

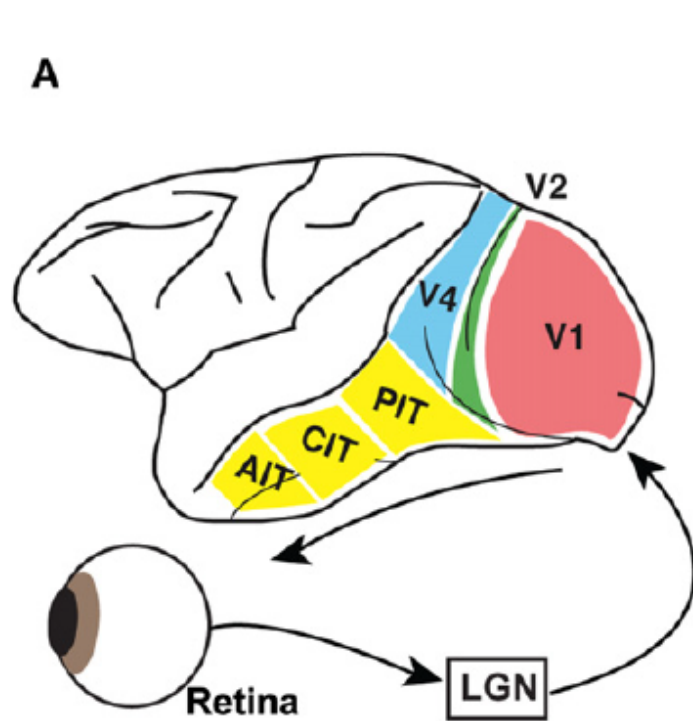
Computation in the Brain

Amitabha Mukerjee

Intro to Cognitive Science

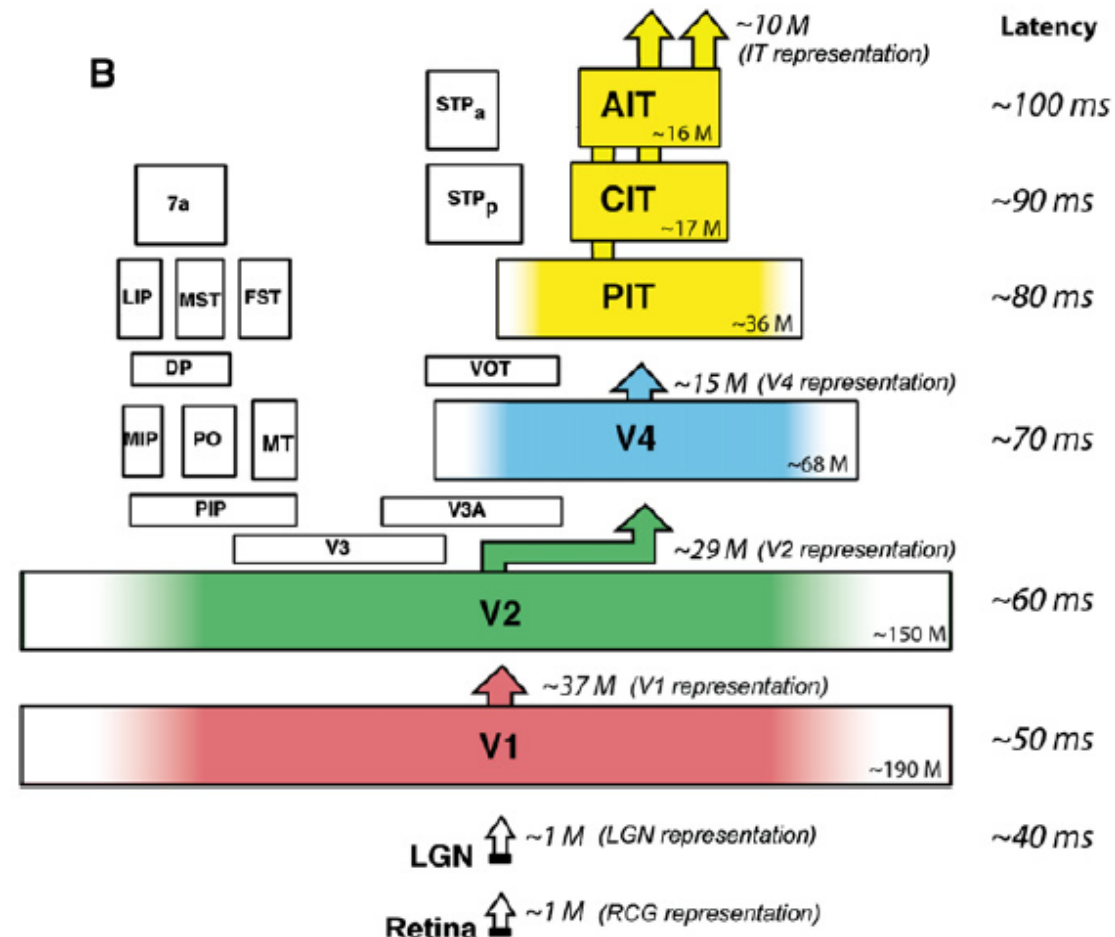
Manifolds in the Visual Brain

Visual Recognition Pathway (Ventral)



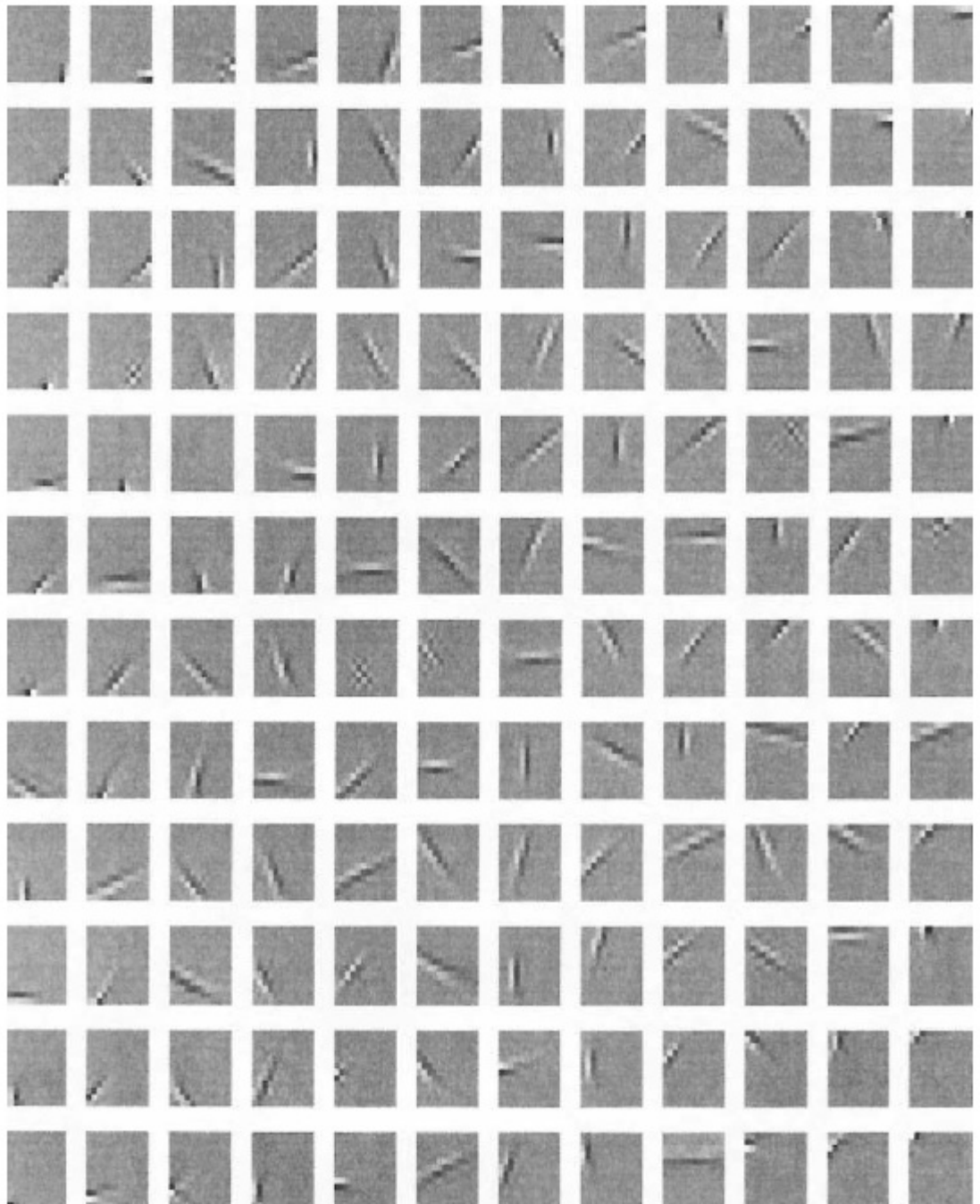
Ventral pathway, Macaque brain

Deep colours: process central 10 deg





Early Responses



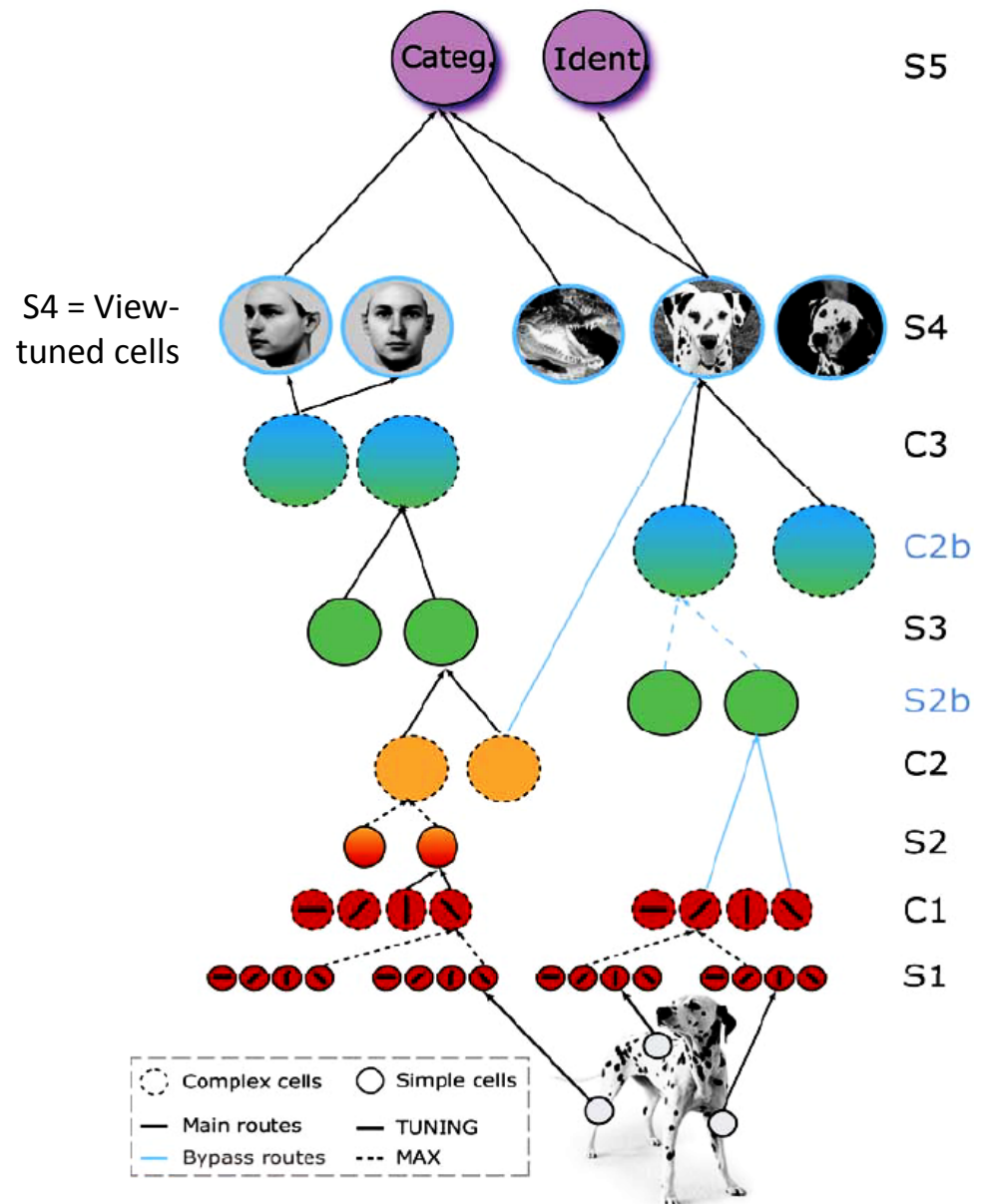
Bell / Sejnowski 02

Computational models (Poggio)

[Serre Oliva Poggio 2007]

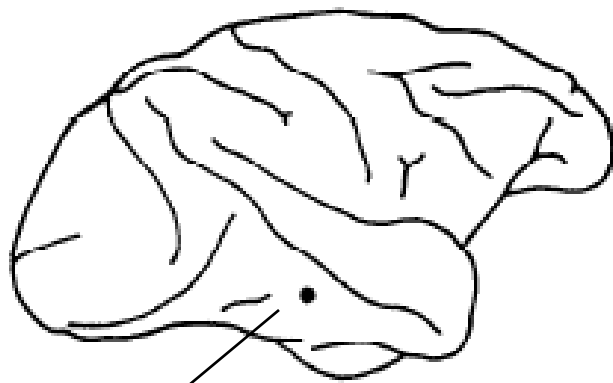
[Serre Wolf Riesenhuber Poggio
2007 pami]

This computational model did as
well on object recognition tasks as
state-of-art CV (bag of words)
models.

