INTELIGENCIA DE NEGOCIO
2020-2021

- **Tema 1. Introducción a la Inteligencia de Negocio**
- **Tema 2. Minería de Datos. Ciencia de Datos**
- **Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales**
- **Tema 4. Preparación de Datos**
- **Tema 5. Modelos de Agrupamiento o Segmentación**
- **Tema 6. Modelos de Asociación**
- **Tema 7. Modelos Avanzados de Minería de Datos.**
- **Tema 8. Big Data**

Modelos avanzados de Minería de Datos

Objetivos:

- Analizar diferentes extensiones del problema de clasificación clásico de acuerdo a diferentes problemas reales que plantean un nuevo escenario en los problemas de clasificación.
- Introducir brevemente estas extensiones.

Inteligencia de Negocio

TEMA 7. Modelos Avanzados de Minería de Datos

1. Clases no balanceadas/equilibradas

2. Características intrínsecas de los datos en clasificación

3. **Clasificación no Estándar: Más allá del aprendizaje clásico**

4. Detección de anomalías

5. Deep Learning

6. Análisis de Sentimientos

Nuevos problemas de clasificación

- TÉCNICAS DE CLASIFICACIÓN: Árboles decisión: C4.5, Sistemas basados en reglas, Clasificación basada en instancias (k-NN, …), regresión logística, SVM, RNN, One-class, modelos probabilísticos,

- Técnicas avanzadas: Ensembles (Bagging, Boosting), Pruning, …
- Multiclases: OVA, OVO

Discretización
Selección de características
Selección de instancias
Reducción de la dimensionalidad

- Datos imperfectos: Valores perdidos, Ruido de clase y variable
- Clases no equilibradas

Baja densidad de datos– small disjuncts
Overlapping entre clases
Dataset Shift - Particionamiento
Medidas de Complejidad

Múltiples etiquetas
Múltiples instancias
Ranking de clases
Clasificación ordinal y monotónica, semisupervisada, multiview learning, …

Clasificación no estándar: más allá del aprendizaje clásico

- Introducción. Aprendizaje supervisado estándar
- Más allá de aprendizaje clásico
- Problemas de aprendizaje no estándar
- Aprendiendo de los nuevos datos
- Comentarios finales

Clasificación no estándar: más allá del aprendizaje clásico

- **Introducción. Aprendizaje supervisado estándar**
- Más allá de aprendizaje clásico
- Problemas de aprendizaje no estándar
- Aprendiendo de los nuevos datos
- Comentarios finales

Aprendizaje supervisado estándar

Involucra

- Entrada, *x*: **vector** de características
- Salida, *y*: **escalar** en espacio discreto (clasificación) o continuo (regresión)
- Dataset: conjunto de parejas *(x, y)*

Aprendizaje: generar una **función** ***f*** que aproxime $f(x) = y$.

Clasificación. Ejemplo

- Ejemplo: **Diseño de un Clasificador para *Iris***
  - Problema simple muy conocido: *clasificación de lirios*.
  - Tres clases de lirios: *setosa*, *versicolor* y *virginica*.
  - Cuatro atributos: *longitud* y *anchura* de *pétalo* y *sépalo*, respectivamente.
  - 150 ejemplos, 50 de cada clase.
  - Disponible en http://www.ics.uci.edu/~mlearn/MLRepository.html

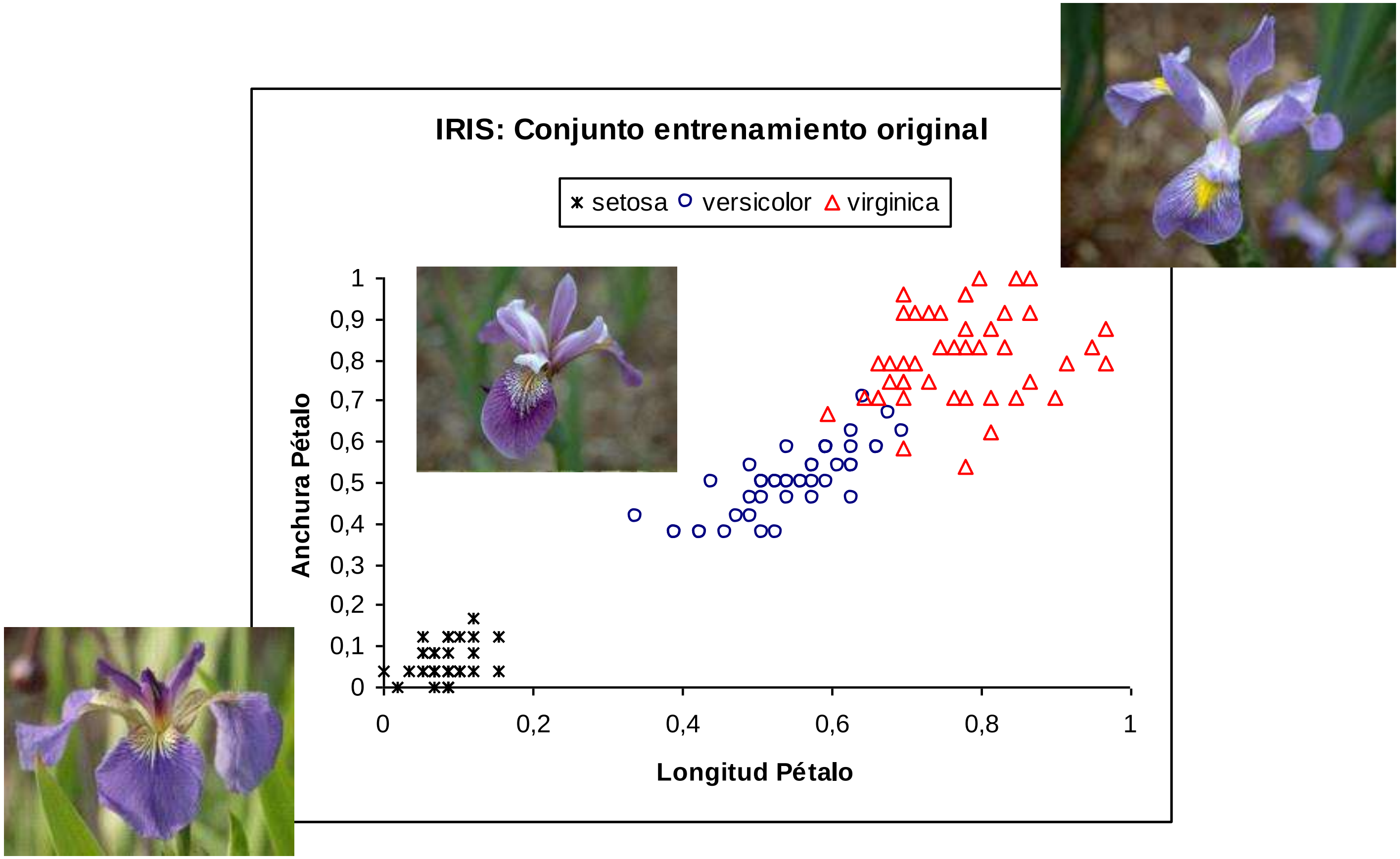
setosa

versicolor

virginica

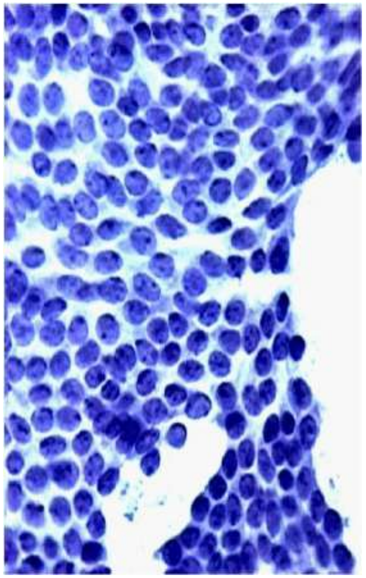
Clasificación. Ejemplo

Ejemplos de conjuntos seleccionados sobre *Iris:*



Clasificación. Ejemplo

Wisconsin Breast Cancer: Predict malignant/benign



Attribute name	Description
RADIUS	*Mean of distances from center to points on the perimeter*
TEXTURE	*Standard deviation of grayscale values*
PERIMETER	*Perimeter of the mass*
AREA	*Area of the mass*
SMOOTHNESS	*Local variation in radius lengths*
COMPACTNESS	*Computed as: perimeter^2/area – 1.0*
CONCAVITY	*Severity of concave portions of the contour*
CONCAVE POINTS	*Number of concave portions of the contour*
SYMMETRY	*A measure of the nuclei's symmetry*
FRACTAL DIMENSION	*'Coastline approximation' – 1.0*
DIAGNOSIS (Target)	*Diagnosis of cell sample: malignant or benign*

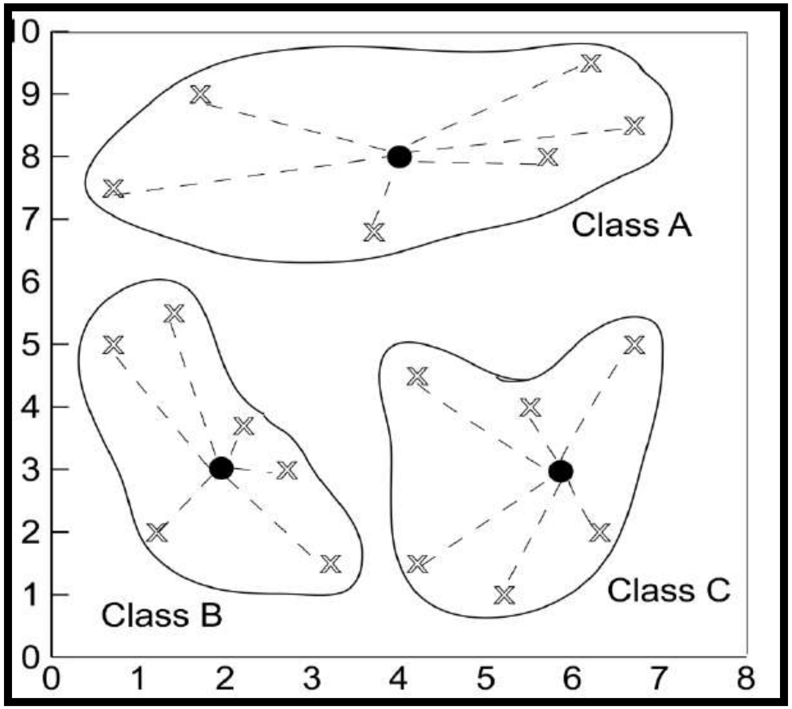
Clasificación. Ejemplo

Handwritting recognition.
Assign a digit from 0 to 9.

Clasificación. Ejemplo

Basado en Particiones

Basado en Distancias



Clasificación no estándar: más allá del aprendizaje clásico

- Introducción. Aprendizaje supervisado estándar
- **Más allá de aprendizaje clásico**
- Problemas de aprendizaje no estándar
- Aprendiendo de los nuevos datos
- Comentarios finales

¿Cómo clasificamos esta escena imagen?

Lago

Bosque

Montaña

Múltiples etiquetas

¿Cómo clasificamos esta escena imagen?

Lago

Bosque

Montaña

Múltiples etiquetas

Motivation: Multi-label objects

Clasificación de noticias

Negocios

Politica

Viajes

Noticias del Mundo

Entretenimiento

Noticias Locales

… …

Aprendizaje supervisado no estándar

Surge por **variaciones** en las estructuras de entrada y salida que **no se ajustan** al problema estándar.

Clasificación no estándar:
más allá del aprendizaje clásico

- Introducción. Aprendizaje supervisado estándar
- Más allá de aprendizaje clásico
- **Problemas de aprendizaje no estándar**
- Aprendiendo de los nuevos datos
- Comentarios finales

Variaciones no estándares

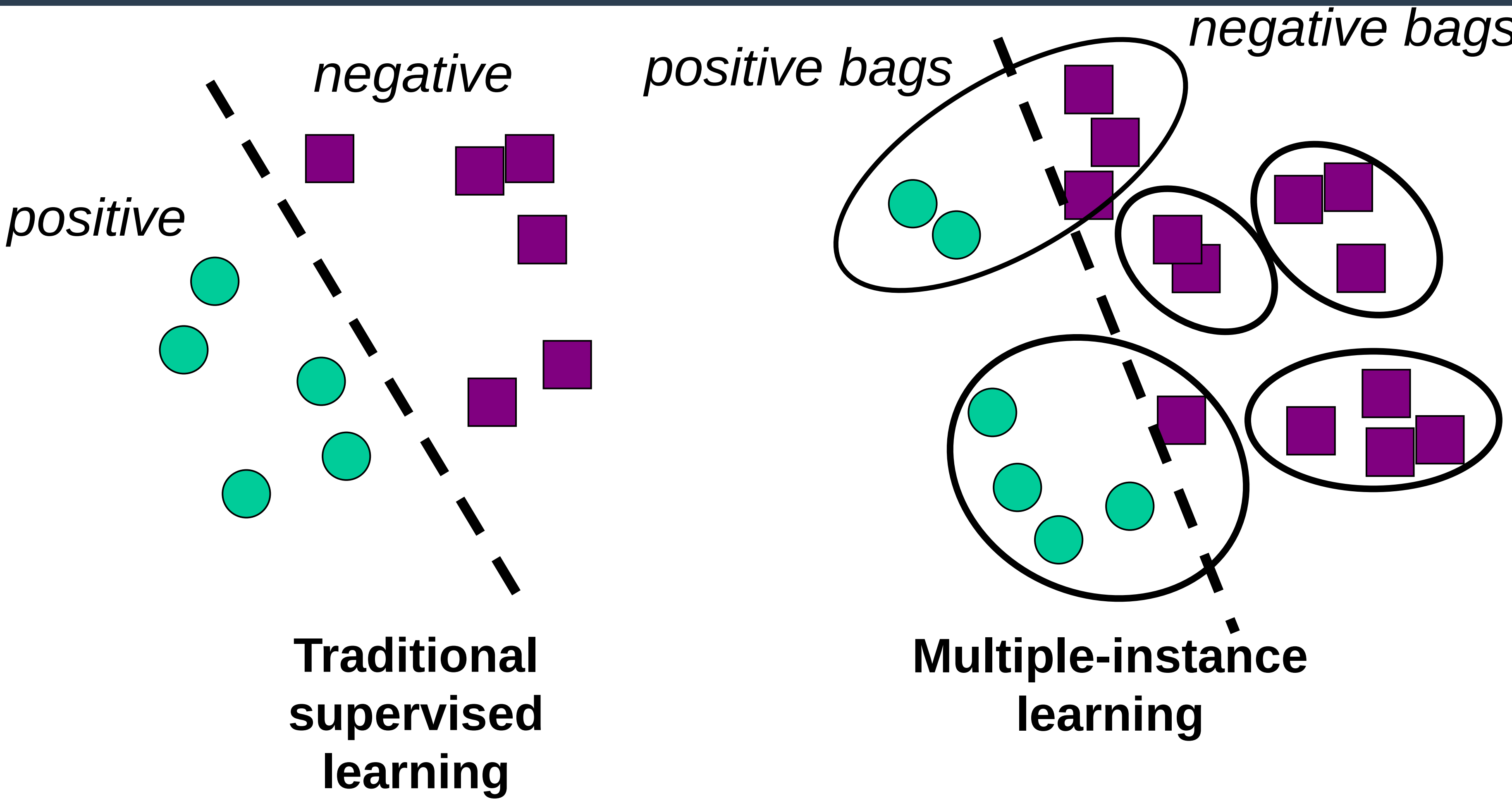
Tipos de variación comunes:

- **Entradas múltiples**: cada instancia contiene un conjunto de vectores de características

Multi-instancia

Multi-vista

Multi-instancia



[Dietterich et al. 1997]

Multi-instancia

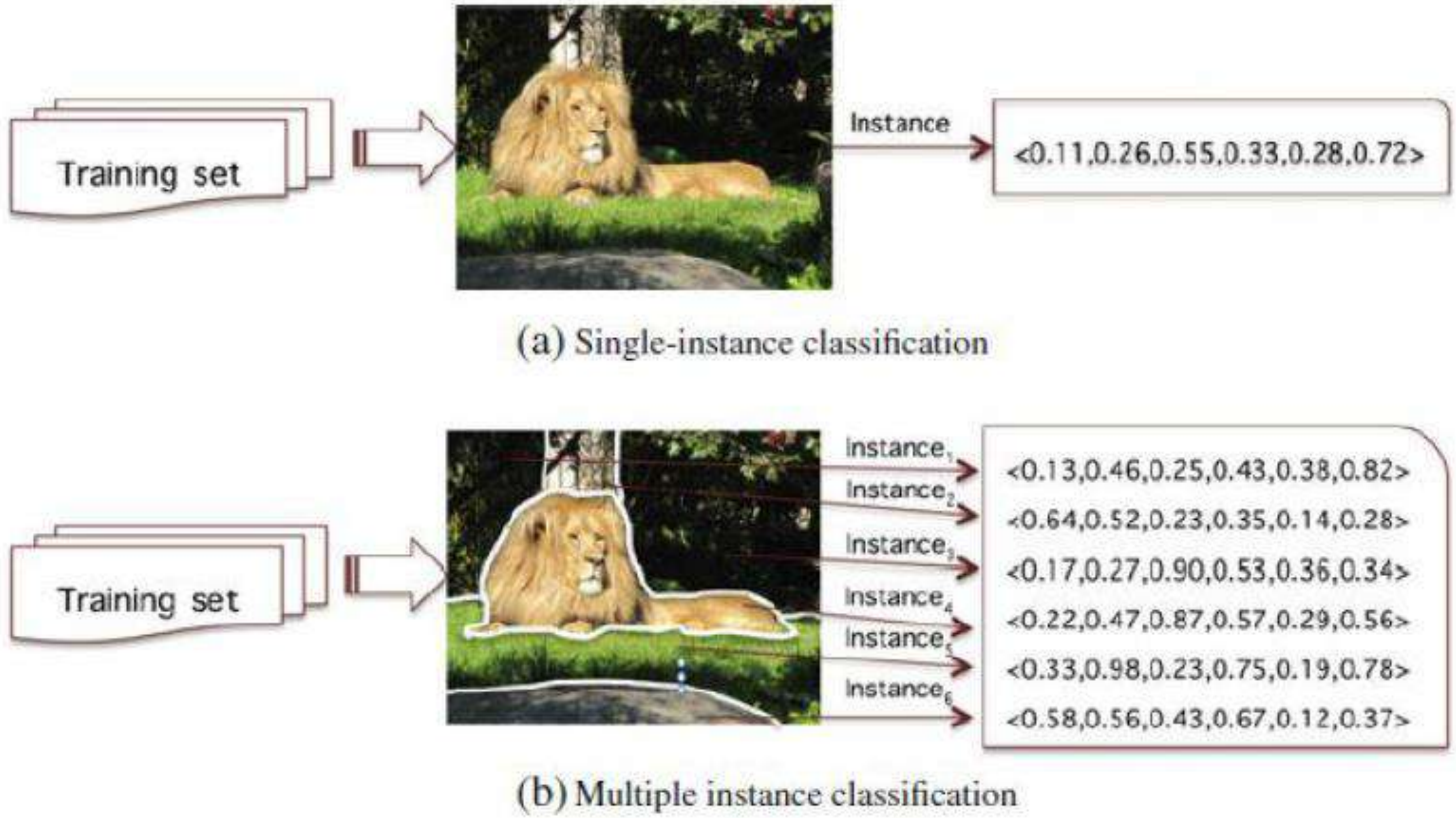


Fig. 3.2 Training data set for classification task

Multi-instancia: predicción de actividad de fármacos

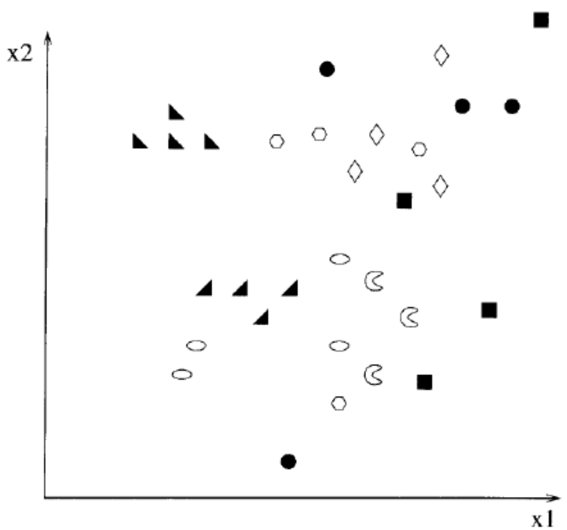
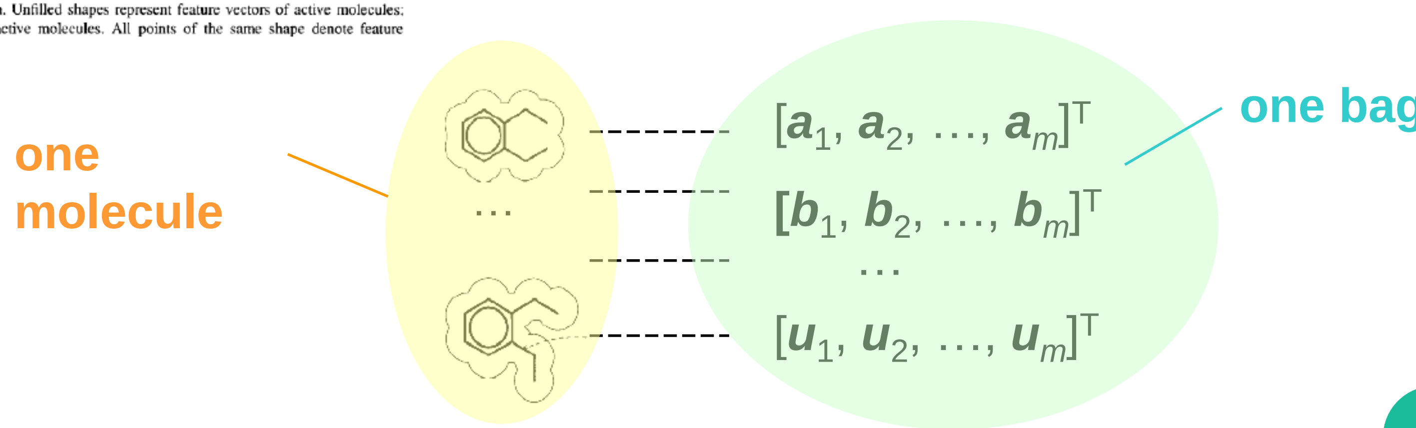
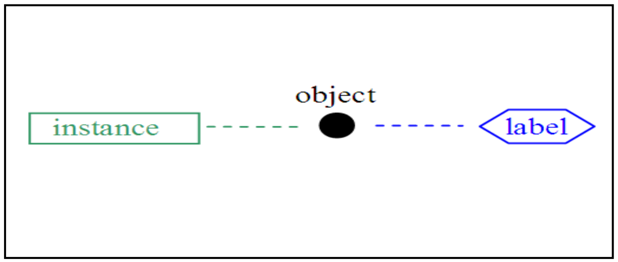


Fig. 14. A multiple instance learning problem. Unfilled shapes represent feature vectors of active molecules; filled shapes represent feature vectors of inactive molecules. All points of the same shape denote feature vectors of the same molecule.

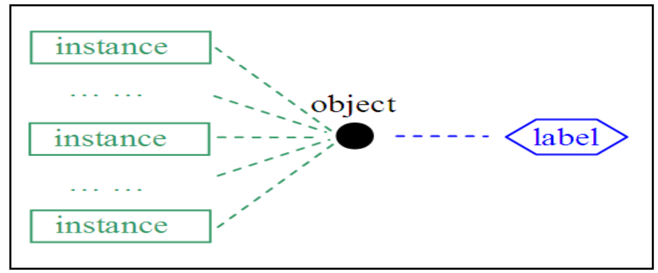
Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, 89(1-2), 31-71.



