

# Inteligencia Artificial

## Introducción

Mario González

## Sistemas que actúan como humanos

- El objetivo es construir un sistema que pase por humano (Test de Touring)
- Capacidades necesarias:
  - Procesamiento de lenguaje natural
  - Representación del conocimiento
  - Razonamiento
  - Aprendizaje

## El cerebro como máquina

- La “hipótesis asombrosa” de Francis Crick (1916–2004) :
  - Nuestros pensamientos, sensaciones, alegrías, dolores consisten enteramente de **actividad fisiológica en los tejidos del cerebro**.
- La conciencia es un **producto biológico** natural, tan vacía de elementos sobrenaturales como la digestión o la circulación de la sangre.

## Que es la inteligencia

- Son los humanos inteligentes?
- Son los perros inteligentes?
- Una lombriz?
- Una ameba?
- **Es una computadora inteligente?**
- **Un smartphone?**
- **Un robot?**
- La inteligencia es la capacidad de entender, asimilar, elaborar información y utilizarla para [resolver problemas](#).
- **Inteligencia: Propriocepción+Interacción con el medio**



## Que es la inteligencia artificial

- Es un área de la ciencia bastante nueva (Touring 1956)
- Su objetivo es replicar las capacidades que consideramos inteligentes
- Involucra áreas genéricas: aprendizaje, percepción, resolución de problemas
- Involucra áreas específicas: jugar al ajedrez, diagnosticar enfermedades, conducir coches

## Definicion de la IA

- Sistemas que actúan como humanos
- Sistemas que piensan como humanos
- Sistemas que actúan razonalmente
- Sistemas que piensan razonalmente

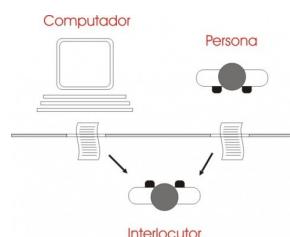
## Es la IA posible

- La posibilidad de la inteligencia artificial plantea problemas filosóficos complejos
- Las máquinas pensantes poseen conciencia?
- Habitación china (Searle, 1980)
- Es la inteligencia una propiedad emergente de los elementos biológicos que la producen?

## Sistemas que actúan razonalmente

- Conseguir unos objetivos dadas unas creencias
- El ejemplo es el agente
  - Un agente percibe y actúa, según el entorno en que está situado

## Test de Touring



## Sistemas que piensan razonalmente

- El pensamiento racional se fundamenta en la lógica
- La lógica formal es la base de los programas inteligentes
- Obstáculos:
  - Es difícil formalizar el conocimiento
  - La capacidad teórica de la lógica es difícil de aplicar en la práctica

## La habitación china (Searle)



## Bases de la IA

- **Filosofía.** Debate sobre la posibilidad de una inteligencia mecánica:
  - Descartes (1596–1650), Leibniz (1646–1716): la mente está ligada al mundo físico.
  - Hume (1779), Russell: el conocimiento es fruto de la percepción, se adquiere por la experiencia (**inducción**) y está representado por teorías lógicas.
- Darwin (1859): anulación del “*de la nada, nada proviene*” a través de la teoría de la evolución por selección natural.

## Áreas de trabajo de la IA

### • Áreas Básicas

- Representación del conocimiento
- Resolución de problemas, Búsqueda

### • Áreas Específicas

- Planificación de tareas
- Tratamiento del Lenguaje Natural
- Razonamiento Automático
- Sistemas Basados en el Conocimiento
- Percepción
- Aprendizaje Automático
- Agentes autónomos

## Aparatos Electrónicos



## Interfaces Inteligentes Recomendación Personalización



- Ambientes inteligentes, computación ubicua, algoritmos de recomendación.

## Bases de la IA

- **Matemáticas.** Las bases filosóficas necesitan reglas formales:
  - Boole (1815–1864), Frege (1848–1925): fundamentos de la lógica matemática.
  - Gödel (1906–1978), Turing (1912–1954): límites de lo computable (teorema de incompletitud).
  - Fermat (1601–1665), Bernoulli (1700–1782), Bayes (1702–1761): probabilidad, razonamiento probabilístico.

## Robots autónomos



- **Tareas complejas:** visión, planificación, coordinación, tiempo real, etc.

## Bases de la IA

- **Economía**
  - ¿Como debemos tomar decisiones ...
  - que nos beneficien?
  - en contra de competidores?
  - cuando el beneficio no es inmediato?
  - Teoría de la decisión/Teoría de juegos/Investigación operativa
- **Neurociencia**
  - ¿Como procesa el cerebro la información?
  - Neuronas/Especialización del cerebro
- **Psicología**
  - ¿Como piensan y actúan las personas?
  - Psicología cognitiva/ciencias cognitivas: Teorías sobre la conducta, bases del comportamiento racional

## Reconocimiento de imágenes



## Trabajo en grupo

- Ampliar y exponer en clase los siguientes temas (elegir uno por grupo):

- 1.Reconocimiento de objetos (visión por computadora)
- 2.Reconocimiento facial
- 3.Reconocimiento de huellas dactilares
- 4.Reconocimiento de escritura a mano (reconocimiento de firmas)
- 5.Ambientes inteligentes
- 6.Computación ubicua
- 7.Electrodomésticos inteligentes
- 8.Algoritmos de recomendación
- 9.Inteligencia artificial y videojuegos
- 10.Inteligencia artificial y robótica

## Bases de la IA

- **Ingeniería computacional:**
  - Para la existencia de la IA es necesario un mecanismo para soportarlo (hardware).
  - También son necesarias herramientas para desarrollar programas de IA.
- **Teoría de control/Cibernética:** construcción de sistemas autónomos
- **Lingüística computacional:**
  - Chomsky: representación del conocimiento, gramática de la lengua.

## Juegos



- Coordinación, cooperación, aprendizaje, adaptación, estrategia, búsqueda de caminos.

## Resolución de Problemas

- La resolución de problemas es una capacidad que consideramos inteligente
- Somos capaces de resolver problemas muy diferentes
  - Encontrar el camino en un laberinto
  - Resolver un crucigrama
  - Jugar a un juego
  - Diagnosticar una enfermedad
  - Decidir si invertir en bolsa
- El objetivo es que un programa también sea capaz de resolverlos

## Resolución de Problemas

- Deseamos definir cualquier tipo de problema de manera que se pueda resolver automáticamente
- **Necesitamos:**
  - Una representación común para todos los problemas
  - Algoritmos que usen alguna estrategia para resolver problemas definidos en esa representación común

## Definición de un Problema

- Si abstraemos los elementos de un problema podemos identificar:
  - Un punto de partida
  - Un objetivo a alcanzar
  - Acciones a nuestra disposición para resolver el problema
  - Restricciones sobre el objetivo
  - Elementos que son relevantes en el problema definidos por el tipo de dominio

## Representación de problemas

- Existen diferentes formas de representar problemas para resolverlos de manera automática
- Representaciones generales
  - Espacio de estados: un problema se divide en un conjunto de pasos de resolución desde el inicio hasta el objetivo
  - Reducción a subproblemas: un problema se puede descomponer en una jerarquía de subproblemas
- Representaciones para problemas específicos
  - Resolución de juegos
  - Satisfacción de restricciones

## Representación de problemas: Estados

- Podemos definir un problema por los elementos que intervienen y sus relaciones
- En cada instante de la resolución de un problema esos elementos tendrán unas características y relaciones específicas
- Denominaremos Estado a la representación de los elementos que describen el problema en un momento
- Distinguiremos dos estados especiales el Estado Inicial (punto de partida) y el Estado Final (objetivo del problema)

## Modificación del estado: operadores

- Para poder movernos entre los diferentes estados necesitamos operadores de transformación
- **Operador:** Función de transformación sobre la representación de un estado que lo convierte en otro estado
- Los operadores definen una relación de accesibilidad entre estados
- Representación de un operador:
  - Condiciones de aplicabilidad
  - Función de transformación
- ¿Qué operadores? ¿Cuántos? ¿Qué granularidad?

## Espacio de estados

- Los estados y su relación de accesibilidad conforman lo que se denomina espacio de estados
- Representa todos los caminos que hay entre todos los estados posibles de un problema
- Podría asimilarse con un mapa de caminos de un problema
- La solución de nuestro problema está dentro de ese mapa

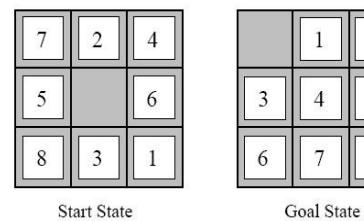
## Solución de un problema en Espacio de Estados

- **Solución:** Secuencia de pasos que llevan del estado inicial al final (secuencia de operadores) o también el estado final
- **Tipos de solución:** una cualquiera, la mejor, todas
- **Coste de una solución:** Gasto en recursos de la aplicación de los operadores a los estados. Puede ser importante o no según el problema y que tipo de solución busquemos

## Descripción de un problema en Espacio de Estados

- Definir el conjunto de estados del problema (explícita o implícitamente)
- Especificar el estado inicial
- Especificar el estado final o las condiciones que cumple
- Especificar los operadores de cambio de estado (condiciones de aplicabilidad y función de transformación)
- Especificar el tipo de solución:
  - La secuencia de operadores o el estado final
  - Una solución cualquiera, la mejor (definición de coste)

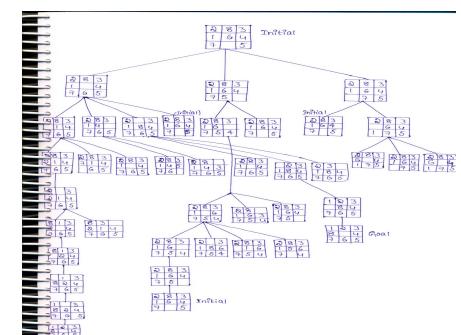
## Ejemplo: 8 puzzle



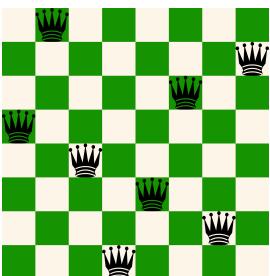
## Ejemplo: 8 puzzle

- **Espacio de estados:** Configuraciones de 8 fichas en el tablero
- **Estado inicial:** Cualquier configuración
- **Estado final:** Fichas en orden específico
- **Operadores:** Mover hueco
- **Condiciones:** El movimiento está dentro del tablero
- **Transformación:** Intercambio entre el hueco y la ficha en la posición del movimiento
- **Solución:** Qué pasos + El menor número

## Ejemplo: 8 puzzle



## Ejemplo: N reinas



<http://www.cosc.canterbury.ac.nz/mukundan/dsal/NQP.html>

## Algoritmo Básico

- El espacio de estados puede ser infinito
- Es necesaria una aproximación diferente para buscar y recorrer árboles y grafos (no podemos tener la estructura en memoria)
- La estructura la construimos a medida que hacemos la búsqueda

## Tipos de algoritmos

- Algoritmos de búsqueda ciega
  - No tienen en cuenta el coste de la solución en la búsqueda
  - Su funcionamiento es sistemático, siguen un orden de visitas y generación de nodos establecido por la estructura del espacio de búsqueda
  - Anchura prioritaria, Profundidad prioritaria, Profundidad iterativa
- Algoritmos de búsqueda heurística
  - Utilizan una estimación del coste de la solución para guiar la búsqueda
  - No siempre garantizan el óptimo, ni una solución
  - Hill-climbing, Branch and Bound, A\*, IDA\*

## Ejemplo: N reinas

- **Espacio de estados:** Configuraciones de 0 a n reinas en el tablero con sólo una por fila, columna y diagonal
- **Estado inicial:** Configuración sin reinas en el tablero
- **Estado final:** Configuración en la que ninguna reina se mata entre si
- **Operadores:** Colocar una reina en una fila, columna y diagonal
  - Condiciones: La reina no es matada por ninguna ya colocada
  - Transformación: Colocar una reina mas en el tablero en una fila y columna determinada
- **Solución:** Una solución, pero no nos importan los pasos

## Algoritmo Básico

**Función:** Búsqueda en espacio de estados()  
**Datos:** El estado inicial  
**Resultado:** Una solución

Seleccionar el primer estado como el estado actual  
**mientras** estado actual = estado final **hacer**  
    Generar y guardar sucesores del estado actual (expansión)  
    Escoger el siguiente estado entre los pendientes (selección)  
**fin**