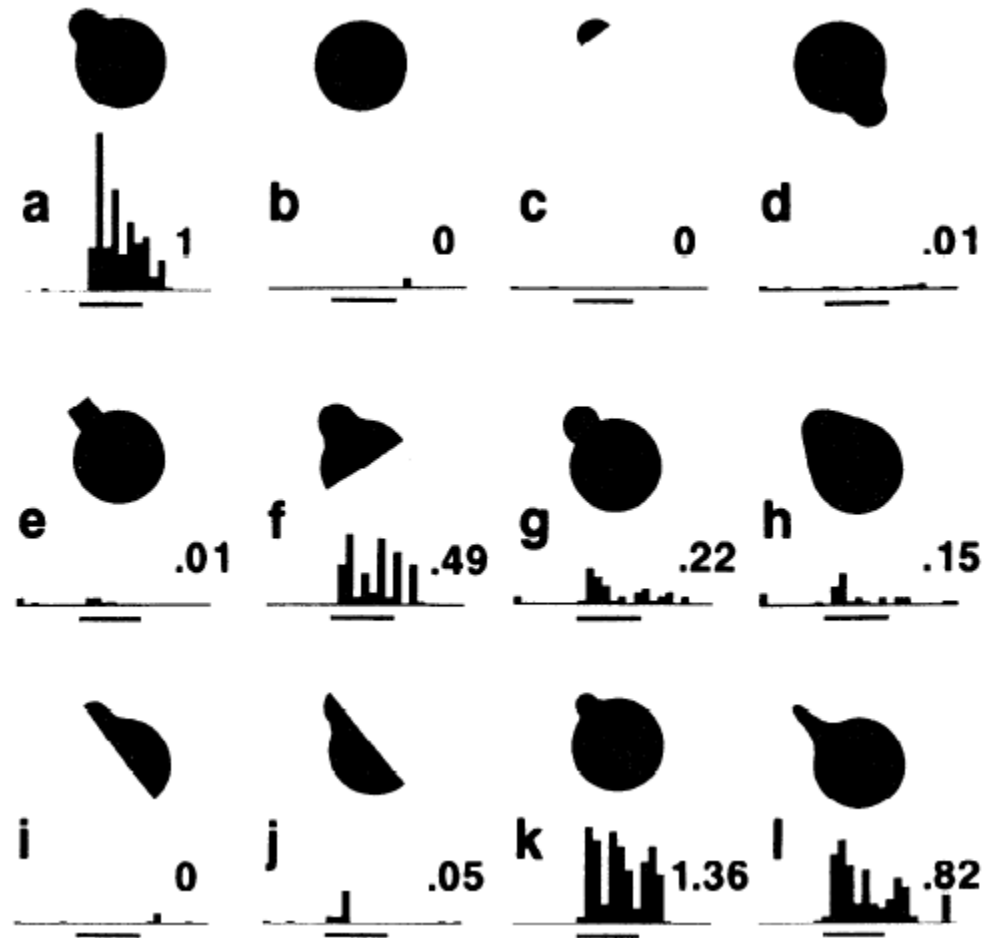


Selectivity for complex features

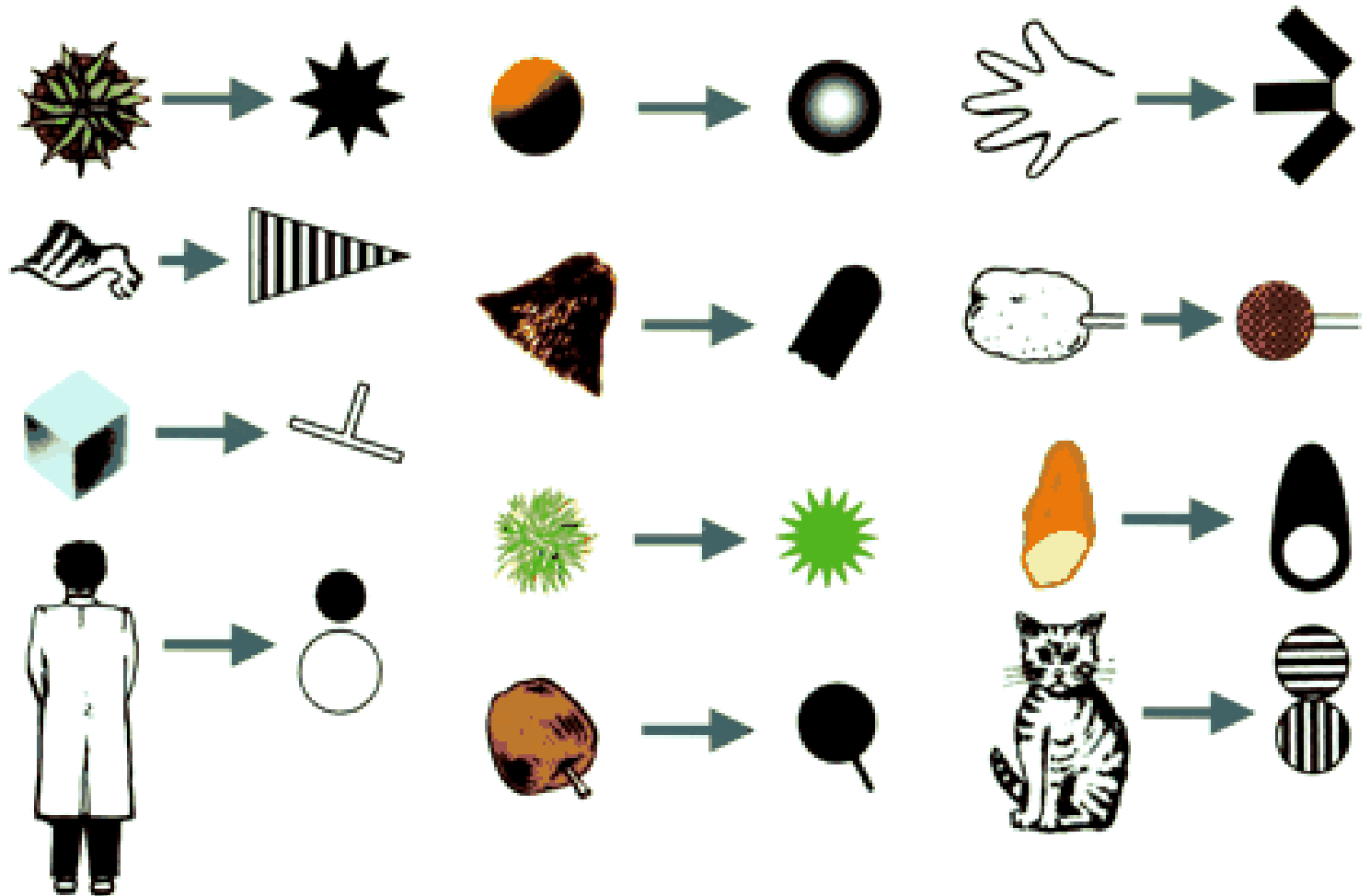
[kobatake-tanaka-94]
neuronal-selectivity-to-complex-
object-features