Multi-instancia

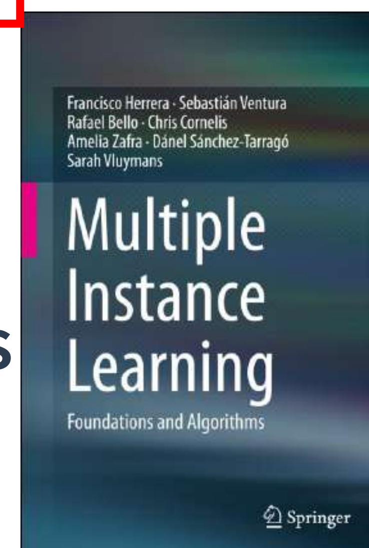


Traditional supervised learning

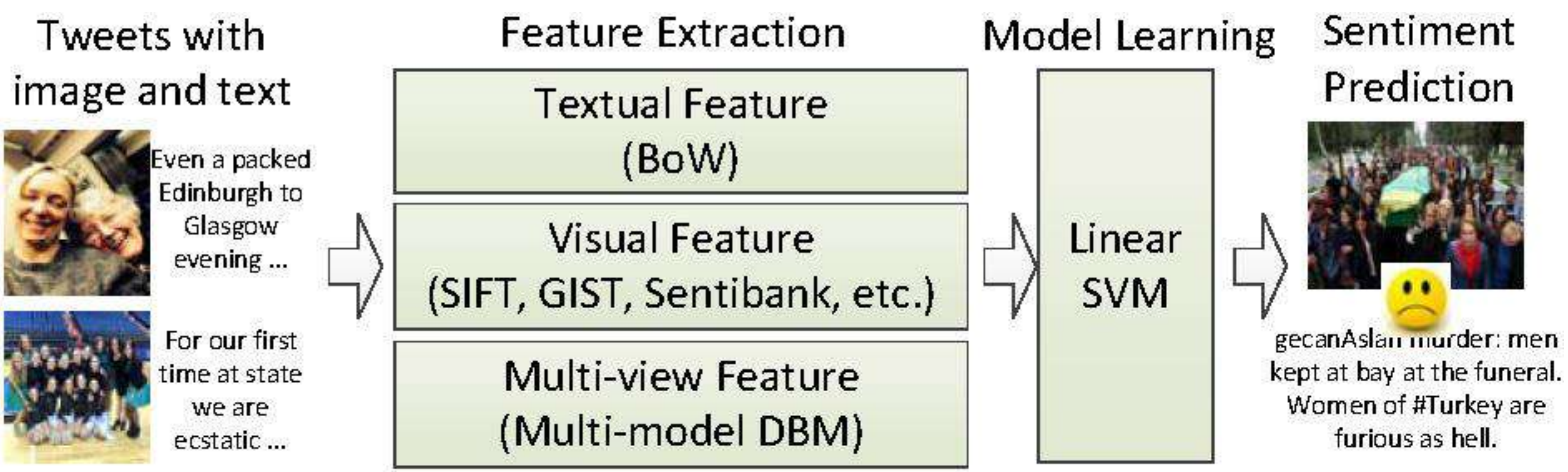


Multi-instance learning

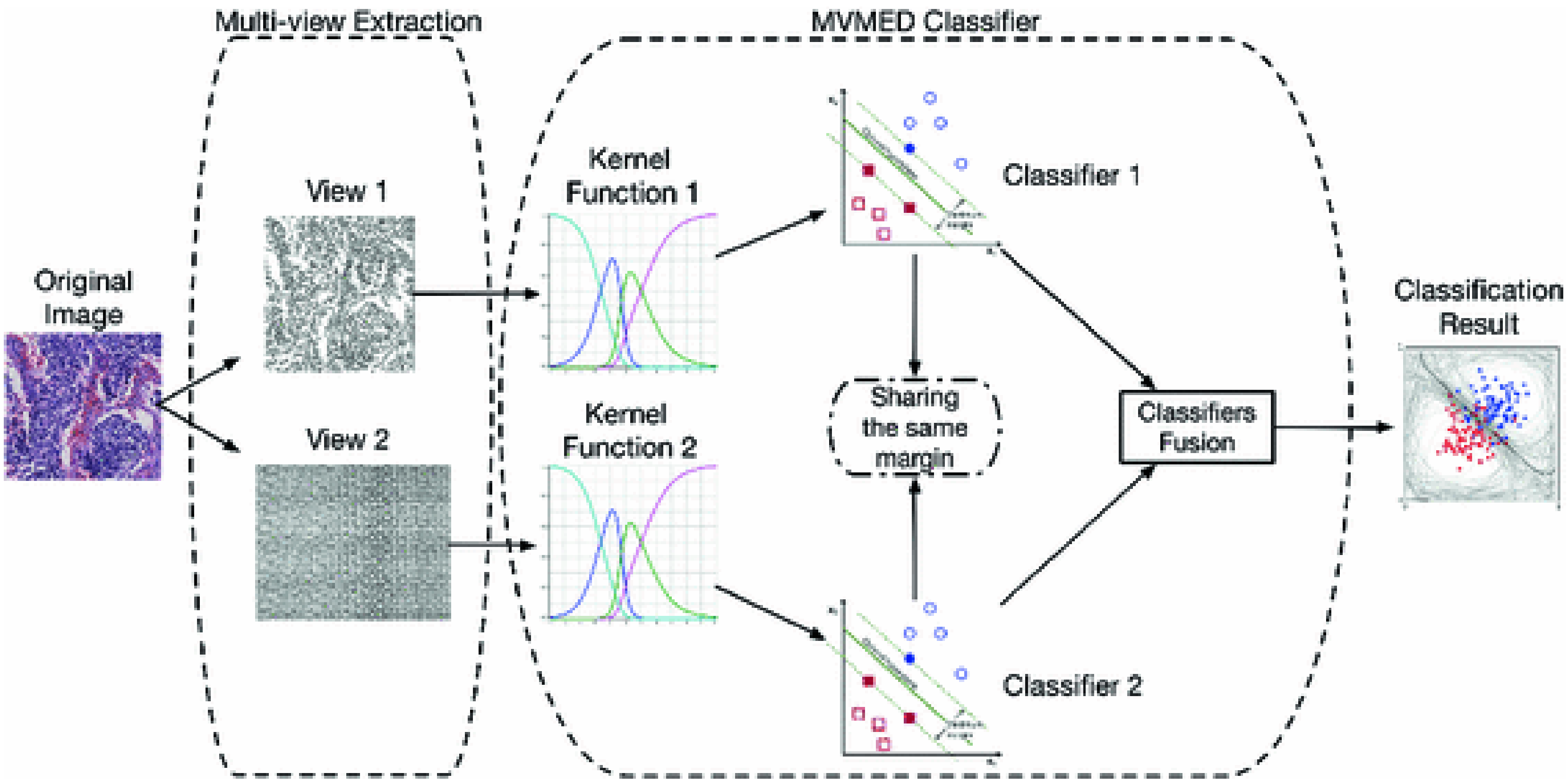
- Predicción de actividad de fármacos
- Clasificación de imágenes por segmentos
- Predicción de bancarrotas



Multi-vista: Análisis de sentimientos



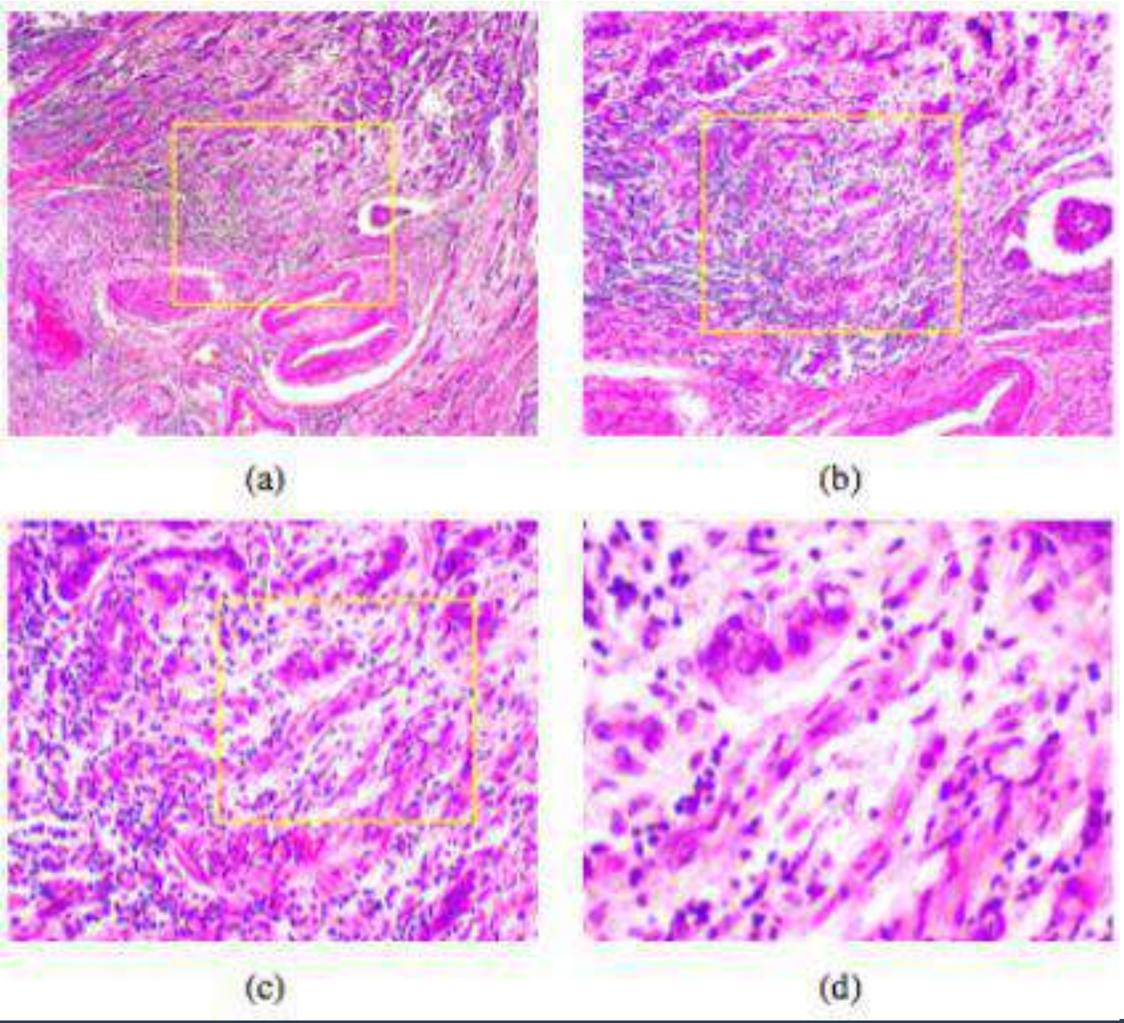
Multi-vista: Análisis de sentimientos



Multi-vista: Corales

Multi-vista: Corales

Multi-vista: Imágenes de cáncer



Muestras de tejido correspondientes a cáncer de mama maligno:

a)Magnificación 40x
b)Magnificación 100x
c)Magnificación 200x
d)Magnificación 400x

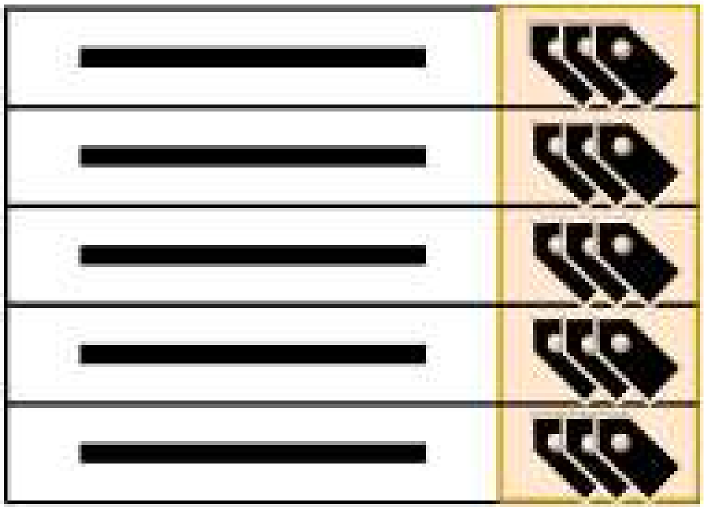
Multi-vista

- Categorización de texto multilingüe
- Detección de caras con varias poses
- Localización de usuarios de redes WiFi
- Clasificación de anuncios (imágenes+texto)
- Clasificación de imágenes con varias vistas basadas en colores y texturas

Variaciones no estándares

Tipos de variación comunes:

- **Salidas múltiples**: cada instancia está asociada a varias etiquetas



Clasificación multi-etiqueta

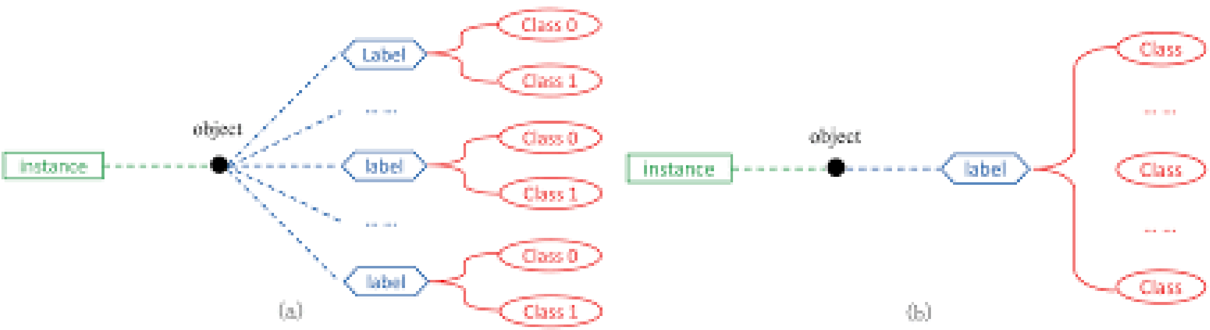
Label distribution learning

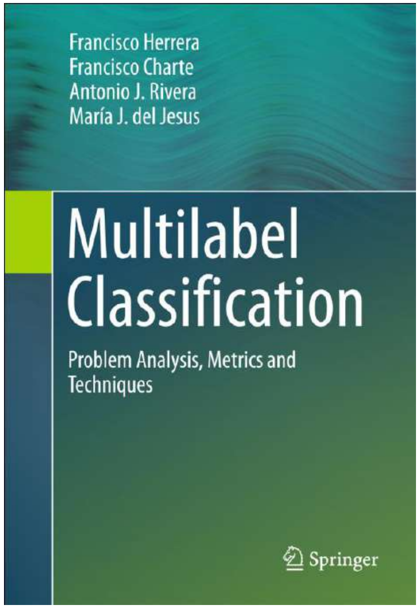
Regresión multi-salida

Label ranking

Clasificación multi-dimensional

Multi-etiqueta





Lago

Bosque

Montaña

Múltiples etiquetas

Multi-etiqueta: asignación de tags a preguntas

(a) Chemistry (b) Chess (c) Coffee

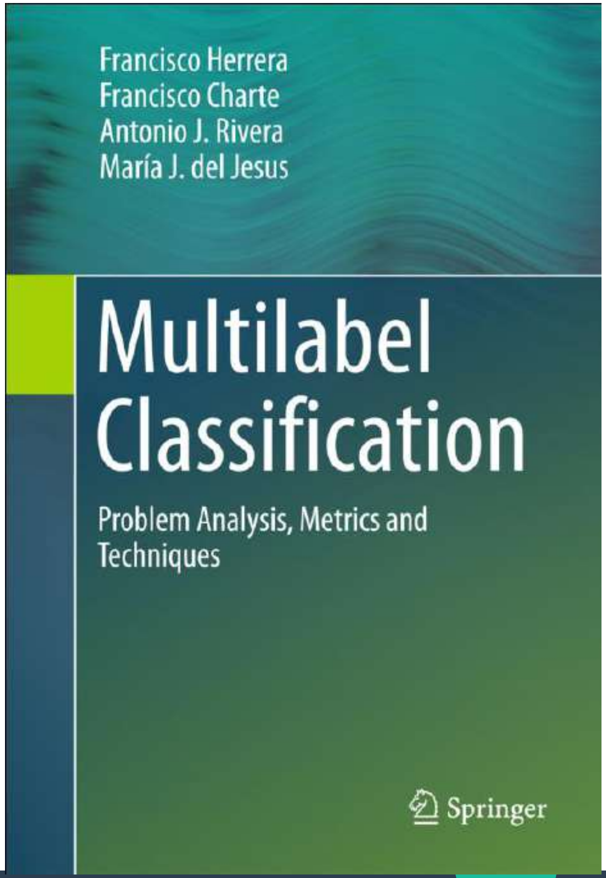
(d) Cooking (e) Computer science (f) Philosophy

Fig. 3: Word clouds for each forum after text mining.

Charte, F., Rivera, A. J., del Jesus, M. J., & Herrera, F. (2015, September). QUINTA: a question tagging assistant to improve the answering ratio in electronic forums. In EUROCON 2015-International Conference on Computer as a Tool (EUROCON), IEEE (pp. 1-6). IEEE.

Multi-etiqueta

- Etiquetado de textos y documentos
- Etiquetado de imágenes y escenas
- Asignación de tags a preguntas en foros
- Clasificación de proteínas



Regresión multi-salida: modelado de ecosistemas

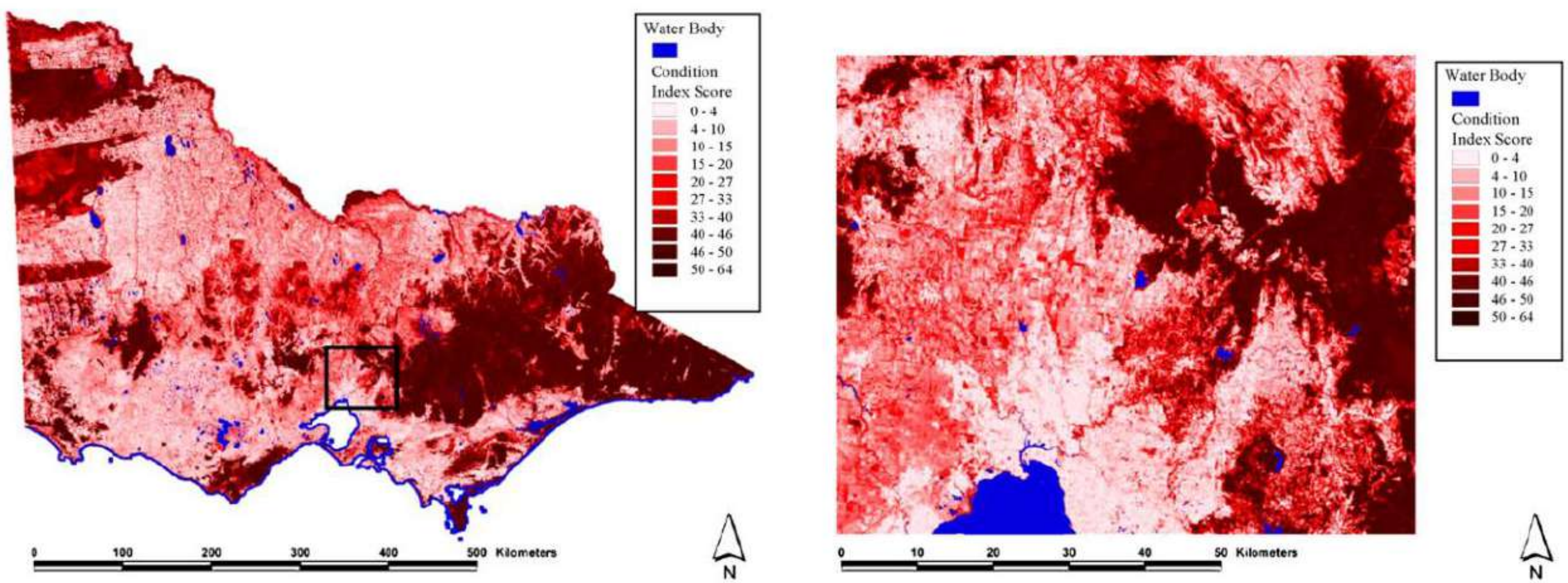


Fig. 7. Map of the condition of indigenous remnant vegetation in Victoria derived from the application of the random forests of MTRTs (left-hand side figure). The dark bordered rectangular inset refers to the area represented at higher resolution at the right-hand side figure.

La *Condition Index Score* es una suma de las 7 variables objetivo: *Weeds, Tree Canopy, Understorey, Recruitment, Logs, Litter* y *Large Tree*.

Kocev, D., Džeroski, S., White, M. D., Newell, G. R., & Griffioen, P. (2009). Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition. Ecological Modelling, 220(8), 1159-1168.

Regresión multi-salida

- Condición de vegetación en ecosistemas naturales
- Predicción de espectros auditivos en túneles de viento
- Estimación de parámetros biofísicos en imágenes de teledetección

LDL: detección de emociones en caras

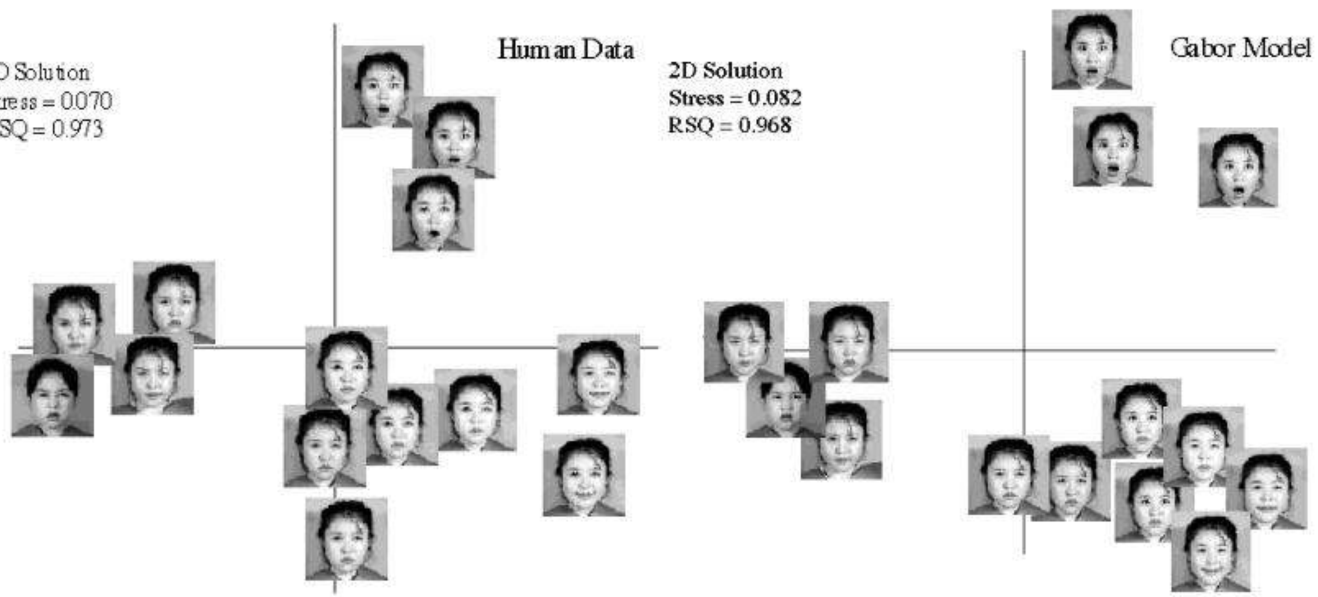


Figure 5. nMDS solutions for Gabor and semantic rating similarities (data from pilot study).

Lyons, M., Akamatsu, S., Kamachi, M., & Gyoba, J. (1998, April). Coding facial expressions with gabor wavelets. In Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on (pp. 200-205). IEEE.

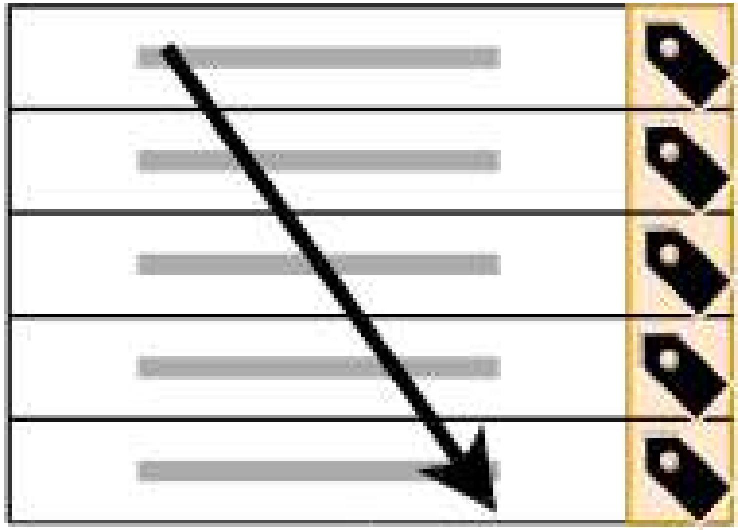
Label Distribution Learning

- Análisis de niveles de expresión de genes en levaduras
- Descripción de emociones en expresiones faciales

Variaciones no estándares

Tipos de variación comunes:

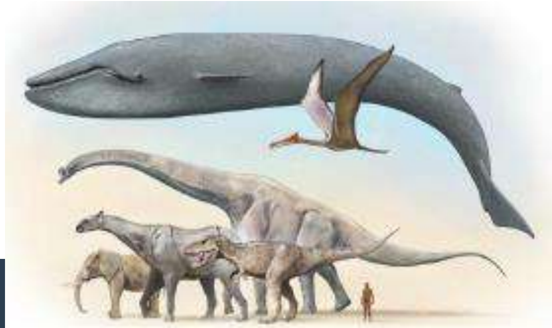
- **Relaciones de orden**: las entradas o las salidas están ordenadas, restricciones de monotonía



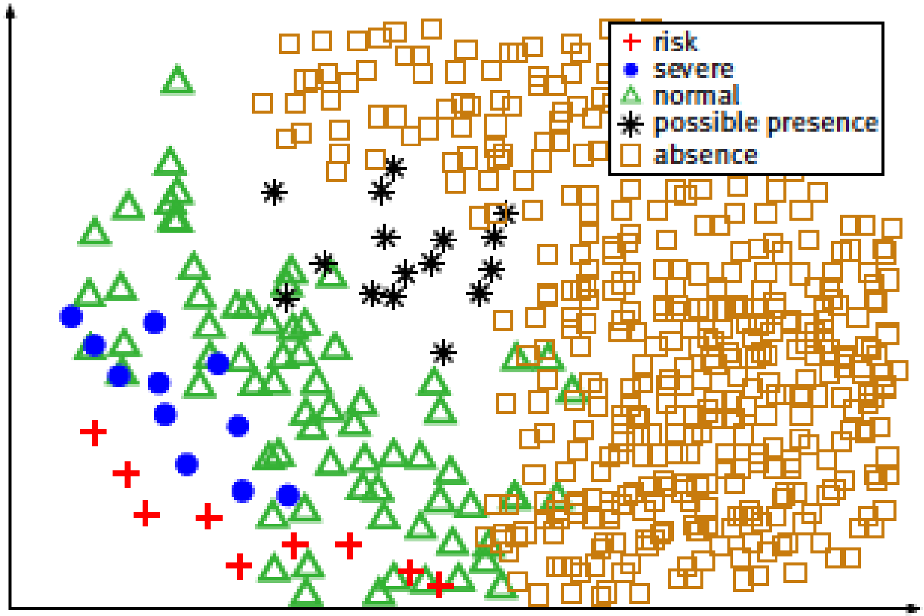
Clasificación ordinal

Clasificación monotónica

Regresión isotónica



Clasificación ordinal: predicción de dolencias



Dolencias basadas en una escala ordinal:
{C_1 = riesgo, C_2 = severa,
C_3 = normal, C_4 = posible presencia, C_5 = ausencia}

Clasificación ordinal

- Clasificación de textos, imágenes
- Investigación médica
- Análisis de riesgos
- Estimación de edad
- Mercados financieros

Regresión isotónica: predicción de gasto de combustible

Attribute	Type	Sign
mpg	continuous	target
cylinders	multi-valued discrete	−
displacement	continuous	−
horsepower	continuous	−
weight	continuous	−
acceleration	continuous	+
model year	multi-valued discrete	+
origin	multi-valued discrete	+

La relación de monotonía entre cada variable predictora y la objetivo tiene un signo (positivo = relación directa, negativo = relación inversa)

Clasificación monotónica, regresión isotónica

- Análisis de satisfacción de consumidores
- Predicción de precios de viviendas
- Evaluación de riesgo de bancarrota
- Predicción de cáncer
- Predicción de gasto de combustible

Variaciones no estándares

Tipos de variación comunes:

- **Información parcial**: en la información de entrenamiento hay instancias sin etiquetar

Positive-unlabeled learning

Aprendizaje semi-supervisado

o no hay instancias de algunas clases

Zero-shot learning

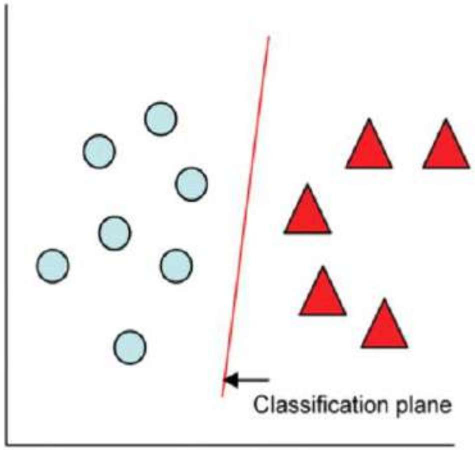
One-class classification

One-shot learning

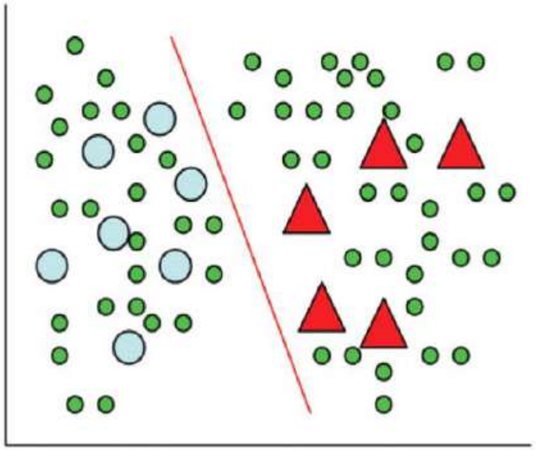
Aprendizaje semi-supervisado

Aplicaciones donde etiquetar datos es costoso:

- Clasificación de páginas web
- Reconocimiento de habla
- Secuencias de proteínas
- ...

