

Combinaciones cónjuntas



- Palabras
- Fixed to one problem

optimización: Find "the best" solution

Ejemplos:

* Una función para unir

+ con variables para maximizar ganancias

! Reducir el tiempo de respuesta

+ Reducir el riesgo

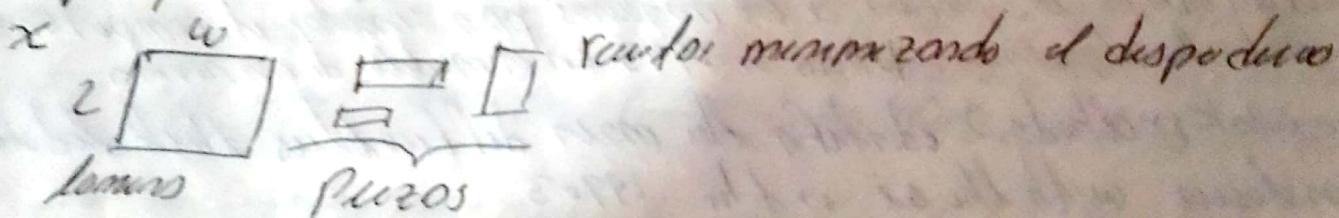
* Problema del agente viajero

A. F. D. visitar n p. de los estados (correos) usando la menor cantidad de rutas

C. B.

* $x = A \cdot B \cdot C \cdot D + D \oplus B + A(B \oplus C) + \dots$

porque valores x es free



* $f(x)$
 $\hookrightarrow R^d \rightarrow \min_{\max}$

- Brute force: calculando todos los salarios posibles

- Géndice Busc.: local options

↳ Combinaciones cónjuntas

Primer ejemplo de salas o los problemas de búsqueda

- Encuentro el mínimo de una sola de una variable

$$F(x) = \sin(e^{5 \sin(3x)} \cos(5x))$$

Question sheet 11

Define intelligence. what is the intelligent behavior of a machine?

La inteligencia es la habilidad que permite reasonar, pensar, "aprender", comprender hechos extensos o aun mas no.

Los inteligentes de los maquinas lo podemos definir como la capacidad computacional que permite aprender o una maquina sobre diferentes cosas

Describe the three test for artificial intelligence and justify its validity.
From mind stand point

Lo prueba de Turing consiste en 2 partes, la primera, un persona redada con un traje de mono. Tiene 2 personas. despues alguno de esos 2 personas se cambia por una maquina. Despues, el entrevistado pregunta si tiene preguntas sobre cualquier tema. En la cual se tienen 5 preguntas que estan enojadas antes de 5 minutos, los maquinas no pasan las pruebas desde un punto de vista natural, si se nota que no es correcto esto prueba, da la querencia de la I.S's capacias de imitar un conversacion fluida.

Define artificial intelligence as a science. what was artificial intelligence born?

Es los años que estudió la capacidad de los maquinas para simular el computacion humano mediante algoritmos. Nacen en 1943 con el modelo de un neurón

Cohort are work methods? identify the most difficult that led to the discoveries with this in the 1970's

los metodos debiles son aquellos en los cuales se hace de acuerdo con el problema al que se aplican solucion. lo principal nro 2 es que no se podian aplicar a problemas reales

Define expert system. what is the main difference between work made on the expert system technology?

Son sistemas que buscan resolver problemas complejos usando conocimiento humano de dichos temas. lo principal nro 3 es el uso de conocimiento humano

List the common characteristics of early expert systems such as DENDRAL, ~~TRANSLATE, MYCIN AND PROSPECTOR~~

- presentaban el conocimiento (solo se informan de su problema)
- permiten la adquisicion de conocimiento
- humana

who are the limitations of expert systems

- solo pueden manejar conocimientos de mundo real o en su forma escrita
- las preguntas de los usuarios no las manejan
- No son flexibles manipuladores
- no saben detectar la coherencia y consistencia
- no aprenden de las experiencias

what are the differences between expert systems and artificial neural networks

los sistemas experto manejan el mundo humano solo para sus propias necesidades, tienen que adaptarse a las experiencias

los red neuronal permite una mayor flexibilidad y aprende de las experiencias

why was the field of AI not robust in the 1980s?

que lo llevó al algoritmo back-propagation

what are the promises which Zadeh, logics and what was Fuzzy set theory introduced?