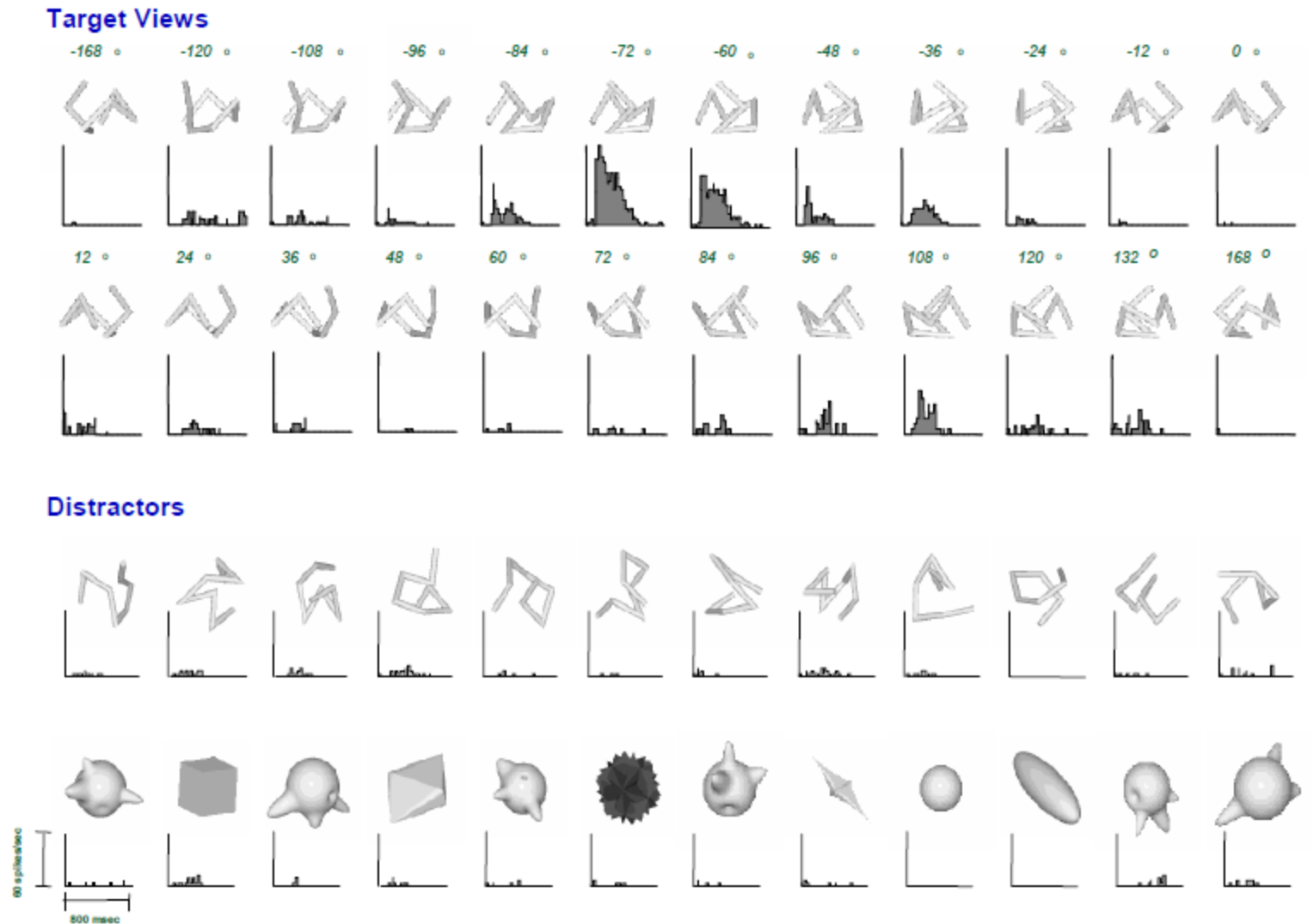
Location of
recorded IT cell



Simplifying the response

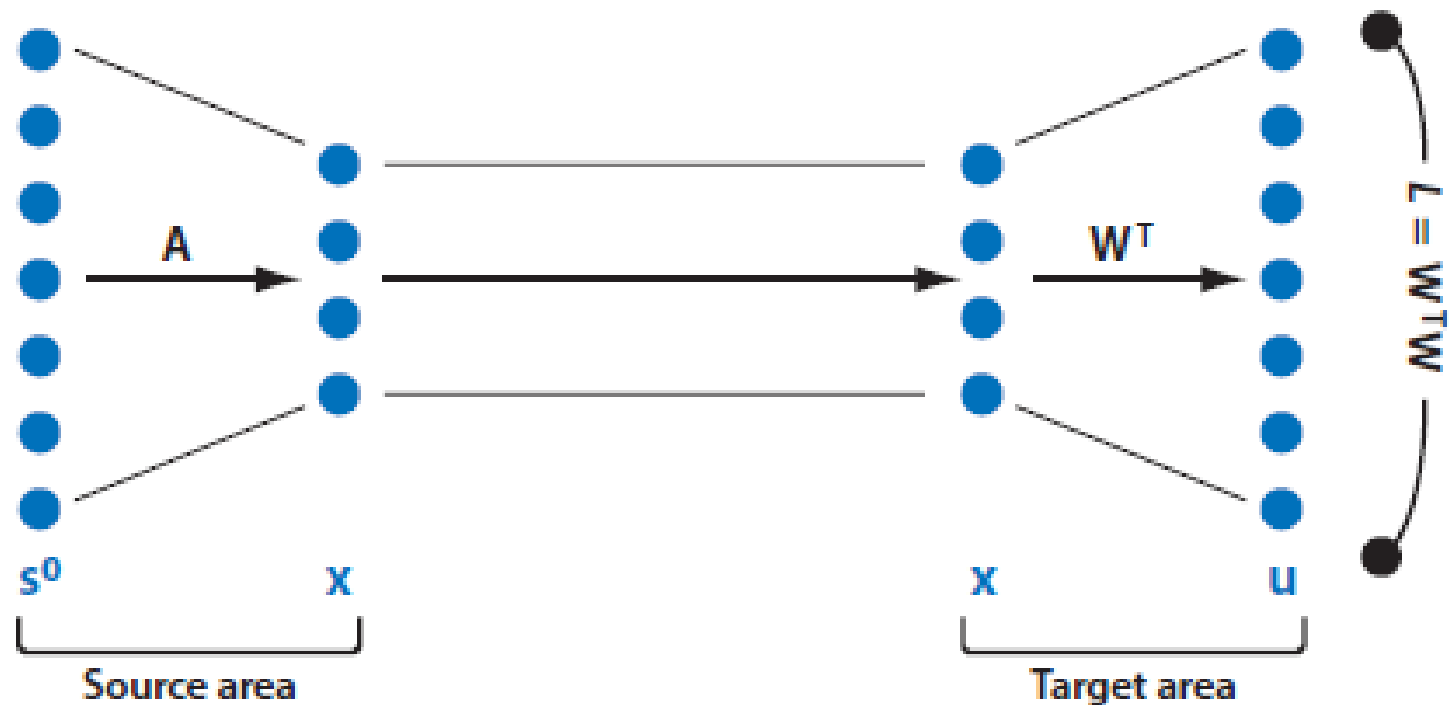


Object-specific View-tuned cells

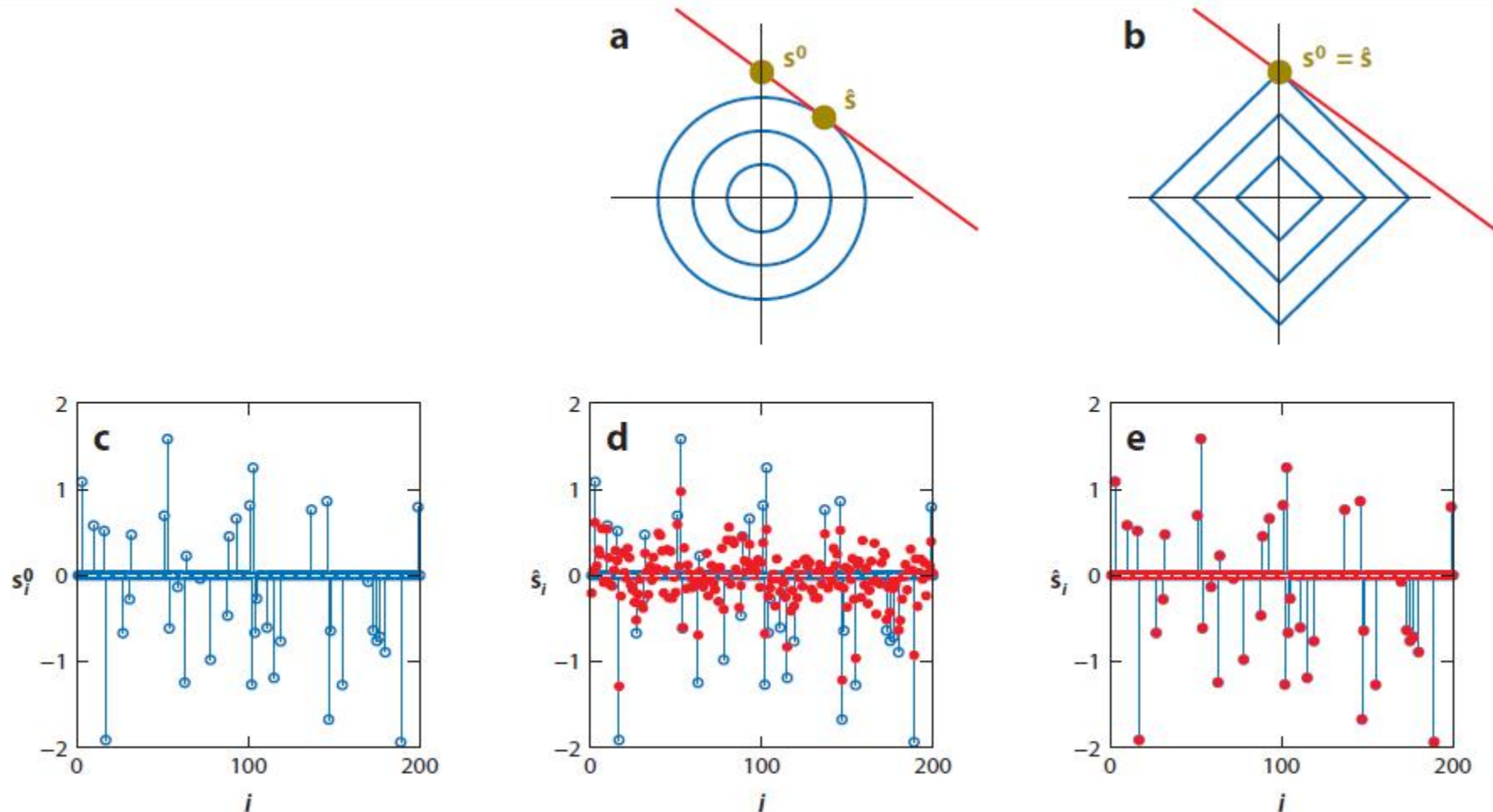


[logothetis pauls poggio 95] shape representation in IT macaque

Dimensionality Reduction and Long-distance neural communication

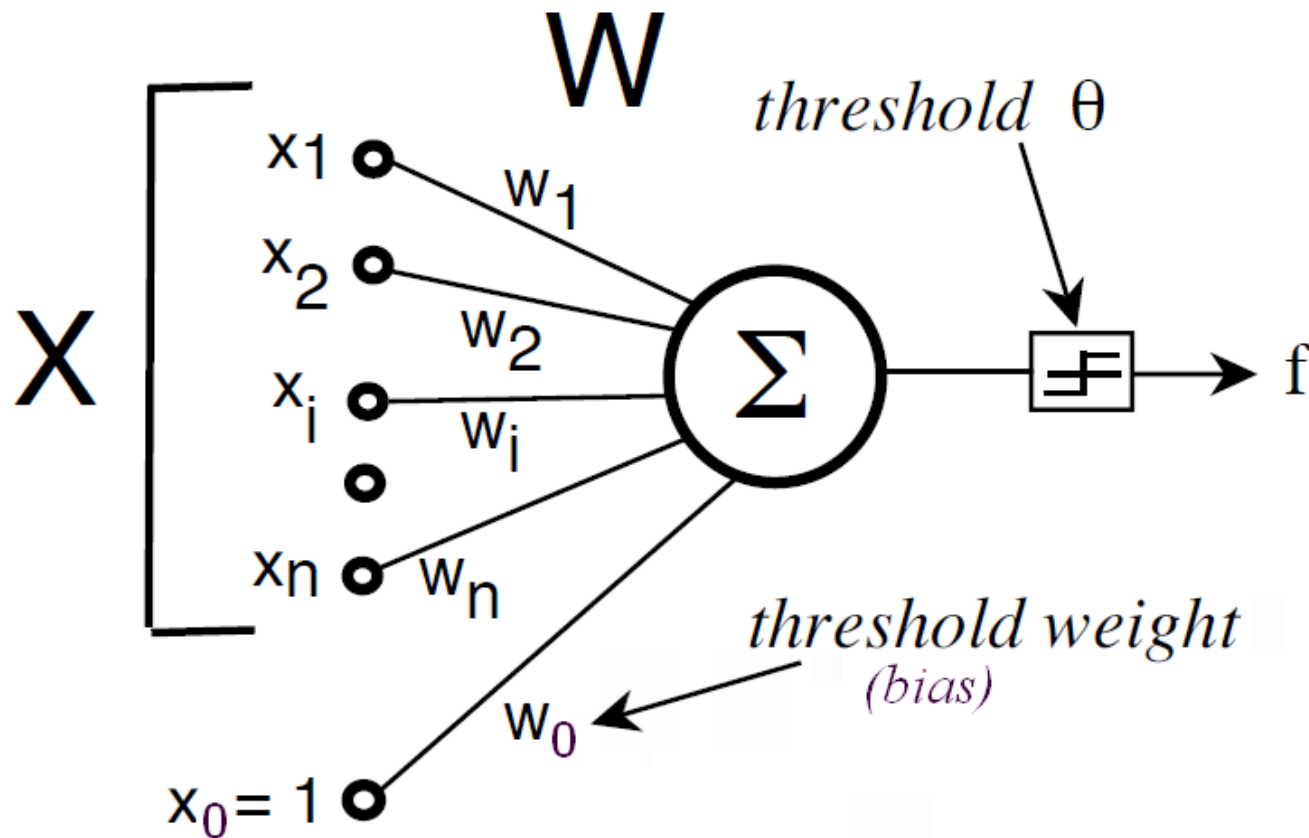


Dimensionality Reduction and Long-distance neural communication



Artificial Neural Networks

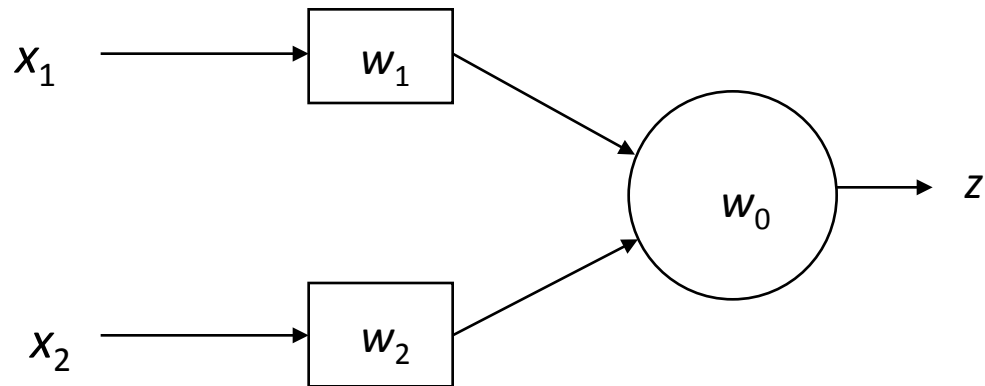
Perceptron (Threshold unit)



$$f = 1 \text{ if } \sum w_i x_i > \theta; \text{ else } = 0$$

Learn weights W
to best match
output on
training set

Perceptron



$$Z = 1 \text{ if } \sum w_i x_i \geq \text{bias}$$

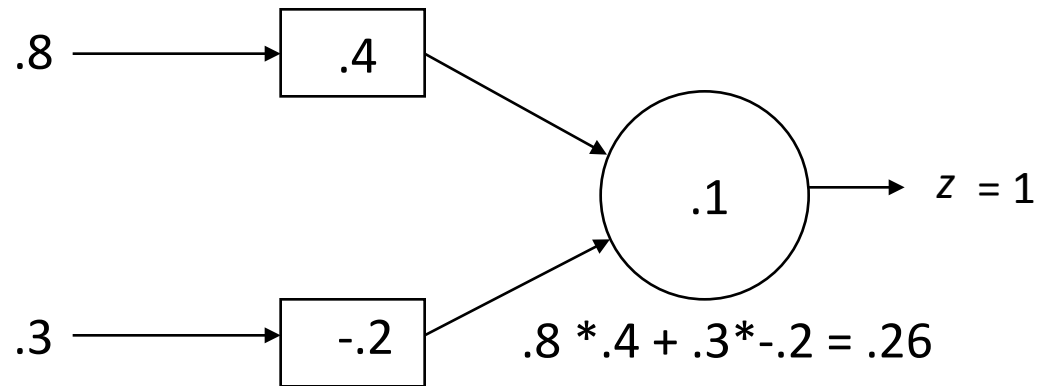
$$\epsilon * (t - z) * x_i$$

x_1	x_2	t
.8	.3	1
.4	.1	0

output $z =$

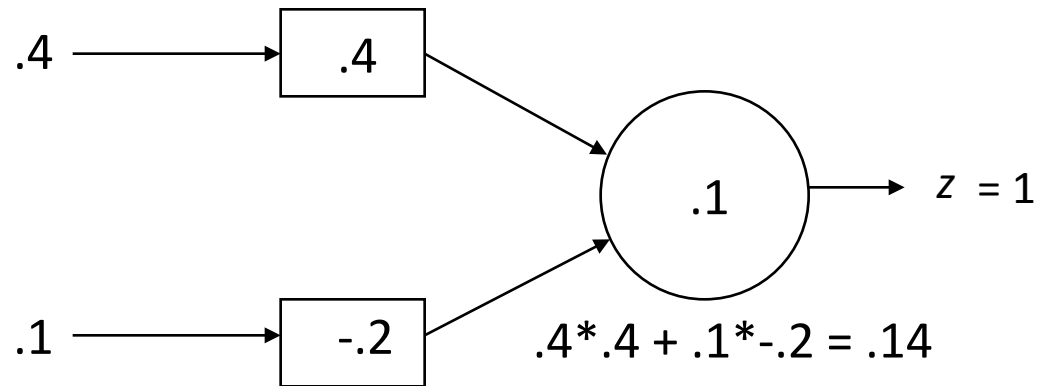
- 1 if $\sum w_i * x_i > \text{bias}$
- 1 if $\sum w_i * x_i < \text{bias}$

Training a Perceptron



x_1	x_2	t
$.8$	$.3$	1
$.4$	$.1$	0

Training a Perceptron



x_1	x_2	t
.8	.3	1
.4	.1	0

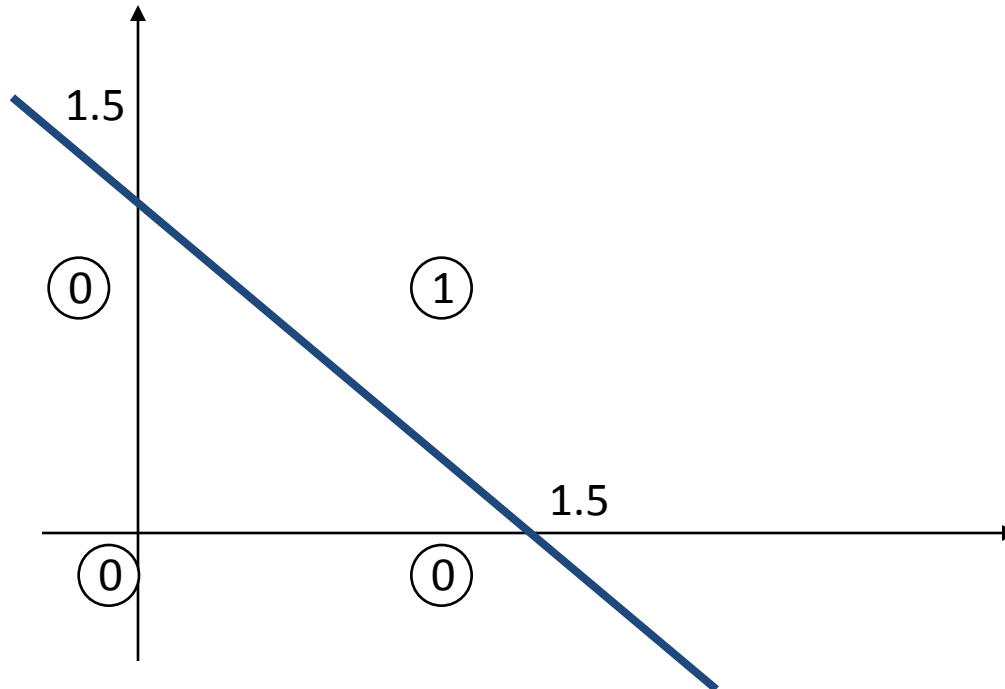
$$\Delta w_i = \varepsilon * (t - z) * x_i$$

Linear Classifier

Heaviside function has threshold at $x=0$. Decision boundary given by:

$$a = w^*x + w_0 = w_0 + w_1 x_1 + w_2 x_2 = 0$$

Thus: $x_2 = -(w_0 + w_1 x_1)/w_2$.





Dimensionality


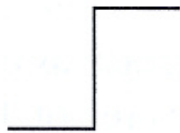



- What fraction of possible functions are Linearly separable?
- Consider Boolean functions
 - n = Number of variables

n	Total Functions 2^{2^n}	Linearly Separable	
0	2	2	
1	4	4	
2	16	14	
3	256	104	
4	65536	1882	$2^{(n/2+16)} < \lambda(n) \leq 2^{n^2}$
5	4.3×10^9	94572	
6	1.8×10^{19}	1.5×10^7	
7	3.4×10^{38}	8.4×10^9	
8	1.2×10^{77}	1.7×10^{13}	