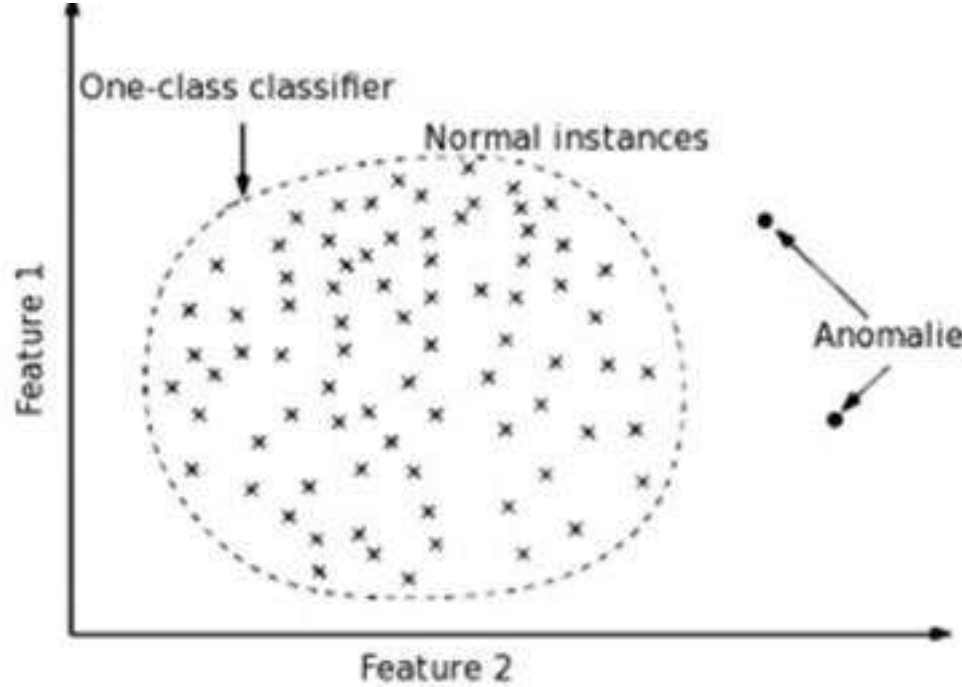


Supervised Learning


Semi-Supervised Learning

One-class (1-NN)

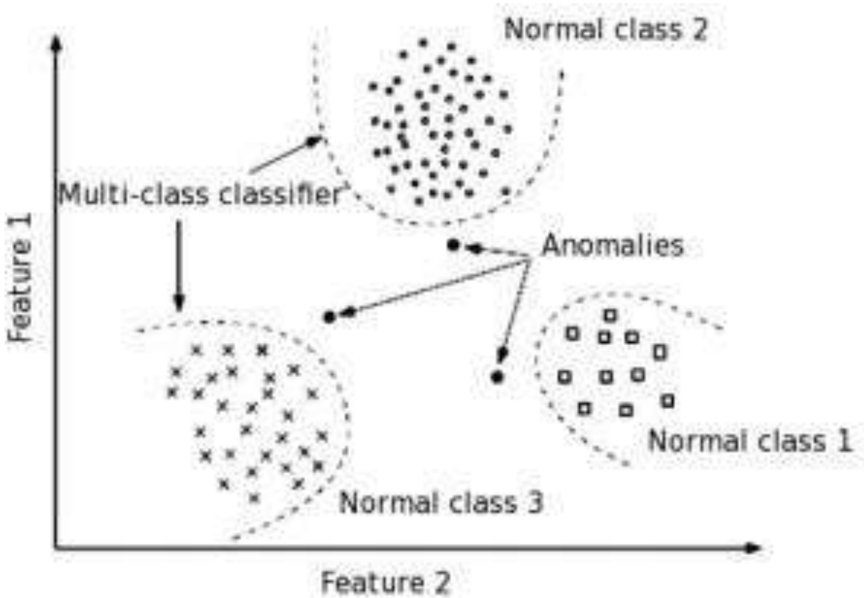
One-class 1-NN **Es un algoritmo semi-supervisado que aprende una función de decisión para la detección de novedades: la clasificación de nuevos datos como similares o diferentes al conjunto de entrenamiento.**

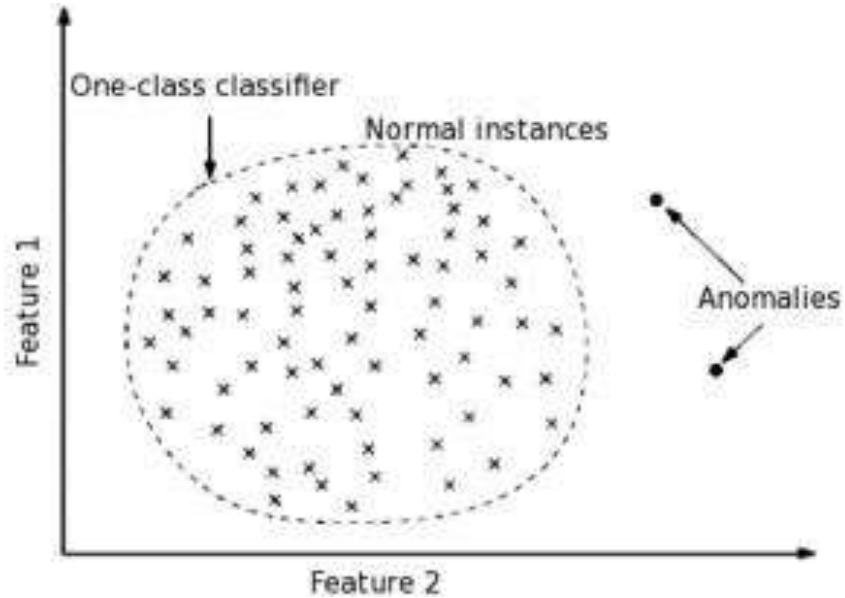


One-class para abordar la detección de fallos

Industrial damage detection
Credit Card Fraud
Cyber Intrusions
Faults in a system

Multiclasificación vs One-class classification





Ejemplo: La detección de daños industriales (la detección de diferentes fallos y averías en sistemas industriales complejos)

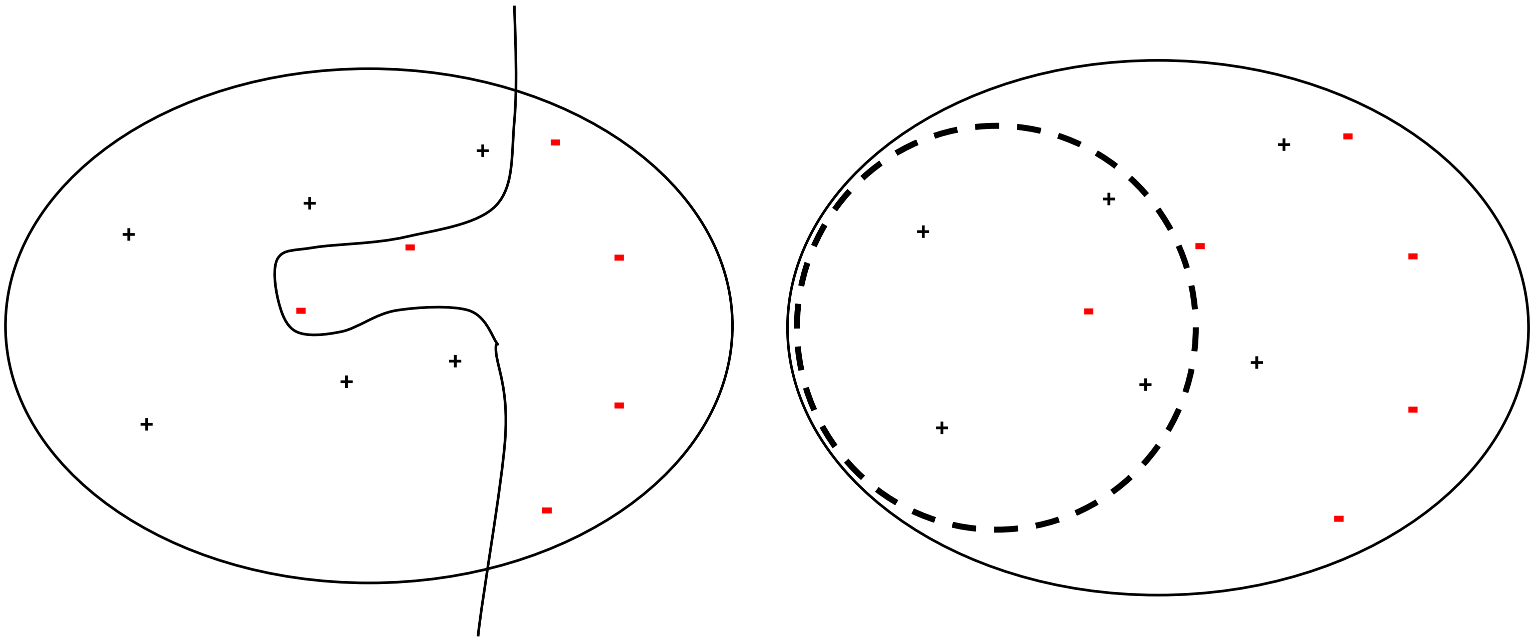
Variaciones no estándares

Otros problemas:

- **Descubrimiento de subgrupos**: buscar relaciones entre las entradas y las salidas
- **Learning to rank**: buscar un ranking o un orden de las instancias
- **Clasificación ordinal circular**: las clases forman una secuencia circular

Descubrimiento de subgrupos

Classification vs. Subgroup discovery



Construcción de reglas de descripción de subgrupos individuales con el objetivo de encontrar patrones interesantes en ejemplos de clases objetivo

Clasificación no estándar: más allá del aprendizaje clásico

- Introducción. Aprendizaje supervisado estándar
- Más allá de aprendizaje clásico
- Problemas de aprendizaje no estándar
- **Aprendiendo de los nuevos datos**
- Comentarios finales

Aprediendo de los nuevos datos

- Depende de la calidad de los datos (ruido, incompletos ...)
- Nuevas métricas
- Se transforman los problemas en problemas clasicos y/o se adaptan los algoritmos
    - kNN extensiones
    - Decision tree – extensiones
    - SVM ….
- Paquetes de software (mldr, mulan, ...)

Clasificación no estándar: más allá del aprendizaje clásico

- Introducción. Aprendizaje supervisado estándar
- Más allá de aprendizaje clásico
- Problemas de aprendizaje no estándar
- Aprendiendo de los nuevos datos
- **Comentarios finales**

Comentarios finales

- La adaptación de las técnicas clásicas de aprendizaje requiere conocerlas para adaptar sus componentes

- Existen herramientas software que permiten utilizar estos nuevos algoritmos.

Queda un gran reto en el diseño de estas nuevas librerías

Inteligencia de Negocio

TEMA 7. Modelos Avanzados de Minería de Datos

1. Clases no balanceadas/equilibradas

2. Características intrínsecas de los datos en clasificación

3. Clasificación no Estándar: Más allá del aprendizaje clásico

4. Detección de anomalías

5. Deep Learning

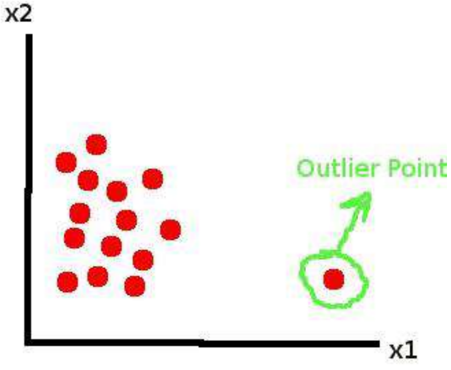
6. Análisis de Sentimientos

Outline

- What are anomalies?
- Anomaly Detection: Taxonomy
- Nearest Neighbor Based Techniques
- One-Class to tackle the Fault Detection
- Concluding Remarks

What are anomalies?

- Anomaly is a pattern in the data that does not conform to the expected behavior
- Also referred to as outliers, exceptions, peculiarities, surprise, etc.
- Anomalies translate to significant (often critical) real life entities
  - Cyber intrusions
  - Credit card fraud
  - Faults in a System



What are anomalies?

Real World Anomalies

- Credit Card Fraud
  - An abnormally high purchase made on a credit card

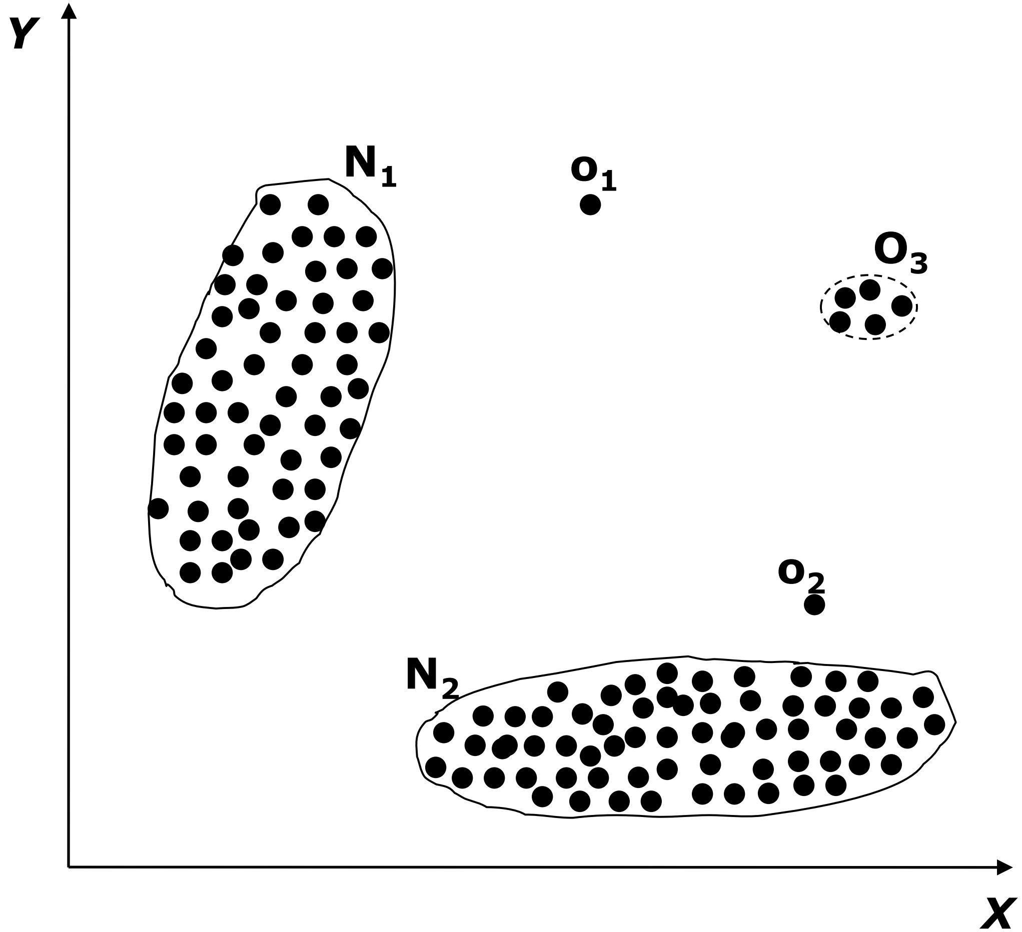
- Cyber Intrusions
  - A web server involved in *ftp* traffic

- Faults in a system
  - An abnormal values from sensors

What are anomalies?

Simple Example

- N_1 and N_2 are regions of normal behavior
- Points o_1 and o_2 are anomalies
- Points in region O_3 are anomalies



What are anomalies?

Related problems

- Rare Class Mining (high imbalanced classes)
- Chance discovery
- Novelty Detection
- Exception Mining
- Noise Removal
- Black Swan*

* N. Talleb, The Black Swan: The Impact of the Highly Probable?, 2007

What are anomalies?

Key Challenges

- Defining a representative normal region is challenging
- The boundary between normal and outlying behavior is often not precise
- The exact notion of an outlier is different for different application domains
- Availability of labeled data for training/validation
- Malicious adversaries
- Data might contain noise
- Normal behavior keeps evolving

What are anomalies?

Aspects of Anomaly Detection Problem

- Nature of input data
- Availability of supervision
- Type of anomaly: point, contextual, structural
- Output of anomaly detection
- Evaluation of anomaly detection techniques

What are anomalies?

Type of Anomaly

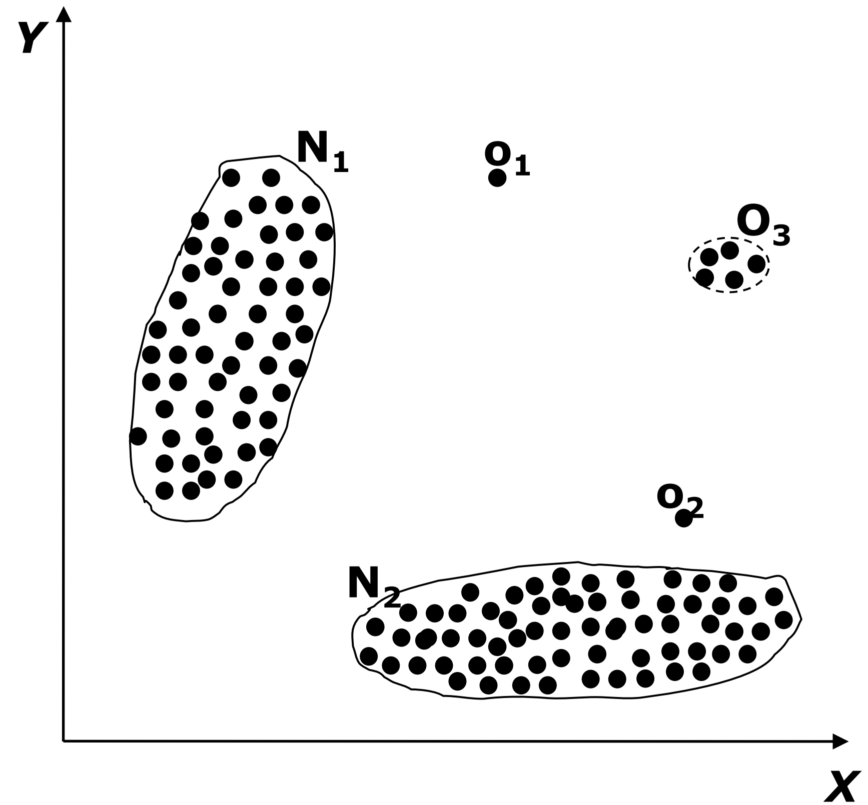
- Point Anomalies
- Contextual Anomalies
- Collective Anomalies

V. CHANDOLA, A. BANERJEE, and VI. KUMAR. **Anomaly Detection: A Survey**
ACM Computing Surveys, Vol. 41, No. 3, Article 15, Publication date: July 2009.
http://doi.acm.org/10.1145/ 1541880.1541882

What are anomalies?

Point Anomalies

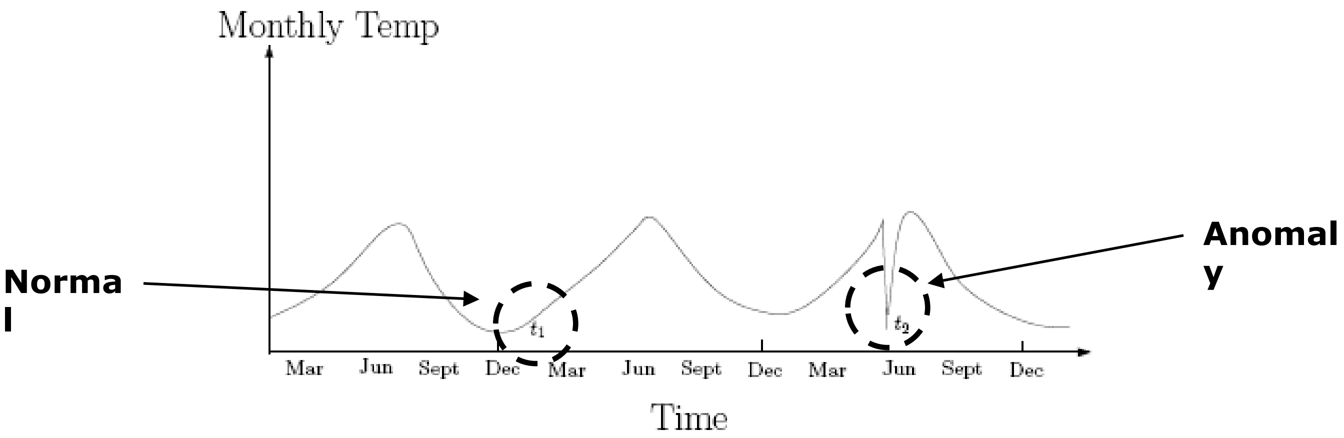
- An individual data instance is anomalous w.r.t. the data



What are anomalies?

Contextual Anomalies

- An individual data instance is anomalous within a context
- Requires a notion of context
- Also referred to as conditional anomalies*

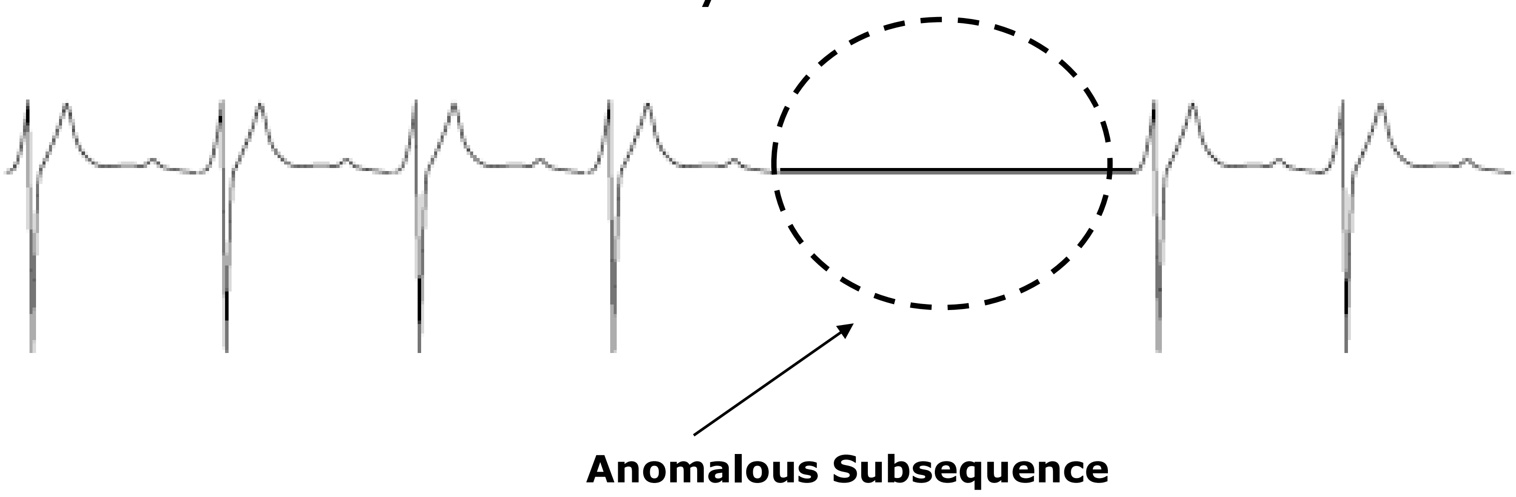


* Xiuyao Song, Mingxi Wu, Christopher Jermaine, Sanjay Ranka, Conditional Anomaly Detection, IEEE Transactions on Data and Knowledge Engineering, 2006.

What are anomalies?

Collective Anomalies

- A collection of related data instances is anomalous
- Requires a relationship among data instances
  - Sequential Data
  - Spatial Data
  - Graph Data
- The individual instances within a collective anomaly are not anomalous by themselves



What are anomalies?

Applications of Anomaly Detection

- Network intrusion detection
- Insurance / Credit card fraud detection
- Healthcare Informatics / Medical diagnostics
- Industrial Damage Detection
- Image Processing / Video surveillance
- Novel Topic Detection in Text Mining
- …

What are anomalies?