- La selección del siguiente nodo determinará el tipo de búsqueda (orden de selección o expansión)
- Es necesario definir un orden entre los sucesores de un nodo (orden de generación)

## Búsqueda en Anchura Prioritaria

- Los nodos se visitan y generan por niveles
- La estructura para los nodos abiertos es una cola (FIFO)
- Un nodo es visitado cuando todos los nodos de los niveles superiores y sus hermanos precedentes han sido visitados
- Características:
  - **Complejidad:** El algoritmo siempre encuentra una solución
  - **Complejidad temporal:** Exponencial respecto al factor de ramificación y la profundidad de la solución  $O(r^p)$
  - **Complejidad espacial:** Exponencial respecto al factor de ramificación y la profundidad de la solución  $O(r^p)$
  - **Optimalidad:** La solución que se encuentra es óptima en número de niveles desde la raíz

## Búsqueda en el espacio de estados

- La resolución de un problema con una representación dada pasa por explorar el espacio de estados
- Partimos del estado inicial evaluando cada paso hasta encontrar un estado final
- En el caso peor exploraremos todos los posibles caminos entre el estado inicial del problema hasta llegar al estado final

## Estructura del espacio de estados

- Primero definiremos una representación del espacio de estados para poder implementar algoritmos que busquen soluciones
  - **Estructuras de datos:** Árboles y Grafos
  - **Estados** = Nodos
  - **Operadores** = Arcos entre nodos (dirigidos)
  - **Árboles:** Solo un camino lleva a un nodo
  - **Grafos:** Varios caminos pueden llevar a un nodo

## Características de los algoritmos

- Características:
  - **Complejidad:** ¿Encontrará una solución?
  - **Complejidad temporal:** ¿Cuanto tardará?
  - **Complejidad espacial:** ¿Cuanta memoria gastará?
  - **Optimalidad:** ¿Encontrará la solución óptima?

## Algoritmo Básico

- **Nodos abiertos:** Estados generados pero aún no visitados
- **Nodos cerrados:** Estados visitados y que ya se han expandido
- Tendremos una estructura para almacenar los nodos abiertos
- Las diferentes políticas de inserción en la estructura determinarán el tipo de búsqueda

## Búsqueda en Profundidad Prioritaria

- Los nodos se visitan y generan buscando los nodos a mayor profundidad y retrocediendo cuando no se encuentran nodos sucesores
- La estructura para los nodos abiertos es una pila (LIFO)
- Para garantizar que el algoritmo acaba debe imponerse un límite en la profundidad de exploración
- Características
  - **Complejidad:** El algoritmo encuentra una solución si se impone un límite de profundidad y existe una solución dentro de ese límite
  - **Complejidad temporal:** Exponencial respecto al factor de ramificación y la profundidad del límite de exploración  $O(r^p)$
  - **Complejidad espacial:** En el caso de no controlar los nodos repetidos el coste es lineal respecto al factor de ramificación y el límite de profundidad  $O(rp)$ . Si tratamos repetidos el coste es igual que en anchura. Si la implementación es recursiva el coste es  $O(p)$
  - **Optimalidad:** No se garantiza que la solución sea óptima

## Búsqueda en Profundidad Limitada

**Procedimiento:** Búsqueda en profundidad limitada (límite: entero)  
Est\_abiertos.insertar(Estado inicial)  
Actual  $\leftarrow$  Est\_abiertos.primer()

**mientras** no es\_final?(Actual) y no Est\_abiertos.vacia?() **hacer**  
    Est\_abiertos.borrar\_primer()  
    Est\_cerrados.insertar(Actual)  
    **si** profundidad(Actual)  $\leq$  límite **entonces**  
        Hijos  $\leftarrow$  generar\_sucessores (Actual)  
        Hijos  $\leftarrow$  tratar\_repetidos (Hijos, Est\_cerrados, Est\_abiertos)  
        Est\_abiertos.insertar(Hijos)  
    **fin**  
    Actual  $\leftarrow$  Est\_abiertos.primer()

## Búsqueda en profundidad iterativa

```
Procedimiento: Búsqueda en profundidad iterativa (límite: entero)
prof ← 1
Actual ← Estado inicial
mientras no es_final?(Actual) y prof<limite hacer
    Est_abiertos.inicializar()
    Est_abiertos.insertar(Estado inicial)
    Actual ← Est_abiertos.primer()
    mientras no es_final?(Actual) y no Est_abiertos.vacia?() hacer
        Est_abiertos.borrar_primer()
        Est_cerrados.insertar(Actual)
        si profundidad(Actual) ≤ prof entonces
            Hijos ← generar_sucesores(Actual)
            Hijos ← tratar_repetidos(Hijos, Est_cerrados, Est_abiertos)
            Est_abiertos.insertar(Hijos)
        fin
        Actual← Est_abiertos.primer()
    fin
    prof ← prof+1
fin
```

## Profundidad Iterativa

- **Compleitud:** El algoritmo siempre encontrará la solución
- **Complejidad temporal:** La misma que la búsqueda en anchura. El regenerar el árbol en cada iteración solo añade un factor constante a la función de coste  $O(r^p)$
- **Complejidad espacial:** Igual que en la búsqueda en profundidad
- **Optimalidad:** La solución es óptima igual que en la búsqueda en anchura

# Breve introducción a Octave.

Manual de iniciación.

[https://docs.google.com/file/d/0B-dmEvN59Z2Z  
SFFBRHZGbjVRZFk/edit?usp=sharing](https://docs.google.com/file/d/0B-dmEvN59Z2ZSFFBRHZGbjVRZFk/edit?usp=sharing)

Un tutorial básico.

[https://docs.google.com/file/d/0B-dmEvN59Z2Z  
SFFBRHZGbjVRZFk/edit?usp=sharing](https://docs.google.com/file/d/0B-dmEvN59Z2ZSFFBRHZGbjVRZFk/edit?usp=sharing)

Un tutorial más completo.

[https://docs.google.com/file/d/0B-dmEvN59Z2Z  
dkpNMzVacWZ1U2M/edit?usp=sharing](https://docs.google.com/file/d/0B-dmEvN59Z2ZdkpNMzVacWZ1U2M/edit?usp=sharing)

Introducción informal.

[https://docs.google.com/file/d/0B-dmEvN59Z2Z  
RTBiNjZDVHBSSUU/edit?usp=sharing](https://docs.google.com/file/d/0B-dmEvN59Z2ZRTBiNjZDVHBSSUU/edit?usp=sharing)  
<http://iimyo.forja.rediris.es/matlab/index.html>

## Introducción a los Algoritmos Genéticos (AG)

- Encuentran soluciones “buenas” en un tiempo razonable
- Representan las posibles soluciones de un problema individuos
- Los algoritmos genéticos generan una **población de individuos (posibles soluciones)** y los hacen **evolucionar** para obtener individuos con genes mas aptos (**mejores soluciones**)

## Descripción

## Inspiración

- En la naturaleza los individuos compiten por los recursos del medio ambiente
- Algunos son mejores que otros, esos son los que tienen mas posibilidades de sobrevivir y propagar su material genético
- En un AG los individuos son evaluados según una función llamada **Fitness function** y los individuos con mejores genes son los que pasarán a la **próxima generación** (iteración)

## Representación de los genes de los individuos

- En la naturaleza:

Una secuencia de nucleótidos  
A C C T G C A G G

- En un algoritmo genético (ejemplos):  
Un valor numérico (178) expresado en binario  
1 0 1 1 0 0 1 0

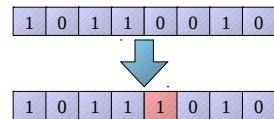
Una secuencia de movimientos  
Adelante Izquierda Izquierda Atras

## Pasos de un algoritmo genético

1. Generar una población de **n** individuos aleatorios
2. Evaluar a todos los individuos según la función de aptitud (*fitness function*)
3. Generar nuevos individuos utilizando funciones como Mutar, Cruzar (*crossover*), Variar, etc.
4. Seleccionar a los individuos que formarán la próxima generación. (Seleccionar a los hijos (*offsprings*) o seleccionar a los n mejores)
5. Volver a 2 hasta que se encuentre un valor predefinido o se hallan cumplido una cantidad predeterminada de iteraciones

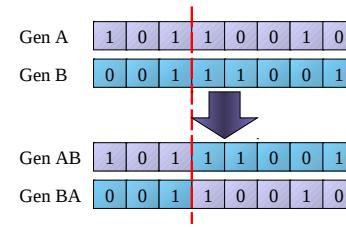
## Mutación

1. Se decide con alguna probabilidad si se aplica la mutación o no
2. De aplicarse se selecciona con alguna probabilidad el bit (o ítem del arreglo) a mutar
3. De no ser un bit (que sería negado) se elige con alguna probabilidad en que muta



## Cruza

1. Se seleccionan las parejas a ser cruzadas
2. Se selecciona la posición del cruce aleatoriamente



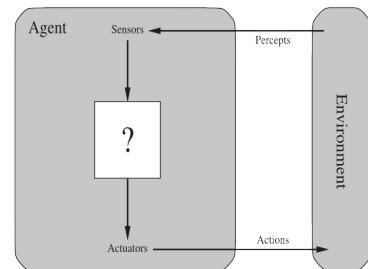
## Selección de parejas

- En función de la función de aptitud (fitness)
- Se utiliza una política de cruce (ej. **selección de individuos por ruleta**)

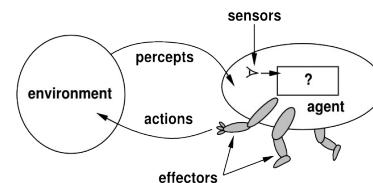
## ¿Cuál robot es más inteligente?



## Modelos y arquitecturas de agentes

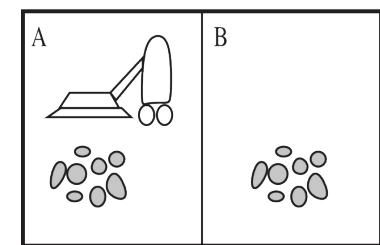


## Modelos y arquitecturas de agentes



Agentes interactúan con el ambiente a través de sensores y actuadores.

## Modelos y arquitecturas de agentes



Un agente aspiradora con un mundo de sólo dos lugares.

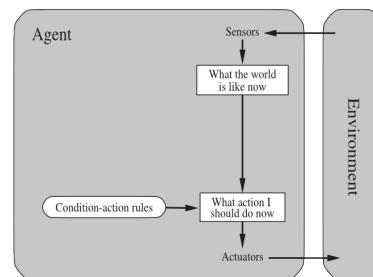
## ¿Cómo se diseña un agente?