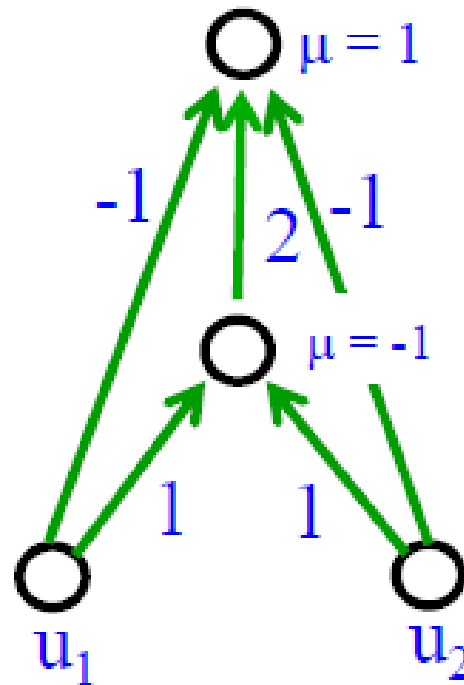
Examples of activation functions

The sigma node

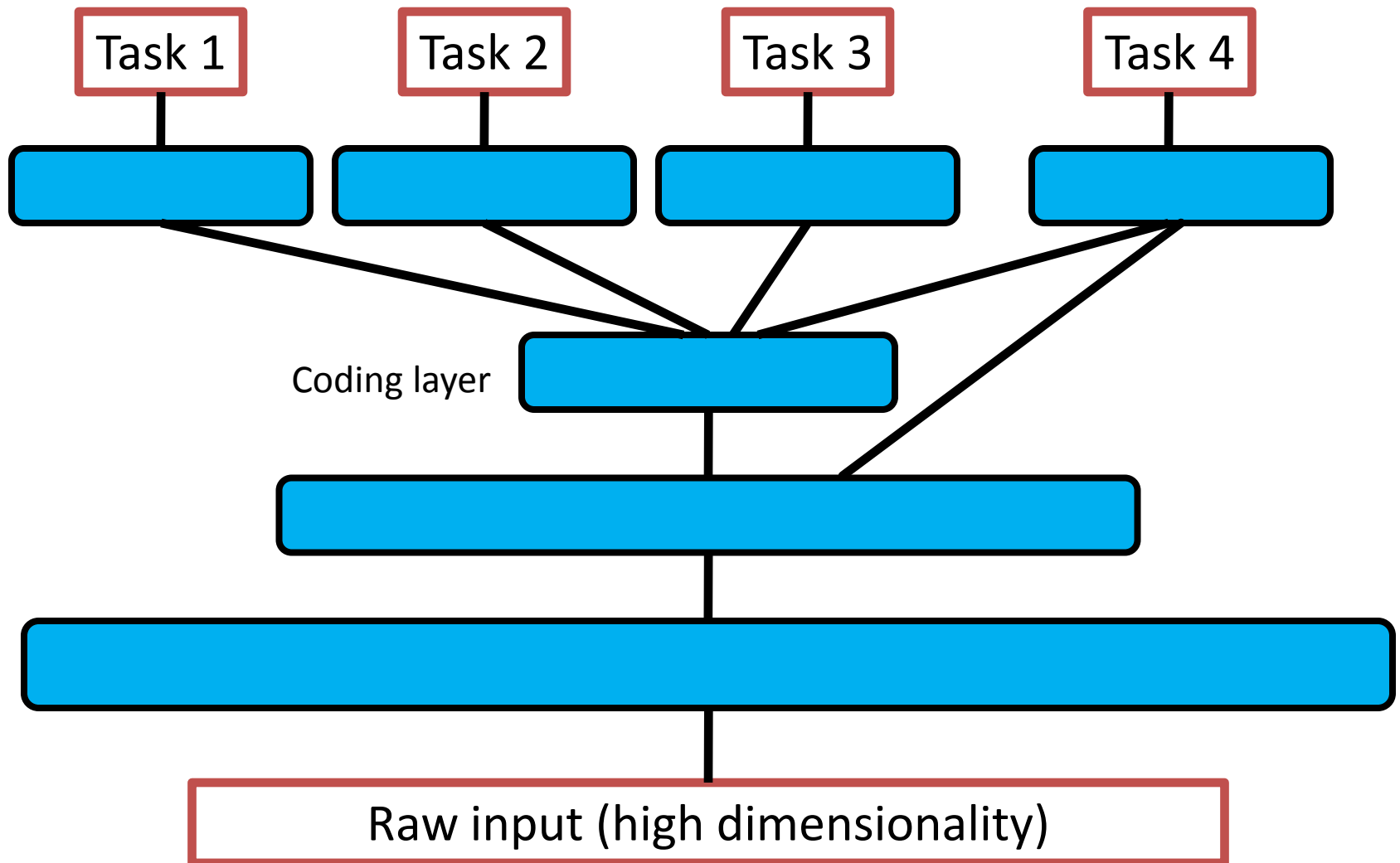
Table 4.2 Examples of frequently used activation functions and their basic implementation in MATLAB.

Type of function	Graph	Mathematical formula	Matlab implementation
Linear		$g^{\text{lin}}(x) = x$	<code>x</code>
Step		$g^{\text{step}}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{elsewhere} \end{cases}$	<code>floor(0.5*(1+sign(x)))</code>
Threshold-linear		$g^{\text{theta}}(x) = x \Theta(x)$	<code>x.*floor(0.5*(1+sign(x)))</code>
Sigmoid		$g^{\text{sig}}(x) = \frac{1}{1+\exp(-x)}$	<code>1./(1+exp(-x))</code>
Radial-basis		$g^{\text{gauss}}(x) = \exp(-x^2)$	<code>exp(-x.^2)</code>

Multi-layer perceptron



Deep Learning



Traditional Learning

■ The traditional model of pattern recognition (since the late 50's)

- ▶ Fixed/engineered features (or fixed kernel) + trainable classifier



hand-crafted
Feature Extractor

"Simple" Trainable
Classifier

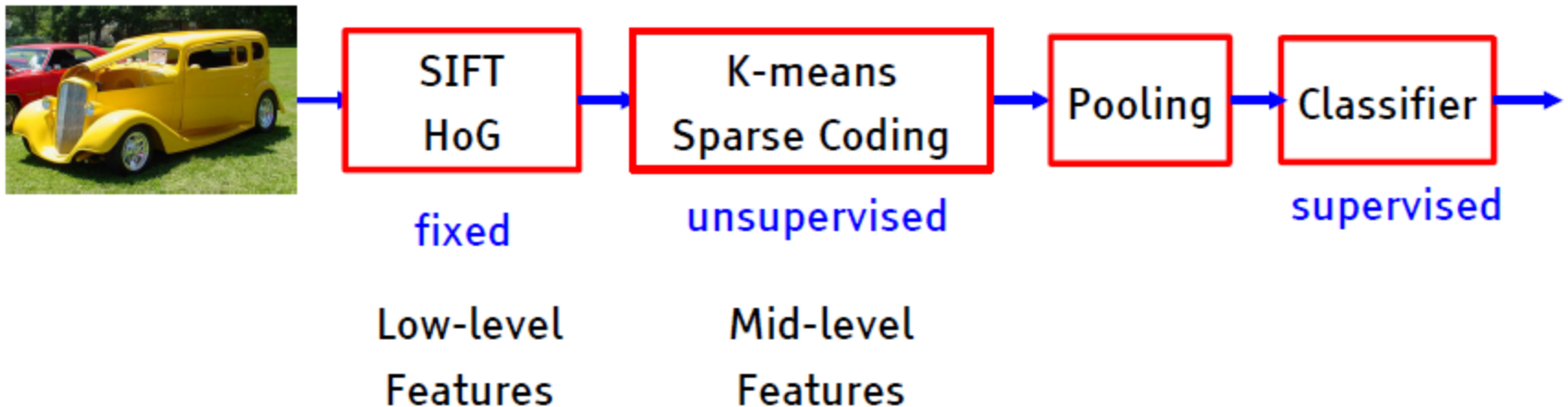
■ End-to-end learning / Feature learning / Deep learning



Trainable
Feature Extractor

Trainable
Classifier

Traditional Learning

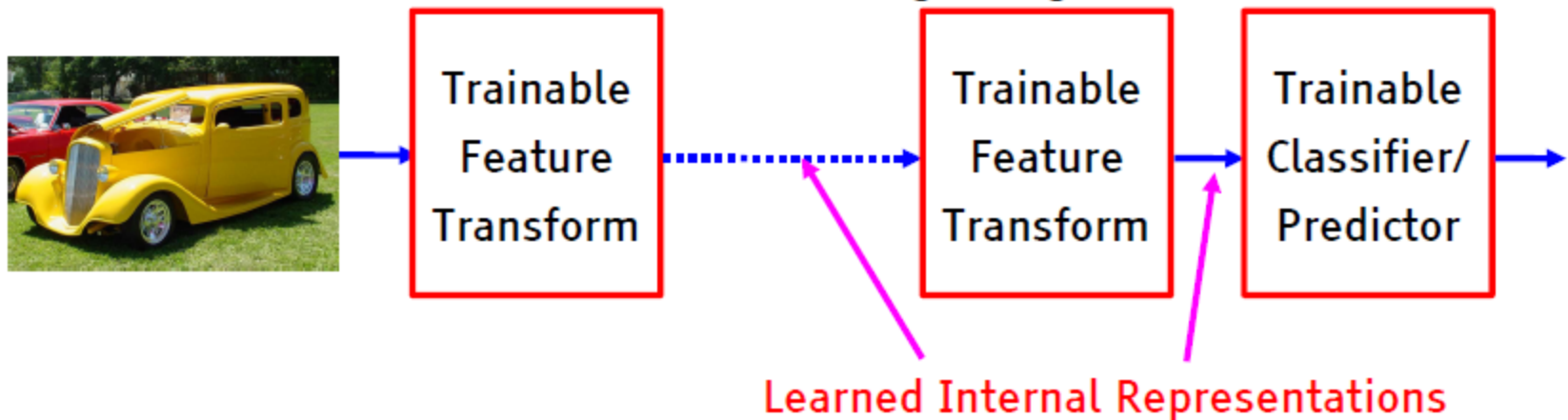


Hand-engineered features

Feature Discovery in Deep Learning

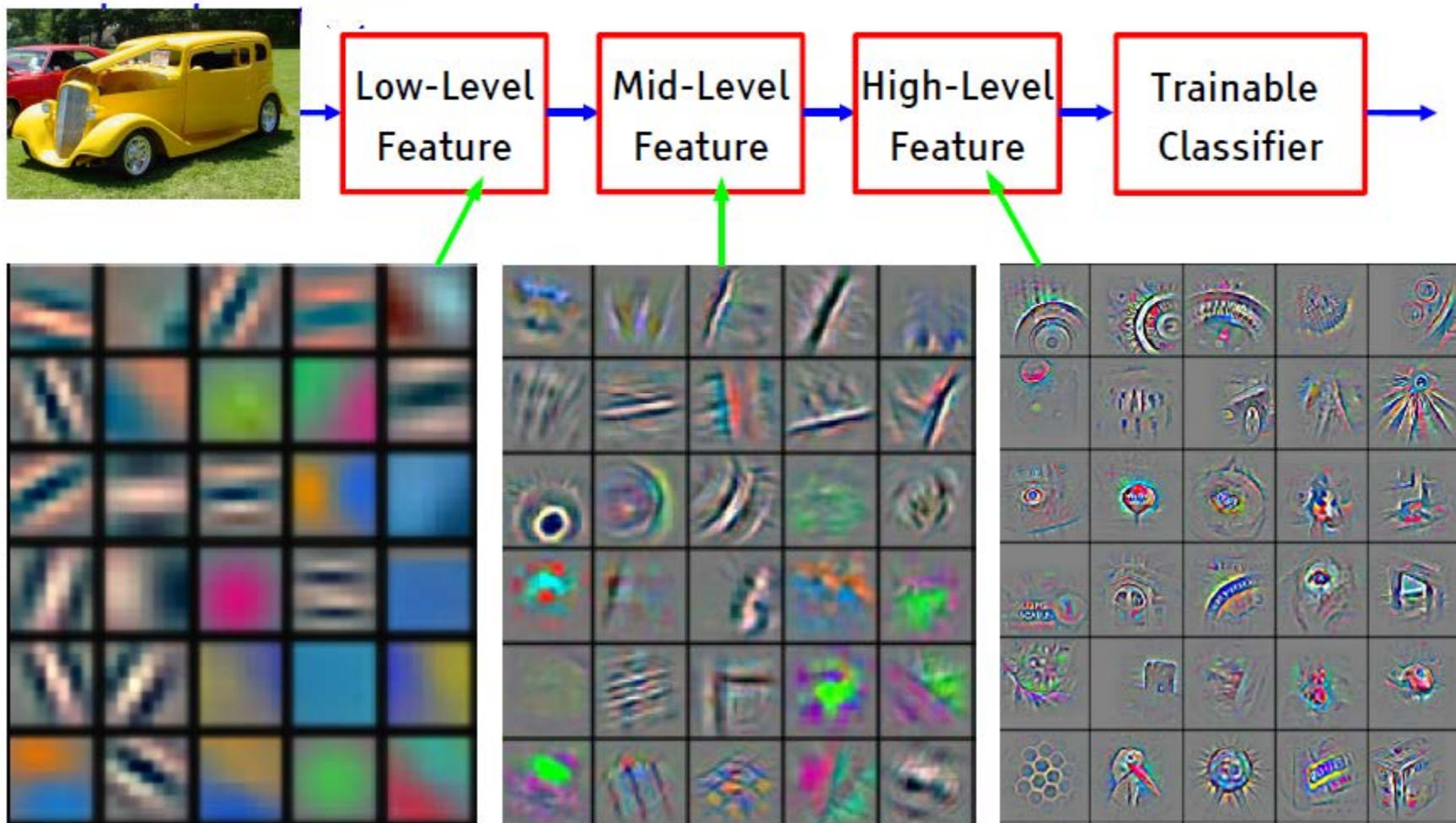
■ A hierarchy of trainable feature transforms

- ▶ Each module transforms its input representation into a higher-level one.
- ▶ High-level features are more global and more invariant
- ▶ Low-level features are shared among categories



- ## ■ How can we make all the modules trainable and get them to learn appropriate representations?

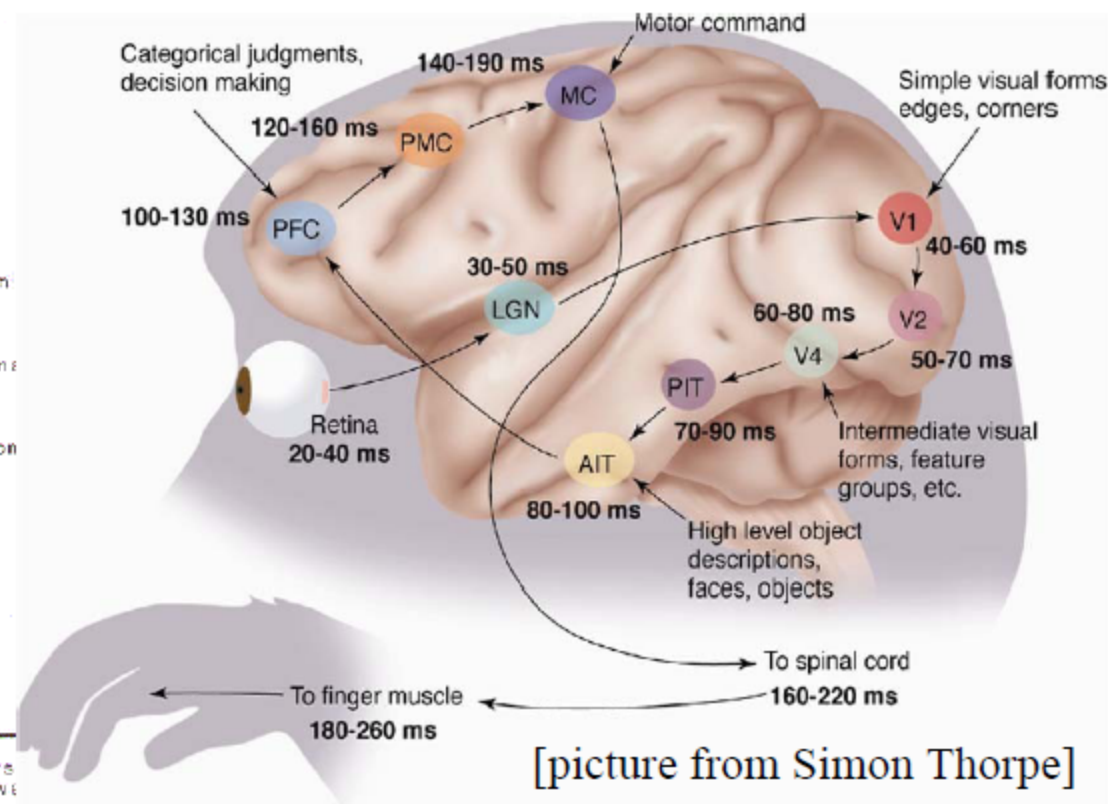
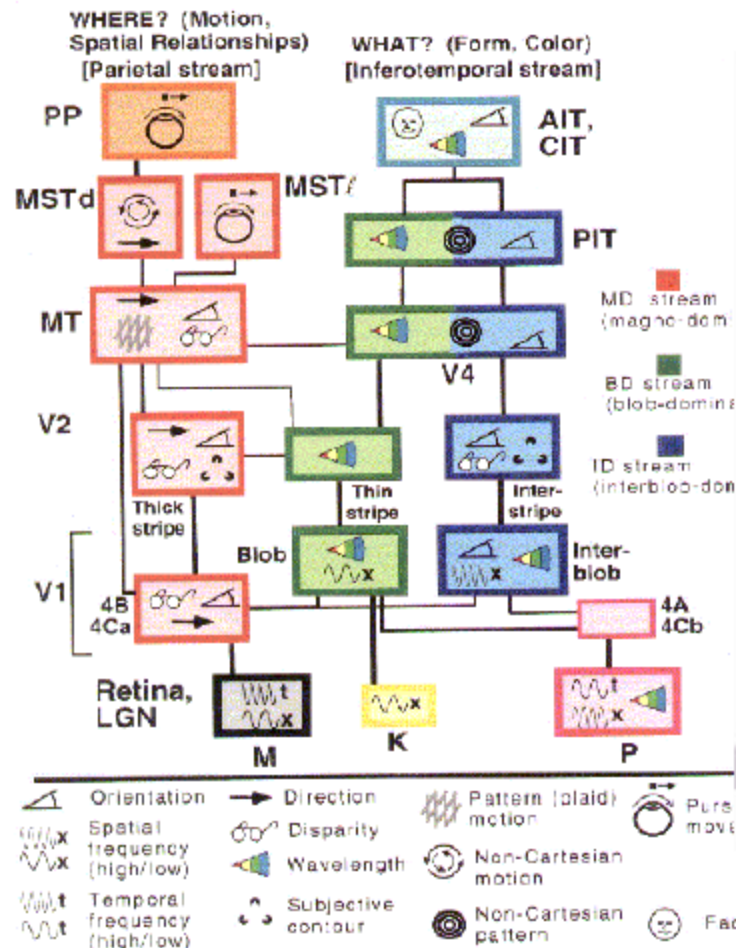
Deep Learning