Los logicos difusos fueron introducidos por Lotfi Zadeh en 1965. Se basa en la noción de conocimiento mediante probabilidades y no binarias como los sistemas expertos.

what are the main advantages of applying fuzzy logic or knowledge based systems?

son más apropiados para el control de un sistema, manejo de datos cognitivos, la habilidad de representar y manipular conocimientos

before the benefits of intelligent experts systems, fuzzy logic and neural computing.

señalaron la simplicidad, robustez, flexibilidad y facilidad

Computo evolutivo

- muchos problemas reales pueden ser modelados mediante los problemas de optimización
- si el problema puede ser modelado con avances matemáticos diformables

$$F(x_1, y_1, z)$$

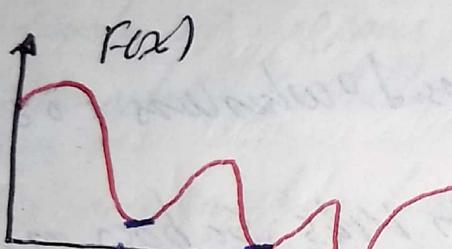
- Se puede encontrar el mínimo o máximo usando otras fórmulas basadas en gradientes

$$\nabla F = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right) >$$

- No todos los problemas pueden ser modelados mediante fórmulas basadas en gradientes

- Fórmulas no diformables,
- más de un óptimo simultáneo
- Fórmulas muchas óptimas locales

Fórmulas no diformables



Punto de inflexión óptimo global

$$\nabla F = 0$$

Son varios algoritmos que pertenecen a la computación evolutiva, algunos de los más comunes son:

ant colony optimization

artificial immune systems

genetic algorithms

ant genetic programming

swarm intelligence

{ Evolutiva Algoritmos

Algoritmos evolutivos

Los elementos básicos son

- Crear un ambiente \rightarrow problema a resolver
- Hogar poblaciones de individuos \rightarrow Selección de individuos
- El ambiente hace que algunos individuos son más aptos que otros para sobrevivir y reproducirse Optimal del individuo

Ejemplo general de un DE

- Iniciarizar los poblaciones
- Evolucionar cada individuo
- Repetir

- Crear padres
- Recombinar padres
- Mutar descendientes
- Evolucionar cada individuo
- Seleccionar nuevos individuos

• lo que difieren es el tipo de DE de otros es su representación y operadores

Gentoo algorithms Usan un orden compuesto por elementos de un alfabeto fijo

Evolutive strategies Usan vectores de variables reales

Evolutive programming Usan un estado binario

Genetic programming Usan árboles

Representación

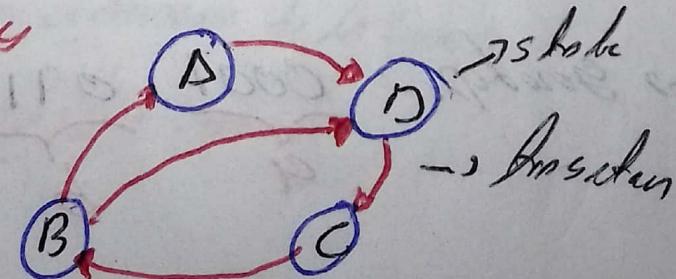
Problema

$$\min f(x,y) = \frac{x^2 + y^2}{4000} - \cos(x) \cos\left(\frac{y}{\sqrt{2}}\right) + 1$$

Gentoo algorithms 10011001101011 alfabeto {0, 1}

Evolutive strategies $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, x_1, x_2 \in \mathbb{R}$

Evolutive programming



Representación: considerar elegir un punto entre el punto de falso o el punto de las soluciones o el problema

Ejercicio: proponer una representación para el siguiente problema

1	2	3	4	5
5	4	3	2	1
3	2	1		

1	2	3	4	5
5	4	3	2	1
3	2	1		

1	1	3	4	5
5	4	3	2	1

Y individuos visitados
individuo visitado

Fórmula de optimización

- Representa las reglas o complejos de problemas
- Usan una medida de utilidad de los individuos en un problema
- Generalmente como resultado un valor real
- Permite comparar soluciones candidatas
- También se le conoce como función objetivo

Fórmulas fáciles

$$F(x) \rightarrow PR$$

1	2	3	4	5
5	4	3	2	1
3	2	1		

$$F(1, 2, 3, 5, 4) \rightarrow PR \text{ optimizado}$$

Mecanismos de selección de programación

Son mecanismos que permiten preservar o lo público para generar mejores soluciones. Un individuo se considera un "padre" si es elegido para crear o otro individuo. La selección de padres es generalmente estocástica.