- Como entidad que interactúa con su entorno:
  - Cómo percibir el entorno
  - Cómo representar el entorno
  - Cómo definir los actuadores
- Utilizando la definición de Newell:
  - Cómo representar los objetivos del agente
  - Cómo describir la toma de decisiones del agente
  - Cómo representar el conocimiento
- Y si consideramos un SMA, según Ferber, además:
  - Leyes que controlan el entorno
  - Objetos ubicados
  - Coordinación de los agentes
  - Acciones permitidas

## Agentes que mantienen un estado interno (modelo del mundo)

- La percepción actual se interpreta a partir del estado anterior utilizando información sobre:
  - cómo evoluciona el mundo independientemente del agente
  - cómo influyen en el mundo las acciones del agente

## Agentes reactivos

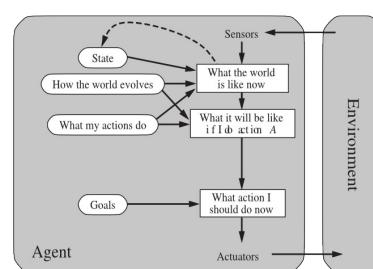


## Agentes reactivos

- El proceso del agente es un ciclo percepción-acción (estímulo/respuesta)
  - Reacciona a la evolución del entorno
- No hay una representación explícita del entorno, de los otros agentes, sus capacidades, etc.
- Las decisiones no tienen en cuenta ni el pasado (no hay historia) ni el futuro (no hay planificación)
- Emerge comportamiento inteligente de la interacción de elementos muy simples

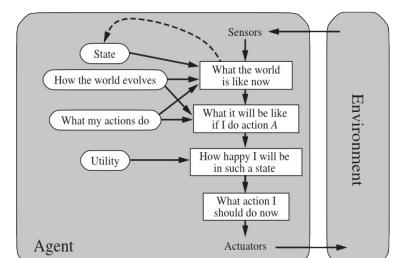
## Agentes basados en objetivos

## Agentes basado en objetivos



## Agentes basados en utilidad

- A veces no es posible tomar una decisión únicamente a partir del estado actual del mundo, ej. Al llegar a un cruce el taxi puede realizar distintas acciones
- Un agente basado en objetivos utiliza una descripción de las metas a alcanzar que le sirve para escoger entre las distintas acciones posibles, ej. el destino del pasajero
- La toma de decisiones basada en objetivos = la toma de decisiones basada en reglas condición-acción:
  - Tiene en cuenta el estado en el futuro
  - Es más flexible
- La búsqueda y la planificación son dos subáreas de la IA dedicadas a la resolución de problemas utilizando objetivos



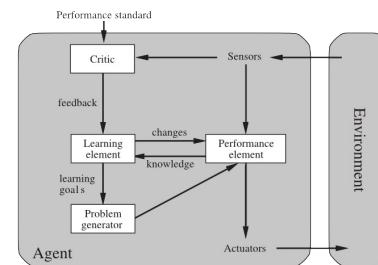
## Agentes dirigidos por utilidad

- Los objetivos únicamente no bastan para asegurar un comportamiento ideal, ej. diferentes secuencias de acciones pueden llevar al destino que el pasajero desea, pero algunas son más rápidas, seguras o baratas que otras
- Un agente basado en la utilidad utiliza un criterio para estimar el grado de satisfacción de un estado para el agente que le sirve para escoger entre distintas acciones válidas
- La utilidad es una función que asocia a un estado un número real, y debe considerar situaciones problemáticas, ej. compromiso adecuado a adoptar en caso de objetivos en conflicto

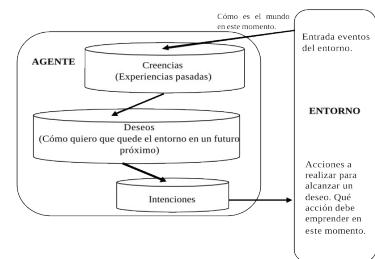
## Tipos de ambiente

	Ajedrez	Conducir un taxi
Observable	Totalmente	Parcial
Determinista vs. estocástico	Determinista (estratégico*)	Estocástico
Episódico vs. secuencial	Secuencial	Secuencial
Estático vs. dinámico	Semi-estático	Dinámico
Discreto vs. continuo	Discreto	Continuo
Individual vs multiagente	Multiagente	Multiagente

## Agentes que aprenden



## Arquitectura BDI (Belief, Desire, Intention)



## Recursos

- MASON

<http://cs.gmu.edu/~eclab/projects/mason/>

- REPAST

[http://repast.sourceforge.net/repast\\_3/tutorials.html#thirdparty](http://repast.sourceforge.net/repast_3/tutorials.html#thirdparty)

- NETLOGO

<http://ccl.northwestern.edu/netlogo/>

## Extracción del conocimiento

### Extracción del conocimiento

- Concepto de minería de datos o data mining:
  - *Proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos* - Witten & Frank
- Nuestro objetivo es pues convertir datos en conocimiento.
- Otras formas de llamarlo:
  - Intelligent Data Analysis
  - KDD – Knowledge Discovery in Databases

3

4

## Modelos

- Nuestro objetivo es analizar los datos para extraer conocimiento.
- Este conocimiento puede ser en forma de reglas, relaciones, asociaciones, patrones o modelos.

7

8

## ESQUEMA GENERAL DEL PROCESO



11

12

$$\text{CONOCIMIENTO} = \text{€}$$

### KDD

- Normalmente nos referimos a KDD como todo el proceso de extracción del conocimiento, donde la minería de datos sería una de las fases.



5

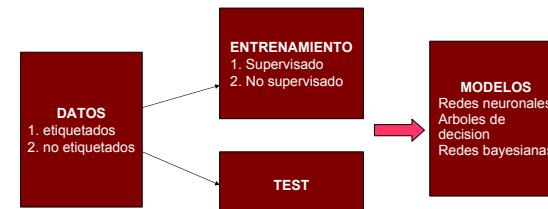
## ESQUEMA GENERAL DEL PROCESO

### Datos

- ¿Cómo nos llegan esos datos?
  - Datos estructurados, semiestructurados o no estructurados.
  - Bases de datos relacionales, html, xml, texto plano...
  - O incluso podemos tener datos multimedia

6

## ESQUEMA GENERAL DEL PROCESO



9

10

## ¿Para qué sirve todo esto? Aplicaciones

## EXTRACCIÓN DE CONOCIMIENTO DE GRANDES BASES DE DATOS

- **DATA WAREHOUSE** - Es una técnica para consolidar y administrar datos de variadas fuentes con el propósito de responder preguntas de negocios y tomar decisiones (bases de datos de empleados de una empresa)
- **DATA MINING** - En este sentido un sistema de minería de datos es una tecnología de soporte para usuario final, cuyo objetivo es extraer conocimiento útil y utilizable a partir de la información contenida en las bases de datos de las empresas.

14

## EXTRACCIÓN DE CONOCIMIENTO DE GRANDES BASES DE DATOS

- Los objetivos:
  - analizar factores de influencia en determinados procesos,
  - predecir variables o comportamientos futuros
  - segmentar o agrupar ítems similares,
  - además de obtener secuencias de eventos que provocan comportamientos específicos.

15

## INDUSTRIA DEL ENTRETENIMIENTO

- HSS: Predicción del posible éxito de un grupo musical
  - <http://www.hitsongscience.com/>
  - Norah Jones, Maroon 5.
  - Aquí hay una forma de ganar mucho dinero.
- Pandora, Last-fm
- <http://www.music-map.com/>
- Composición musical
- Predicción de los ganadores del festival de Sundance

19

## ROBOTS AUTÓNOMOS

- Conducción autónoma de vehículos
- Visión artificial

23

## NEGOCIOS

- **CRM** - la estrategia de negocio centrada en anticipar, conocer y satisfacer las necesidades y los deseos presentes y previsibles de los clientes.
- Predicción automatizada de tendencias y comportamientos – Marketing predictivo y personalizado.
- Nos regalan tarjetas en todos los sitios.

16

## OTROS

- Deportes - Predicción de lesiones antes de contratar a un futbolista – Redondo en el Milán.
- Policía - Detección y predicción de posibles policías corruptos

20

## CIENCIA

- **Astrofísica** - Ingentes cantidades de datos – Clasificación, predicción. [Proyecto telescopio MAGIC](#)
- **Diagnóstico en Psiquiatría y Medicina**: Buscar en qué se parecen los pacientes de alguna enfermedad, es decir, aprender por casos que ya han sido diagnosticados. [Proyecto del DSM IV](#)
- **Predicción en psicología** - proyecto de maltratadores.
- **Bioinformática (ADN)** - Ingentes cantidades de datos – Clasificación, predicción.

17

## RECONOCIMIENTO DE PATRONES

- Reconocimiento de voz
- Clasificación de documentos
- OCR (Reconocimiento Óptico de Caracteres)
  - Reconocimiento de escritura, firmas, talones, cheques...
- Reconocimiento de caras humanas
- Reconocimiento de huellas digitales o el iris, con aplicación en seguridad informática.

21

## APLICACIONES COMPUTACIONALES

- Aplicaciones que aprenden y mejoran con la experiencia: Mozilla – spam.
- **AUTO-PROGRAMACIÓN** de robots - Aprendizaje mediante redes neuronales (corrección de errores de un brazo robótico industrial)
- **WEB** - Motores de búsqueda (mindset de Yahoo), navegación personalizada (hiperenlaces personalizados), tiendas virtuales (Amazon)
- **SISTEMAS ADAPTATIVOS** (e-learning).

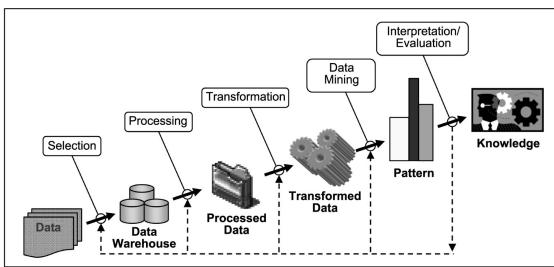
18

## LINGÜÍSTICA

- Traductores de inglés
  - ¿Cómo funciona Google translate?
- Comprensión de texto – resúmenes de prensa

22

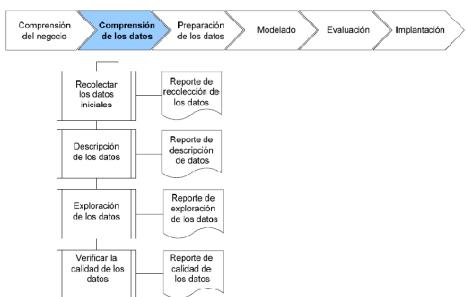
## Proceso KDD



## Los pasos del proceso de KDD

- Limpieza de datos:** Valores perdidos, datos ruidosos, y los datos inconsistentes
- Integración de datos:** la fusión de datos de múltiples "almacenes de datos"
- Selección de datos:** seleccionar los datos relevantes para el análisis
- Transformación de datos:** agregación (ventas diarias a semanales o mensuales) o generalización (calle a ciudad, edad a joven, de mediana edad a adultos mayores)
- Minería de datos:** aplicar métodos inteligentes para extraer patrones
- Evaluación de patrones:** patrones interesantes deben rebatir creencias o confirmar una hipótesis que el usuario desea validar
- Presentación del conocimiento:** técnicas de visualización y representación para comunicar el conocimiento extraído a los usuarios

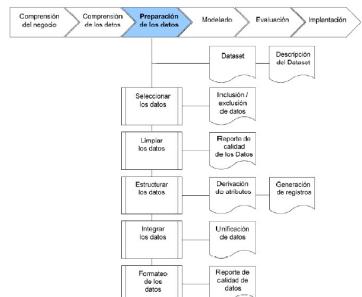
## Comprensión de los datos



## Preproceso de los datos (Tareas)

- Data cleaning** o limpieza de datos, es el proceso orientado a eliminar datos con ruido o incorrectos
- Data integration**, trata de integrar diferentes fuentes de datos en un almacén coherente y homogéneo como un data warehouse o un data cube.
- Data transformation**, o transformación de los datos como, por ejemplo, una normalización.
- Data reduction**, o reducción de los datos, orientado a reducir el tamaño de los datos mediante la agregación y/o eliminación de características redundantes, o clustering.

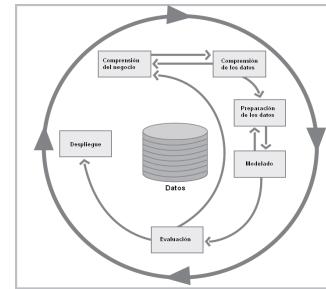
## Preparación de los datos



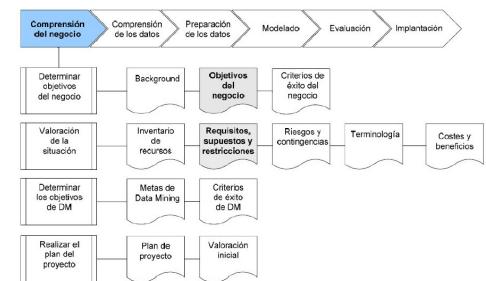
## Data cleaning

- Los datos del mundo real suelen presentarse en una forma incompleta, inconsistente y ruidosa, limpiaremos los datos
  - eliminando datos que faltan (sustituir por la media),
  - suavizando el efecto del ruido (binning, clustering),
  - eliminando datos fuera de rango (outliers)
  - corrigiendo inconsistencias.

## Cross Industry Standard Process for Data Mining (CRISP-DM)



## Comprensión del negocio



## Comprensión de los datos

### Recopilar datos

- Lista de los conjuntos de datos adquiridos (lugares, los métodos utilizados para la adquisición, problemas encontrados y las soluciones alcanzadas).

### Describir los datos

- Compruebe el volumen de datos y examinar sus propiedades macroscópicas.
- La accesibilidad y disponibilidad de los atributos. Los tipos de atributos, gama, correlaciones, las identidades.
- Comprender el significado de cada atributo y valor del atributo en términos de negocio.
- Para cada atributo, calcular estadísticas básicas (por ejemplo, la distribución, promedio, máximo, mínimo, desviación estándar, varianza, la moda, la asimetría).