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Ventral Visual Pathway

- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT
- Lots of intermediate representations



[Gallant & Van Essen]

Face Recognition and Manifolds

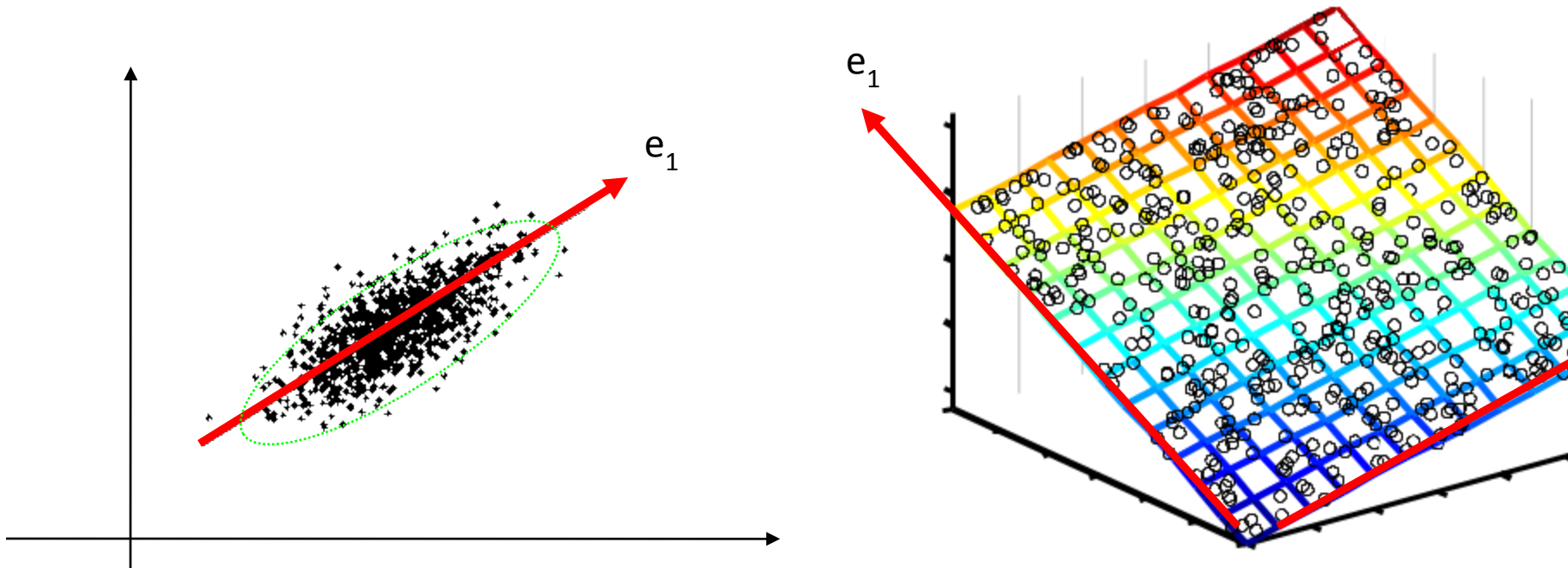


Eigenfaces: Linear Dimensionality Reduction

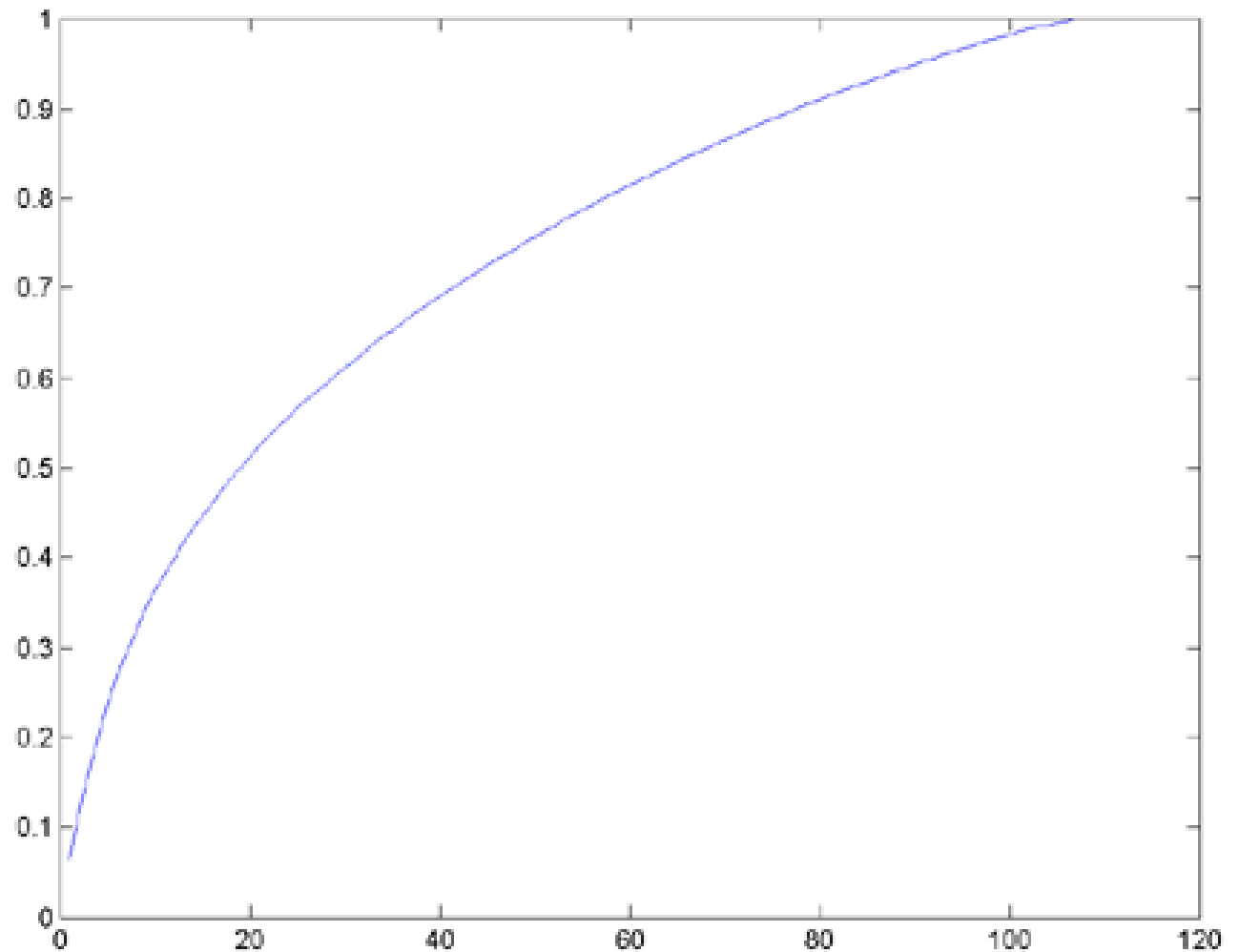
PCA: project data onto subspace of maximum variance

$[A]$ = **top** eigenvectors of covariance matrix $[XX^T]$

$$Y = [A] X$$



Percentage of Variance Explained



Percentage of Variance Explained

- $k=1$ to 25



Percentage of Variance Explained

$k=1,9,17, 25, 33... 89$



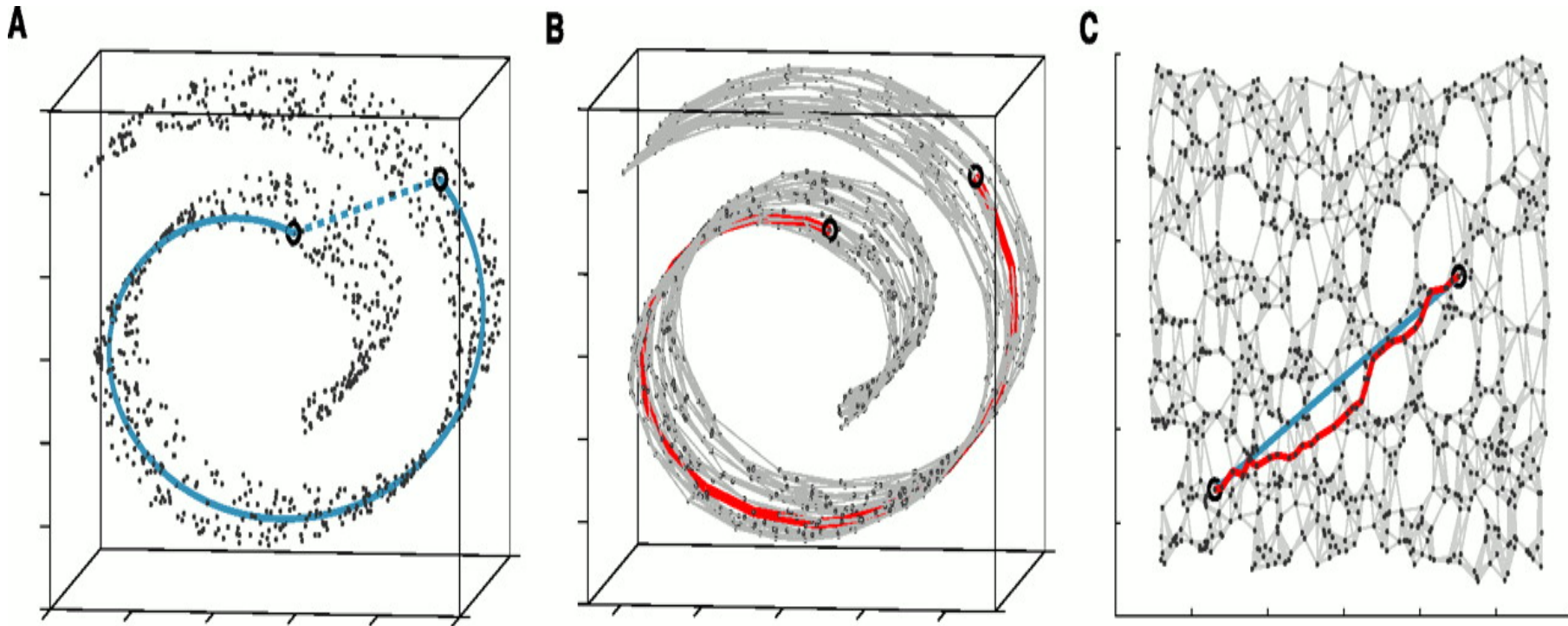
Percentage of Variance Explained



Non-Linear Dimensionality Reduction (NLDR)

algorithm: ISOMAP

Data may lie on non-linear “manifolds”



Geodesic = shortest path along manifold

Isomap Algorithm

A Global Geometric Framework for Nonlinear Dimensionality Reduction

Joshua B. Tenenbaum,^{1*} Vin de Silva,² John C. Langford³

Scientists working with large volumes of high-dimensional data, such as global climate patterns, stellar spectra, or human gene distributions, regularly confront the problem of dimensionality reduction: finding meaningful low-dimensional structures hidden in their high-dimensional observations. The human brain confronts the same problem in everyday perception, extracting from its high-dimensional sensory inputs—30,000 auditory nerve fibers or 10^6 optic nerve fibers—a manageably small number of perceptually relevant features. Here we describe an approach to solving dimensionality reduction problems that uses easily measured local metric information to learn the underlying global geometry of a data set. Unlike classical techniques such as principal component analysis (PCA) and multidimensional scaling (MDS), our approach is capable of discovering the nonlinear degrees of freedom that underlie complex natural observations, such as human handwriting or images of a face under different viewing conditions. In contrast to previous algorithms for nonlinear dimensionality reduction, ours efficiently computes a globally optimal solution, and, for an important class of data manifolds, is guaranteed to converge asymptotically to the true structure.

A canonical problem in dimensionality reduction from the domain of visual perception is illustrated in Fig. 1A. The input consists of many images of a person's face observed under different pose and lighting conditions, in no particular order. These images can be thought of as points in a high-dimensional vector space, with each input dimension corresponding to the brightness of one pixel in the image or the firing rate of one retinal ganglion cell. Although the input dimension-

ality may be quite high (e.g., 4096 for these 64 pixel by 64 pixel images), the perceptually meaningful structure of these images has many fewer independent degrees of freedom. Within the 4096-dimensional input space, all of the images lie on an intrinsically three-dimensional manifold, or constraint surface, that can be parameterized by two pose variables plus an azimuthal lighting angle. Our goal is to discover, given only the unordered high-dimensional inputs, low-dimensional representations such as Fig. 1A with coordinates that capture the intrinsic degrees of freedom of a data set. This problem is of central importance not only in studies of vision (1–5), but also in speech (6, 7), motor control (8, 9), and a range of other physical and biological sciences (10–12).