Industrial Damage Detection

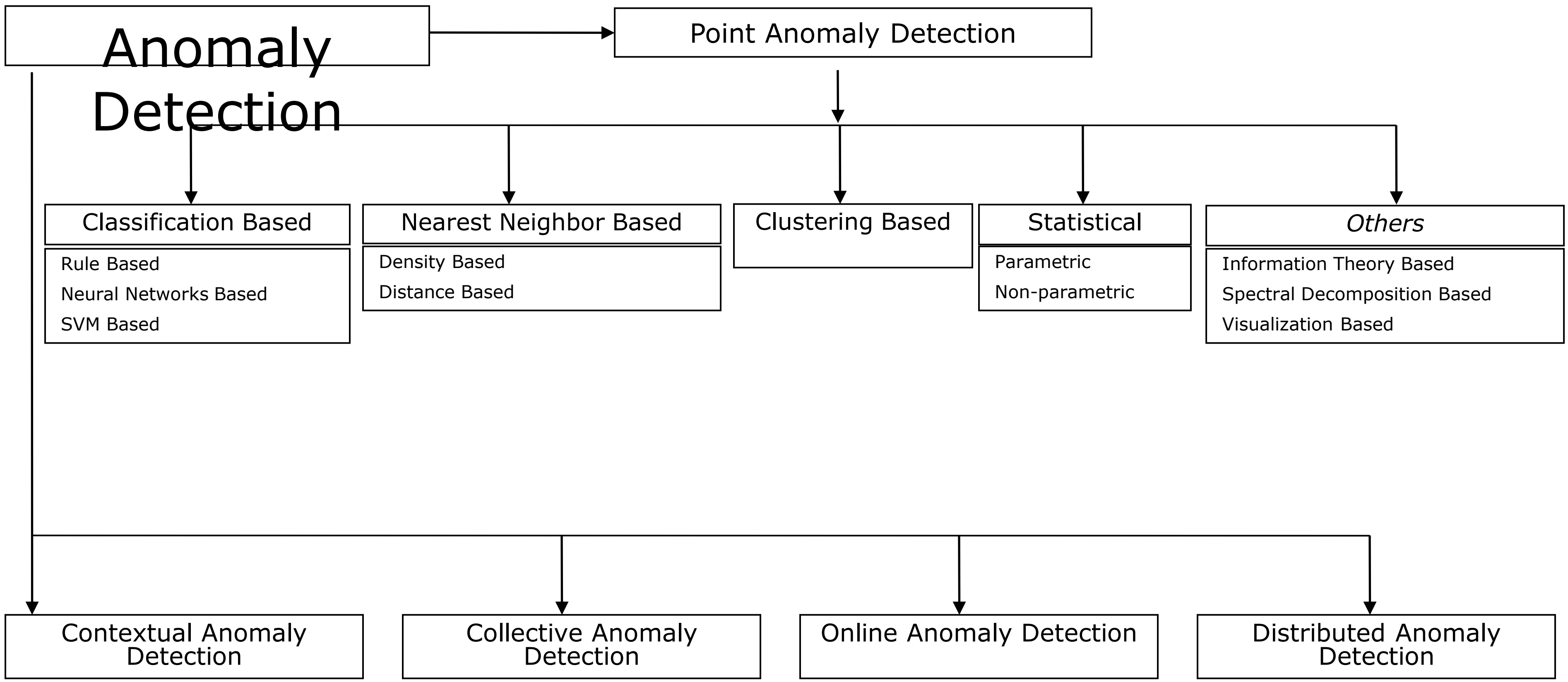
- Industrial damage detection refers to detection of different faults and failures in complex industrial systems, structural damages, intrusions in electronic security systems, suspicious events in video surveillance, abnormal energy consumption, etc.
  - Example: Wind Turbines
    - Fault detection / Anomalies in performance
  - Example: Aircraft Safety
    - Anomalous Aircraft (Engine) / Fleet Usage
    - Anomalies in engine combustion data
    - Total aircraft health and usage management

- **Key Challenges**
  - **Data is extremely huge, noisy and unlabelled**
  - **Most of applications exhibit temporal behaviour**
  - **Detecting anomalous events typically require immediate intervention**

Outline

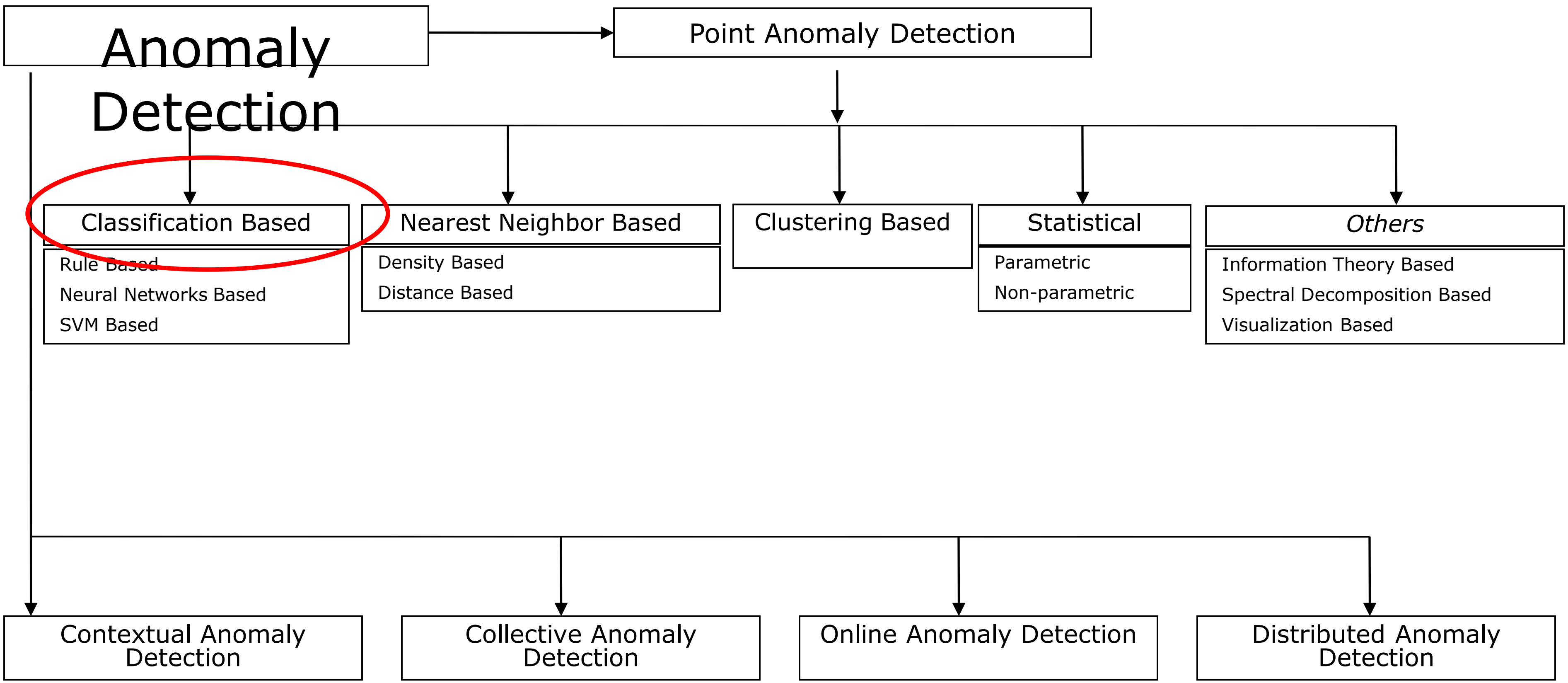
- What are anomalies?
- Anomaly Detection: Taxonomy
- Nearest Neighbor Based Techniques
- One-Class to tackle the Fault Detection
- Concluding Remarks

Anomaly Detection: Taxonomy



* Anomaly Detection – A Survey, Varun Chandola, Arindam Banerjee, and Vipin Kumar, ACM Computing Surveys, Vol. 41, No. 3, Article 15, Publication date: July 2009.

Anomaly Detection: Taxonomy



* Anomaly Detection – A Survey, Varun Chandola, Arindam Banerjee, and Vipin Kumar, ACM Computing Surveys, Vol. 41, No. 3, Article 15, Publication date: July 2009.

Anomaly Detection: Taxonomy

Classification Based Techniques

- **Main idea:** build a classification model for normal (and anomalous (rare)) events based on labeled training data, and use it to classify each new unseen event
- Classification models must be able to handle skewed (imbalanced) class distributions
- Categories:
  - *Supervised classification techniques*
    - Require knowledge of both ***normal*** and ***anomaly*** class
    - Build classifier to distinguish between normal and known anomalies
  - *Semi-supervised classification techniques*
    - Require knowledge of ***normal*** class only!
    - Use modified classification model to learn the normal behavior and then detect any deviations from normal behavior as anomalous

Anomaly Detection: Taxonomy

Classification Based Techniques

- **Advantages:**
  - ***Supervised classification techniques***
    - Models that can be easily understood
    - High accuracy in detecting many kinds of known anomalies
  - ***Semi-supervised classification techniques***
    - Models that can be easily understood
    - Normal behavior can be accurately learned
- **Drawbacks:**
  - ***Supervised classification techniques***
    - Require both labels from both normal and anomaly class
    - Cannot detect unknown and emerging anomalies
  - ***Semi-supervised classification techniques***
    - Require labels from normal class
    - Possible high false alarm rate - previously unseen (yet legitimate) data records may be recognized as anomalies

Anomaly Detection: Taxonomy

Supervised Classification Techniques

- Rule based techniques
- Model based techniques
  - Neural network based approaches
  - Support Vector machines (SVM) based approaches
  - Bayesian networks based approaches
- Imbalanced classification
  - Manipulating data records (oversampling / undersampling / generating artificial examples)
  - Cost-sensitive classification techniques
  - Ensemble based algorithms (SMOTEBoost, RareBoost

Anomaly Detection: Taxonomy

Rule Based Techniques

- **Creating new rule based algorithms**
- **Adapting existing rule based techniques**
  - Robust C4.5 algorithm [John95]
  - Adapting multi-class classification methods to single-class classification problem
- **Association rules**
  - Rules with support higher than pre specified threshold may characterize normal behavior
  - Anomalous data record occurs in fewer frequent itemsets compared to normal data record
  - Frequent episodes for describing temporal normal behavior [Lee00,Qin04]
- **Case specific feature/rule weighting**
  - Increasing the rule strength for all rules describing the rare class or features strength for highlighting the minority class.

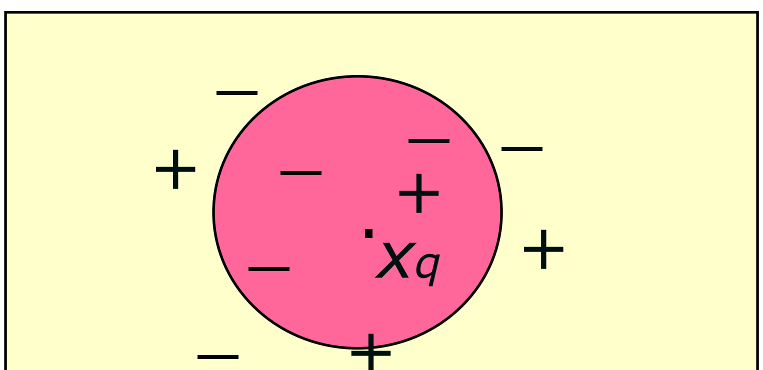
Outline

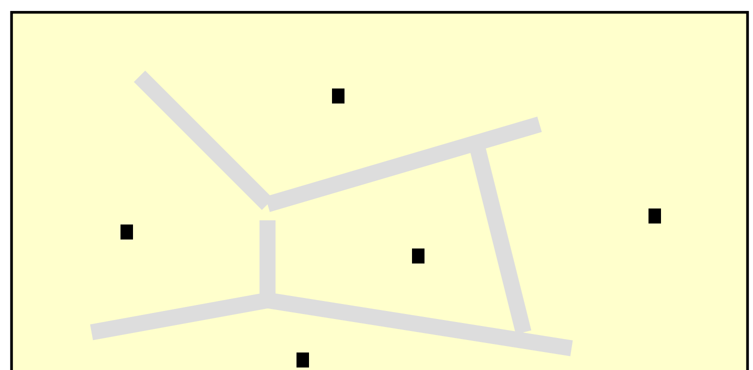
- What are anomalies?
- Anomaly Detection: Taxonomy
- Nearest Neighbor Based Techniques
- One-Class to tackle the Fault Detection
- Concluding Remarks

Nearest Neighbor Based Techniques

K Nearest Neighbor (KNN)

- All instances correspond to points in the n-D space.
- The nearest neighbor are defined in terms of Euclidean distance.
- The target function could be discrete- or real- valued.
- For discrete-valued, the *k*-NN returns the most common value among the k training examples nearest to x_q.
- Voronoi diagram: the decision surface induced by 1-NN for a typical set of training examples.





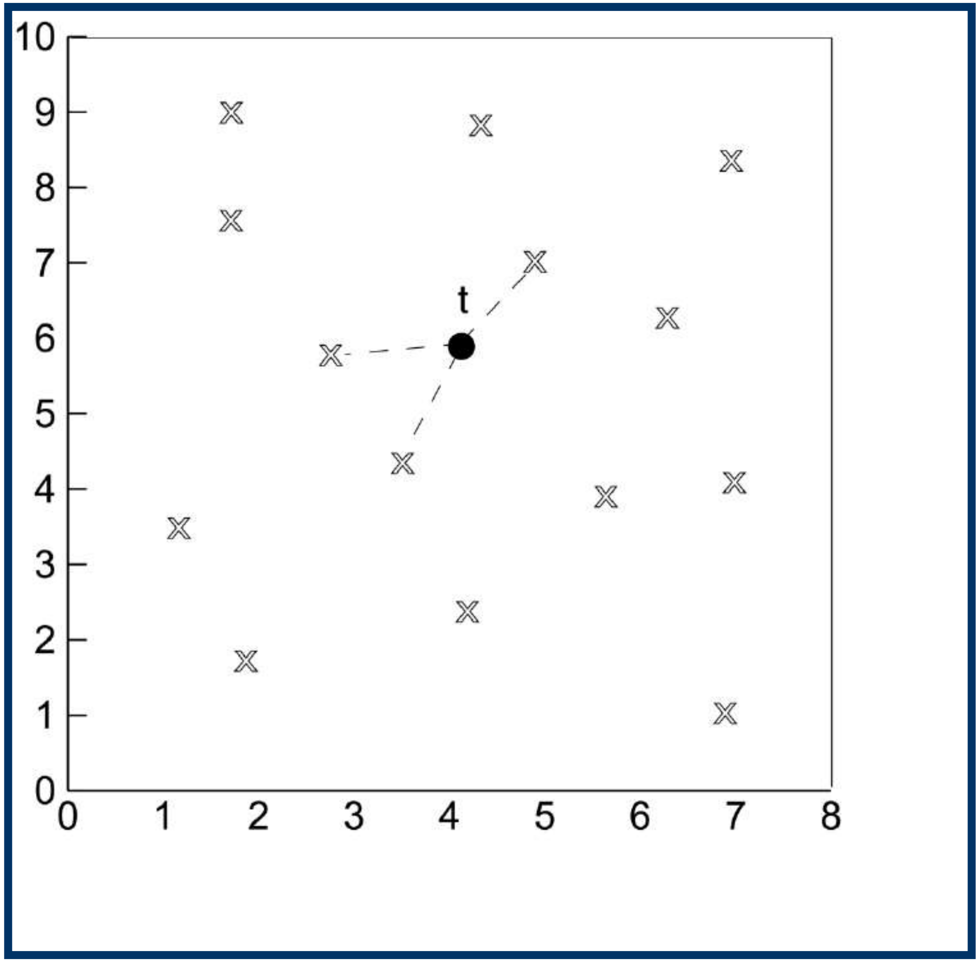
Nearest Neighbor Based Techniques

K Nearest Neighbor (KNN)

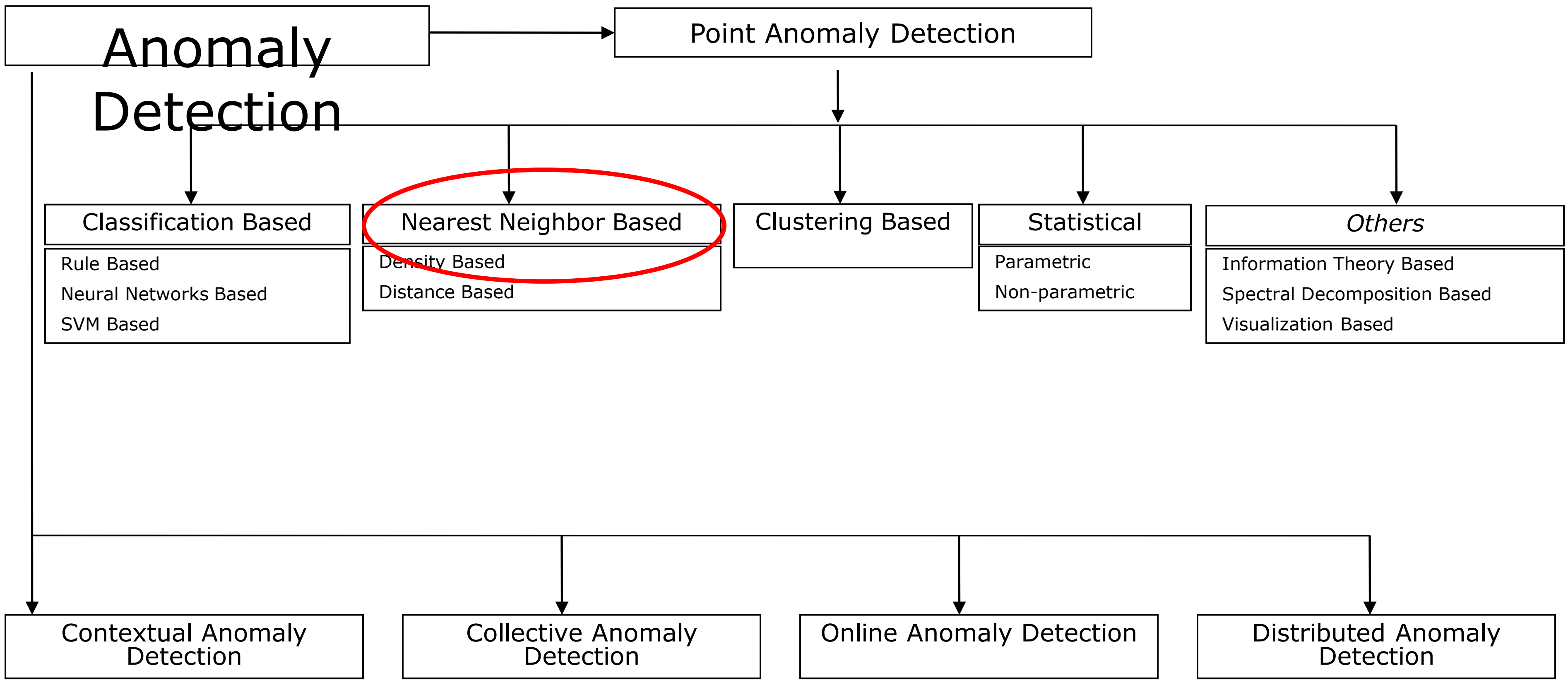
- Training set includes classes.
- Examine K items near item to be classified.
- New item placed in class with the most number of close items.
- O(q) for each tuple to be classified. (Here q is the size of the training set.)

Nearest Neighbor Based Techniques

K Nearest Neighbor (KNN)



Nearest Neighbor Based Techniques



* Anomaly Detection – A Survey, Varun Chandola, Arindam Banerjee, and Vipin Kumar, ACM Computing Surveys, Vol. 41, No. 3, Article 15, Publication date: July 2009.

Nearest Neighbor Based Techniques

- ***Key assumption***: normal points have close neighbors while anomalies are located far from other points
- General two-step approach
  1. Compute neighborhood for each data record
  2. Analyze the neighborhood to determine whether data record is anomaly or not
- **Categories:**
  - Distance based methods
    - Anomalies are data points most distant from other points
  - Density based methods
    - Anomalies are data points in low density regions

Nearest Neighbor Based Techniques

- **Advantage**
  - Can be used in unsupervised or semi-supervised setting (do not make any assumptions about data distribution)
- **Drawbacks**
  - If normal points do not have sufficient number of neighbors the techniques may fail
  - Computationally expensive
  - In high dimensional spaces, data is sparse and the concept of similarity may not be meaningful anymore. Due to the sparseness, distances between any two data records may become quite similar => Each data record may be considered as potential outlier!

Nearest Neighbor Based Techniques

- **Distance based approaches**
  - A point *O* in a dataset is an *DB*(*p*, *d*) outlier if at least fraction *p* of the points in the data set lies greater than distance *d* from the point *O**
- **Density based approaches**
  - Compute local densities of particular regions and declare instances in low density regions as potential anomalies
  - Approaches
    - Local Outlier Factor (LOF)
    - Connectivity Outlier Factor (COF
    - Multi-Granularity Deviation Factor (MDEF)

Nearest Neighbor Based Techniques

Distance based Outlier Detection

- *Nearest Neighbor (NN) approach*
  - For each data point d compute the distance to the *k-th* nearest neighbor d_k
  - Sort all data points according to the distance d_k
  - Outliers are points that have the largest distance d_k and therefore are located in the more sparse neighborhoods
  - Usually data points that have top n% distance d_k are identified as outliers
    - n – user parameter
  - Not suitable for datasets that have modes with varying density

Nearest Neighbor Based Techniques

Density Based Approaches: Local Outlier Factor (LOF)

- For each data point q compute the distance to the k-th nearest neighbor (*k-distance*)
- Compute *reachability distance* (*reach-dist*) for each data example q with respect to data example p as:

$$\text{reach-dist}(q, p) = \max\{k\text{-distance}(p), d(q,p)\}$$

- Compute *local reachability density* (*lrd*) of data example q as inverse of the average reachabaility distance based on the *MinPts* nearest neighbors of data example q

$$lrd(q) = \frac{MinPts}{\sum_p reach\_dist_{MinPts}(q, p)}$$

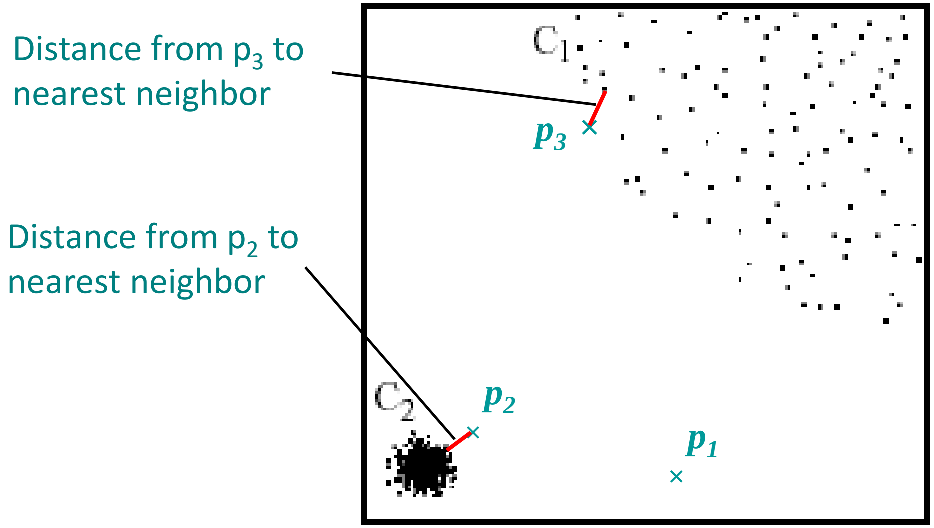
- Compute $LOF(q)$ as ratio of average local reachability density of q's k-nearest neighbors and local reachability density of the data record q

$$LOF(q) = \frac{1}{MinPts} \cdot \sum_p \frac{lrd(p)}{lrd(q)}$$

Nearest Neighbor Based Techniques

Advantages of Density based Techniques

- *Local Outlier Factor (LOF) approach*
  - Example:



In the *NN* approach, p_2 is not considered as outlier, while the *LOF* approach find both p_1 and p_2 as outliers

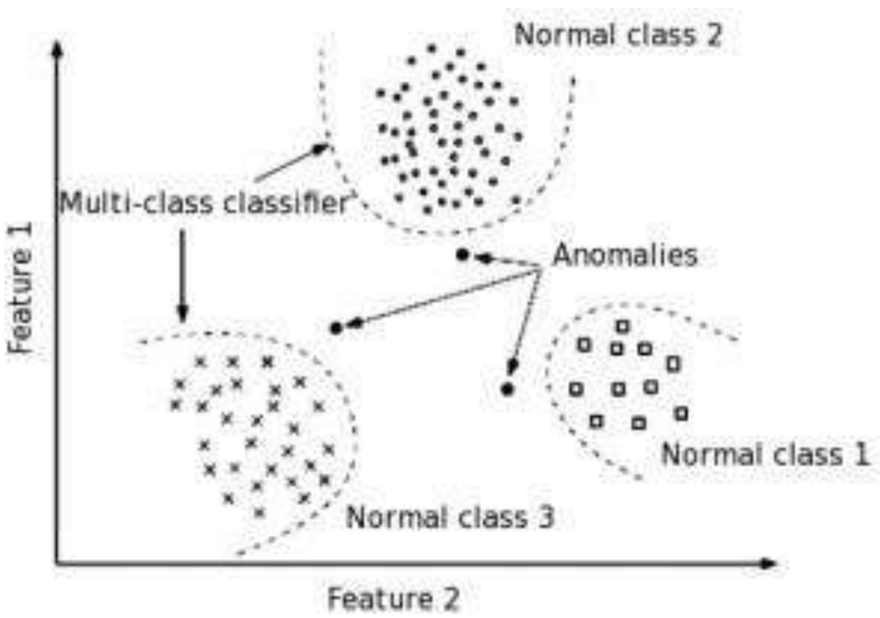
NN approach may consider p_3 as outlier, but LOF approach does not

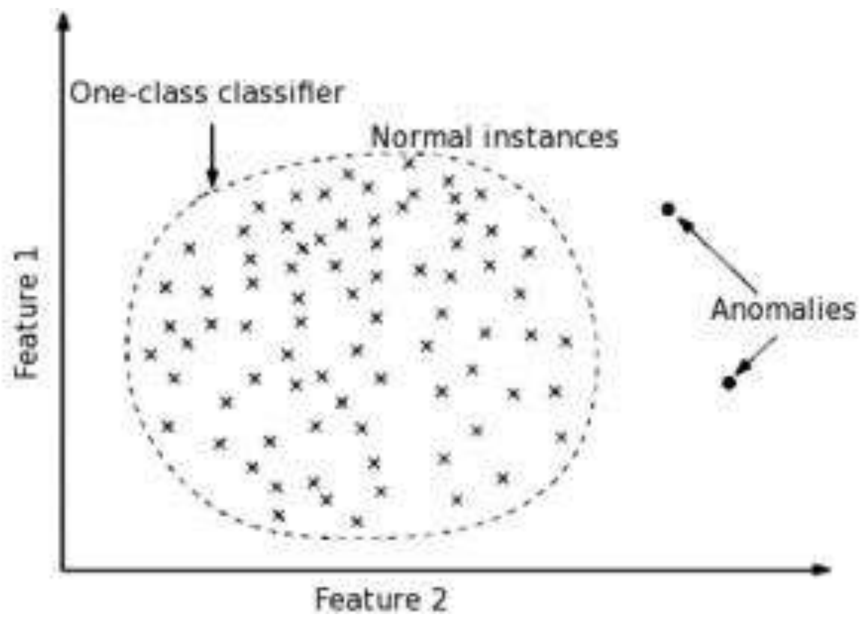
Outline

- What are anomalies?
- Anomaly Detection: Taxonomy
- Nearest Neighbor Based Techniques
- One-Class to tackle the Fault Detection
- Concluding Remarks

One-Class to tackle the Fault Detection

Several classes vs One-class classification





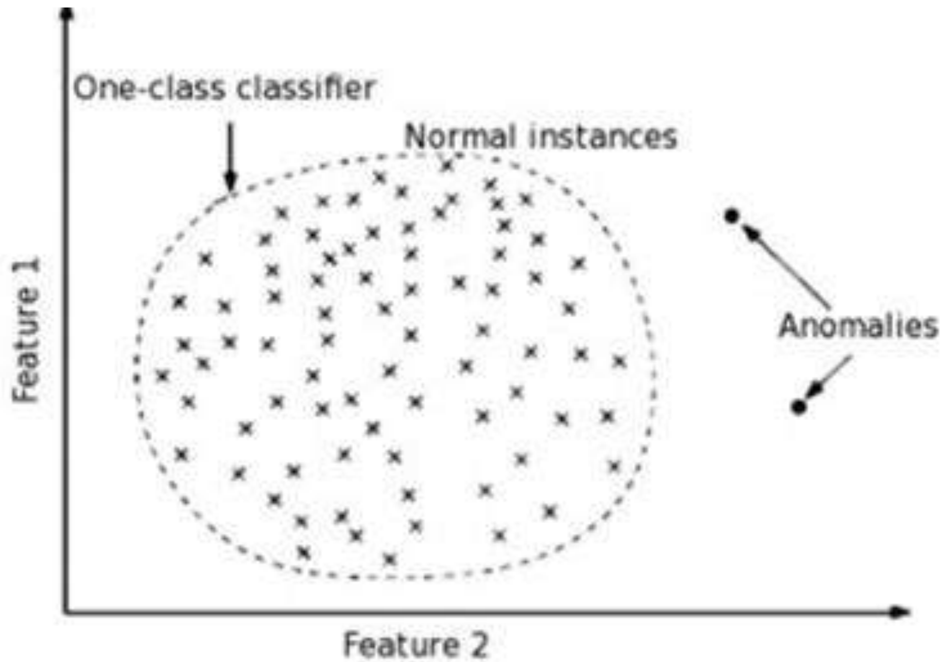
One-Class to tackle the Fault Detection

Classification Based Techniques

- **Advantages:**
  - ***Supervised classification techniques***
    - Models that can be easily understood
    - High accuracy in detecting many kinds of known anomalies
  - ***Semi-supervised classification techniques (One-class)***
    - Models that can be easily understood
    - Normal behavior can be accurately learned
- **Drawbacks:**
  - ***Supervised classification techniques***
    - Require both labels from both normal and anomaly class
    - Cannot detect unknown and emerging anomalies
  - ***Semi-supervised classification techniques (One-class)***
    - Require labels from normal class
    - Possible high false alarm rate - previously unseen (yet legitimate) data records may be recognized as anomalies

One-Class to tackle the Fault Detection

One-class 1-NN is an semi-supervised algorithm that learns a decision function for novelty detection: classifying new data as similar or different to the training set.