## Conversión de datos

- Transformamos unos datos en otros equivalentes y así dejarlos listos para la minería
- Suavizado o eliminación del ruido (utilizamos binning, clustering, regresión, etc)
- Agregación: agregamos valores de atributos, e.g. agregamos ventas diarias en semanales y/o mensuales (i.e. ver los data cubes, en donde se usa típicamente este tipo de transformación).
- Generalización: mediante jerarquías de conceptos, sustituimos valores categóricos o numéricos por otros valores más abstractos (e.g. calle por ciudad, 30 por mediana edad).
- Normalización: escalamos el atributo a un cto. de valores apropiado según el caso
- Construcción de atributos: mediante el cual construimos nuevos atributos cuando esto es conveniente para el proceso de minería (area=altura x anchura).

## Normalización

### Min-max

$$v' = \frac{v - min_{\lambda}}{max_{\lambda} - min_{\lambda}} (new\_max_{\lambda} - new\_min_{\lambda}) + new\_min_{\lambda}$$

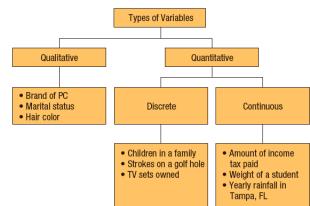
### Media cero (z normalisation)

$$v' = \frac{v - mean_{\lambda}}{stand\_dev_{\lambda}}$$

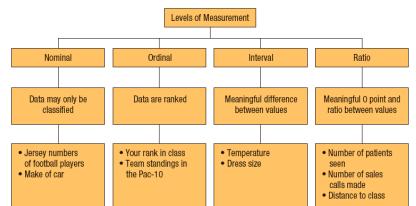
### Escalado decimal

$$v' = \frac{v}{10^j} \quad j \text{ es el menor entero tal que } max(|v'|) < 1 \\ \text{Rango: } -986 \text{ a } 917 \Rightarrow j=3, -986 \rightarrow -0.986, 917 \rightarrow 0.917$$

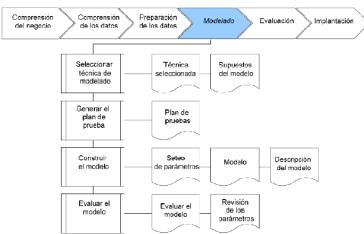
## Tipos de variables



## Tipos de medida



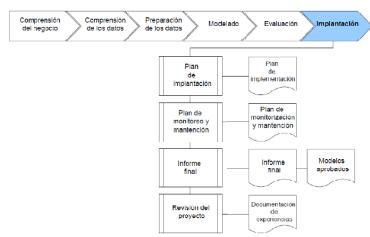
## Modelado



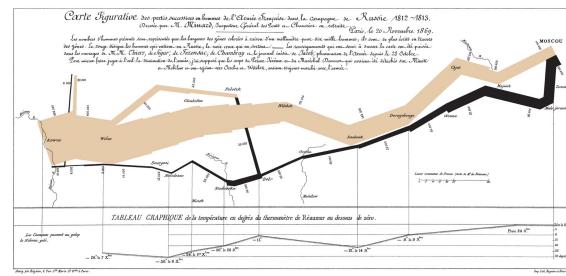
## Evaluación



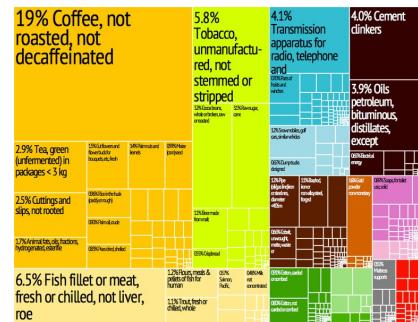
## Implementación



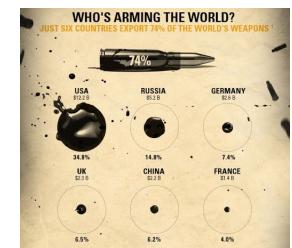
## Visualización



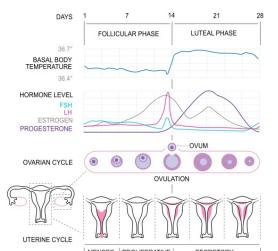
## Visualización



## Visualización

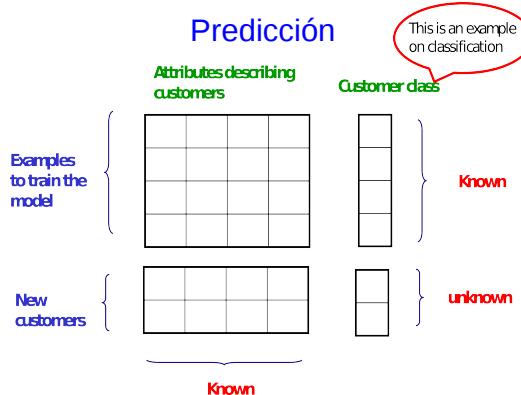


## Visualización



## Predicción

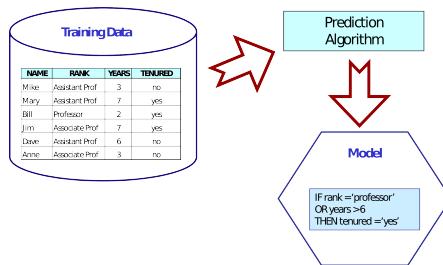
- Objetivo: predecir el valor de una variable dada (variable objetivo)
- Si la variable objetivo es:
  - Categórica: tenemos un problema de **clasificación**
  - Numérica: tenemos un problema de **regresión**
- Para cada registro en el conjunto de datos se determina el valor del atributo de clase
- Se construye un modelo basado en el conjunto de entrenamiento, a continuación, utiliza el modelo en la predicción de nuevos datos



## Paso 1: Construcción del modelo

- Se supone que cada muestra (registro, instancia o ejemplo) tiene un valor de destino: **atributo**
- El conjunto de muestras se utiliza para la construcción del modelo: **conjunto de entrenamiento**
- El modelo se representa como reglas de clasificación, árboles de decisión, o fórmulas matemáticas

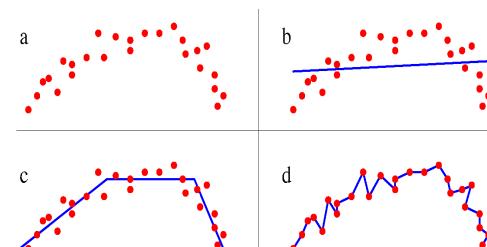
## Paso 1: Construcción del modelo



## Subconjuntos de datos

- Conjunto de entrenamiento - usado en la construcción del modelo
- Conjunto de prueba - utilizado en la validación del modelo
- Entrenamiento / prueba (70% / 30%)

## Overfitting



¿El modelo que mejor se ajusta a los datos de entrenamiento (los puntos rojos) es la que mejor predice? **No existe tal garantía**

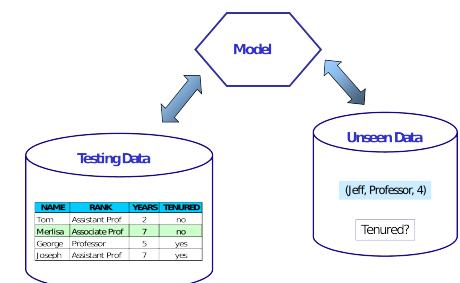
## Clasificación

- **Aplicaciones típicas:**
  - Aprobación de crédito: se clasifica la solicitud de crédito como de bajo riesgo, alto riesgo o riesgo medio
  - Se determina si un acceso local en un equipo es legal o ilegal
  - Marketing dirigido (enviar o no un catálogo)
  - Diagnóstico médico
  - Clasificación de texto (spam, no spam)
  - Reconocimiento de texto (reconocimiento óptico de caracteres)

## Regresión

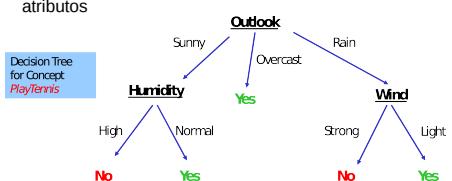
- **Aplicaciones típicas:**
  - Mercado de valores: predecir el precio de la acción para el futuro
  - Energía exigente en una presa
  - Velocidad del viento: energía eólica
  - Travel predicción tiempo: para la planificación de las empresas de transporte
  - Nivel de agua en un río: la seguridad y la prevención
  - Los ingresos tributarios: presupuesto público

## Paso 2: Aplicación del modelo



## Árboles de decisión

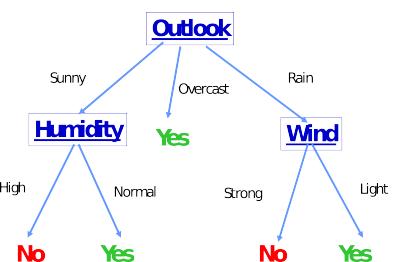
- **Nodo interno** representa una pregunta sobre un atributo
- **Rama** corresponde a un valor de atributo y representa el resultado de una pregunta
- **Nodo hoja** representa una etiqueta o distribución de clase
- Cada **camino** es una asociación de valores de uno o más atributos



## Multiples árboles de decisión

- Si los atributos son adecuados, se puede construir un árbol de decisión que clasifica correctamente todas las instancias
- Pueden generarse muchos árboles de decisión correctos
- Muchos algoritmos prefieren árbol más simple (navaja de Occam)
  - El principio establece que no se debe hacer más suposiciones que el mínimo necesario
  - El árbol más simple captura la mayor generalización y representa las relaciones esenciales

## Árboles de decisión



## Por qué Árboles de decisión

- Los árboles de decisión son especialmente atractivos para la minería de datos por tres razones:
  - Debido a su representación intuitiva, son fáciles de entender por los seres humanos.
  - Pueden construirse relativamente rápido en comparación con otros métodos.
  - La precisión de los clasificadores usando árboles de decisión es comparable o superior a otros modelos.

## Árboles de decisión: construcción

- La generación del árbol de decisión consiste en dos fases:
  - **Construcción**
    - Al principio, todos los ejemplos de entrenamiento están en la raíz
    - Partición recursiva de los ejemplos basándose en los atributos seleccionados
  - **Poda**
    - Identificar y eliminar las ramas que reflejan el ruido o valores atípicos
- Uso del árbol de decisión: clasificación de una muestra desconocida
  - Prueba de los valores de atributo de la muestra contra el árbol de decisión

## Elegir buenos atributos

- ¡Muy importante!
  - 1.Si no existe un atributo fundamental, el árbol de decisión no aprenderá los conceptos
  - 2.Si dos instancias tienen la misma representación, pero pertenecen a diferentes clases, no es adecuado usar árboles de decisión

Name	Cough	Fever	Pain	Diagnosis
Ernie	No	Yes	Throat	Flu
Bert	No	Yes	Throat	Appendicitis

## Eligiendo atributos de separación

- Se evalúa el atributo en función de su poder de separación entre las instancias cercanas
- Los valores de un buen atributo deben distinguir entre los casos cercanos de distinta clase y tener valores similares para los casos cercanos de una misma clase
- Los valores numéricos se pueden discretizar

## Escogiendo el atributo

- Muchas variantes:
  - del aprendizaje automático: **ID3** (iterative Dichotomizer), **C4.5** (Quinlan 86, 93)
  - de las estadísticas: **CART** (árboles de clasificación y regresión) (Breiman et al 84)
  - de reconocimiento de patrones: **CHAID** (Chi-squared Automated Interaction Detection) (Magidson 94)
- La principal diferencia: división (split) criterio
  - ¿Qué atributo para poner a prueba en cada nodo en el árbol? El atributo que es más útil para la clasificación de ejemplos.