The classical techniques for dimensionality reduction, PCA and MDS, are simple to implement, efficiently computable, and guaranteed to discover the true structure of data lying on or near a linear subspace of the high-dimensional input space (13). PCA finds a low-dimensional embedding of the data points that best preserves their variance as measured in the high-dimensional input space. Classical MDS finds an embedding that preserves the interpoint distances, equivalent to PCA when those distances are Euclidean. However, many data sets contain essential nonlinear structures that are invisible to PCA and MDS (4, 5, 11, 14). For example, both methods fail to detect the true degrees of freedom of the face data set (Fig. 1A), or even its intrinsic three-dimensionality (Fig. 2A).

Here we describe an approach that combines the major algorithmic features of PCA and MDS—computational efficiency, global optimality, and asymptotic convergence guarantees—with the flexibility to learn a broad class of nonlinear manifolds. Figure 3A illustrates the challenge of nonlinearity with data lying on a two-dimensional “Swiss roll”: points far apart on the underlying manifold, as measured by their geodesic, or shortest path, distances, may appear deceptively close in the high-dimensional input space, as measured by their straight-line Euclidean distance. Only the geodesic distances reflect the true low-dimensional geometry of the manifold, but PCA and MDS effectively see just the Euclidean structure; thus, they fail to detect the intrinsic two-dimensionality (Fig. 2B).

Our approach builds on classical MDS but seeks to preserve the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The crux is estimating the geodesic distance between faraway points, given only input-space distances. For neighboring points, input-space distance provides a good approxima-

¹Department of Psychology and ²Department of Mathematics, Stanford University, Stanford, CA 94305, USA. ³Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15217, USA.

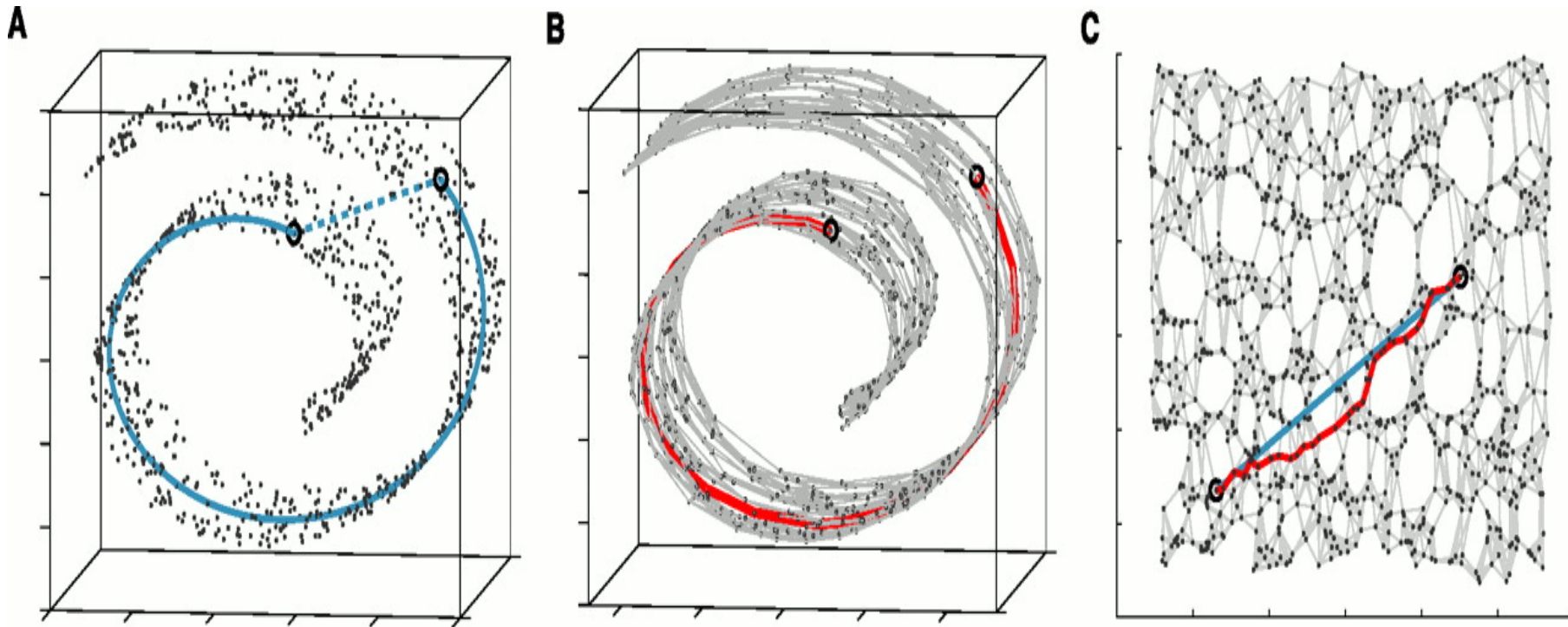
*To whom correspondence should be addressed. E-mail: jbt@psych.stanford.edu

Isomap Algorithm

- Identify neighbors.
 - points within epsilon-ball (ϵ -ball)
 - k nearest neighbors (k -NN)
- Construct neighborhood graph.
 - x connected to y if $neighbor(x,y)$.
 - edge length = $distance(x,y)$
- Compute shortest path between nodes
 - Dijkstra / Floyd-Warshall algorithm
- Construct a lower dimensional embedding.
 - Multi-Dimensional Scaling (MDS)

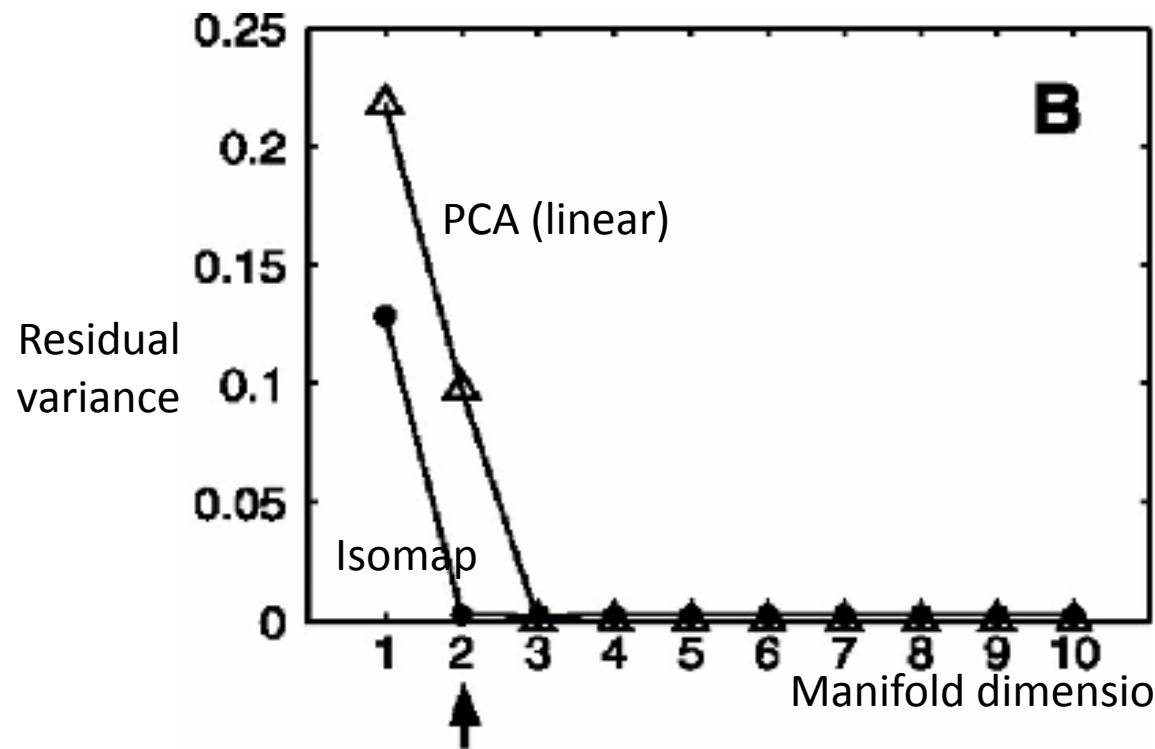
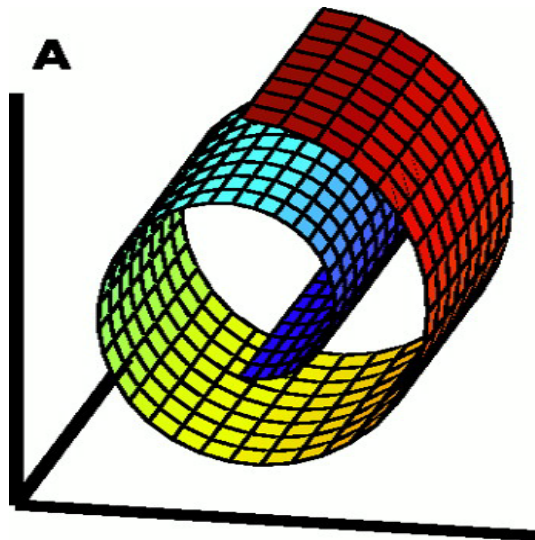
[Tenenbaum, de Silva and Langford 2001]

Isomap Algorithm



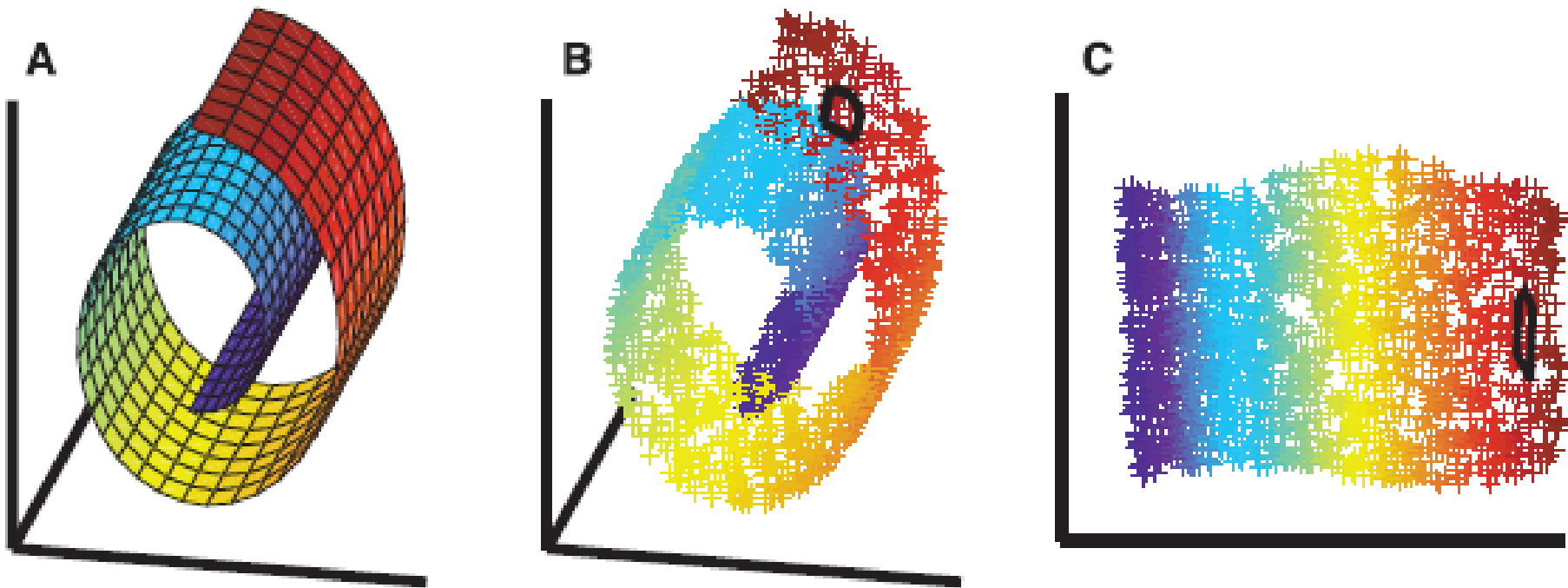
Geodesic = shortest path along manifold

Residual Variance and Dimensionality



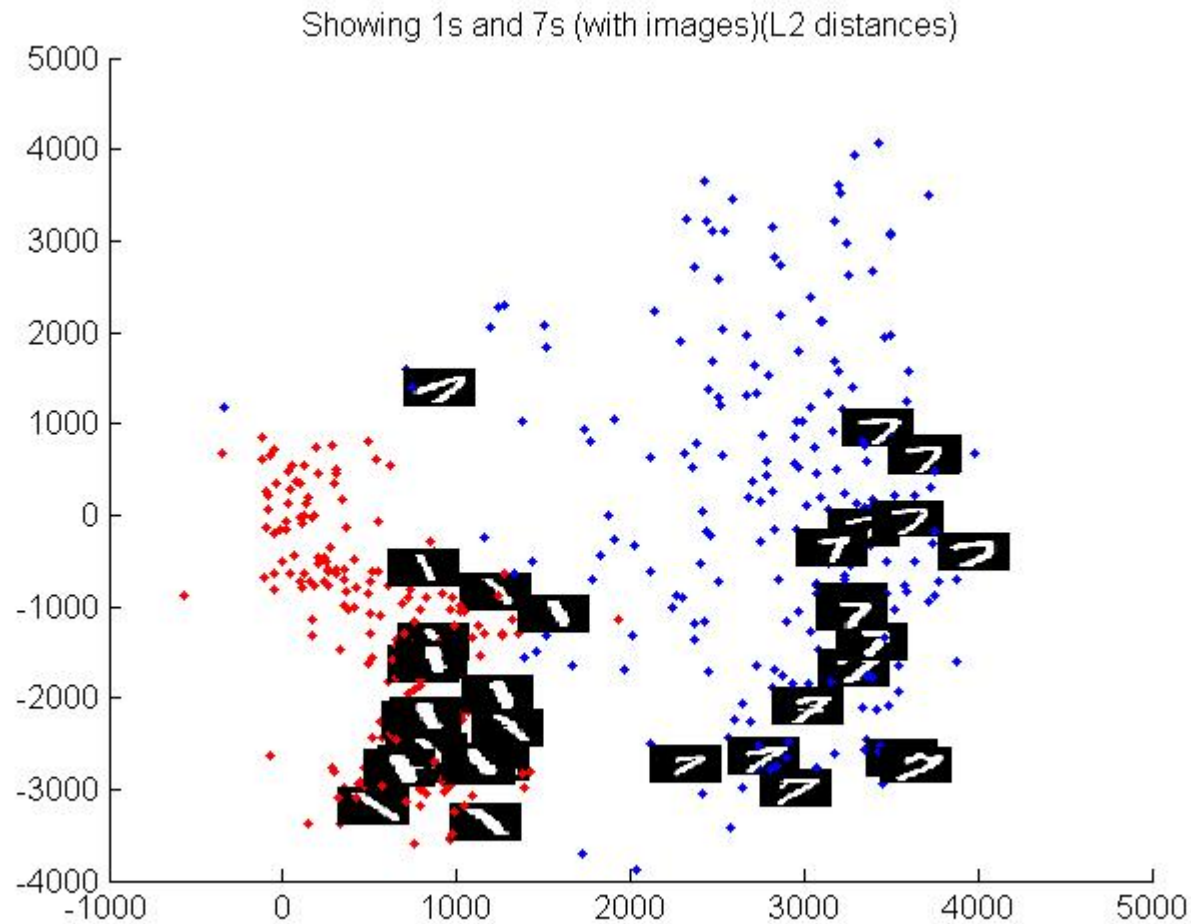
residual variance = $1 - r^2(D_g, D_y)$; r = linear correlation coefficient
 D_g = geodesic distance matrix; D_y = manifold distance