Mecanismos de variación

Permite crear nuevos individuos

Recombinación: Permite combinar partes de dos padres para crear descendientes.

Mutación: Un individuo se modifica sin alterar su estructura en el problema.

Algoritmos evolutivos

Problemas: Encuentro óptimo de la función

$$F(x,y) = (x-1)^2 + (y+3)^2$$

variables: 2 Tds: usa 8 bits para representar entero
Type: Real

5	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Entero

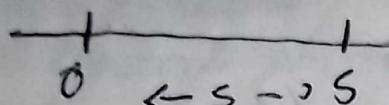
1 00000000 *número*
00000100 *positivo*

$s = 0$ positivo

$s = 1$ negativo

5 00000000 número = -5

5 11111111 número = 5



$\frac{s}{127}$

≈ 0.039

0000 000 $\rightarrow 0$

0000 001 $\rightarrow 0.039$

0000 010 $\rightarrow 0.078$

Algoritmo Evolutivo

Población inicial (N)

clase v $\downarrow N/2$ $w/2$
padres / madres

hijos N

notar un par de cambios

(S-1)

new padres e hijos $2N$

selección N individuos

Repetir hasta que no sea más

necesario

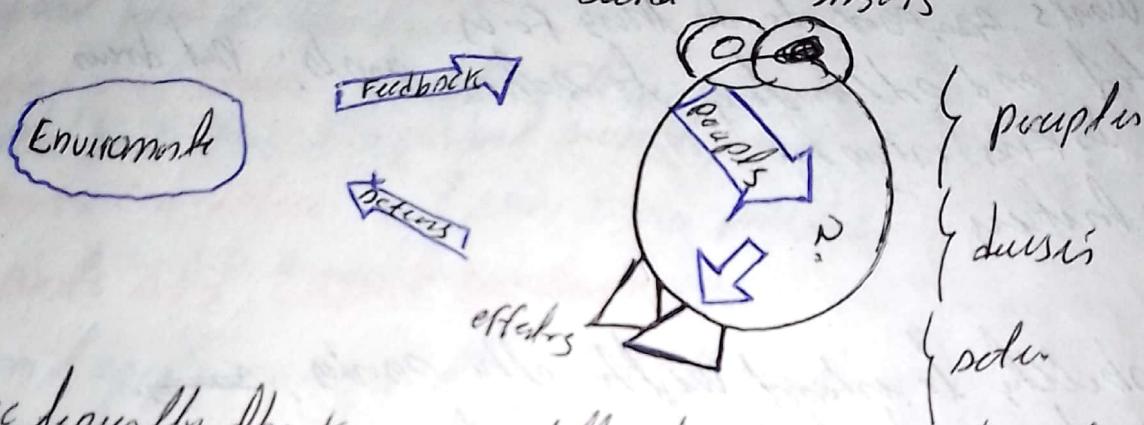
$$x_1^2 + x_2^2 = 1$$

What is an Agent?

The main point about agents is they are autonomous: capable of independent action

- Thus, an agent is a computer system capable of autonomy within its environment, in order to achieve its delegated goals
- We think of an agent as being as a decoupled component interacting with its environment.

sense → decide → act - sense - decide ... sensors



We usually think of an intelligent agent as exhibiting 3 types of behavior: react, pro-act, sense

Accessible vs. Inaccessible An accessible environment is one in which the agent can obtain complete, accurate, up-to-date information about the environment's state. Most real-world environments (including, for example, the everyday physical world and the internet) are not accessible in this sense.

Deterministic vs. non-deterministic A deterministic environment is one in which any action has single guaranteed effect. There is no uncertainty about the state that will result from performing an action.

Static vs. dynamic A static environment is one that can be assumed to remain unchanged except by the performance of actions by the agent. In contrast, a dynamic environment is one that has other forces operating on it, and which brings changes in areas beyond the agent's control. The physical world is a highly dynamic environment, as is the Internet.