## Criterio de separación

- Ganancia de información (Information gain)
  - Se supone que todos los atributos son categóricos (ID3)
  - Puede ser modificado para atributos de valor continuo (C4.5)
- Índice de Gini (CART, IBM IntelligentMiner)
  - Todos los atributos se suponen de valor continuo
  - Asume que existen varias posibilidades de separación para cada atributo
  - Puede ser modificado para los atributos categóricos

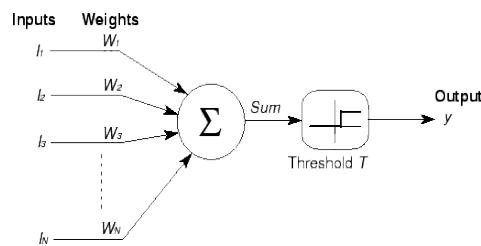
## C4.5 (Quinlan)

- Se trata de un algoritmo de referencia
- Innovaciones del C4.5:
  - permitir atributos numéricos
  - lidar con valores perdidos
  - poda para hacer frente a los datos ruidosos
- C4.5 - uno de los más conocidos y algoritmos de aprendizaje más utilizados
  - Última versión de investigación: C4.8, implementado en Weka como J4.8 (Java)
  - Sucesor comercial: C5.0 (disponible en Rulequest)

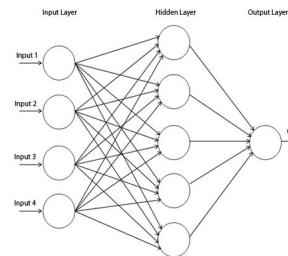
## Perceptrón multicapa: Historia

1957	Rosenblatt comienza el desarrollo del Perceptrón (simple).
1960	Widrow y Hoff desarrollan el modelo Adaline (ADaptive LINear Elements).
1969	Minsky y Papert prueban que el Perceptrón no es capaz de resolver problemas sencillos (XOR).
1974	Werbos desarrolla la idea básica del algoritmo de retro-propagación (BP).
1986	Rumelhart y Hinton redescubren y mejoran el algoritmo de BP.

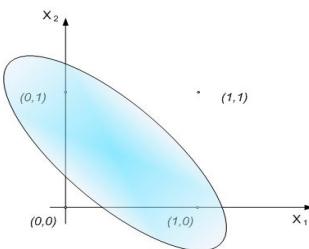
## Neurona McCullochs-Pitts



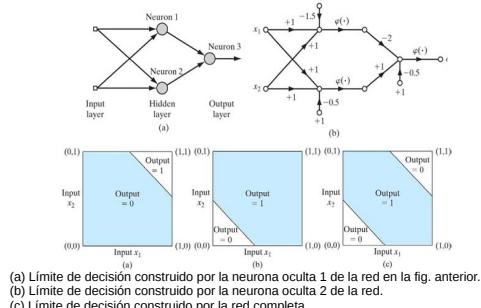
## Perceptrón multicapa



## El problema del XOR



## Combinación de perceptrones simples



## Entrenamiento por el método de gradiente

- Concepto:
  - Mover los pesos en la dirección en que se reduce el error, dirección que es opuesta a su gradiente con respecto a los pesos
- Ecuación básica:
 
$$w(n+1) = w(n) - \mu \nabla_w \xi(w(n))$$
- Aplicación:
  - Caso sencillo: perceptrón simple (least mean squares)
  - Caso más general: perceptrón multicapa (back-propagation)

## Recomendaciones

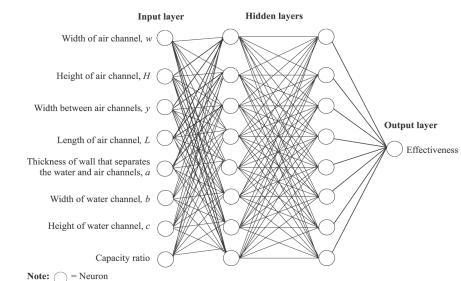
- Escalar la entrada:** normalizar los patrones de entrenamiento de modo que la media (sobre el conjunto de entrenamiento) de cada atributo sea cero y la misma varianza 1.
- Inicializar los Pesos:** aleatoriamente utilizando el método Nguyen-Widrow.
- Tasa de aprendizaje:** El valor de la tasa de aprendizaje determina la velocidad a la que la red alcanza un mínimo. En la práctica, dado que las redes son rara vez totalmente capacitados para un mínimo error de entrenamiento la tasa de aprendizaje puede afectar a la calidad de la red final.
- Número de unidades ocultas:** Una regla práctica para elegir el número de neuronas en la capa oculta es

$$h = \frac{\text{instances} * 0.05}{\text{InputUnits} + \text{OutputUnits}}$$

## Recomendaciones

- Momentum:** permite a la red aprender más rápidamente (las superficies de error a menudo tienen mesetas). El enfoque consiste en alterar la regla de aprendizaje de retropropagación estocástico para incluir una fracción de la actualización de peso anterior.
- Decaimiento de pesos:** un método de simplificación de la red y evitar sobreajuste es imponer una heurística para que las conexiones que no se usan tengan pesos pequeños.

## Perceptrón multicapa



## Regiones para varias capas

Estructura	Tipos de regiones de decisión	Problema XOR	Separación en clases	Formas regiones más generales
Una capa	hemiplano limitado por hiperplano	A vs B	Separación lineal	Separación lineal
Dos capas	Regiones convexas abiertas o cerradas	A vs B vs A	Separación lineal	Separación no lineal
Tres capas	Arbitrarias (Complejidad limitada por N°. de Nodos)	A vs B vs A	Separación no lineal	Separación no lineal

# Clustering

## 1 Clustering

- [What is Clustering?](#)
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods

## 2 What is Clustering?

- **Clustering** of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data.  

- **Cluster**: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters  

- Clustering is unsupervised classification: no predefined classes

## 3 What is Clustering?

- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms
- Use cluster detection when you suspect that there are natural groupings that may represent groups of customers or products that have lot in common.
- When there are many competing patterns in the data, making it hard to spot a single pattern, creating clusters of similar records reduces the complexity within clusters so that other data mining techniques are more likely to succeed.

## 4 Examples of Clustering Applications

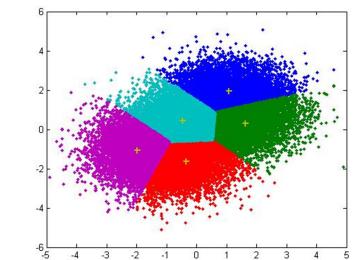
- **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **Land use**: Identification of areas of similar land use in an earth observation database
- **Insurance**: Identifying groups of motor insurance policy holders with a high average claim cost
- **City-planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Earthquake studies**: Observed earth quake epicenters should be clustered along continent faults

## 5 Clustering definition

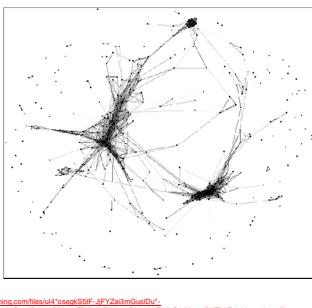
- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that:
  - data points in one cluster are more similar to one another (**high intra-class similarity**)
  - data points in separate clusters are less similar to one another (**low inter-class similarity**)
- Similarity measures: e.g. Euclidean distance if attributes are continuous.

## 6 Requirements of Clustering in Data Mining

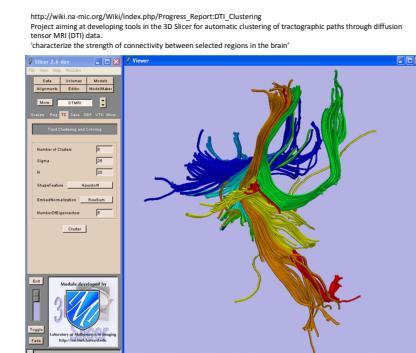
- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability



[http://webscripts.softpedia.com/screenshots/Efficient-K-Means-Clustering-using-JIT\\_1.png](http://webscripts.softpedia.com/screenshots/Efficient-K-Means-Clustering-using-JIT_1.png)



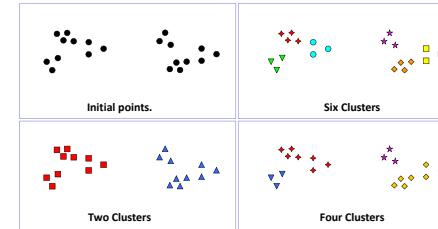
[https://openimage.net/files/1447/reports/DTI\\_MRI\\_ClusteredGraph.pdf](https://openimage.net/files/1447/reports/DTI_MRI_ClusteredGraph.pdf)



9

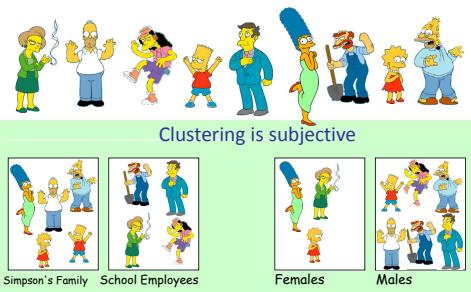
10

## 7 Notion of a Cluster is Ambiguous



11

## 8 What is a natural grouping among these objects?



(c) Eamonn Keogh, eamonn@cs.ucr.edu

## Clustering

- What is Cluster Analysis?

### Types of Data in Cluster Analysis

- A Categorization of Major Clustering Methods

- Partitioning Methods

- Hierarchical Methods

13

## Data Matrix

- Represents n objects with p variables (attributes, measures)
  - A relational table

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

14

Name	Energy	Protein	Fat	Codium	Iron
Brisket	340	20	28	9	2.6
Hamburger	245	21	17	9	2.7
Roast beef	420	15	39	7	2
Beefsteak	375	19	32	9	2.6
Canned beef	180	22	10	17	3.7
Broiled chicken	170	20	3	6	1.4
Canned chicken	170	25	7	12	1.5
Beef heart	160	26	5	14	5.9
Roast lamb leg	265	20	20	9	2.6
Roast lamb shoulder	300	18	25	9	2.3
Smoked ham	340	20	28	9	2.5
Pork roast	340	19	29	9	2.5
Pork tenderloin	205	19	30	9	2.4
Beef tongue	205	18	14	7	2.5
Veal cutlet	185	23	9	9	2.7
Baked beefsteak	135	22	4	25	0.6
Raw clams	70	11	1	82	6
Canned clams	70	7	1	74	5.4
Canned crabmeat	90	14	2	10	3.8
Fried haddock	135	16	5	15	0.5
Broiled mackerel	200	19	13	5	1
Canned mackerel	155	16	9	157	1.8
Fried perch	195	16	11	14	1.3
Canned salmon	120	17	5	159	0.7
Canned sardines	140	22	9	367	2.5
Canned tuna	170	25	7	7	1.2
Canned shrimp	110	23	1	98	2.6

15

## Dissimilarity Matrix

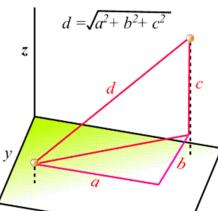
- Proximities of pairs of objects
- $d(i,j)$ : dissimilarity between objects i and j
- Nonnegative
- Close to 0: similar

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

16

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27				
1	0.27																													
2	0.31	0.42																												
3	0.31	0.42	0.32																											
4	0.37	0.27	0.32	0.41																										
5	0.41	0.33	0.52	0.46	0.32																									
6	0.41	0.33	0.52	0.46	0.32	0.45																								
7	0.41	0.32	0.52	0.45	0.32	0.45	0.30																							
8	0.24	0.20	0.38	0.28	0.30	0.34	0.34	0.30																						
9	0.22	0.26	0.32	0.25	0.37	0.39	0.39	0.36	0.32																					
10	0.13	0.15	0.17	0.16	0.18	0.15	0.15	0.15	0.15	0.15																				
11	0.18	0.29	0.36	0.20	0.39	0.42	0.42	0.52	0.22	0.18																				
12	0.20	0.30	0.28	0.19	0.49	0.43	0.53	0.27	0.22	0.19	0.18																			
13	0.35	0.25	0.49	0.39	0.22	0.29	0.24	0.32	0.32	0.34	0.35	0.36	0.38	0.25																
14	0.45	0.35	0.56	0.49	0.31	0.23	0.25	0.38	0.38	0.45	0.45	0.46	0.47	0.34	0.29															
15	0.65	0.70	0.67	0.68	0.68	0.55	0.64	0.66	0.65	0.64	0.66	0.51	0.52	0.50	0.23															
16	0.51	0.43	0.54	0.56	0.44	0.47	0.37	0.48	0.44	0.45	0.51	0.50	0.51	0.37	0.39	0.28	0.41													
17	0.35	0.28	0.44	0.48	0.30	0.27	0.28	0.28	0.41	0.29	0.35	0.35	0.35	0.29	0.28	0.28	0.53	0.56	0.33	0.26										
18	0.46	0.38	0.50	0.46	0.46	0.33	0.34	0.46	0.39	0.45	0.45	0.45	0.45	0.46	0.33	0.35	0.32	0.32	0.27	0.27										
19	0.52	0.44	0.58	0.42	0.29	0.36	0.49	0.45	0.46	0.52	0.51	0.52	0.38	0.40	0.29	0.46	0.49	0.28	0.33	0.24	0.32									
20	0.51	0.41	0.65	0.55	0.37	0.44	0.40	0.49	0.45	0.50	0.52	0.53	0.40	0.42	0.58	0.62	0.51	0.49	0.43	0.35	0.41									
21	0.42	0.31	0.51	0.45	0.44	0.32	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31									
22	0.35	0.36	0.40	0.37	0.38	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39									
23	0.45	0.46	0.47	0.48	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49									
24	0.45	0.46	0.47	0.48	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49									
25	0.45	0.46	0.47	0.48	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49									
26	0.45	0.46	0.47	0.48	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49									
27	0.45	0.46	0.47	0.48	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49									

17



[http://uk.geocities.com/ahf\\_alternate/dist.htm#S2](http://uk.geocities.com/ahf_alternate/dist.htm#S2)

21

- City-block (Manhattan) distance.** This distance is simply the sum of differences across dimensions. In most cases, this distance measure yields results similar to the Euclidean distance. However, note that in this measure, the effect of single large differences (outliers) is damped (since they are not squared).
- The properties stated for the Euclidean distance also hold for this measure.