Manifold discovery



Locally Linear Embedding (Saul and Roweis 01)

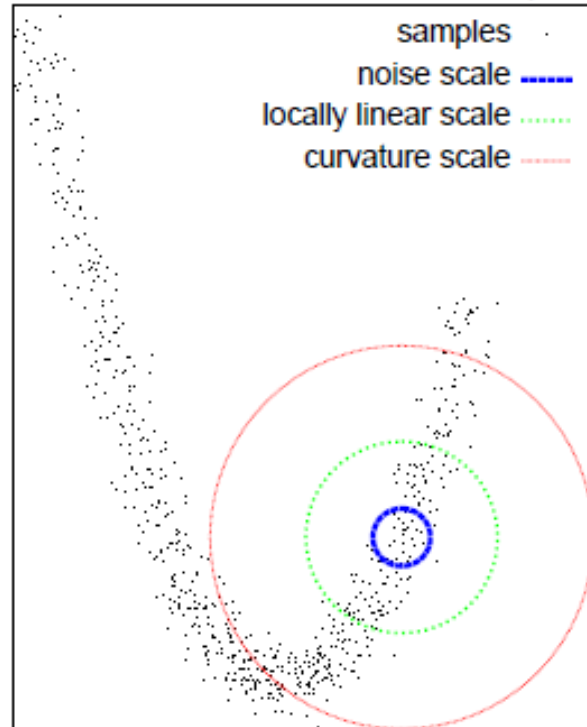
Manifold clustering



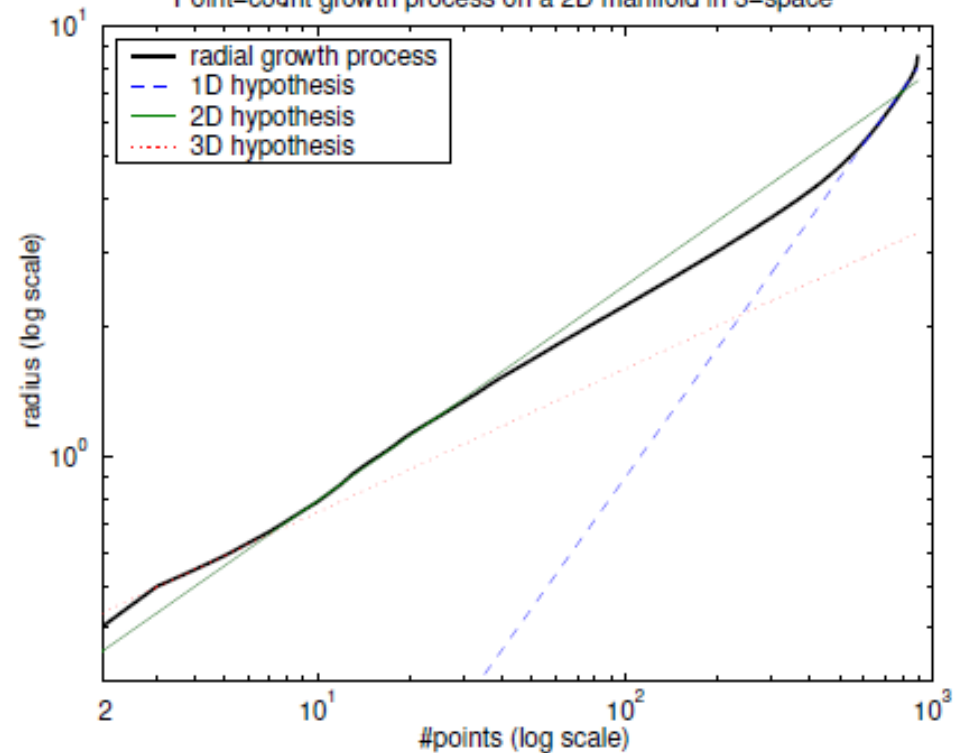
Clustering on Isomap embedding

r-ball density \rightarrow intrinsic dimensionality

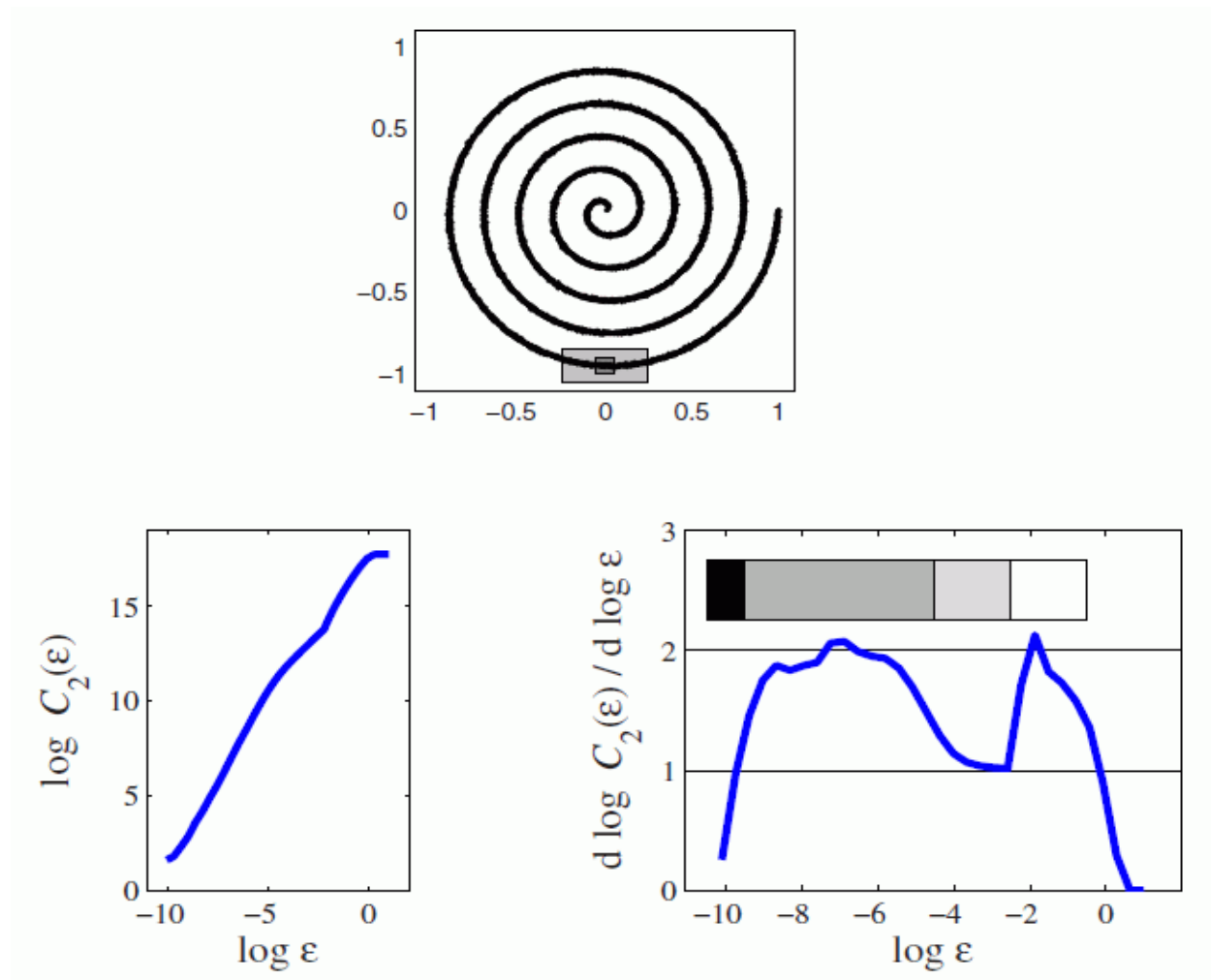
Scale behavior of a 1D manifold in 2-space



Point-count growth process on a 2D manifold in 3-space



Effect of scale on dimensionality

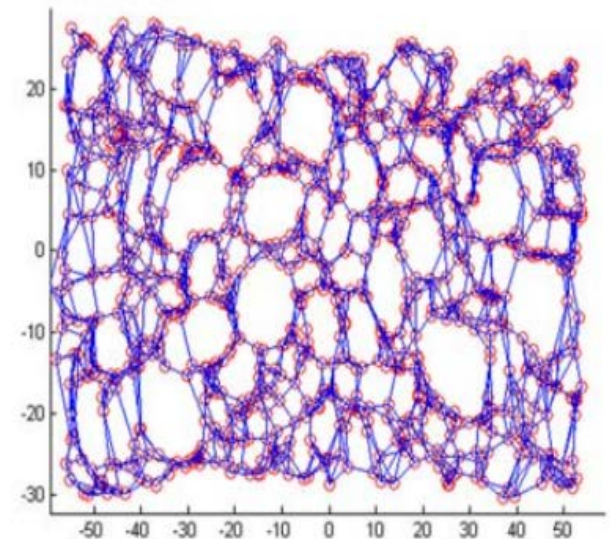
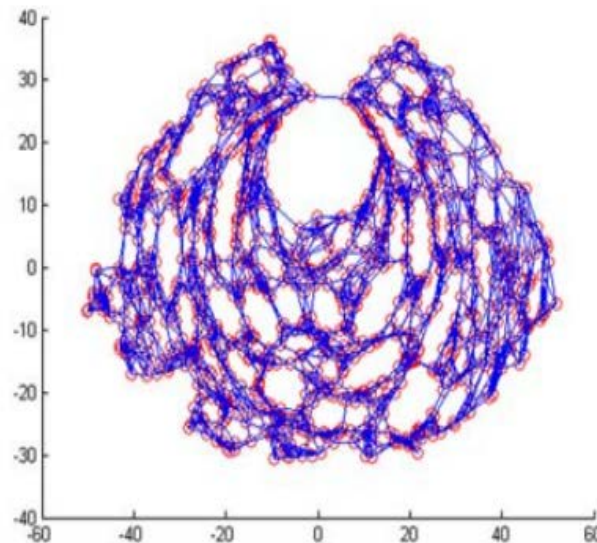


Short Circuits & Neighbourhood selection

neighbourhood size

too big: short-circuit errors

too small: isolated patches



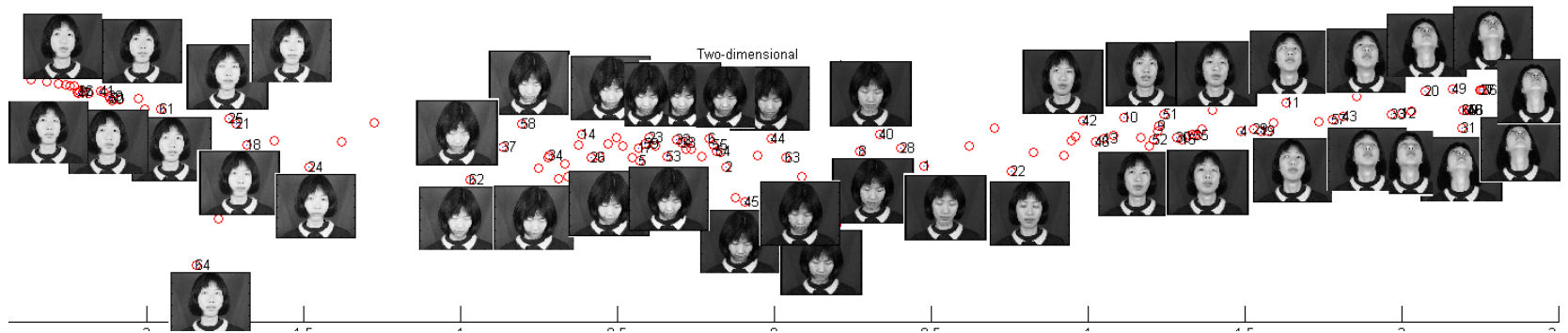
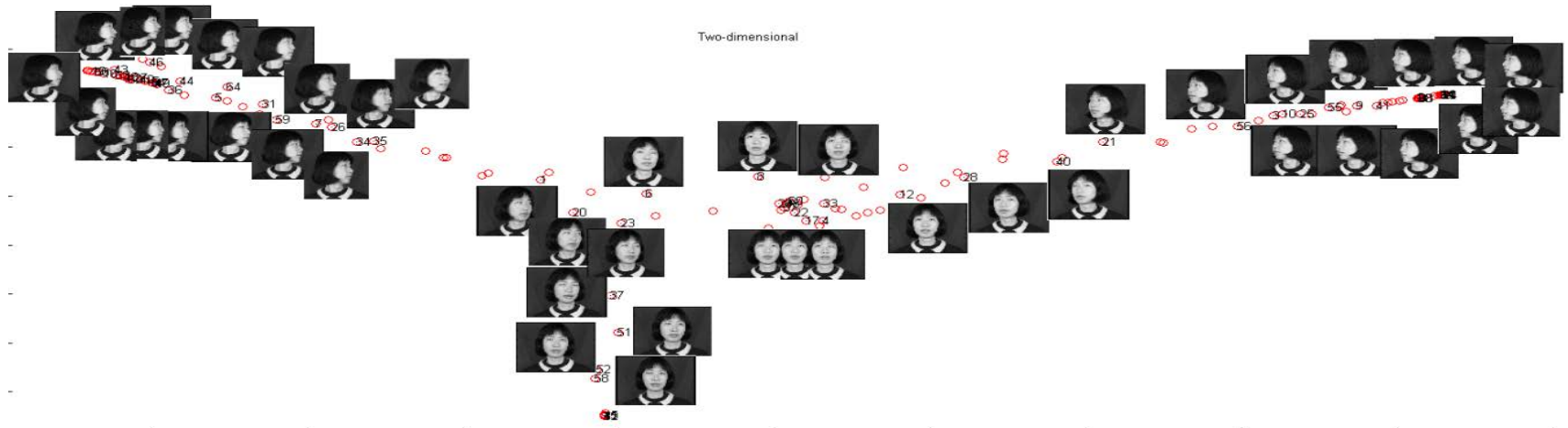
[saxena, gupta mukerjee 04]