Discrete vs. continuous An environment is discrete if the environment's number of actions and percept is it

Permanently If a program's environment is guaranteed to be fixed so program can just execute blindly

The real world is not like that: much movements are dynamic
 software is hard to build for dynamic domains: programs must take into
 account possibility of failure and itself whether is worth executing
 A random succession is one that maintains an ongoing feedback while is
 with out movement. It responds to changes that occur in it (in time or
 the response is successful)

Proactiveness

Reacting to an environment is easy

But we generally wants ~~to~~ want to do many things
 More goal oriented and self-motivating to achieve goals; but draws
 So little energy: Is kind of similar

Recognizing opportunities

Social ability

Agents is the ability to interact with other agents, ~~agents~~
 Agents are cooperative, coordinate and negotiate

At least it means the ability to communicate

Cooperation is working together as team to achieve a shared goal
 often prompted either by the fact that no one agent can achieve the
 goal alone, or that cooperation will obtain a better result

Coordination is managing the interdependencies between activities

For example, if there is a non-shareable resource that you want to use
 and I want to use, then we need to coordinate

Negotiation is the ability to work agreements on matters of communication
 typically involves offer and counter-offer, with compromises made by
 participants

Some other properties mobility, versatility, Diversification, Robustness,
 learning/adaptation

Agents and objects

One agent does objects by another man.²

- object:

- encapsulates some state
- contains local message passing
- has methods, corresponding to operations that may be performed on this state

Agents are subconscious: agents embody stronger notion of action than objects, and in particular they decide for themselves whether, or not to perform an action or request from another agent

Agents are smart capable of flexibly (react, predict, search) behaviour - the OB model has nothing to say about such types of behavior

Agents are active: not passive service providers

Agents and expert systems

Don't agents just expert systems - by another name?

Expert systems typically disentangled "experts" about some (abstract) domain of discourse

Differences between Agents and expert systems

- agents are situated in an environment
- agents act

Some real-time (temporally pros control) expert systems are agents

Intelligent agents and AI

Don't agents just the AI project?

isn't building intelligent systems that can ~~understand~~ ultimately understand natural language, recognise and understand signs use common sense, think, understand, etc - all of which are very hard

so, don't we need to solve all of AI to build an agent

thus building an agent, we simply want a system that can choose the right action to perform. Typically in a limited domain

We do not have to solve all the problems at once but it can be solved
in little bit by one other way

Properties of Environment

Stochastic vs Deterministic

A determinstic environment is one in which the next possible complete
action will be the action that maximizes value.
Most real-life complex environments are stochastic. The more
deterministic environments are simple. The more stochastic and
complex it is.

Deterministic vs non-deterministic

As we have already mentioned, a deterministic environment is one in which
any action has single guaranteed effect after some time according to which
state that will result from previous action.

The physical world can be all stochastic and sometimes be regarded as non-
deterministic environment.

No deterministic environments present much problems for the agent
desire.

Episodic vs non episodic

In an episodic environment, the performance of the agent is dependent on a
number of discrete episodes, with no link between the performance of an
agent in different surroun.

Episodic environments are simple form the agent develops perspective
because the agent can learn about what needs to perform best only on the
current episode. It need not worry about interaction between plus
before episodes.

Static vs dynamic

A static environment is one that can be assumed to remain unchanged
except by the performance of actions by the agent.

A dynamic environment is one that has other processes operates on it and
have changes in ways beyond the agent's control.

The physical world is a highly dynamic environment.

Desarrollo vs continuo

Un desarrollo es discreto si tiene un fin fijo, finito número de actos y progresivo. Russell y Norvig dan ejemplos como el de un agente que recorre un laberinto, y el de un coche dirigido por GPS.

Agentes vs sistemas naturales

When explaining human activity, we use statements like the following:

John took his umbrella because she believed it was raining and she wanted to stay dry.

- These statements make use of a folk psychology, by which human behavior is predicted and explained by attributing abilities such as believing, wanting, hoping, fearing,

Abstract architecture for agents

- Describir el entorno mediante unión de un tipo de descripción abstracta de estados $E = \{e_1, e_2, \dots, e_n\}$
- Agente en poseerá la lista de posibles actos availables. Entonces, qué transformar el estado del entorno

$$A_c = \{a_1, a_2, \dots, a_m\}$$

La acción, r , de un agente en su entorno es una secuencia de estados y entorno

$$r: e_0 \xrightarrow{a_0} e_1 \xrightarrow{a_1} e_2 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} e_n$$

R Runs

let

- R sea el conjunto de todos los posibles secuencias finitas ($\cup E$ y A_c)
- R^{sc} sea el subconjunto de R que no contenga self loops
- R^c sea el subconjunto de R^{sc} que no cumple con ciertas condiciones

Entornos

Los estados transforman los datos representativos del entorno

$$\tau: R^c \rightarrow \delta(E)$$

Note the environment are
history dependent
non-deterministic

If $\tau(\epsilon) = 0$, then more possible success states to or successor function has order

An environment E_{env} is thus a triple $E_{\text{env}} = \langle E, \epsilon_0, \tau \rangle$ where E is set of environments states, ϵ_0 is initial state, and τ is state transition function

Agents

Agent is a function which maps runs to actions

$$A_g : R^E \rightarrow \Delta_c$$

This agent makes a decision about what action to perform based on the history of the system that it has witnessed so far.

