideas practical.
- The applications are many in speech and video analysis
- The impact in creating Cognitive Information Processing systems for engineering can be enormous.