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

22

- Manhattan distance** = distance if you had to travel along coordinates only.

$$\text{euc.: } \text{dist}(x,y) = \sqrt{(4^2+3^2)} = 5$$

manh.:  $\text{dist}(x,y) = 4+3 = 7$

23

- Minkowski distance.** Sometimes one may want to increase or decrease the progressive weight that is placed on dimensions on which the respective objects are very different. This measure enables to accomplish that and is computed as:

$$d(i,j) = \left( |x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q \right)^{1/q}$$

24

## Similarity/Dissimilarity Between Objects

### Weighted distances

- If we have some idea of the relative importance that should be assigned to each variable, then we can weight them and obtain a weighted distance measure.

$$d(i, j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + \dots + w_p(x_{ip} - x_{jp})^2}$$

25

## Binary Variables

- Binary variable has only two states: 0 or 1
- A binary variable is **symmetric** if both of its states are equally valuable, that is, there is no preference on which outcome should be coded as 1.
- A binary variable is **asymmetric** if the outcome of the states are not equally important, such as positive or negative outcomes of a disease test.
- Similarity that is based on symmetric binary variables is called **invariant similarity**.

## Binary Variables

- A contingency table for binary data

		Object <i>j</i>		sum
		1	0	
Object <i>i</i>	1	<i>a</i>	<i>b</i>	
	0	<i>c</i>	<i>d</i>	<i>c+d</i>
		sum	<i>a+c</i>	<i>b+d</i>
				<i>p</i>

object i	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0
object j	0	1	1	0	0	1	0	1	0	1	0	1	0	0	0	1	1	1	0

## Binary Variables

- Symmetric binary dissimilarity:
- $$d(i, j) = \frac{b + c}{a + b + c + d}$$
- Jaccard coefficient (asymmetric binary dissimilarity):
- $$d(i, j) = \frac{b + c}{a + b + c}$$

28

## Dissimilarity between Binary Variables

### Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- gender is a symmetric attribute
- the remaining attributes are asymmetric binary
- let the values Y and P be set to 1, and the value N be set to 0

$$\left\{ \begin{array}{l} d(jack, mary) = \frac{0+1}{2+0+1} = 0.33 \\ d(jack, jim) = \frac{1+1}{1+1+1} = 0.67 \\ d(jim, mary) = \frac{1+2}{1+1+2} = 0.75 \end{array} \right.$$

29

## Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1:** simple matching
  - # of matches, *m*: # of variables
- Method 2:** use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

$$d(i, j) = \frac{p-m}{p}$$

30

## Ordinal Variables

- On ordinal variables order is important
  - e.g. Gold, Silver, Bronze
- Can be treated like continuous
  - the ordered states define the ranking  $1, \dots, M_i$
  - replacing  $x_{if}$  by their rank  $r_{if} \in \{1, \dots, M_i\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by
- compute the dissimilarity using methods for continuous variables

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

31

Col 1	Gender	Age	Waterfme	Swedish	Number of actions	Other species & s	Surgeon id	Surgeon's name	Expertise/Qualification	Multi/Procedural	OtherDiagnosis	OtherTreatment	Date lastUpdate						
1	1	54	0	2	36	1	0	0	80	737	0	0	0						
2	1	54	0	2	36	1	0	0	1160	0	1	0	0						
3	1	58	1	97	36	1	1	0	1160	0	20	1	0						
4	1	73	0	236	2	2	1548	0	0	1103	1	1	0	0					
5	1	36	0	20	3	493	0	0	894	1	515	259	0						
6	0	89	0	12	5	1	3317	0	0	233	1	8	101	1	0				
7	1	61	0	140	5	1	1873	0	0	1182	0	32	25	0	0				
8	0	56	0	78	12	1	2339	0	0	592	1	0	0	0	1				
9	1	77	0	29	20	3	440	1	3	0	944	1	14	43	1	0			
10	1	57	0	12	5	1	3317	0	0	220	0	32	25	0	0				
11	1	60	0	41	5	1	3317	1	2	0	2109	0	20	156	1	0			
12	1	57	0	13	5	2	3317	0	0	233	1	6	79	0	0				
13	0	59	0	35	13	1	3226	0	0	1156	0	25	40	0	0				
14	0	60	0	24	13	2	1326	1	1	0	1156	0	54	13	1	0			
15	0	60	0	34	24	2	2919	2	3	1	826	0	138	253	1	0			
16	0	39	0	31	26	1	2324	2	2	1	2180	1	19	24	1	1			
17	0	36	0	84	5	1	3308	0	0	3	0	0	0	0	0				
18	0	36	0	109	1	1	0	231	0	7	32	0	0	0	0				
19	0	81	0	14	20	2	478	1	2	0	986	1	209	222	1	0	0		
20	0	81	0	38	20	2	478	0	0	1860	0	21	57	0	0				

33

```
> data <- read.table("dataset.csv", header=TRUE, sep=",")
> data$Priority <- as.factor(data$Priority)
> data$SpecialtyId <- as.factor(data$SpecialtyId)
> data$ICD1 <- as.factor(data$ICD1)
> data$SurgeonId <- as.factor(data$SurgeonId)
> library(cluster)

> diss_matrix <- daisy(data, type = list(asymm=c(11,16,17,18),
symm=c(2,13), ordratio = 4), metric = "gower", stand = TRUE)
```

34

## dissimilarity matrix

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																			
2	0.38																		
3	0.47	0.15																	
4	0.54	0.52	0.56																
5	0.54	0.52	0.70	0.46															
6	0.33	0.43	0.50	0.47	0.56														
7	0.33	0.42	0.45	0.39	0.52	0.42													
8	0.55	0.57	0.64	0.56	0.52	0.54	0.64												
9	0.51	0.59	0.53	0.43	0.38	0.57	0.55	0.64											
10	0.51	0.59	0.53	0.43	0.38	0.57	0.55	0.64	0.68										
11	0.79	0.43	0.38	0.56	0.69	0.35	0.34	0.55	0.47	0.50									
12	0.35	0.51	0.59	0.35	0.40	0.23	0.32	0.41	0.51	0.43	0.34								
13	0.45	0.45	0.55	0.52	0.58	0.48	0.35	0.37	0.67	0.45	0.41	0.41							
14	0.48	0.47	0.58	0.64	0.44	0.49	0.51	0.53	0.51	0.56	0.36	0.48	0.24						
15	0.57	0.64	0.60	0.71	0.76	0.57	0.63	0.57	0.62	0.49	0.61	0.51	0.36						
16	0.64	0.64	0.64	0.68	0.71	0.65	0.67	0.65	0.66	0.64	0.60	0.54	0.40	0.40					
17	0.53	0.55	0.58	0.56	0.66	0.59	0.59	0.58	0.65	0.66	0.60	0.55	0.45	0.45	0.45	0.45	0.45	0.45	0.45
18	0.63	0.64	0.61	0.61	0.52	0.43	0.57	0.49	0.55	0.48	0.53	0.56	0.40	0.40	0.47	0.60	0.45	0.45	0.45
19	0.54	0.51	0.58	0.50	0.53	0.44	0.43	0.46	0.58	0.36	0.51	0.43	0.31	0.38	0.51	0.60	0.34	0.46	0.35
20	0.81	0.38	20	2	478	0	0	1860	0	21	57	0	0	0	0	0	0	0	0

35

## Clustering

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods

36

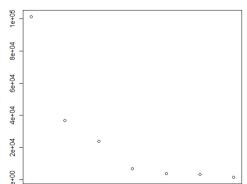


### Example (cont.)

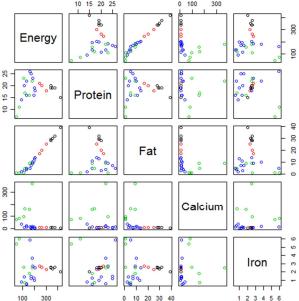
```
> cl3 <- kmeans(data[,2:6],centers=3)
...
> cl8 <- kmeans(data[,2:6],centers=8)
```

```
> errors <-
c(mean(cl2$withinss),
mean(cl3$withinss),
mean(cl4$withinss),
mean(cl5$withinss),
mean(cl6$withinss),
mean(cl7$withinss),
mean(cl8$withinss))
```

```
> plot(errors)
```



49



> plot(data[,2:6], col = cl4\$cluster)

50

Name	Energy	Protein	Fat	Calcium	Iron	cl4\$cluster	
Braised beef	340	20	8	9	2.6	1	
Roast beef	420	15	39	7	2	1	
Beefsteak	375	19	32	9	2.6	1	
Roast lamb leg	265	20	28	9	2.5	1	
Roast lamb shoulder	300	18	25	9	2.3	1	
Smoked ham	340	20	28	9	2.5	1	
Pork roast	340	19	29	9	2.5	1	
Pork simmered	255	20	30	9	2.4	1	
Hamburger	245	21	17	9	2.7	2	
Roast lamb leg	265	20	28	9	2.6	2	
Roast lamb shoulder	300	18	25	9	2.3	2	
Raw clams	70	11	1	82	6	3	
Canned clams	45	7	1	74	5.4	3	
Canned salmon	125	16	17	18	3.8	3	
Canned salmon	120	17	5	159	0.7	3	
Canned sardines	180	22	9	367	2.5	3	
Canned shrimp	110	23	1	98	2.6	3	
Canned beef	180	22	10	17	3.7	4	
Broiled chicken	115	20	3	8	1.4	4	
Canned chicken	170	25	7	12	1.5	4	
Beef heart	160	26	5	14	5.9	2	
Beef tongue	205	18	14	7	2.5	4	
Veal cutlet	185	23	9	9	3.7	4	
Baked bluefish	135	22	4	25	0.6	2	
Canned crabmeat	90	14	2	38	0.8	2	
Canned mackerel	200	13	5	9	15	0.5	2
Fried mackerel	195	16	11	14	1.3	2	
Canned tuna	170	25	7	7	1.2	4	
Canned tuna	170	25	7	7	1.2	4	

> data\_w\_cl 4<-

cbind(data,cl\$cluster)

> data\_w\_cl4

Name	Energy	Protein	Fat	Calcium	Iron	cl3\$cluster
Raw clams	70	11	1	82	6	3
Canned clams	45	7	1	74	5.4	3
Canned salmon	155	16	17	18	3.8	3
Canned salmon	120	17	5	159	0.7	3
Canned sardines	180	22	9	367	2.5	3
Canned shrimp	110	23	1	98	2.6	3
Canned beef	180	22	10	17	3.7	4
Broiled chicken	115	20	3	8	1.4	4
Canned chicken	170	25	7	12	1.5	4
Beef heart	160	26	5	14	5.9	2
Beef tongue	205	18	14	7	2.5	4
Veal cutlet	185	23	9	9	3.7	4
Baked bluefish	135	22	4	25	0.6	2
Canned crabmeat	90	14	2	38	0.8	2
Canned mackerel	200	13	5	15	0.5	2
Fried mackerel	195	16	11	14	1.3	2
Canned tuna	170	25	7	7	1.2	2
Canned tuna	170	25	7	7	1.2	4