One-Class to tackle the Fault Detection

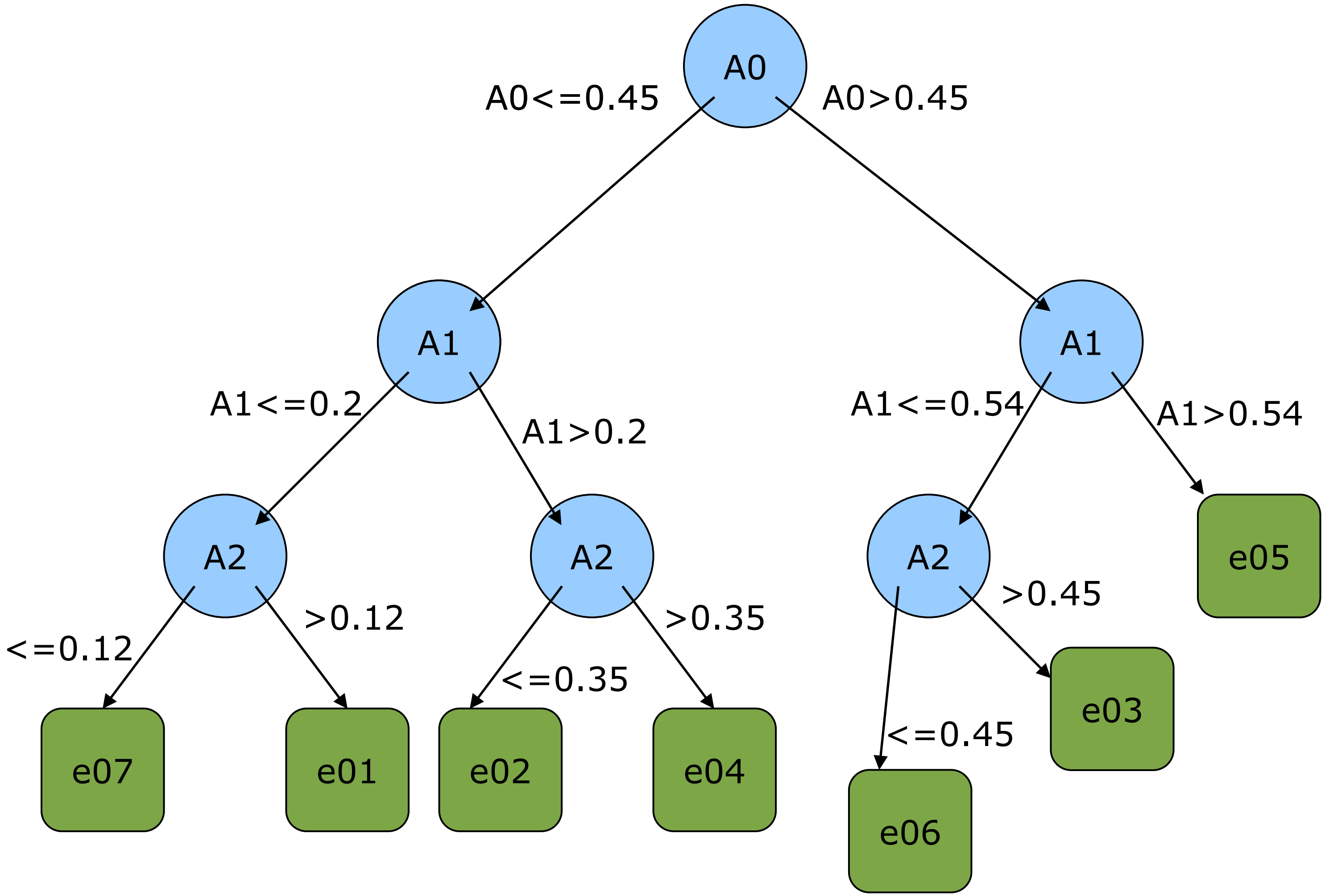
Pseudocode of one-class kNN

When a new test example A needs to be tested

1.- Find its nearest neighbor (NN), which we call B, by using a fast NN technique: **k-d tree***.
2.- The tentative class of A is the class of B.
3.- Find the nearest neighbor of B in the trainining set using a **k-d tree***, call it C.
4.- For each attribute *attr* in the dataset, perform the following calculations:
If (abs (A[*attr*] – B [*attr*]) > threshAttr * abs (B[*attr*]-C[*attr*])):
Example A does not belong to any class and is considered an anomaly, Otherwise
It is assigned to its tentative class.

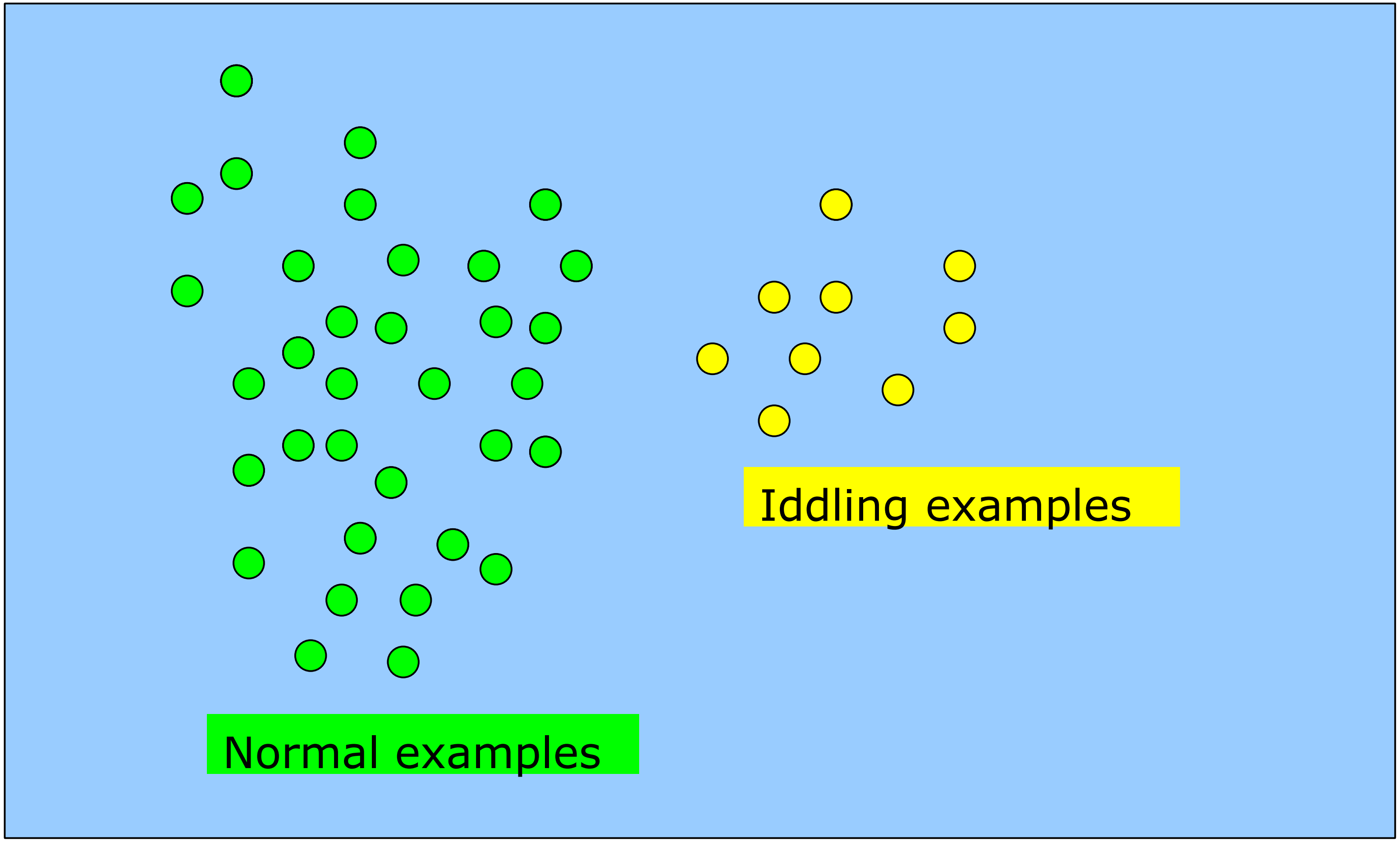
One-Class to tackle the Fault Detection

Constructing a k-d tree



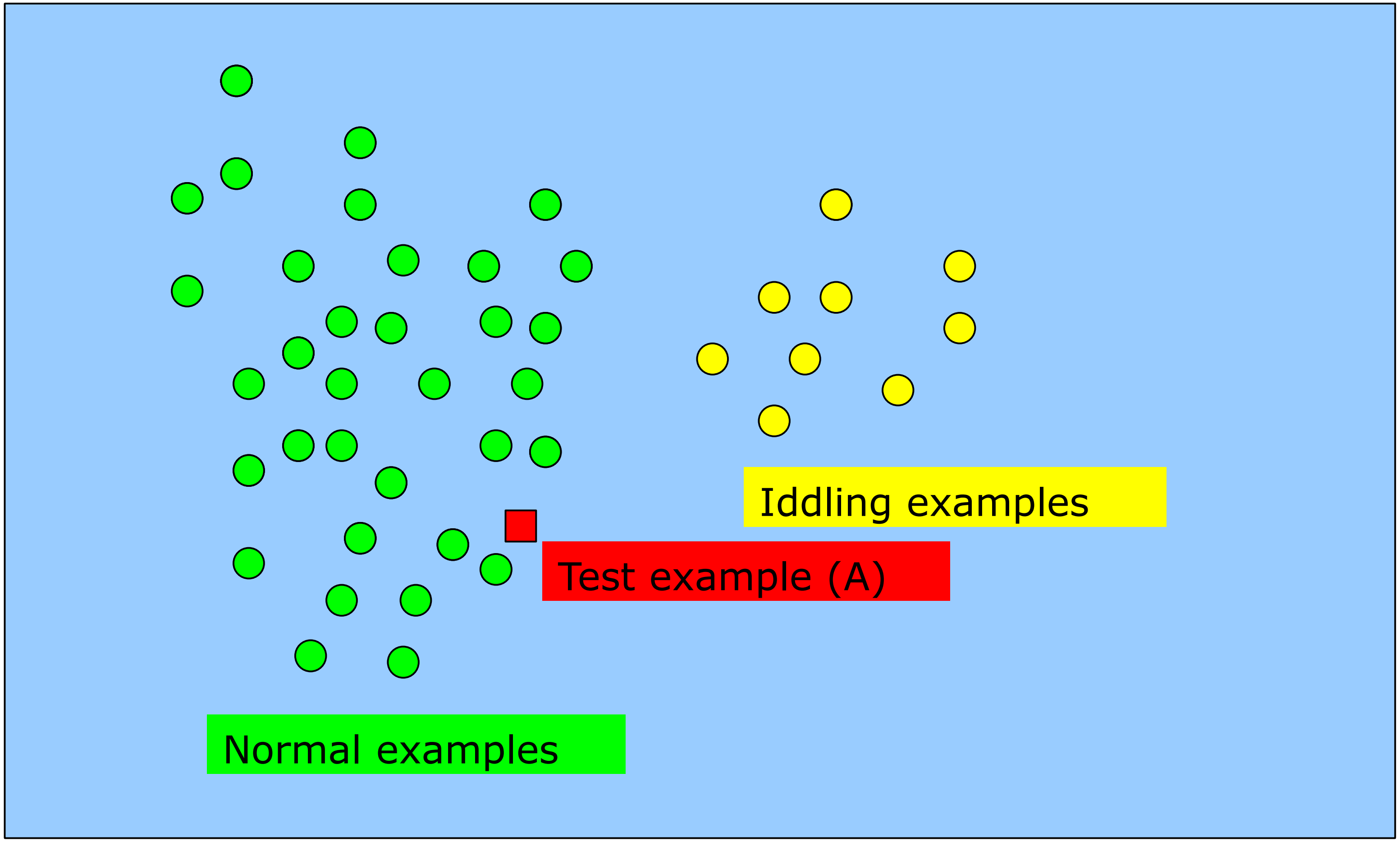
One-Class to tackle the Fault Detection

Visually: Training examples



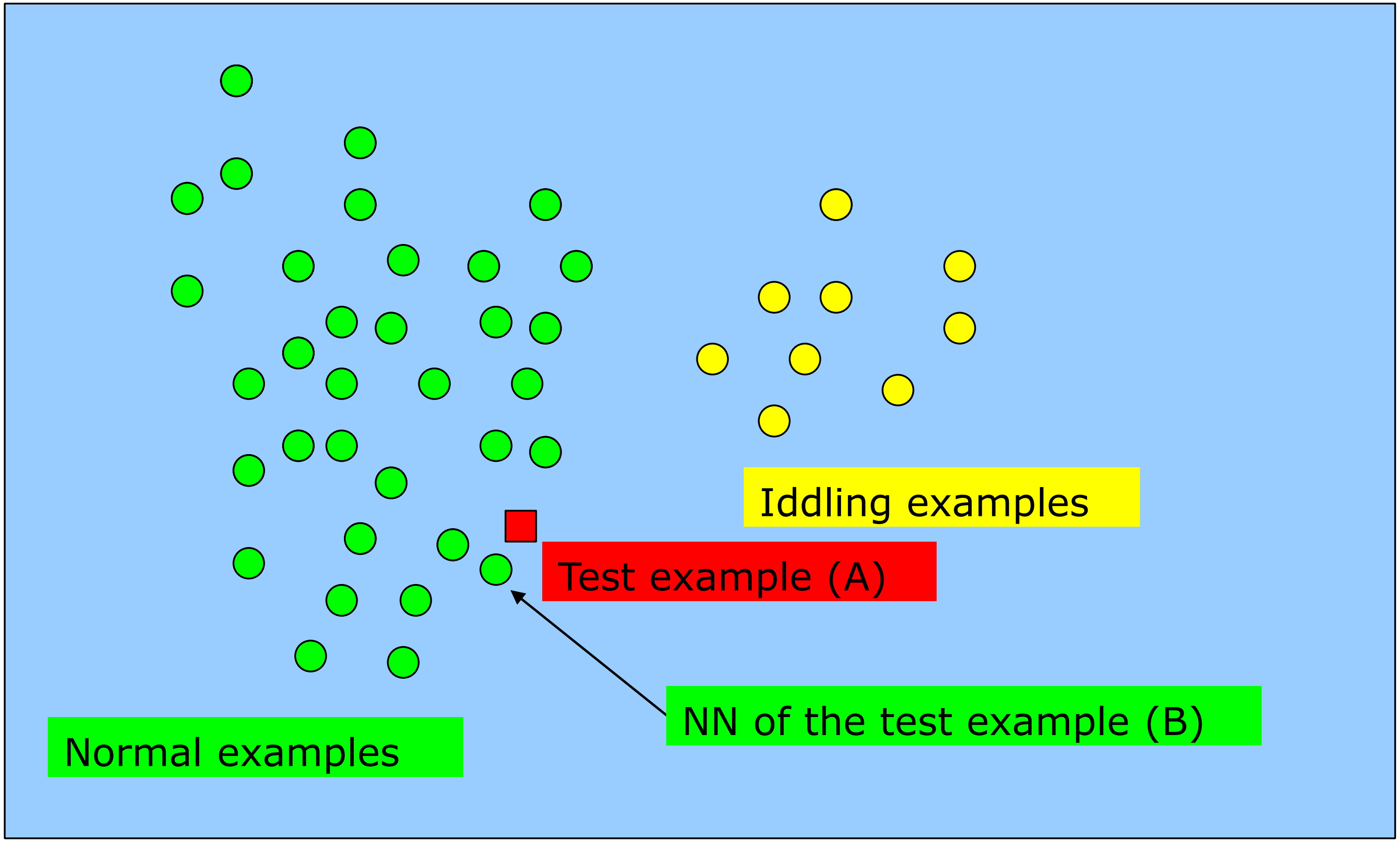
One-Class to tackle the Fault Detection

Visually: Training + 1 test example



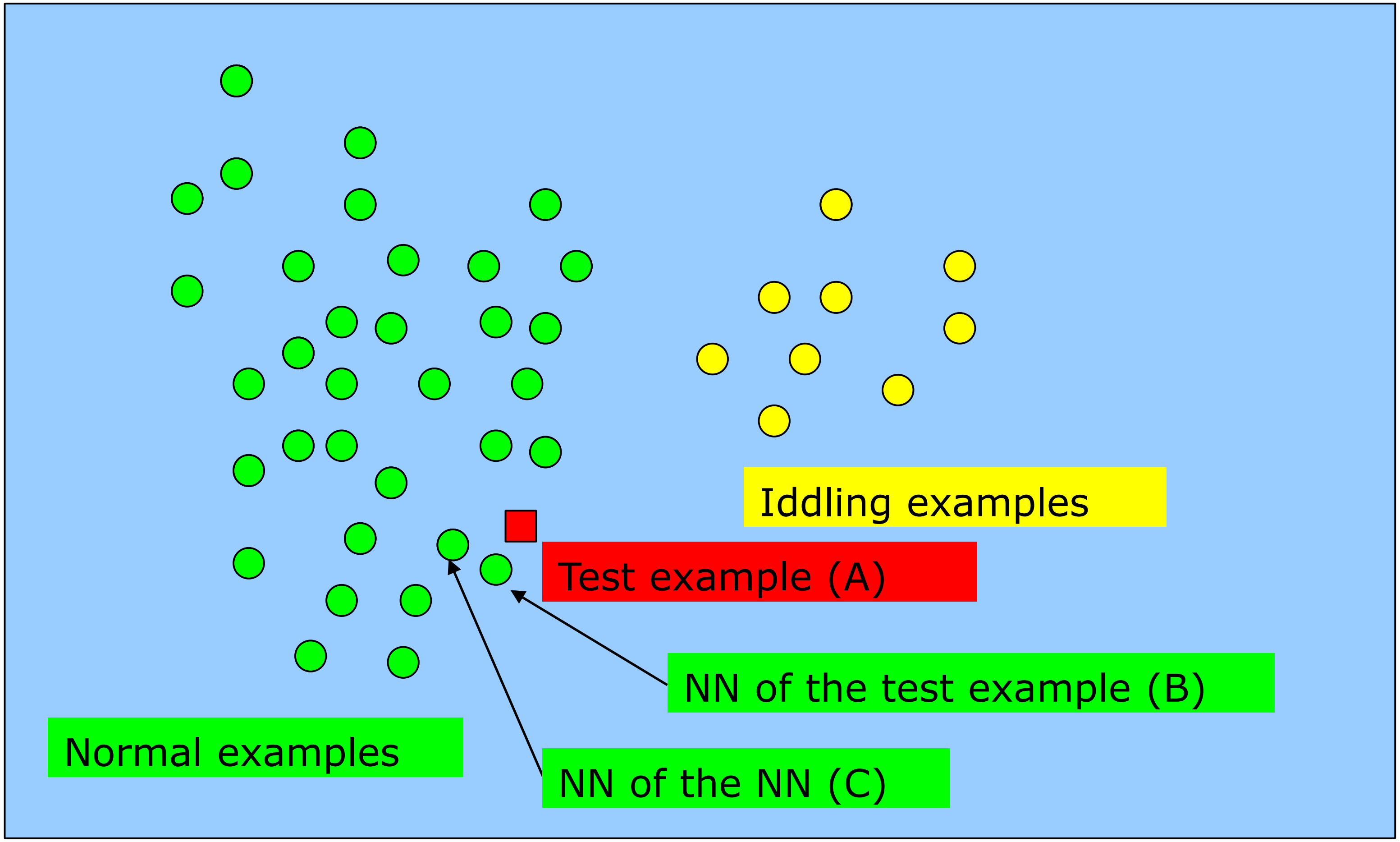
One-Class to tackle the Fault Detection

Visually: Finding the NN of the test example



One-Class to tackle the Fault Detection

Visually: Finding the NN of B (finding C)



One-Class to tackle the Fault Detection

One-class kNN: Reading the output

Test example 24199 has been found to be an anomaly

These are the values for all the attributes on the test example:

225.125	1500	364.523	41.8	42.3

Values for all attributes of example B

Its NN in training is example 57679,

223.575	1497.06	370.553	41.6	42.2

This test example is labeled as an anomaly because
Attribute 3 should be in range [364.553 , 376.553], but its actual value is 364.523

Range where the attribute should be. It is calculated as:

[B[*attr*]-threshAttr*abs(B[*attr*]-C[*attr*]), B [*attr*]+threshAttr*abs(B[*attr*]-C[*attr*])]

One-Class to tackle the Fault Detection

Brief tutorial on k-d trees

Basic idea: binary tree where each node splits the data in two subgroups with roughly half the size (divide and conquer)

How? Take an attribute, split the data points by the median value: The examples with value under or equal to the median are placed on the subtree to one side, those with values over the median go to the subtree on the other side.

The size of the tree is O(n), the average time to find a match (a Nearest Neighbor, the process is explained in the next slide) is O(log(n)). In this context, n refers to the number of examples in the training set.

The time to find a match is on average O(log(n)) only when the k-d tree works well.
For a k-d tree to work well, the number of examples must be much larger than the number of attributes (n should be >= 2^{nAttr}), and said examples should be approximately randomly distributed.

Both of these conditions hold in the HMS-GAMESA data, so the k-d tree is a very good solution for this problem.

One-Class to tackle the Fault Detection

Constructing a k-d tree (I)

Example data (each row corresponds to an example, each column is an attribute):

Example ID	A0	A1	A2	A3
e01	0.10	0.06	0.20	0.30
e02	0.30	0.33	0.35	0.51
e03	0.50	0.65	0.54	0.45
e04	0.45	0.14	0.56	0.89
e05	0.52	0.17	0.67	0.64
e06	0.53	0.40	0.45	0.11
e07	0.29	0.54	0.12	0.54

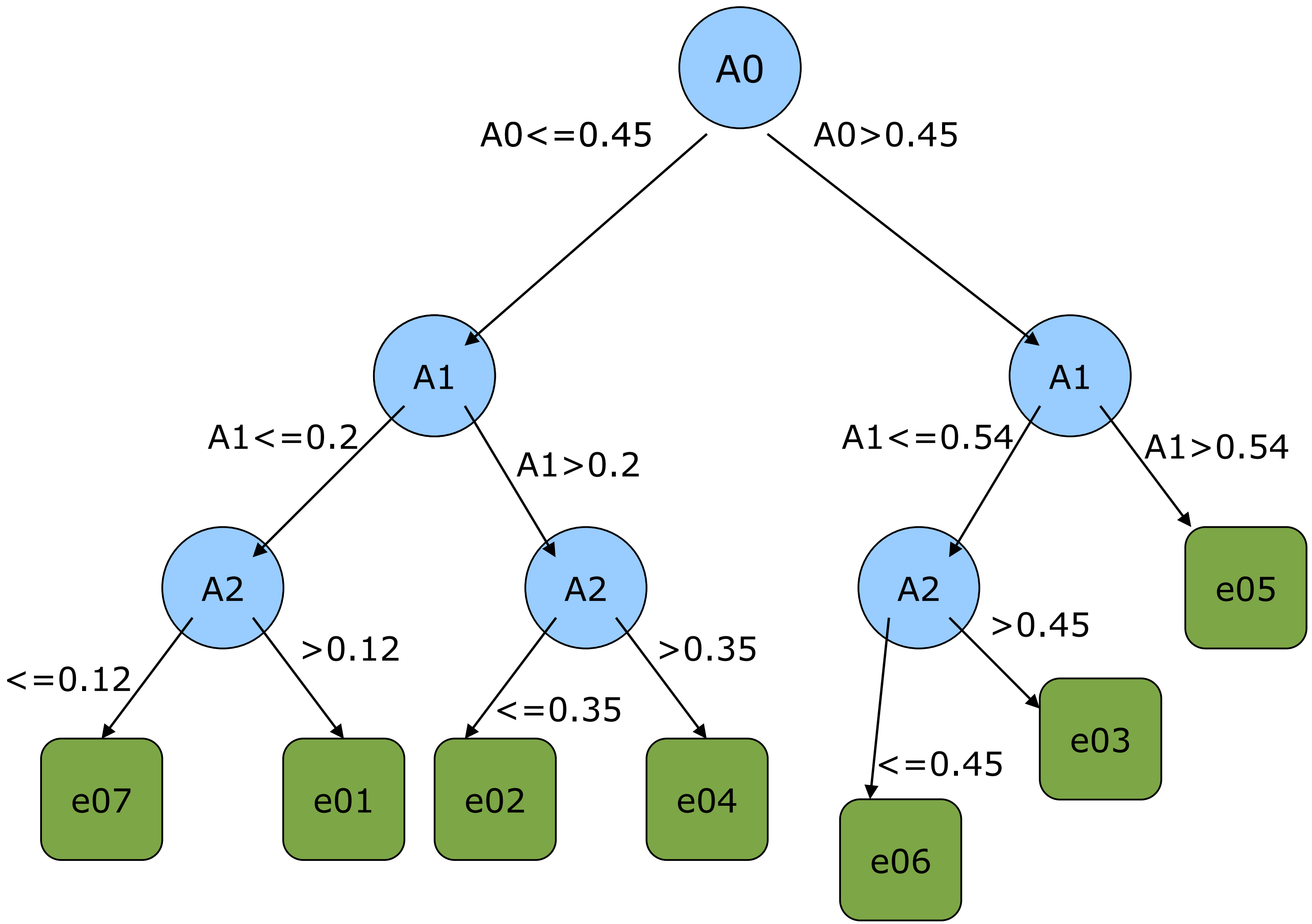
Root node: take attribute A0. Median = 0.45. e01, e02, e04 and e07 go to the left subtree, e03, e05 and e06 to the right one.

Second level: attribute A1. On the left subtree the median is 0.20, e01 and e07 go left and e02, e04 go right. On the right subtree, the median is 0.54. e03 and e06 go left, e05 goes right.

Repeat the process until all examples are on leaves.

One-Class to tackle the Fault Detection

Constructing a k-d tree (II)



One-Class to tackle the Fault Detection

Finding a Nearest Neighbor in the k-d tree

1.- Starting with the root node, the algorithm moves down the tree recursively: it goes left or right depending on whether the point is less than or greater than the current node in the split dimension.

2.-Once the algorithm reaches a leaf node, it saves that node point as the "current best"

3.- The algorithm unwinds the recursion of the tree, performing the following steps at each node:

3.1.- If the current node is closer than the current best, then it becomes the current best.

3.2.- The algorithm checks whether there could be any points on the other side of the splitting plane that are closer to the search point than the current best. In concept, this is done by intersecting the splitting hyperplane with a hypersphere around the search point that has a radius equal to the current nearest distance. Since the hyperplanes are all axis-aligned this is implemented as a simple comparison to see whether the difference between the splitting coordinate of the search point and current node is less than the distance (overall coordinates) from the search point to the current best.