Let S_g be the set of all agents

Systems

A system is open contains an agent and an environment

Any system will have associated with it a set of possible runs; we denote the set of runs of agent A_g in environment E_{env} by $R(A_g, E_{\text{env}})$

Assume $R(A_g, E_{\text{env}})$ contains only runs that have ended

Formally, a sequence

$$(c_0, l_0, e_1, d, l_1, e_2, \dots)$$

represents a run of an agent A_g in environment $E_{\text{env}} = \langle E, \epsilon_0, \tau \rangle$ if:

- 1 c_0 is the state initial
- 2 $l_0 = A_g(c_0)$; and
- 3 for $n > 0$

~~effect~~

$e_0 \in \tau(l(c_0, l_0, e_1, d, l_1, \dots))$ when
 $d_0 = A_g((c_0, l_0, \dots, c_0))$

Perceptual Agents

Some agents decide what to do without referring to their history - they base their decision making entirely on the present, with no reference at all to the past - we call such agent perceptual.

$$\text{act} : E \rightarrow A$$

\rightarrow the percept is a perceptive agent

$$\text{act} : \begin{cases} \text{off} & \text{if } C = \text{longer than OK} \\ \text{on} & \text{otherwise} \end{cases}$$

Perception

Now introduce percept system



The see function is the agent's ability to observe its environment, whereas the action function represents the agent's decision making process.

Output of the see function is a percept

$$\text{See} : E \rightarrow \text{per}$$

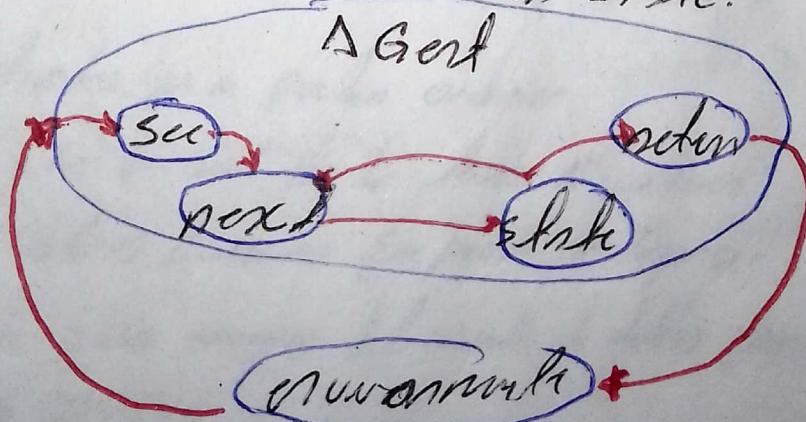
which maps environment states to percept and action is now a function

$$\text{act} : \text{per}^* \rightarrow A$$

which maps sequences of percepts to actions

Agent with state

We now consider agents that memorize state:



Perception

These agents have some internal state s_t , which is typically used to review information about the current state and history.

Let F be the set of all internal states of the agent.

The perception function see is a state-based and is unchanged.

Actions

The action-selection function action is now defined as a mapping

$$\text{action} : F \rightarrow A$$

From internal states to actions

Next state Function

A factor next is introduced, which maps an internal state and action to an internal state:

$$\text{next} : F \times A \rightarrow F$$

Agent loop

agent starts in some initial state s_0

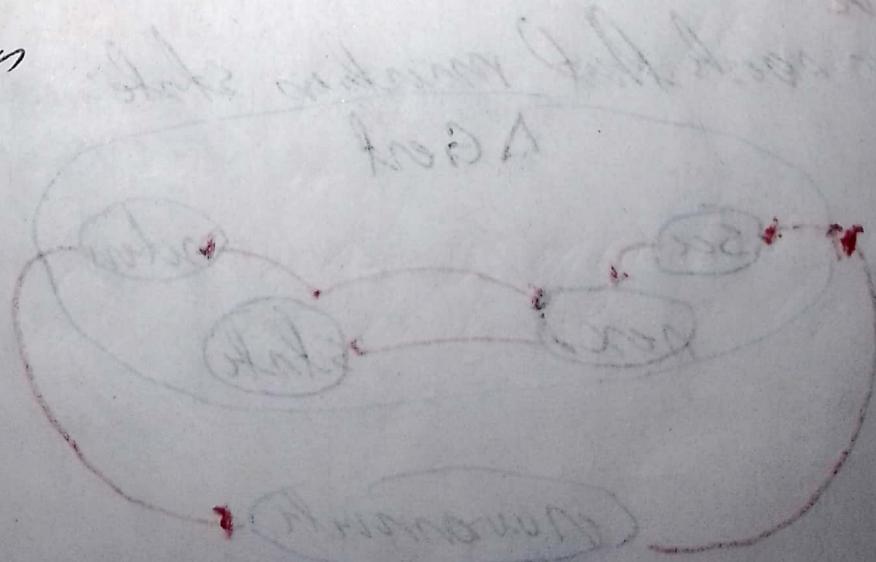
1 repeat forever

- observe environment state, and generate a percepts through see

- update internal state via next factor

- select action via $\text{action}(.)$

- perform action



Mantenimiento

sprocedimientos: En comprensión, se dice que los códigos "Sprocedimientos", si es posible modificar en cambio en su comportamiento, y este cambio tiene que ser descompuesto para algunos de los que se manejan en el sistema.

El Sprocedimiento o código "ROZON" para un propósito, es decir, puede deliberaamente

Conjunto de datos

Es la primera parte para la manejo de datos, y es una colección de datos obtenidos de algún dominio específico.

Normalmente, los datos se pueden ver como un tablón, donde cada columna es un atributo, y cada renglón corresponde con una observación o muestra.

Una muestra es una observación que se registra como un renglón (que siempre viene con un número) en el conjunto de datos. En los sistemas, se puede considerar como serían: Instancias, podríamos, objeto o registro.

Un **conjunto de datos**, se dice que esto es elegible de si uno de sus atributos permite categorizar a las instancias en él.

Un **conjunto de datos** se dice que no es elegible de si ninguno de sus atributos permite categorizar a las instancias en él.

Nota: Generalmente, los datos son elegibles por el experto, y se han visto en sus sistemas.

Típos de atributos Obsérvese los valores de los otros conjuntos de datos siguientes:

Típos de atributos

NOMINAL → ejemplos

ORDINAL → ejemplos que se pueden ordenar

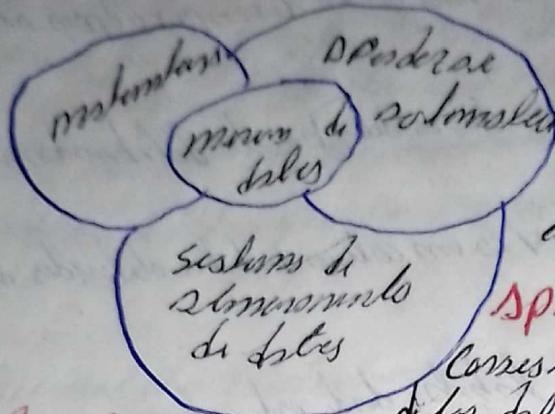
BINARIO → ejemplos que solo pueden tener 2 valores

INTERVALO → Valores numéricos que pueden ordenarse

Predicción se refiere a usar campos del mundo de datos para predecir los valores futuros o valores de otros campos

Los muros de datos incluyen una zona de conocimiento

- **No homogéneos**



los primeros se utilizan para predecir futuros
comportamientos probables, leyes, tendencias,
y su uso extensivo de la probabilidad.
estos datos, al igual que los resultados
en muros de datos

Aprendizaje automático

Sistemas de almacenamiento de datos

Corresponde básicamente a los modelos o patrones
de los datos. Es una zona de los muros de datos
que promueve otras componentes aprendizaje

automático.

los muros primarios de los muros de datos se crean principalmente en bases
de datos estructurados, donde las tablas se crean datos de partes
heterogéneas.

Proceso de extracción de datos

El proceso general para la extracción de conocimiento es el siguiente:

- Selección del conjunto de datos
- Pre-procesamiento de los datos
- Selección y exploración de muros de datos
- Extracción de conocimiento
- Interpretación y conclusión de los datos

Los muros de **aprendizaje supervisado** utilizan conjuntos de datos etiquetados
utilizan conjuntos de datos etiquetados para generar un modelo capaz de realizar
predicciones sobre la base de datos abiertos.

Los muros de **aprendizaje no supervisado** generan modelos que describen
la distribución de los datos, o punto de acuerdo de los datos no etiquetados