Name	Energy	Protein	Fat	Calcium	Iron	cl3\$cluster
Raw clams	70	11	1	82	6	3
Canned clams	45	7	1	74	5.4	3
Canned salmon	155	16	17	18	3.8	3
Canned salmon	120	17	5	159	0.7	3
Canned sardines	180	22	9	367	2.5	3
Canned shrimp	110	23	1	98	2.6	3

### Davies-Bouldin Validity Index

R Package 'clusterSim' – index.DB

Let Ci be a cluster of vector. Let Xi be a vector on Ci and Si be a measure of scatter within the cluster

$$S_i = \sqrt{\frac{1}{T_i} \sum_{j=1}^T |X_j - A_i|^q}$$

Let Ai be the centroid of Ci and Mi,j a measure of separation between cluster Ci and Cj

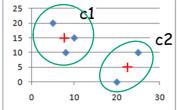
$$M_{i,j} = \sqrt{\sum_{k=1}^N |a_{k,i} - a_{k,j}|^p} \quad R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \quad R_i = \max_{j \neq i} R_{i,j}$$

$$DB = \frac{1}{N} \sum_{i=1}^N R_i$$

DB should be small for good clustering

### Example

X	Y	Cluster
8	10	C1
5	20	C1
20	0	C2
10	15	C1
25	10	C2



$$\begin{aligned} C1 &\left\{ \begin{array}{l} a_{1x} = \frac{8+5+10}{3} = 7.67 \\ a_{1y} = \frac{10+20+15}{3} = 15 \end{array} \right. \\ C2 &\left\{ \begin{array}{l} a_{2x} = \frac{20+15+0}{3} = 12.5 \\ a_{2y} = 5 \end{array} \right. \\ S_1 &= \sqrt{\frac{1}{3}((8-7.67)^2 + (10-15)^2 + \dots + (15-15)^2)} = 4.57 \\ S_2 &= \sqrt{\frac{1}{2}((20-22.5)^2 + (0-5)^2 + \dots)} = 5.59 \\ M_{1,2} &= \sqrt{(7.67-22.5)^2 + (15-5)^2} = 17.89 \\ R_{1,2} &= \frac{4.57 + 5.59}{17.89} = 0.568 \\ DB &= \frac{1}{2}(0.568 + 0.568) = 0.568 \end{aligned}$$

53

54

### Silhouette Validation Method

R Package 'clusterSim' – index.DB

For each point i, let a(i) be the average dissimilarity of i with all other points within the same cluster.

Any measure of similarity can be used (e.g. Euclidian distance)

Then, find the average dissimilarity of i with the data on another single cluster. Repeat this for every other cluster.

b(i) is the lowest average similarity of i to any other cluster.

We can no define for each point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

1 <= s(i) <= 1 and an s(i) close to one means that the datum is appropriately clustered. The average s(i) of a cluster is a measure of how tightly grouped the data is in the cluster.

Therefore, the average s(i) of the entire dataset is a measure of how appropriately the data has been clustered.

55

### Hartigan index

R Package 'clusterSim' – index.DB

Assume a dataset with N samples X1,...,XN. Each sample with M dimensions

For k clusters the overall fitness can be expressed as the square of error for all samples, where d is the distance between sample Xj and the centroid Xci of its cluster.

$$err(k) = \sum_{i=1}^k \sum_{j=1, j \neq i}^N (d(X_j, X_{ci}))^2$$

The Hartigan index H(k) is defined as follows

$$H(k) = \frac{(n-k-1)err(k) - err(k+1)}{err(k+1)}$$

The multiplier correction term of (N-k-1) is a penalty factor for large number of cluster partitioning. The optimal k number is the one that maximizes the H(k).

56

### Some links

HARTIGAN - Data for Clustering Algorithms

<http://orion.math.iastate.edu/burkardt/data/hartigan/hartigan.html>

### R packages

cluster - <http://cran.r-project.org/web/packages/cluster/index.html>

clusterSim - <http://cran.r-project.org/web/packages/clusterSim/index.html>

57

58

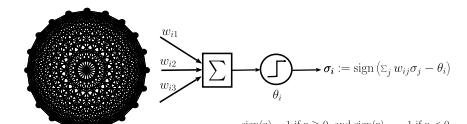
## Naturaleza de la memoria asociativa

- Dos conceptos o estímulos están relacionados cuando la experiencia de uno, conduce a los efectos del otro.
- También se le llama memoria de direcciónamiento por contenido.



El problema básico es:

- Almacenar un conjunto  $P$  de patrones  $\xi_i^\mu$ .
- Cuando un nuevo patrón  $\zeta_i$  es presentado a la red, responder produciendo como salida el patrón almacenado que más se parezca a  $\zeta_i$ .



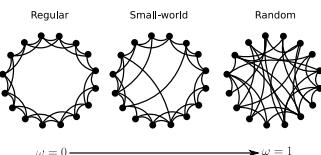
## Información en bloque en redes neuronales atractoras

Mario González  
mar.gonzalez03@estudiante.uam.es

June 13, 2008

### Resumen

El objetivo de este trabajo es examinar el rol de la topología small-world en un red neuronal atractora (RNA) de tipo Hopfield, y presentar una evaluación de su comportamiento en el aprendizaje/recuperación, cuando un estímulo espacialmente estructurado es presentado a la red.

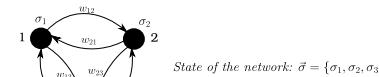


- Matriz de topología:  $C = \{C_{ij}\}$ .
- $C_{ij} \in \{0, 1\}$  se divide en conexiones locales  $K_L$  y aleatorias  $K_R$ .
- Parámetros de la red:

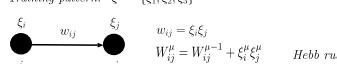
  - Tasa de conectividad:  $\gamma = K/N$ .
  - Tasa de aleatoriedad:  $\omega = K_R/K$ .

## Etapa de aprendizaje

Un conjunto de patrones  $\xi_i^\mu$ ,  $\mu = 1, \dots, P$ , es presentado a la red durante la etapa de entrenamiento, y los pesos de las conexiones  $w_{ij}$  son ajustados de acuerdo a las correlaciones pre/post sinápticas.



Training pattern:  $\zeta = \{\xi_1, \xi_2, \xi_3\}$



## Etapa de recuperación

- La red empieza en una configuración inicial  $\sigma_i = \zeta_i$ , y la red evoluciona en el tiempo de acuerdo a:

$$\sigma_i^t = \text{sign} \left( \sum_i h_i^{t-1} - \theta \right), \quad h_i^t \equiv \sum_j J_{ij} \sigma_j^t.$$

Las conexiones sinápticas entre las neuronas  $i, j$  están dadas por  $J_{ij} \equiv C_{ij} W_{ij}$ .

Todas las unidades son actualizadas de forma simultánea (síncrona) en cada tiempo de iteración.

■ La red evoluciona hacia el patrón que más se asemeja al estímulo presentado ( $\zeta_i$ ).

## Etapa de recuperación

- La red empieza en una configuración inicial  $\sigma_i = \zeta_i$ , y la red evoluciona en el tiempo de acuerdo a:

$$\sigma_i^t = \text{sign} \left( \sum_i h_i^{t-1} - \theta \right), \quad h_i^t \equiv \sum_j J_{ij} \sigma_j^t.$$

Las conexiones sinápticas entre las neuronas  $i, j$  están dadas por  $J_{ij} \equiv C_{ij} W_{ij}$ .

Todas las unidades son actualizadas de forma simultánea (síncrona) en cada tiempo de iteración.

■ La red evoluciona hacia el patrón que más se asemeja al estímulo presentado ( $\zeta_i$ ).

## Medidas de recuperación

- El solapamiento entre el estado de la red ( $\sigma_i$ ) y un patrón ( $\zeta_i$ ) puede ser usado como una medida del rendimiento de la red:

$$m \equiv \frac{1}{N} \sum_i \xi_i \sigma_i$$

- El solapamiento entre el estado de la red ( $\sigma_i$ ) y un patrón ( $\zeta_i$ ) puede ser usado como una medida del rendimiento de la red:

$$m \equiv \frac{1}{N} \sum_i \xi_i \sigma_i$$



Izquierda: Patrón original. Centro:  $m = 0.4$ . Derecha:  $m = -1$ .

## Etapa de aprendizaje

## Etapa de recuperación

## La red de Hopfield

### Introducción

Naturaleza de la memoria asociativa

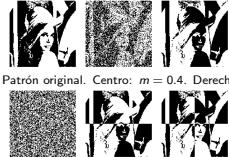
La red de Hopfield

La red de Hopfield

## Medidas de recuperación

El solapamiento entre el estado de la red ( $\sigma_i$ ) y un patrón ( $\xi_i$ ) puede ser usado como una medida del rendimiento de la red:

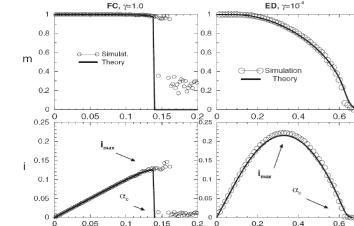
$$m \equiv \frac{1}{N} \sum_i \xi_i \sigma_i$$



Izquierda: Patrón original. Centro:  $m = 0.4$ . Derecha:  $m = -1$ .  
Izquierda: Solapamiento  $m = 0$ . Centro: 2-bloques ( $m = 0$ ). Derecha: 4-bloques ( $m = 0$ ).

## Medidas de información

Comparación entre el solapamiento  $m(\alpha)$  y la tasa de información  $i(\alpha)$  como medidas de la capacidad de recuperación de la red.



Tomado de Dominguez et al.(2007).

## Interacción con sesgo

$$W_{ij} = cW'_{ij} + (1 - c)\bar{W}$$

$W'_{ij}$  son generados aleatoriamente como  $\pm 1$  con igual probabilidad.  
 $\bar{W} = 1$ ,  $c \in (0, 1)$



En un ferromagneto, todos los  $w_{ij}$  son positivos.

$W_{ij} = cW'_{ij} + (1 - c)\bar{W}$   
 $W'_{ij}$  son generados aleatoriamente como  $\pm 1$  con igual probabilidad.  
 $\bar{W} = 1$ ,  $c \in (0, 1)$

Actividad global ( $a$ ) y en bloque ( $d$ )  
 $a_I^t = \frac{1}{L} \sum_{i \in I} \sigma_i^t$   
 $a = \langle a_I \rangle_b$ ,  $d^2 = v = \langle a_I^2 \rangle_b - a^2$



En un ferromagneto, todos los  $w_{ij}$  son positivos.

## Interacción con sesgo

### Interacción con sesgo

$$W_{ij} = cW'_{ij} + (1 - c)\bar{W}$$

$W'_{ij}$  son generados aleatoriamente como  $\pm 1$  con igual probabilidad.  
 $\bar{W} = 1$ ,  $c \in (0, 1)$

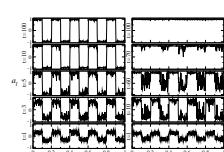
### Actividad global ( $a$ ) y en bloque ( $d$ )

$$a_I^t = \frac{1}{L} \sum_{i \in I} \sigma_i^t$$

$$a = \langle a_I \rangle_b$$
,  $d^2 = v = \langle a_I^2 \rangle_b - a^2$

Fases representativas: Actividad global (R), con  $a \neq 0$ ,  $d = 0$ . Actividad en bloque (B), con  $a = 0$ ,  $d > 0$ . Actividad cero (Z), con  $a = 0$ ,  $d = 0$ .

## Interacción ferromagnética/Hebbiana



Paneles izquierdos: Evolución de la red hacia un estado ordenado en bloques,  $\omega = 0.1$  y  $c = 0.8$ . Paneles derechos: Evolución de la red hacia un estado ordenado global,  $\omega = 0.3$  y  $c = 0.8$ .  $N = 10^5$  y  $\gamma = 10^{-3}$ ,  $N_w = 2 \times 10^5$ . Condición inicial:  $a_I^{t=0} \sim \pm 0.2$ .

## Interacción Hebbiana

$$W_{ij} = cW'_{ij} + (1 - c)\xi_i \xi_j$$

En un ferromagneto, todos los  $w_{ij}$  son positivos.