3.2.1.- If the hypersphere crosses the plane, there could be nearer points on the other side of the plane, so the algorithm must move down the other branch of the tree from the current node looking for closer points, following the same recursive process as the entire search.

3.2.2.- If the hypersphere doesn't intersect the splitting plane, then the algorithm continues walking up the tree, and the entire branch on the other side of that node is eliminated.

4.- When the algorithm finishes this process for the root node, then the search is complete

Outline

- What are anomalies?
- Anomaly Detection: Taxonomy
- Nearest Neighbor Based Techniques
- One-Class to tackle the Fault Detection
- Concluding Remarks

Conclusions

- Anomaly detection can detect critical information in data
- Highly applicable in various application domains
- Nature of anomaly detection problem is dependent on the application domain
- Need different approaches to solve a particular problem formulation
- The nearest neighbor based techniques are very appropriate for different problems, but they need to be tuned to this problem.

Inteligencia de Negocio

TEMA 7. Modelos Avanzados de Minería de Datos

1. Clases no balanceadas/equilibradas

2. Características intrínsecas de los datos en clasificación

3. Clasificación no Estándar: Más allá del aprendizaje clásico

4. Detección de anomalías

5. **Deep Learning**

6. Análisis de Sentimientos

Deep Learning

Deep Learning : El aprendizaje profundo es un conjunto de algoritmos que intenta modelar abstracciones de alto nivel en los datos mediante el uso de arquitecturas compuestas de transformación no lineales múltiples.

Bibliografía:

L. Deng and D. Yu.
Deep Learning methods and applications.
Foundations and Trends in Signal Processing
Vol. 7, Issues 3-4, 2014.

Nota: Deep Learning introduce el uso de estructuras de aprendizaje que requieren de arquitecturas de procesamiento eficiente y distribuido (GPU, Spark, ...) y muestra resultados importantes en el procesamiento de imágenes, habla, lenguaje natural, ...

Deep Learning

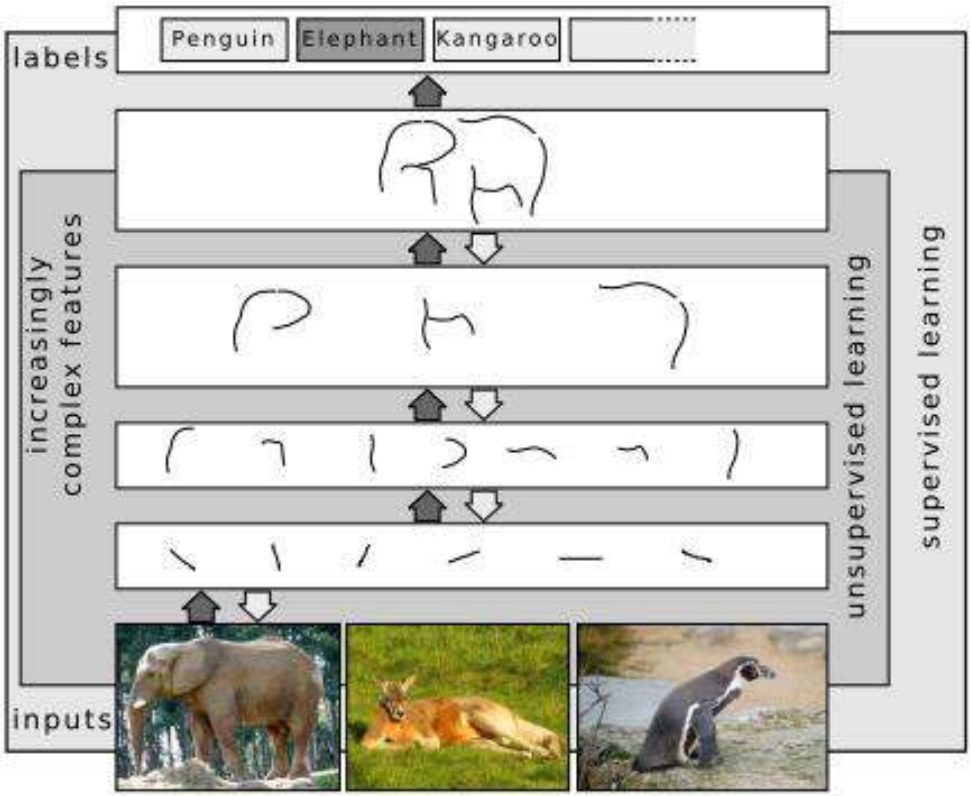
Deep Learning (deep structure learning): machine learning algorithms based on learning multiple levels of representation/abstraction.

Amazing improvements in error rate in object recognition, object detection, speech recognition, and more recently, in natural language processing/understading.

Yoshua Bengio
http://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf

Deep Learning
(also called Hierarchical Learning)

Hierarchical Learning



- Natural progression from low level to high level structure as seen in natural complexity
- Easier to monitor what is being learnt and to guide the machine to better subspaces
- Usually best when input space is locally structured – spatial or temporal: images, language, etc. vs arbitrary input features

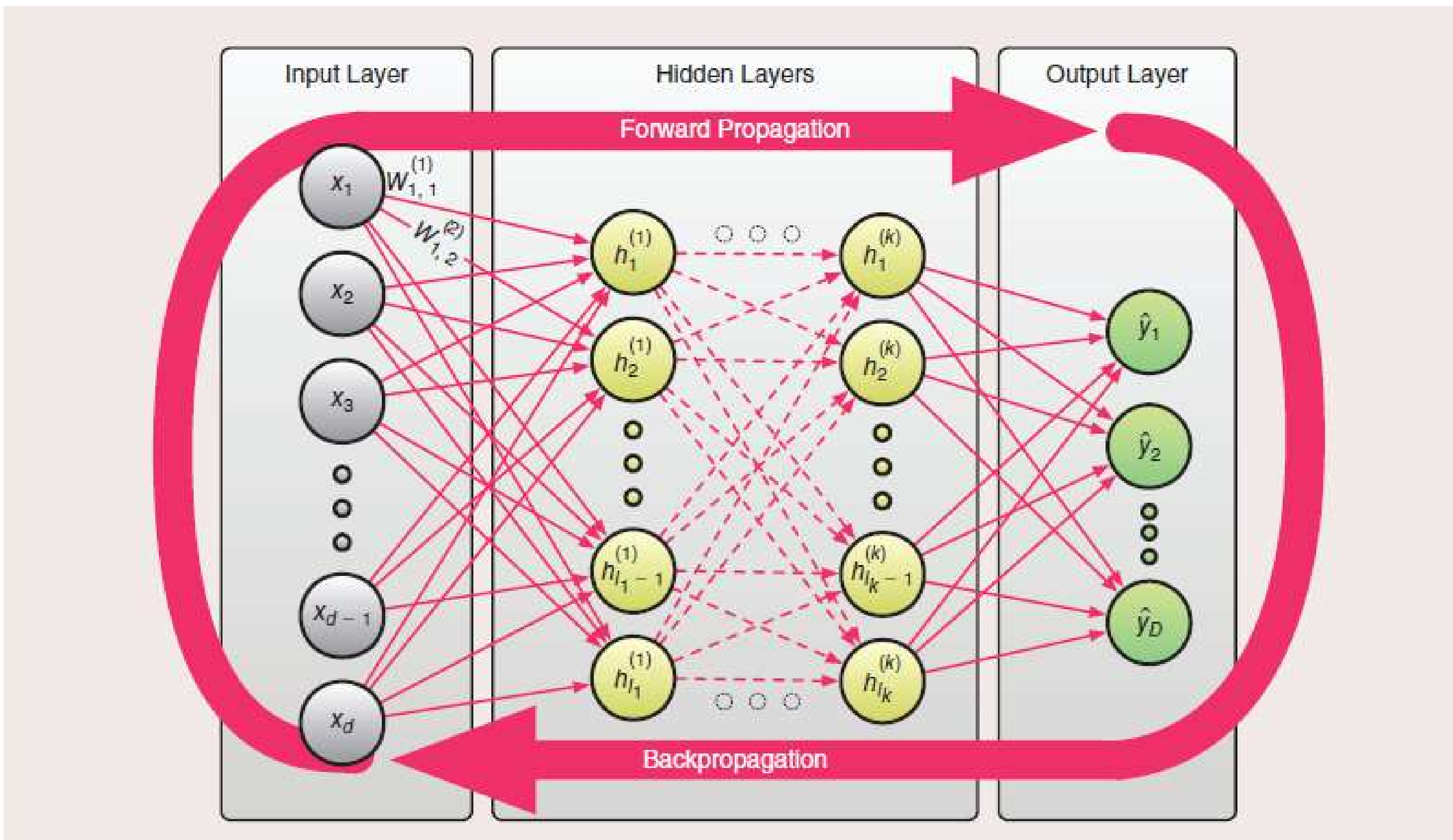
Deep Learning

Human information processing mechanisms (e.g., vision and audition) suggest the need of deep architectures for extracting complex structure and building internal representation from rich sensory inputs.

Historically, the concept of deep learning originated from artificial neural network research. (Hence, one may occasionally hear the discussion of **"new-generation neural networks."**) Feed-forward neural networks or MLPs with many hidden layers, which are often referred to as deep neural networks (DNNs), are good examples of the models with a deep architecture.

Deep Learning

Classical NN: Backpropagation



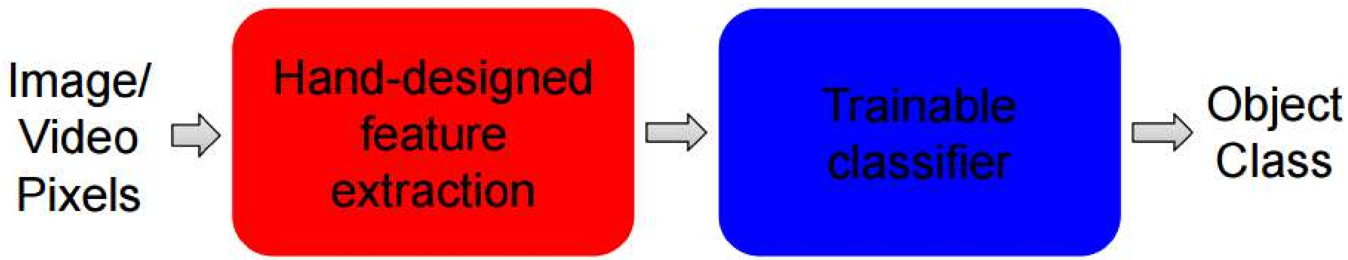
Credits: The Evolution of Neural Learning Systems: A Novel Architecture Combining the Strengths of NTs, CNNs, and ELMs. N Martinel, C Micheloni… - IEEE SMC Magazine, 2015 - ieeexplore.ieee.org

Machine learning: Shallow-structured arquitectures

- Gaussian mixture models (GMMs),
- Linear or nonlinear dynamical systems,
- Conditional, random fields (CRFs)
- Maximum entropy (MaxEnt) models,
- Support vector machines (SVMs)
- Logistic regression/kernel regression
- Multilayer perceptrons (MLPs) with a single hidden layer including extreme learning machines (ELMs).

These architectures typically contain at most one or two layers of nonlinear feature transformations.

Traditional recognition approaches

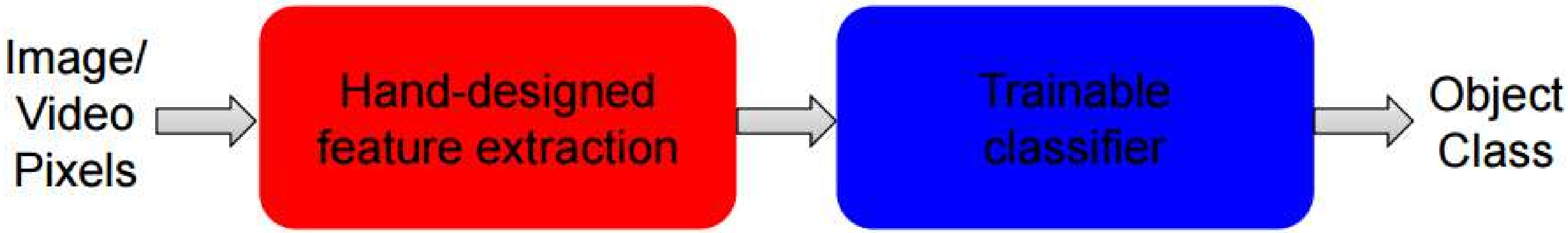


Features are not learning

Deep Learning

“Shallow” vs. “deep” architectures

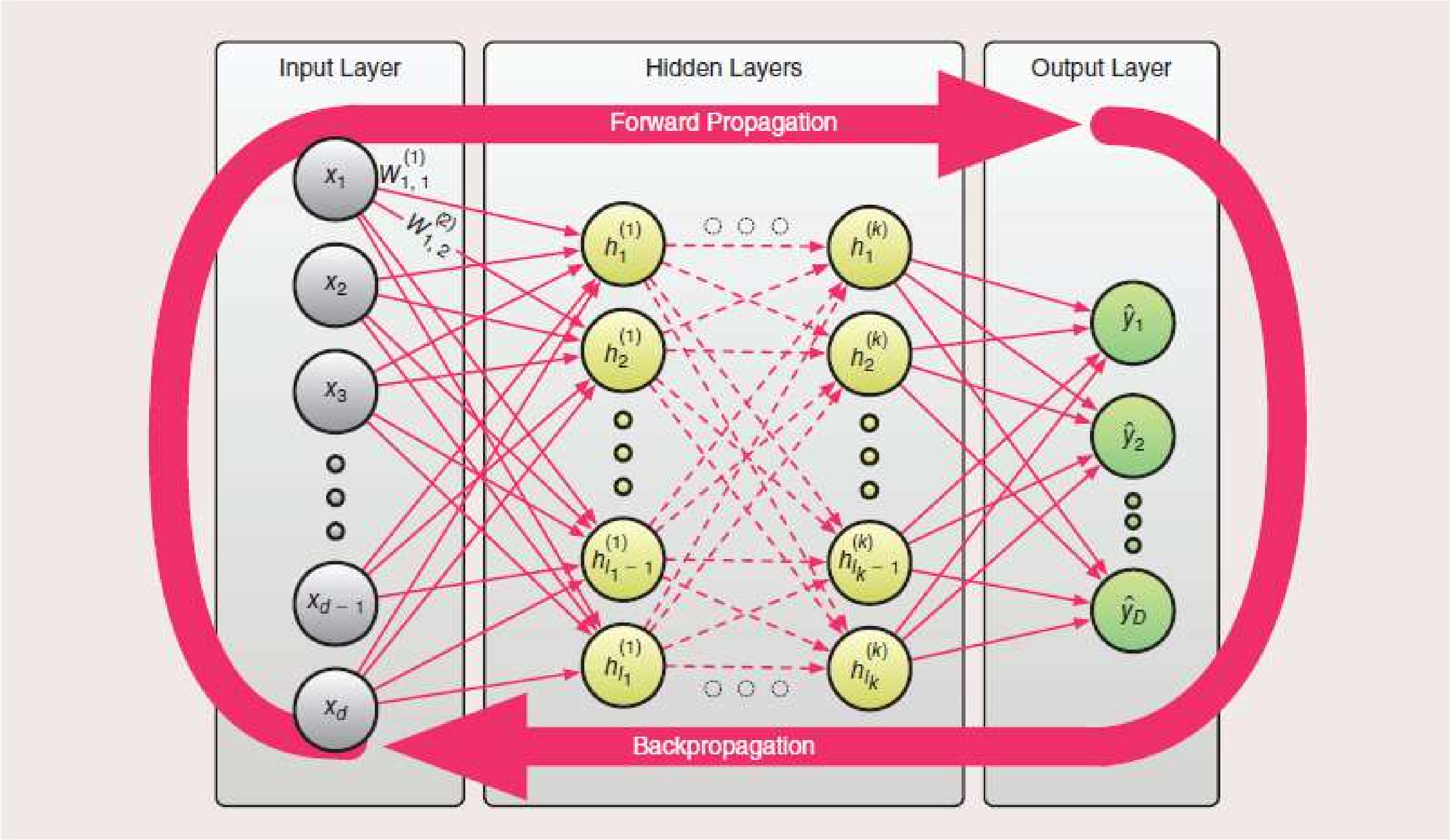
Traditional recognition: “Shallow” architecture



Deep learning: “Deep” architecture

Deep Learning

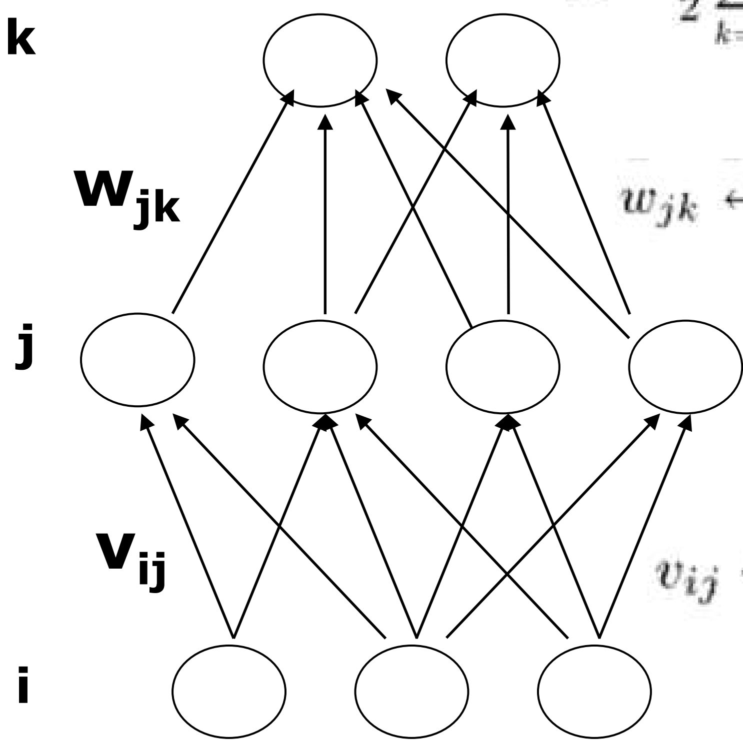
Backpropagation



Credits: The Evolution of Neural Learning Systems: A Novel Architecture Combining the Strengths of NTs, CNNs, and ELMs. N Martinel, C Micheloni… - IEEE SMC Magazine, 2015 - ieeexplore.ieee.org

Deep Learning

Backpropagation



$$E = \frac{1}{2}\sum_{k=1}^{N}(t_k - y_k)^2$$

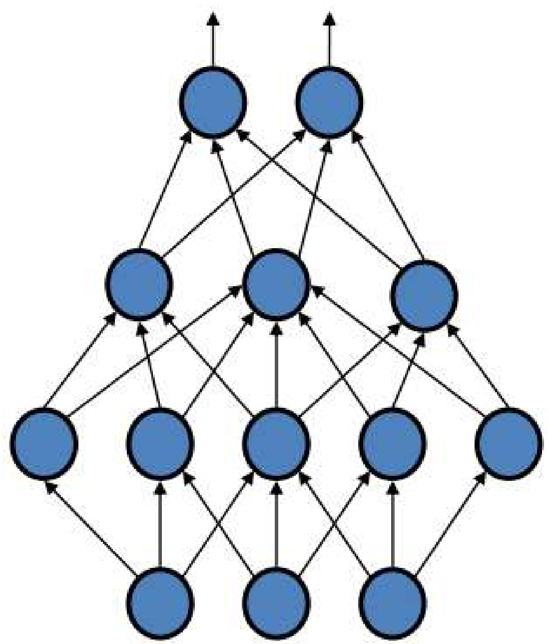
$$w_{jk} \leftarrow w_{jk} - \eta \frac{\partial E}{\partial w_{jk}}$$

$$v_{ij} \leftarrow v_{ij} - \eta \frac{\partial E}{\partial v_{ij}}$$

- Minimize error of calculated output
- Adjust weights
  - Gradient Descent
- Procedure
  - Forward Phase
  - Backpropagation of errors
- For each sample, multiple epochs

Problems with Backpropagation

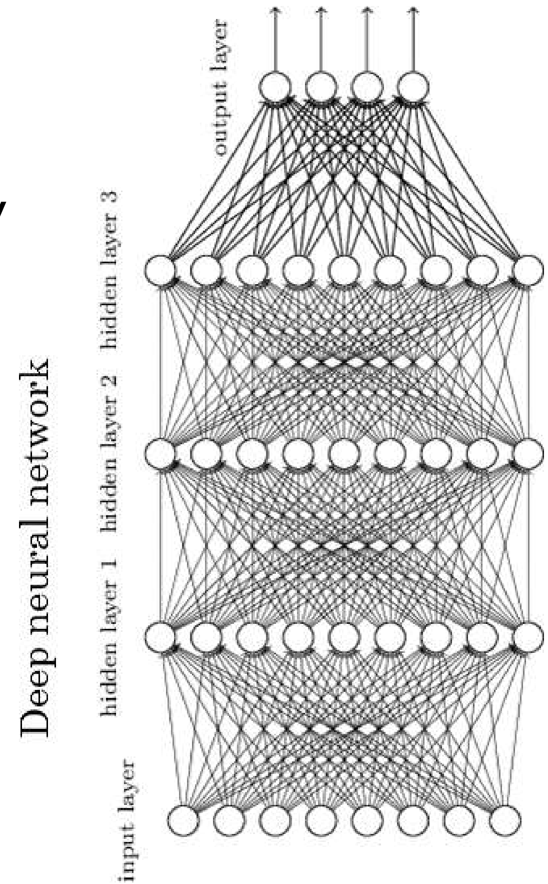
- Multiple hidden Layers
- Get stuck in local optima
  - start weights from random positions
- Slow convergence to optimum
  - large training set needed
- Only use labeled data
  - most data is unlabeled



Deep Learning

Deep Architecture **(Train networks with many layers)**

- Multiple hidden layers
- Motivation (why go deep?)
  - Approximate complex decision boundary
    - Fewer computational units for same functional mapping
  - Hierarchical Learning
    - Increasingly complex features
  - Work well in different domains
    - Vision, Audio, …



Deep Learning

Deep Learning: Automating Feature Discovery

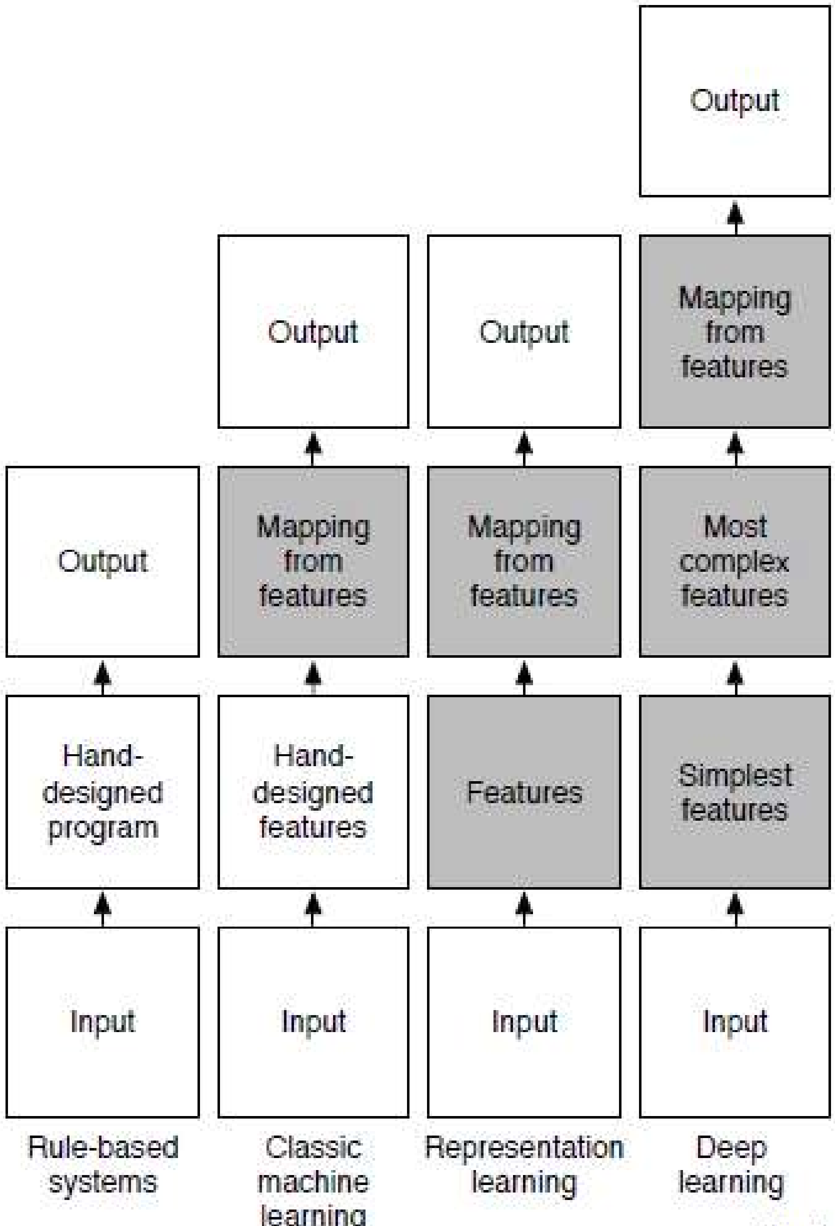


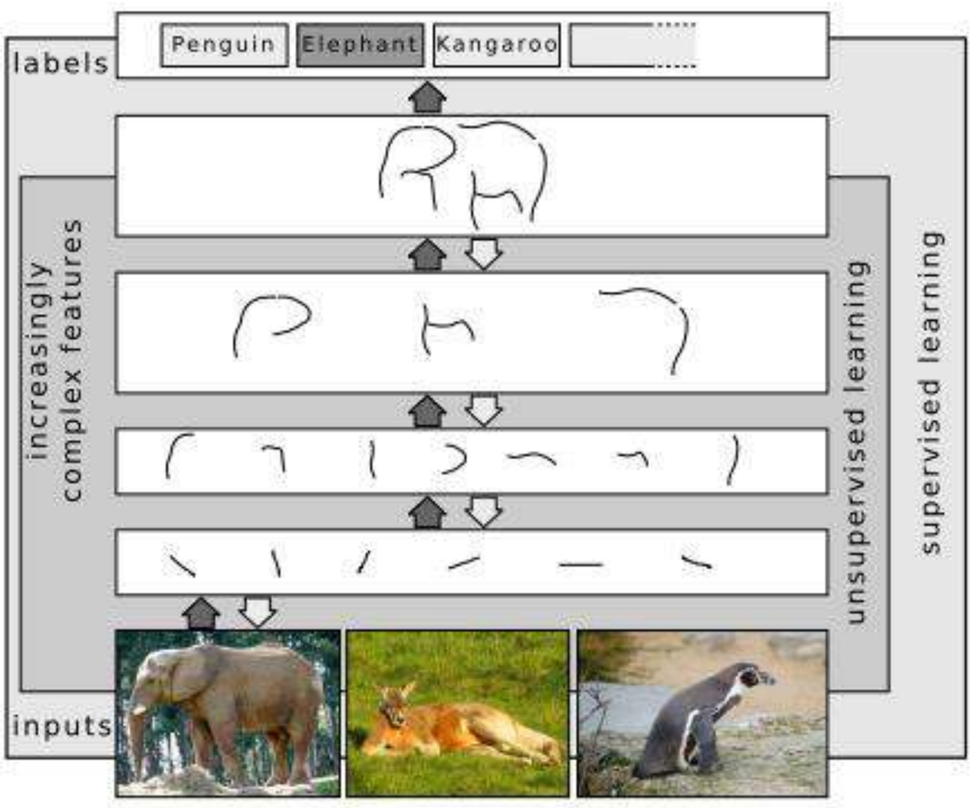
Fig: I. Goodfellow

Yoshua Bengio

Credits: http://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf

Deep Learning
(Hierarchical Learning)