Head Rotation Motion

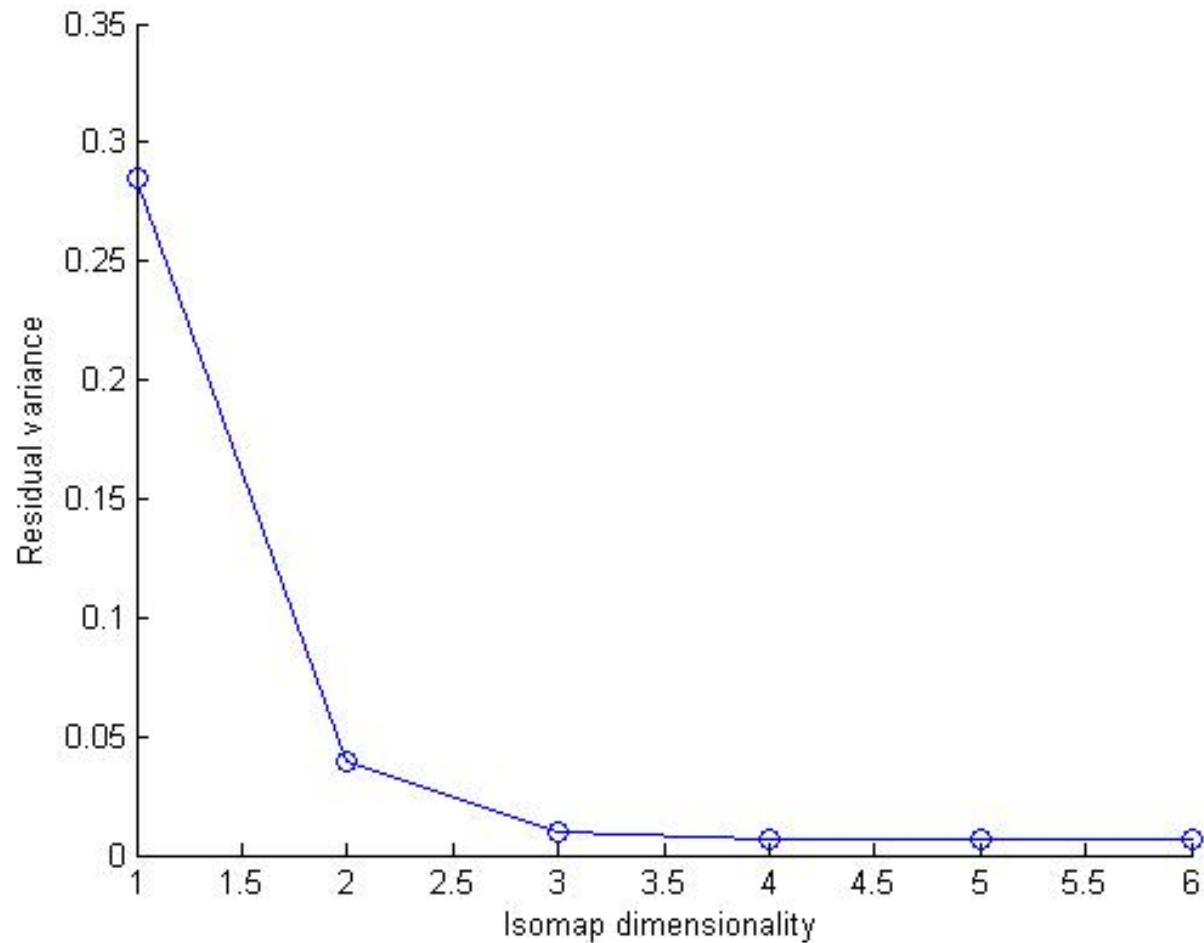


Head Rotation Motion-1

Results



Head Rotation Motion-1 Results



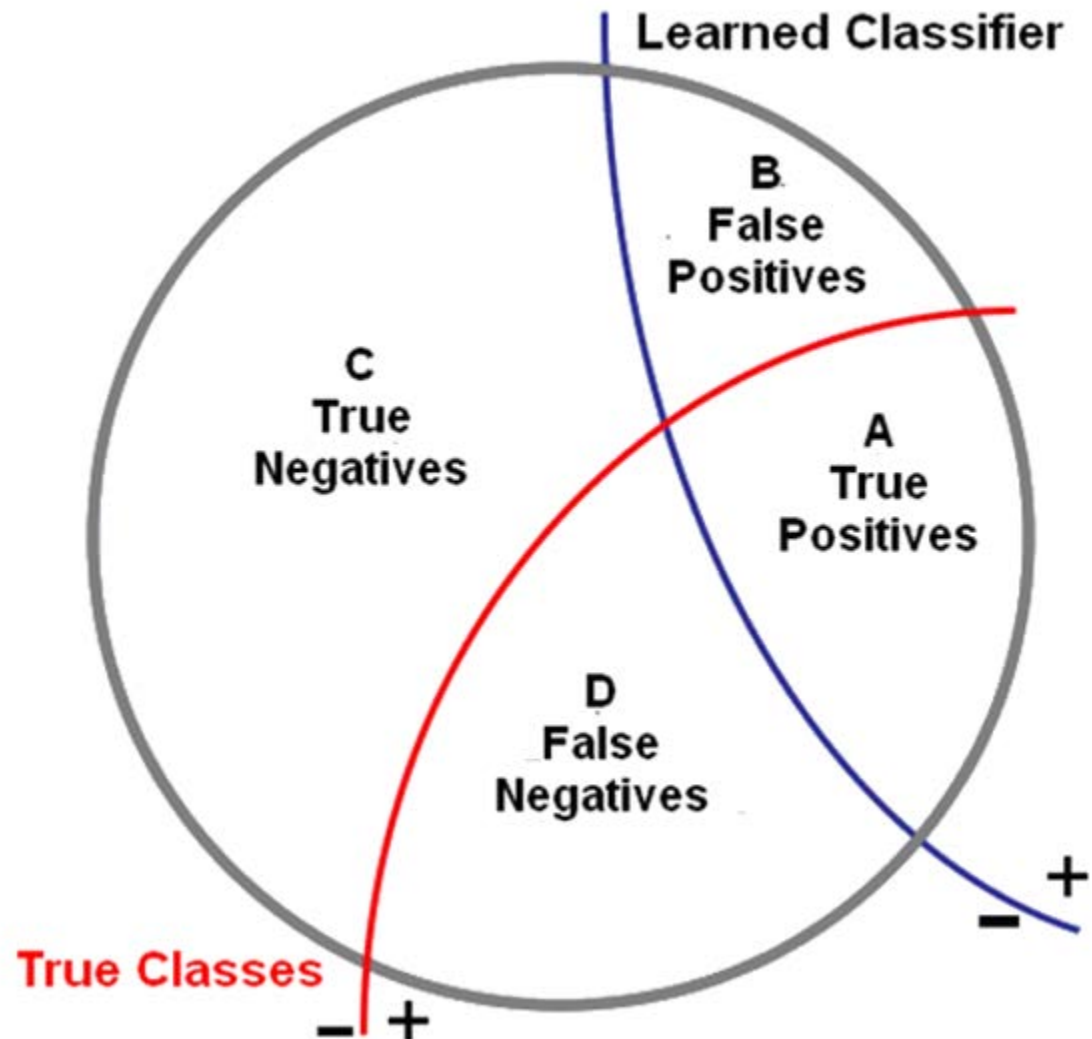
Binary classifier: Precision vs Recall

Precision:

$\text{TP} / \text{Retrieved Positives}$

Recall:

$\text{TP} / \text{Actual Positives}$



Information Theory

Building Expectations

Twenty Questions

Knower: thinks of object (point in a probability space)

Guesser: asks knower to evaluate random variables

Stupid approach:

Guesser: Is it my left big toe?

Knower: No.

Guesser: Is it Valmiki?

Knower: No.

Guesser: Is it Aunt Lakshmi?

...

Expectations & Surprisal

Turn the key: expectation: lock will open

Exam paper showing: could be 100, could be zero.

random variable: function from set of marks
to real interval $[0,1]$

Interestingness \propto unpredictability

$$\text{surprisal (r.v. = } x) = -\log_2 p(x)$$

$$= 0 \text{ when } p(x) = 1$$

$$= 1 \text{ when } p(x) = \frac{1}{2}$$

$$= \infty \text{ when } p(x) = 0$$

Expectations in data

A: 00010001000100010001... 0001000100010001000100010001

B: 01110100110100100110... 1010111010111011000101100010

C: 00011000001010100000... 0010001000010000001000110000

Structure in data → easy to remember

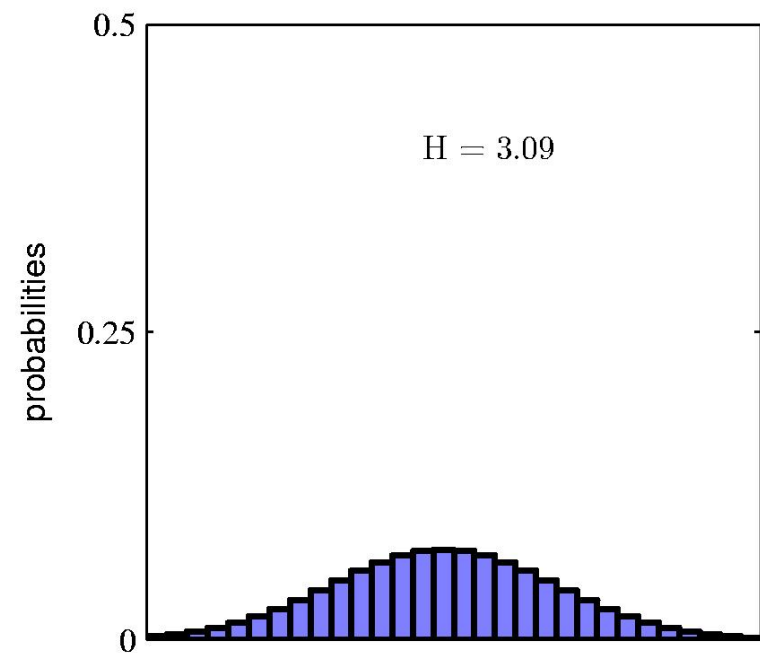
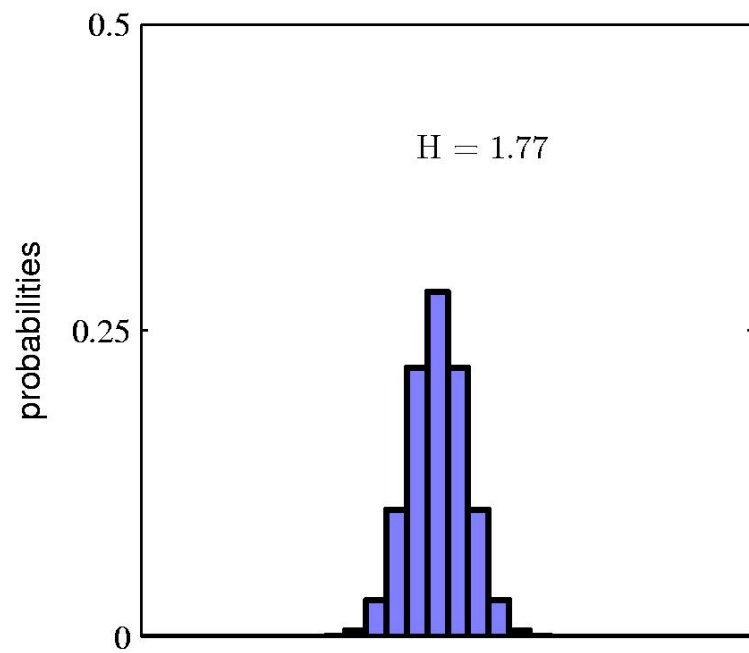
Entropy

$$H[x] = - \sum_x p(x) \log_2 p(x)$$

Used in

- coding theory
- statistical physics
- machine learning

Entropy



Entropy

In how many ways can N identical objects be allocated M bins?

$$W = \frac{N!}{\prod_i n_i!}$$

$$H = \frac{1}{N} \ln W \simeq - \lim_{N \rightarrow \infty} \sum_i \left(\frac{n_i}{N} \right) \ln \left(\frac{n_i}{N} \right) = - \sum_i p_i \ln p_i$$

Entropy maximized when $\forall i : p_i = \frac{1}{M}$

Entropy in Coding theory

x discrete with 8 possible states; how many bits to transmit the state of x?

All states equally likely

$$H[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits.}$$

Coding theory

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
code	0	10	110	1110	111100	111101	111110	111111

$$\begin{aligned} H[x] &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{16} \log_2 \frac{1}{16} - \frac{4}{64} \log_2 \frac{1}{64} \\ &= 2 \text{ bits} \end{aligned}$$

$$\begin{aligned} \text{average code length} &= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + 4 \times \frac{1}{64} \times 6 \\ &= 2 \text{ bits} \end{aligned}$$

Entropy in Twenty Questions

Intuitively : try to ask q whose answer is 50-50

Is the first letter between A and M?

question entropy = $p(Y)\log p(Y) + p(N)\log p(N)$

For both answers equiprobable:

$$\text{entropy} = -\frac{1}{2} * \log_2(\frac{1}{2}) - \frac{1}{2} * \log_2(\frac{1}{2}) = 1.0$$

For $P(Y)=1/1028$

$$\text{entropy} = -\frac{1}{1028} * -10 - \text{eps} = 0.01$$

Information Theory for Language

Shannon entropy

Shannon Entropy

- Predict the next word/letter, given $(n-1)$ previous letters or words : $F_n = \text{entropy} = \text{SUM}_i (p_i \log p_i)$
- probabilities p_i (of n -grams) from corpus:
 - F_0 (only alphabet) = $\log_2 27$ = 4.76 bits per letter
 - F_1 (1-gram frequencies p_i) = 4.03 bits
 - F_2 (bigram frequencies) = 3.32 bits
 - F_3 (trigrams) = 3.1 bits
 - F_{word} = 2.62 bits
(avg word entropy = 11.8 bits per 4.5 letter word)

Shannon Entropy : Human

- Ask human to guess the next letter:

THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
----ROO-----NOT-V-----I-----SM----OBL---

READING LAMP ON THE DESK SHED GLOW ON
REA-----O-----D----SHED-OLD--O-

POLISHED WOOD BUT LESS ON THE SHABBY RED CARPET
P-L-S-----O---BU--L-S-O-----SH-----RE-C-----

- 69% guessed on 1st attempt [“-” = 1st attempt]

Shannon Entropy : Human

- Count number of attempts:

T H E R E I S N O R E V E R S E O N A M O T O R C Y C L E A
1 1 1 5 1 1 2 1 1 2 1 1 1 5 1 1 7 1 1 1 2 1 3 2 1 2 2 7 1 1 1 1 4 1 1 1 1 1 3 1
F R I E N D O F M I N E F O U N D T H I S O U T
8 6 1 3 1 1 1 1 1 1 1 1 1 1 1 6 2 1 1 1 1 1 1 2 1 1 1 1 1 1
• R A T H E R D R A M A T I C A L L Y T H E O T H E R D A Y
4 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 1 1 1 1 1

Claude E. Shannon. "Prediction and Entropy of Printed English", *Bell System Technical Journal* 30:50-64. 1951.