## Interacción Hebbiana

$$W_{ij} = cW'_{ij} + (1 - c)\xi_i \xi_j$$

Solapamiento global ( $m$ ) y de bloque ( $\delta$ )  
 $m_I^t \equiv \frac{1}{L} \sum_{i \in I} \xi_i \sigma_i^t$ ,  
 $m \equiv \langle m_I \rangle_b$ ,  $\delta^2 = v \equiv \langle m_I^2 \rangle_b - m^2$ .



## Interacción Hebbiana

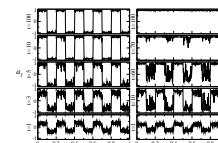
$$W_{ij} = cW'_{ij} + (1 - c)\xi_i \xi_j$$

Solapamiento global ( $m$ ) y de bloque ( $\delta$ )  
 $m_I^t \equiv \frac{1}{L} \sum_{i \in I} \xi_i \sigma_i^t$ ,  
 $m \equiv \langle m_I \rangle_b$ ,  $\delta^2 = v \equiv \langle m_I^2 \rangle_b - m^2$ .



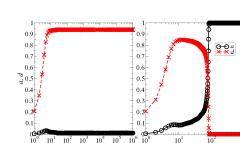
Fases representativas: Recuperación global (R), con  $m \neq 0$ ,  $\delta = 0$ . Recuperación de bloque (B), con  $m = 0$ ,  $\delta > 0$ . Recuperación cero (Z), con  $m = 0$ ,  $\delta = 0$ .

## Interacción ferromagnética/Hebbiana

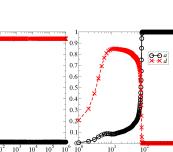


Paneles izquierdos: Evolución de la red hacia un estado ordenado en bloques,  $\omega = 0.1$  y  $c = 0.8$ . Paneles derechos: Evolución de la red hacia un estado ordenado global,  $\omega = 0.3$  y  $c = 0.8$ .  $N = 10^5$  y  $\gamma = 10^{-3}$ ,  $N_w = 2 \times 10^5$ . Condición inicial:  $a_I^{t=0} \sim \pm 0.2$ .

## Interacción ferromagnética/Hebbiana



Paneles izquierdos: Evolución de la red hacia un estado ordenado en bloques,  $\omega = 0.1$  y  $c = 0.8$ . Paneles derechos: Evolución de la red hacia un estado ordenado global,  $\omega = 0.3$  y  $c = 0.8$ .  $N = 10^5$  y  $\gamma = 10^{-3}$ ,  $N_w = 2 \times 10^5$ . Condición inicial:  $a_I^{t=0} \sim \pm 0.2$ .



## Diagramas de fases y de flujo

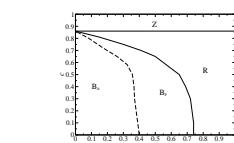


Diagrama de fases. B ( $B_A$ ,  $B_F$ ): región de recuperación de bloques. R: región de recuperación global. Z: región de información cero. Red con  $N = 2^{16}$  y  $\gamma = 2^{-10}$ . Condición inicial:  
 $B_A$ :  $m^{t=0} = 0$ ,  $\delta^{t=0} = 0.1$ ,  
 $B_F$ :  $m^{t=0} = 0$ ,  $\delta^{t=0} = 1$ .

## Diagramas de fases y de flujo

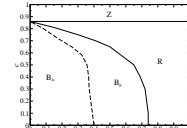


Diagrama de fases. B ( $B_A$ ,  $B_F$ ): región de recuperación de bloques. R: región de recuperación global. Z: región de información cero. Red con  $N=2^{16}$  y  $\gamma = 2^{-10}$ . Condición inicial:  
 $B_A : m^t=0 = 0, \delta^t=0 = 0.1$ ,  
 $B_F : m^t=0 = 0, \delta^t=0 = 1$ .

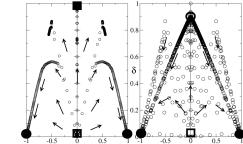
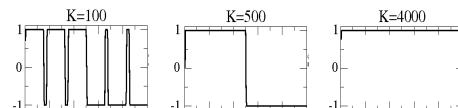


Diagrama de flujo. Izquierda: fase de bloques ( $\omega = 0.1$ ,  $c = 0.75$ ), derecha: fase global ( $\omega = 0.3$ ,  $c = 0.75$ ). Círculos sólidos: puntos estables. Círculo hueco: punto de silla. Cuadrado sólido: punto marginalmente estable. Cuadrado hueco: repulsor marginal.

## Estabilidad de los bloques y dilución de la red



Evolución de la red hacia la fase global o de bloque de acuerdo a la dilución de la red  $\gamma$ . La red empieza con  $b = 40$  bloques no contiguos de actividad positiva/negativa ( $a_i^{t=0} \sim \pm 0.3$ ). Red con  $N = 10^4$ ,  $\omega = 0.1$ ,  $c = 0.4$ ,  $N_w = 40$ .

## Red hebbiana y medidas de información

La red almacena  $P = \alpha K$  patrones binarios,  $\xi^\mu \equiv \xi_i^\mu \in \{\pm 1\}$ ,  $\mu = 1, \dots, P$ , con un nivel de actividad media ( $\langle \xi_i^\mu \rangle = 0$ ,  $\alpha$  es la tasa de carga).

### Actualización de los pesos (regla de Hebb)

$$W_{ij}^\mu = W_{ij}^{\mu-1} + \xi_i^\mu \xi_j^\mu$$

## Red hebbiana y medidas de información

La red almacena  $P = \alpha K$  patrones binarios,  $\xi^\mu \equiv \xi_i^\mu \in \{\pm 1\}$ ,  $\mu = 1, \dots, P$ , con un nivel de actividad media ( $\langle \xi_i^\mu \rangle = 0$ ,  $\alpha$  es la tasa de carga).

### Actualización de los pesos (regla de Hebb)

$$W_{ij}^\mu = W_{ij}^{\mu-1} + \xi_i^\mu \xi_j^\mu$$

La tarea de la red es recuperar un patrón  $\tilde{\xi} \equiv \tilde{\xi}^\mu$ , empezando con una estructura en bloques  $\{\xi^\mu\}$ . Se usa  $m, \delta$  junto con  $i_m, i_v$  para evaluar el rendimiento de la red.

## Red hebbiana y medidas de información

La red almacena  $P = \alpha K$  patrones binarios,  $\xi^\mu \equiv \xi_i^\mu \in \{\pm 1\}$ ,  $\mu = 1, \dots, P$ , con un nivel de actividad media ( $\langle \xi_i^\mu \rangle = 0$ ,  $\alpha$  es la tasa de carga).

### Actualización de los pesos (regla de Hebb)

$$W_{ij}^\mu = W_{ij}^{\mu-1} + \xi_i^\mu \xi_j^\mu$$

La tarea de la red es recuperar un patrón  $\tilde{\xi} \equiv \tilde{\xi}^\mu$ , empezando con una estructura en bloques  $\{\xi^\mu\}$ . Se usa  $m, \delta$  junto con  $i_m, i_v$  para evaluar el rendimiento de la red.

### Información global ( $i_m$ )

$$i_m(\alpha, m) \equiv \alpha M[\sigma; \xi],$$

$$M[\sigma; \xi] = S[\sigma] - S[\sigma|\xi]$$

$$S[\sigma] = 1[\text{bit}]$$

$$S[\sigma|\xi] = -\frac{1-m}{2} \log_2 \frac{1+m}{2} - \frac{1-m}{2} \log_2 \frac{1-m}{2}$$

### Información de bloques ( $i_v$ )

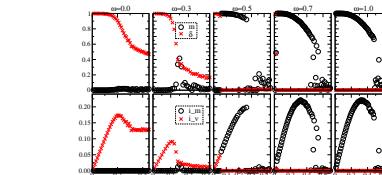
$$i_v(\alpha, m) \simeq \alpha \log(1+v),$$

$$M[\sigma; \{\tilde{\xi}^\mu\}] \sim S[m|z] - S[z]$$

$$\text{Señal: } m_t, \text{ con varianza } v.$$

$$\text{Ruido: } z, \text{ con } v_z \sim 1.$$

- El solapamiento global permanece cerca de  $m^* = 1$  para redes aleatorias.
- La solución en bloque existe para redes con conexiones locales,  $\delta^* = 1$ .



Solapamiento global y de bloque  $m, \delta$  (arriba), e información  $i_m, i_v$  (abajo), vs.  $\alpha$ . Red con  $N = 10^5$ ,  $\gamma = 10^{-3}$ ,  $y = b = 2$ . Estado inicial:  $m^{t=0} = 0.04$ ,  $\delta^{t=0} = 1$ .

## Conclusiones

- Un nuevo tipo de solución para una red atractora ha sido estudiado: la fase de actividad/recuperación de bloques (B).
- Esta fase de bloques es estable, con un cuenco de atracción suficientemente grande.
- Para redes regulares la correlación espacial, entre las neuronas permiten la recuperación del patrón en bloques de información.
- Para el régimen small-world, la red es capaz de extraer la información local contenida en los bloques, y transmitirla a lo largo de la red, recuperando el patrón global.

- Realizar un análisis sistemático del problema con una red estocástica caracterizada por una temperatura.
- Examinar otras topologías: redes libre de escala, modulares, etc.
- Usar neuronas multiestado y neuronas heterogéneas con diferentes umbrales.
- Utilizar modelos de neuronas más realistas como por ejemplo modelos de integración y disparo.
- Encontrar aplicaciones en áreas de interés como procesamiento de señales (reconocimiento de imágenes, voz, etc.), código genético, análisis de mercado de valores, etc.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

Daniel J. Amit. *Modeling brain function: the world of attractor neural networks*. Cambridge University Press, New York, NY, USA, 1989. ISBN 0-521-36100-1.

J. Hertz, J. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Boston, 1991.

P. N. McGraw and M. Menzinger. Topology and computational performance of attractor neural networks. *Phys. Rev. E*, 68:047102, 2003.

D. Dominguez, K. Koroutchev, E. Serrano, and F. B. Rodriguez. Information and topology in attractor neural network. *Neural Computation*, 19(4):956–973, 2007.

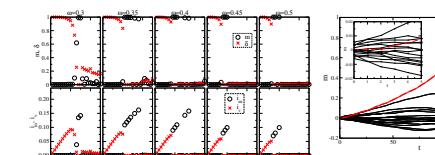
D. Dominguez, M. Gonzalez, E. Serrano, and F. B. Rodriguez. Structured information in small-world neural networks. *submitted to Phys. Rev. E*, 2008.

Mario Gonzalez, David Dominguez, and Francisco B. Rodriguez. Block activity in metric neural networks. *WASET Proceeding*, 27:56–59, 2008a.

Mario Gonzalez, David Dominguez, and Francisco B. Rodriguez. Learning block memories with metric networks. *WASET Proceeding*, 27:60–63, 2008b.

## Información de bloques y recuperación global

- Para valores intermedios del parámetro de aleatoriedad ( $0.3 \leq \omega \leq 0.5$ ), la red es capaz de transmitir la información contenida en los bloques y recuperar el patrón global.



Izquierda: Solapamiento global y de bloque  $m, \delta$  (arriba), e información  $i_m, i_v$  (abajo), vs.  $\alpha$ . Red con  $N = 10^5$ ,  $\gamma = 10^{-3}$ ,  $y = b = 2$ . Estado inicial:  $m^{t=0} = 0.04$ ,  $\delta^{t=0} = 1$ . Derecha: Evolución del solapamiento de la red con todos los patrones almacenados.

## Diagrama de fases

- La recuperación global del patrón es posible para un rango de valores del parámetro de aleatoriedad  $0.3 \leq \omega \leq 0.5$ , y valores de carga,  $\alpha \leq 0.18$ .

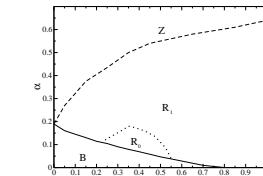


Diagrama de fases ( $\omega, \alpha$ ) para una red con  $N = 10^5$ ,  $\gamma = 10^{-3}$ ,  $y = b = 2$  bloques. B: Fase de bloque.  $R_0(R_1)$ : Fase de recuperación global con  $m^{t=0} = 0, \delta^{t=0} = 1$  ( $m^{t=0} = 1, \delta^{t=0} = 0$ ). Z: Fase de información cero.