Hierarchical Learning/deep structure learning: Automating Feature Discovery



From simplest features to complex one

Fron unsupervised learning to supervised learning

Deep Architecture (Train networks with many layers)

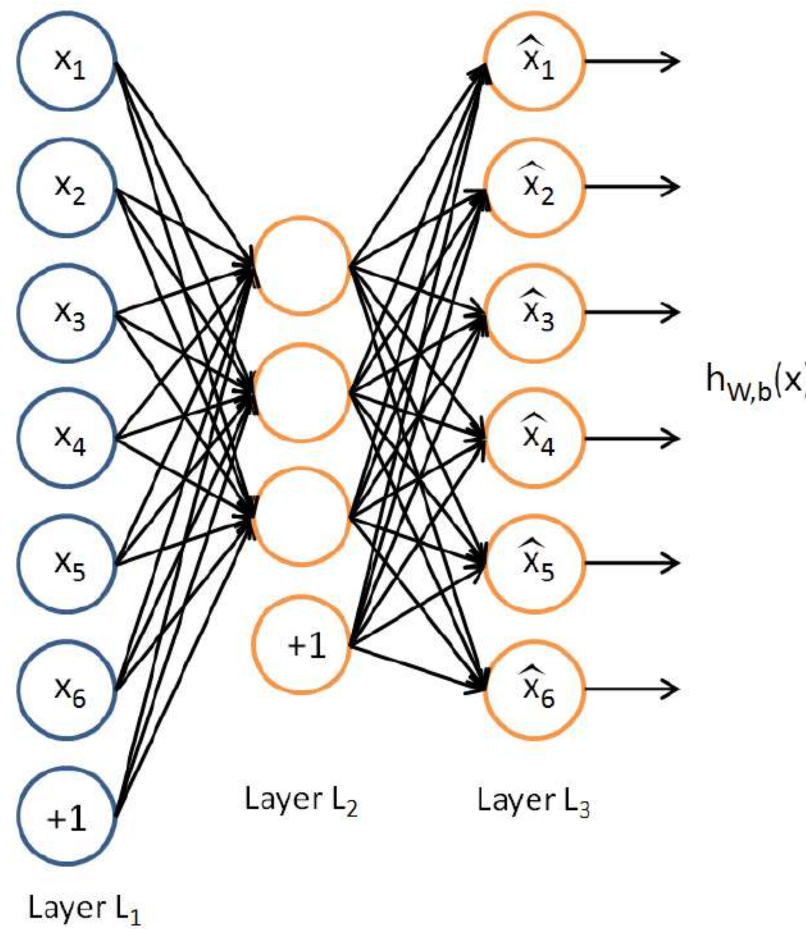
- Some Models:
  - Deep networks for unsupervised or generative learning: deep belief network (DBN), stack of restricted Boltzmann machines (RBMs), autoencoder …
  - Deep networks for supervised learning: Deep Neural Networks (DNN), Convolutional neural network (CNN). …
  - Hybrid deep networks: DBN-DNN (when DBN is used to initialize the training of a DNN, the resulting network is sometimes called the DBN–DNN)

Deep Learning

- **Autoencoder**

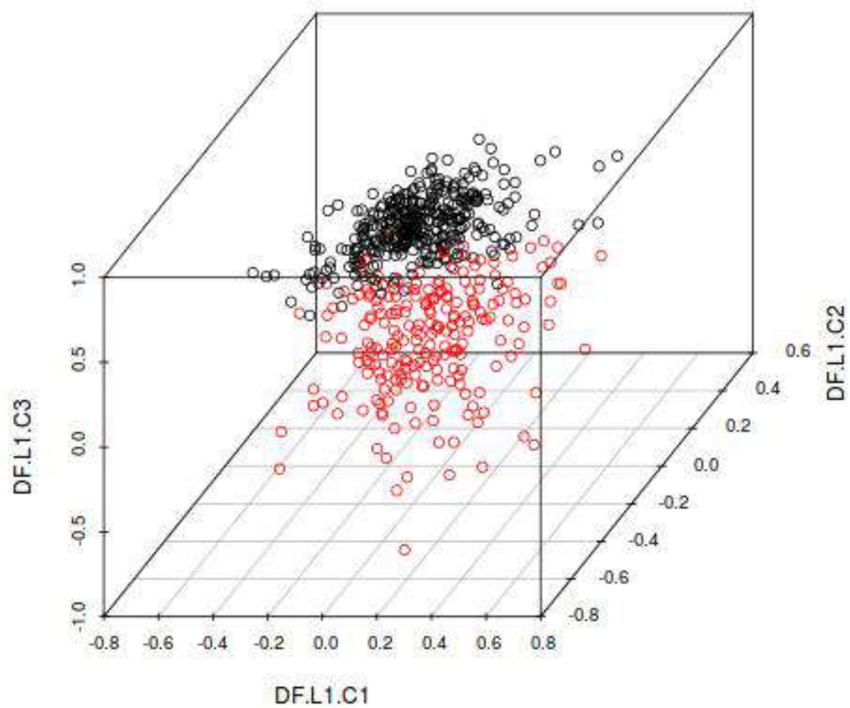
An **autoencoder** neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs.

The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction.



Deep Learning

- **Autoencoder** (autoencoder de h2o, una sola capa interna de 3 neuronas y 1000 "epochs". En todos los autoencoders uso la tangente hiperbólica como función de activación. WDBC (569 instancias con 30 atributos de entrada)

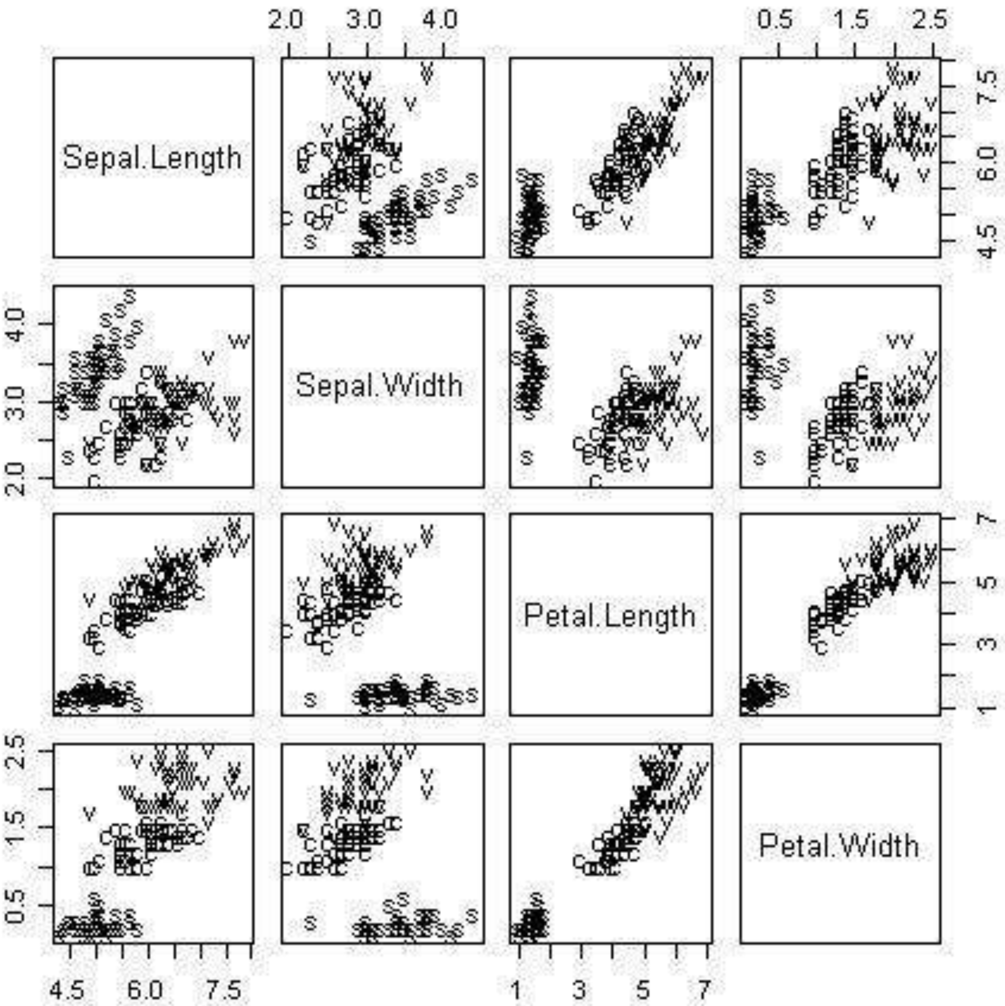


Credito: D. Charte

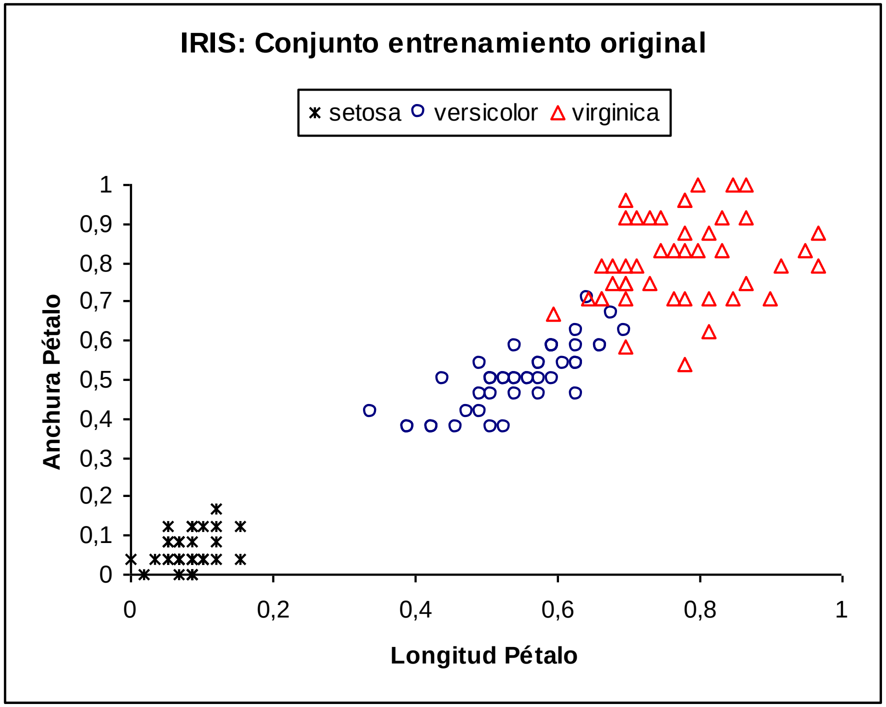
Ejemplo: Diseño de un Clasificador para *Iris*

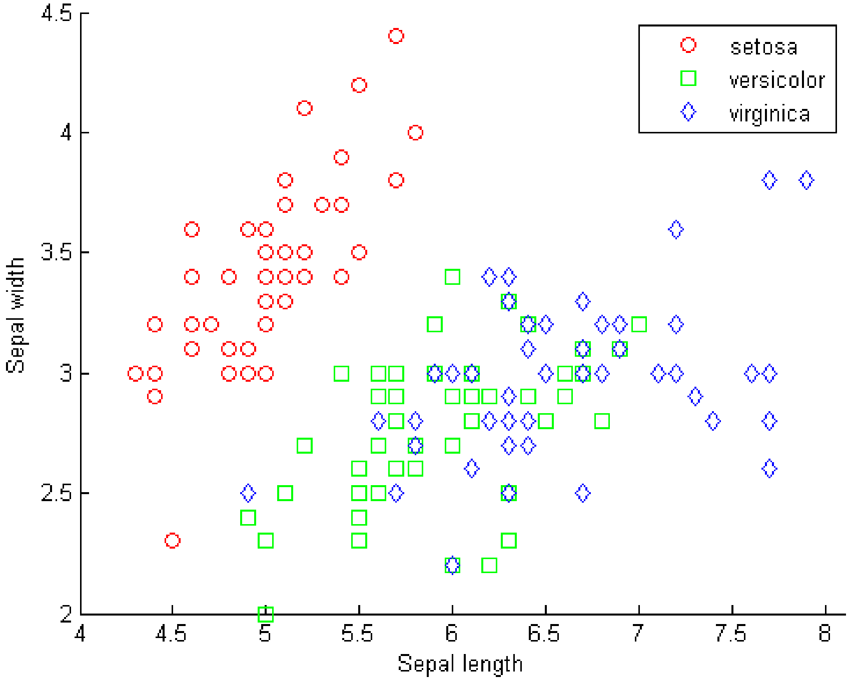
- **Problema simple muy conocido: *clasificación de lirios*.**
- **Tres clases de lirios: *setosa, versicolor* y *virginica*.**
- **Cuatro atributos: *longitud* y *anchura* de *pétalo* y *sépalo*, respectivamente.**
- **150 ejemplos, 50 de cada clase.**
- **Disponible en http://www.ics.uci.edu/~mlearn/MLRepository.html**

setosa, versicolor (C) y virginica



setosa, versicolor y virginica

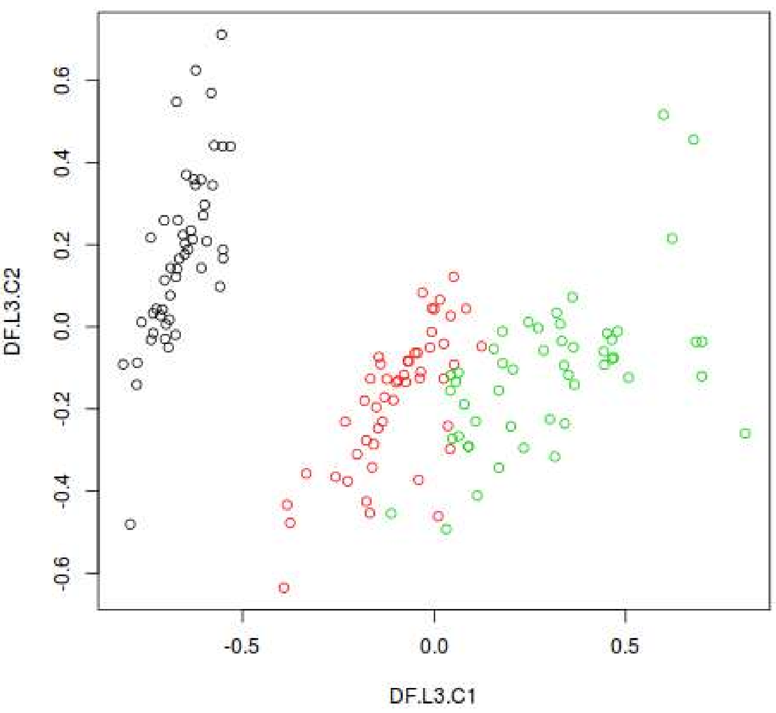


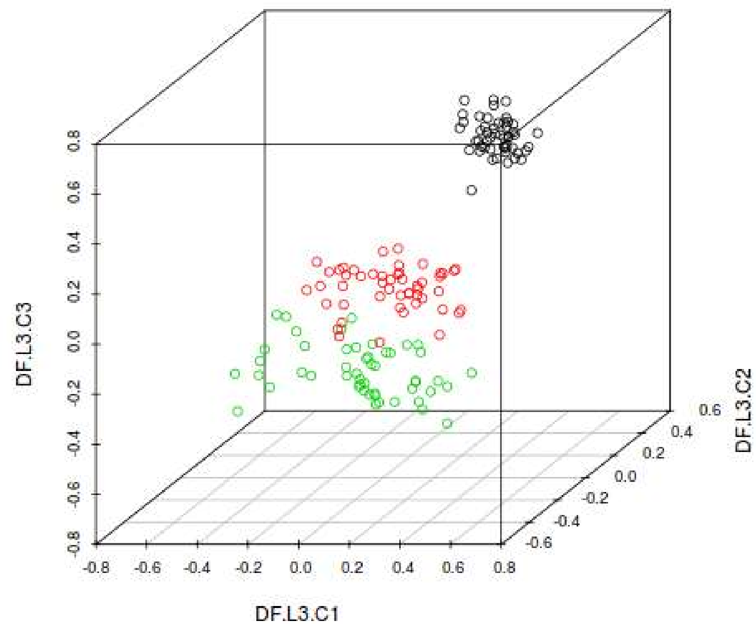


Deep Learning

- **Autoencoder** (autoencoder de h2o, salida de la capa intermedia capas internas de [8, 5, 3, 5, 8] neuronas y 100 "epochs" (el tridimensional), y [8, 5, 2, 5, 8] neuronas con 1000 "epochs" (el bidimensional).

setosa*, *versicolor* y *virginica





Credito: D. Charte

Deep Learning

- **Hybrid deep networks: DBN-DNN**

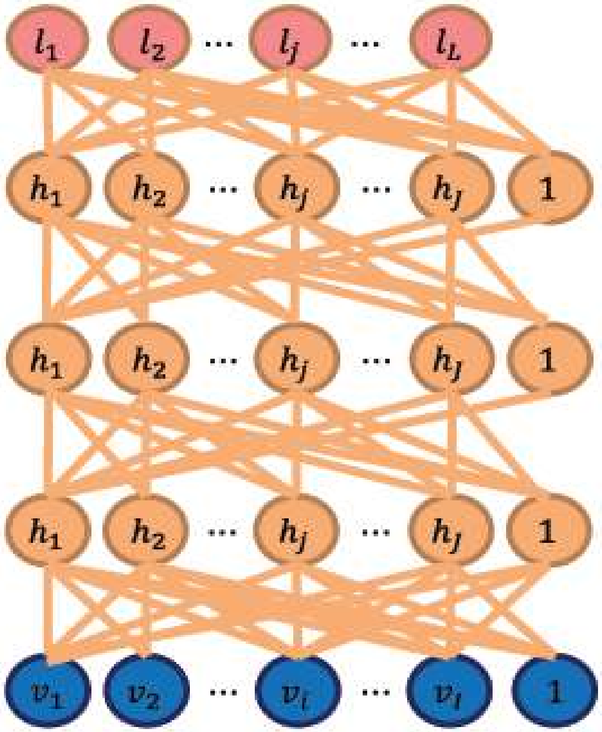


Figure 5.2: An illustration of the DBN-DNN architecture.

Credits: L. Deng and D. Yu. Deep Learning methods and applications. Foundations and Trends in Signal Processing. Vol. 7, Issues 3-4, 2014, pag. 246

Convolutional Neural Networks (Supervised)

Each module consists of a convolutional layer and a pooling layer.

Typically tries to compress large data (images) into a <u>smaller set of robust features, based on local variations.</u>

Basic convolution can still create many features.

CNNs have been found highly effective and been commonly used in computer vision and image recognition.

Convolutional Neural Networks

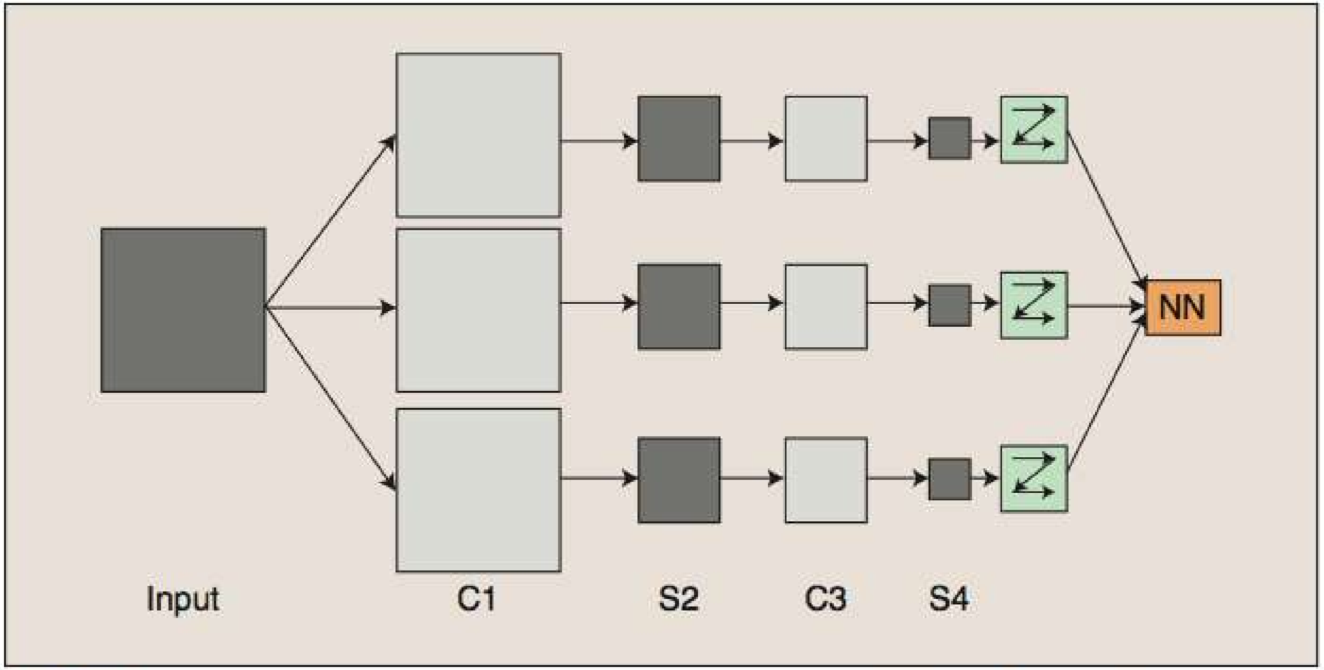
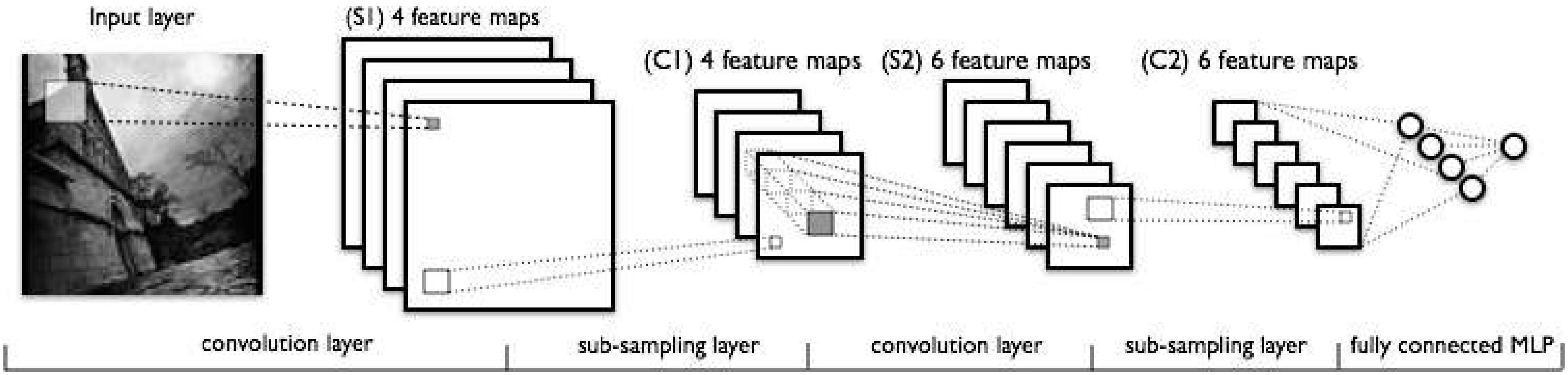


FIGURE 2 Conceptual example of convolutional neural network. The input image is convolved with three trainable filters and biases as in Figure 1 to produce three feature maps at the C1 level. Each group of four pixels in the feature maps are added, weighted, combined with a bias, and passed through a sigmoid function to produce the three feature maps at S2. These are again filtered to produce the C3 level. The hierarchy then produces S4 in a manner analogous to S2. Finally these pixel values are rasterized and presented as a single vector input to the "conventional" neural network at the output.

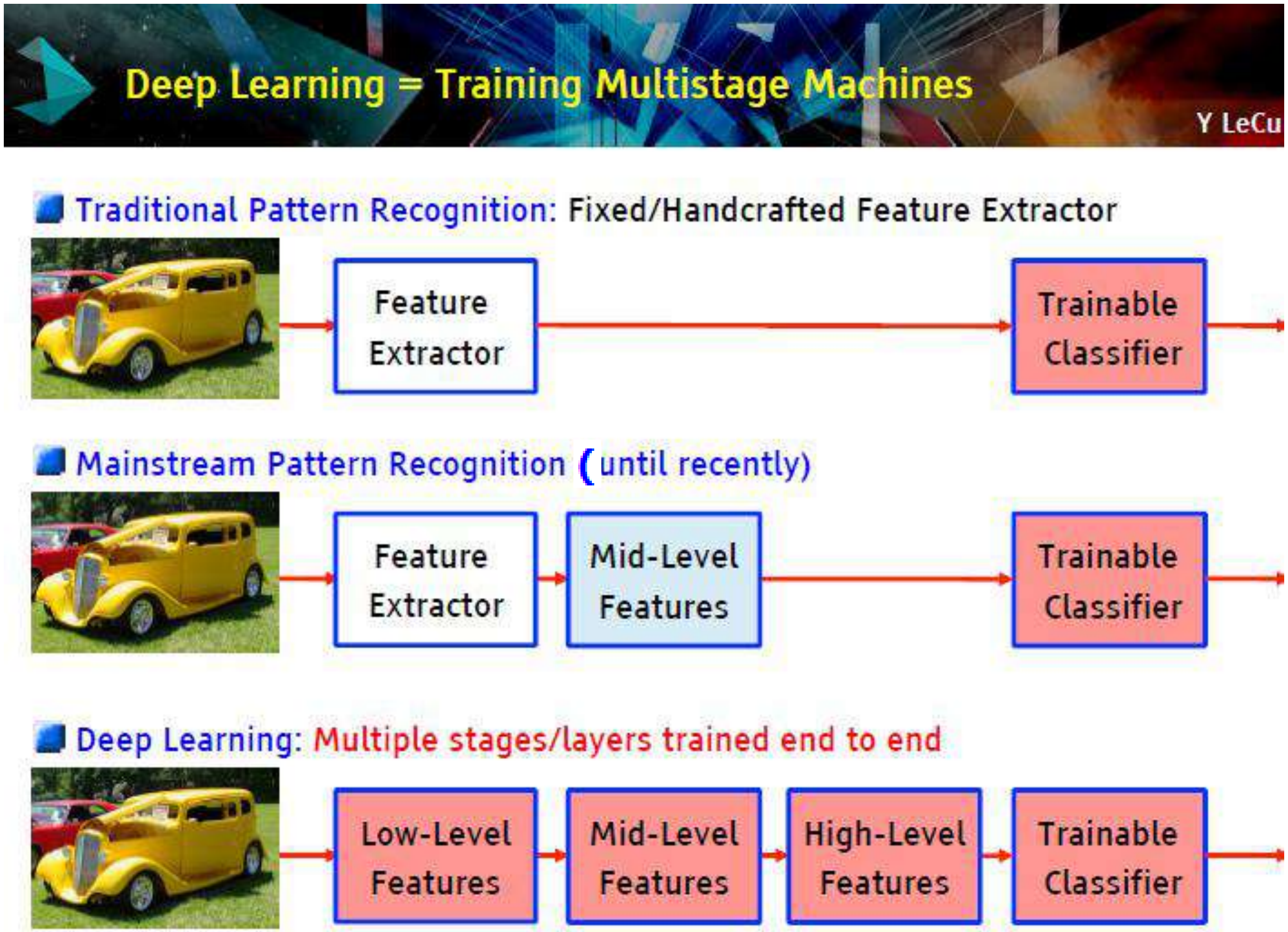
C layers are convolutions, S layers pool/sample

Deep Learning

Convolutional Neural Networks

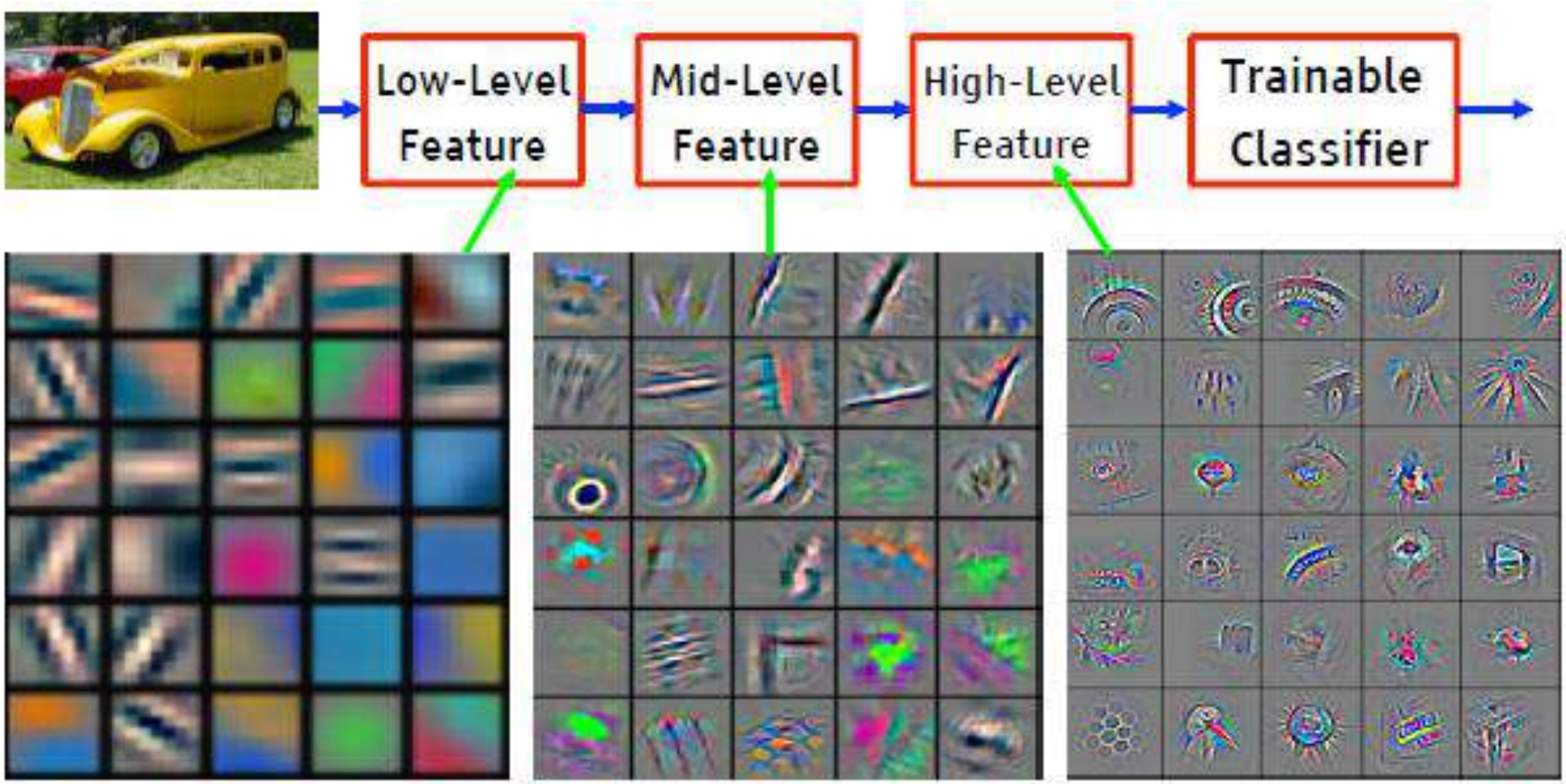


Deep Learning



Credits: http://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf

Deep Learning



Credits: http://www.iro.umontreal.ca/~bengioy/talks/DL-Tutorial-NIPS2015.pdf

Deep Learning

De la academia a la industria: DNNresearch Inc y Google Deepmind

Google Brain is a deep learning research project at Google

En 2013, Google adquirió la compañía DNNresearch Inc creada por uno de los pioneros de Deep Learning (Geoffrey Hinton).

En enero de 2014 se hizo con el control de la 'startup' Deepmind Technologies una pequeña empresa londinense en la trabajaban que algunos de los mayores expertos en 'deep learning'.

Deep Mind: Start up-2011

Demis Hassabis, Shane Legg y Mustafa Suleyman

Convolutional Neural Networks

NIPS2012, un caso de éxito de CNN para el challenge ILSVRC 2010

ImageNet Classification with Deep Convolutional Neural Networks
Part of: [Advances in Neural Information Processing Systems 25 (NIPS 2012)]

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

ImageNet is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labelers using Amazon's Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.

Deep Learning
Retos en los Juegos "inteligentes"

Cornell University Library

arXiv.org > cs > arXiv:1312.5602

Search or Article

Computer Science > Learning

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller

(Submitted on 19 Dec 2013)

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

Comments: NIPS Deep Learning Workshop 2013
Subjects: **Learning (cs.LG)**
Cite as: arXiv:1312.5602 [cs.LG]
(or arXiv:1312.5602v1 [cs.LG] for this version)

Submission history

From: Volodymyr Mnih [view email]
[v1] Thu, 19 Dec 2013 16:00:08 GMT (221kb,D)

http://arxiv.org/abs/1312.5602

Deep Learning
Retos en los Juegos "inteligentes"

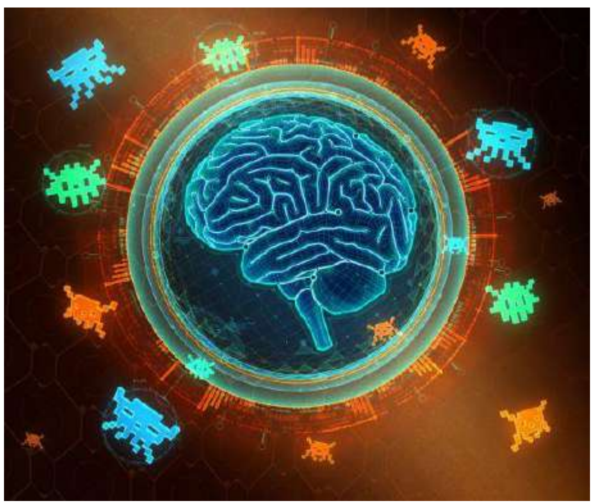
INTELIGENCIA ARTIFICIAL »

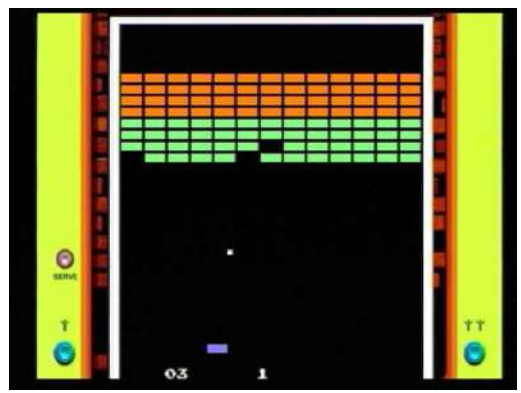
Este programa juega mejor a los 'marcianitos' que un humano

- Expertos en inteligencia artificial de Google crean un algoritmo que aprende por sí solo a jugar con decenas de videojuegos de los años 80 como 'Space Invaders' o el 'Comecocos'

Google DeepMind

El algoritmo se enfrentó a 49 juegos Arcade de los años 80 superando a otros algoritmos y a un jugador humano profesional





Juegos Arcade (Breakout)

http://elpais.com/elpais/2015/02/25/ciencia/1424860455_667336.html

http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html

Deep Learning
Retos en los Juegos “inteligentes”

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Aut

Archive > Volume 518 > Issue 7540 > Letters > Article

NATURE | LETTER

日本語要約

Human-level control through deep reinforcement learning

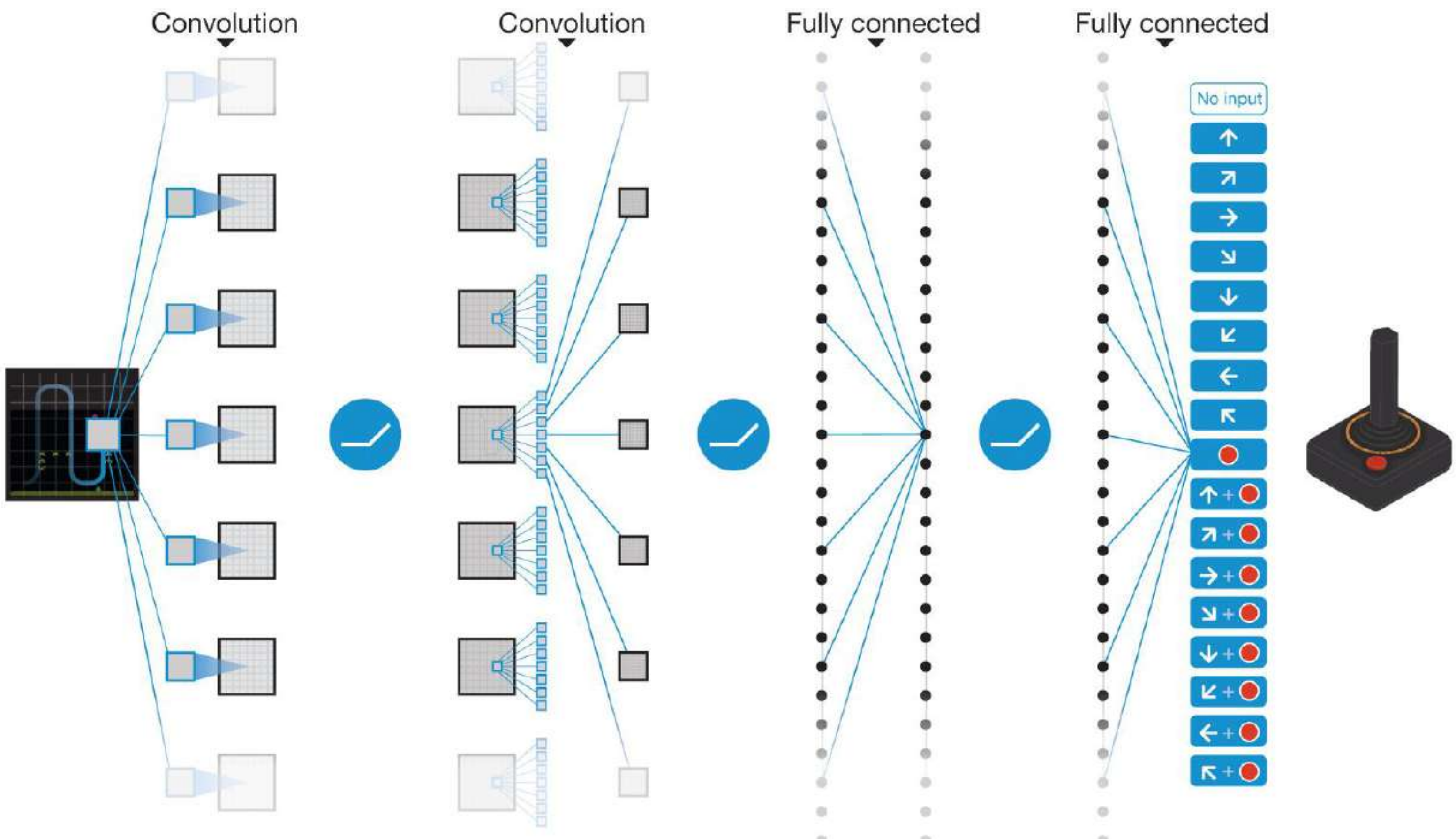
Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg & Demis Hassabis

Affiliations | Contributions | Corresponding authors

Nature **518**, 529–533 (26 February 2015) | doi:10.1038/nature14236
Received 10 July 2014 | Accepted 16 January 2015 | Published online 25 February 2015

http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html

Schematic illustration of the convolutional neural network.



V Mnih *et al. Nature* 518, 529-533 (2015)
doi:10.1038/nature14236

nature

http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html

Deep Learning
Retos en los Juegos “inteligentes”

Cornell University Library

arXiv.org > cs > arXiv:1509.01549

Search or Arti

Computer Science > Artificial Intelligence

Giraffe: Using Deep Reinforcement Learning to Play Chess

Matthew Lai

(Submitted on 4 Sep 2015 (v1), last revised 14 Sep 2015 (this version, v2))

This report presents Giraffe, a chess engine that uses self-play to discover all its domain-specific knowledge, with minimal hand-crafted knowledge given by the programmer. Unlike previous attempts using machine learning only to perform parameter-tuning on hand-crafted evaluation functions, Giraffe's learning system also performs automatic feature extraction and pattern recognition. The trained evaluation function performs comparably to the evaluation functions of state-of-the-art chess engines - all of which containing thousands of lines of carefully hand-crafted pattern recognizers, tuned over many years by both computer chess experts and human chess masters. Giraffe is the most successful attempt thus far at using end-to-end machine learning to play chess.

Comments: MSc Dissertation
Subjects: **Artificial Intelligence (cs.AI)**; Learning (cs.LG); Neural and Evolutionary Computing (cs.NE)
Cite as: arXiv:1509.01549 **[cs.AI]**
(or **arXiv:1509.01549v2 [cs.AI]** for this version)

Submission history

From: Matthew Lai [view email]
[v1] Fri, 4 Sep 2015 18:21:52 GMT (393kb,D)
[v2] Mon, 14 Sep 2015 15:42:35 GMT (393kb,D)

http://arxiv.org/abs/1509.01549

Deep Learning
Retos en los Juegos "inteligentes"

Imperial College London
Department of Computing

Giraffe: Using Deep Reinforcement Learning to Play Chess

by

Matthew Lai

Submitted in partial fulfilment of the requirements for the MSc Degree in Advanced Computing of Imperial College London

September 2015

Giraffe

Editar | 0 | 22

Home * Engines * Giraffe

Salvador Dalí - The Burning Giraffe [7]

Giraffe,
an experimental open source chess engine by Matthew Lai under the GNU General Public License, compliant to the Chess Engine Communication Protocol, written in C++11 and based on deep learning, which is topic of his Master's thesis in August 2015 [1] [2]. Giraffe uses the Eigen linear algebra library [3], and Pradyumna Kannan's magic move generator [4] [5]. As employee of Google DeepMind, Matthew Lai announced the discontinuation of the Giraffe project in January 2016 [6].

Description

Giraffe's evaluation function is a deep neural network trained by TDLeaf [8]. Its feature representation includes a map of static exchange evaluations for all squares and sides [9], a structure already proposed by Russell M. Church and Kenneth W. Church in *Plans, Goals, and Search Strategies for the Selection of a Move in Chess* [10]. Probability-based evaluation scores are not in centipawns nor linear to material, and span a +-10,000 range, with mate scores of +- 30,000. The search recently changed from traditional depth-based iterative deepening to assigning number of nodes (or time) to child nodes [11]. Node budget allocation will also become neural network based.

Table of Contents

http://arxiv.org/abs/1509.01549 https://chessprogramming.wikispaces.com/Giraffe

Deep Learning

Retos en los Juegos "inteligentes"

https://www.technologyreview.com/s/541276/deep-learning-machine-teaches-itself-chess-in-72-hours-plays-at-international-master/

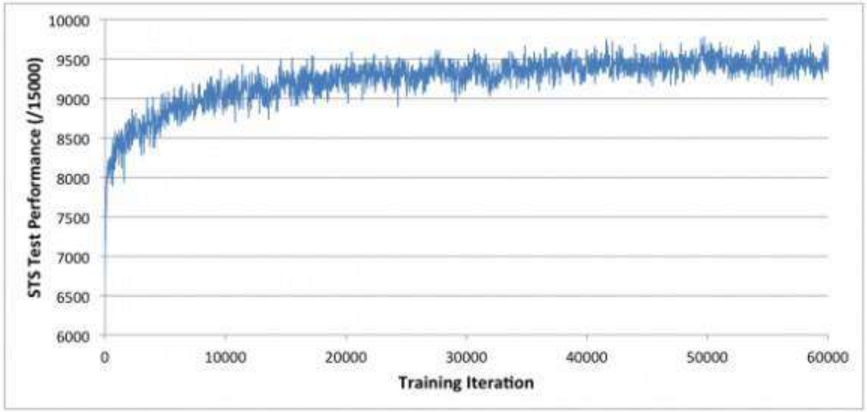
VIEW

30 COMMENTS

Emerging Technology From the arXiv
September 14, 2015

Deep Learning Machine Teaches Itself Chess in 72 Hours, Plays at International Master Level

In a world first, a machine plays chess by evaluating the board rather than using brute force to work out every possible move.



Ref: arxiv.org/abs/1509.01549 :
Giraffe: Using Deep Reinforcement Learning to Play Chess

Algunos datos:
von Neumann introduced the minimax algorithm in 1928
363 features
The evaluator network converges in about 72 hours on a machine with 2x10-core Intel Xeon E5-2660v2 CPU.
Giraffe is able to play at the level of an FIDE International Master

Deep Learning
Retos en los Juegos "inteligentes"

nature International weekly journal of science

NATURE | ARTICLE

日本語要約

Share/bookma

Mastering the game of Go with deep neural networks and tree search

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis

Affiliations | Contributions | Corresponding authors

Nature **529**, 484–489 (28 January 2016) | doi:10.1038/nature16961
Received 11 November 2015 | Accepted 05 January 2016 | Published online 27 January 2016

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

Deep Learning
Retos en los Juegos “inteligentes”

http://elpais.com/elpais/2016/01/26/ciencia/1453766578_683799.html

≡ EL PAÍS — Materia

COMPUTACIÓN ›

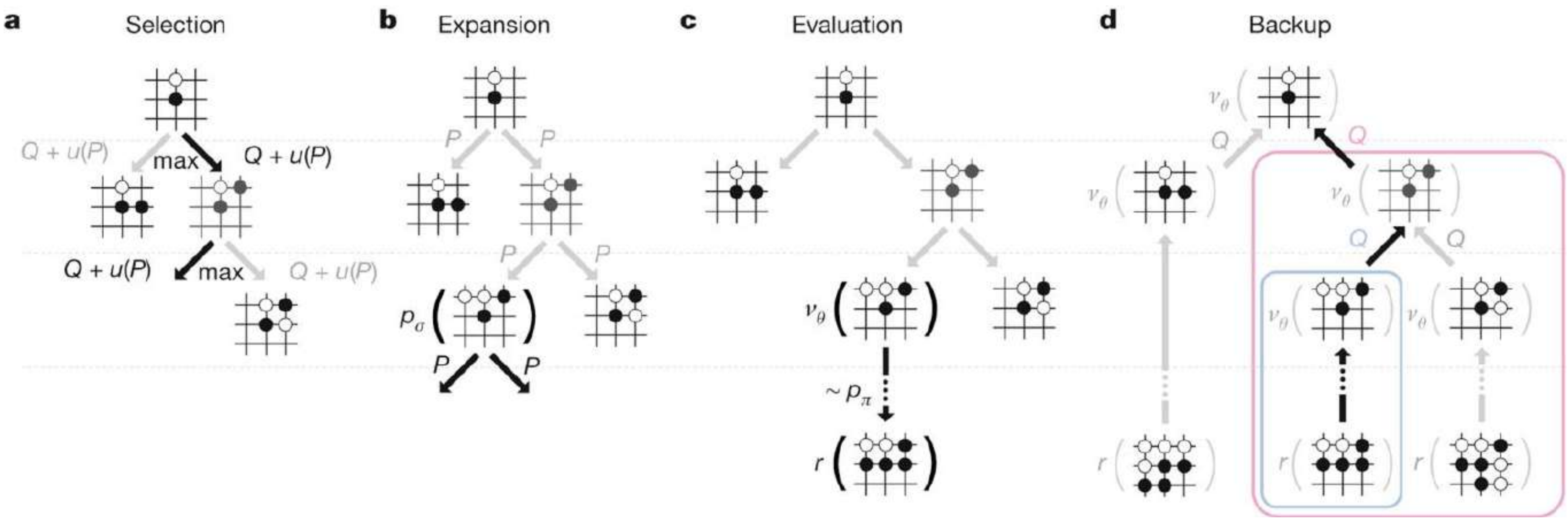
La inteligencia artificial conquista el último tablero de los humanos

Una máquina vence por primera vez a un jugador profesional del milenario juego chino Go

Ganar al Go es mucho más difícil que al ajedrez, ya que las variables son prácticamente infinitas; la fuerza bruta de un supercomputador, tratando de analizar de forma exhaustiva todas las posibilidades, sería inviable

Deep Learning

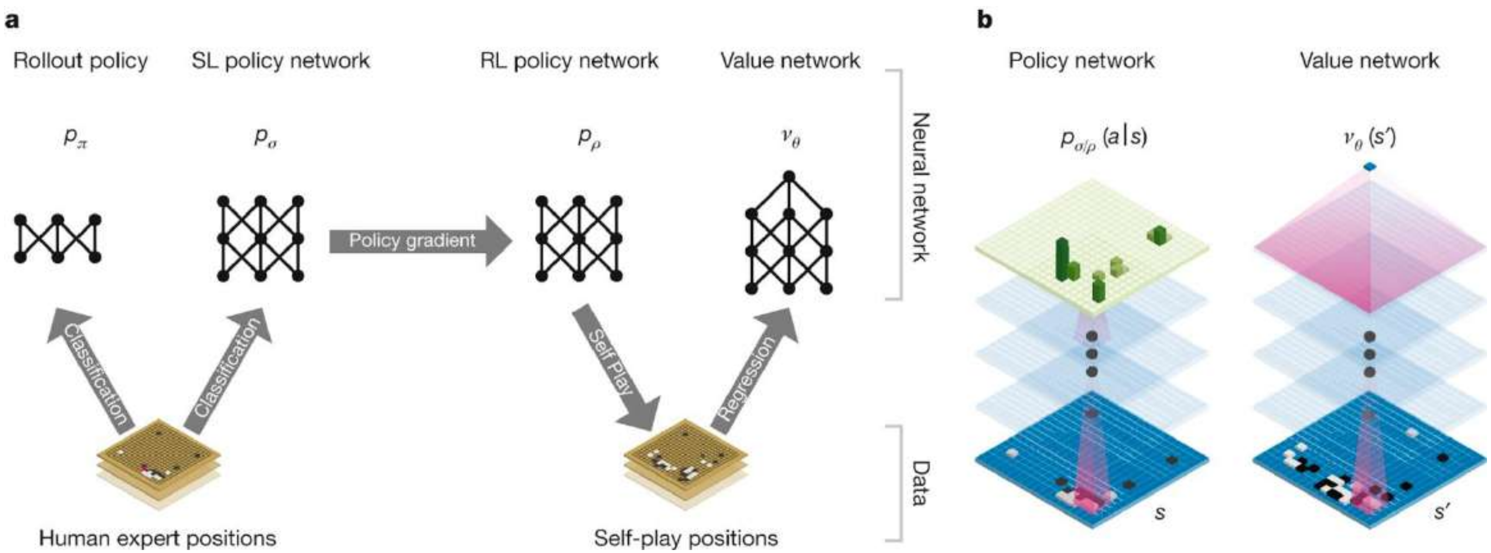
Retos en los Juegos "inteligentes"



Each simulation traverses the tree by selecting the edge with maximum action value *Q*, plus a bonus *u*(*P*) that depends on a stored prior probability *P* for that edge. b, The leaf node may be expanded; the new node is processed once by t...

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

Neural network training pipeline and architecture

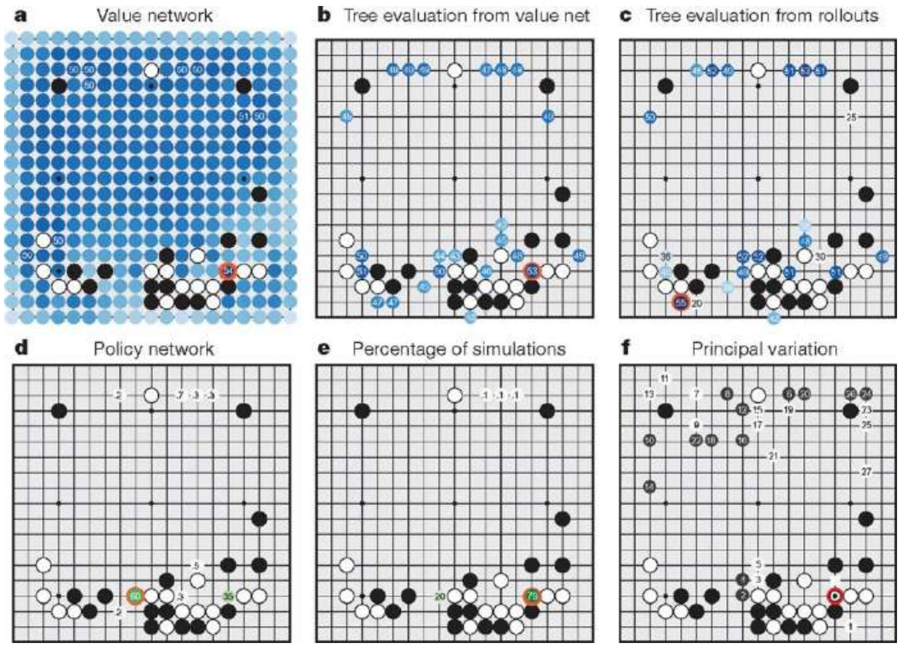


D Silver *et al. Nature* 529, 484–489 (2016)
doi:10.1038/nature16961

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

Deep Learning
Retos en los Juegos “inteligentes”

How AlphaGo (black, to play) selected its move in an informal game against Fan Hui



D Silver *et al. Nature* 529, 484–489 (2016)
doi:10.1038/nature16961

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

nature

Deep Learning

Retos en los Juegos “inteligentes”

DEEP LEARNING

Google AI algorithm masters ancient game of Go

Deep-learning software defeats human professional for first time.

Elizabeth Gibney

27 January 2016

PDF | Rights & Permissions

The computer that mastered Go

http://www.nature.com/news/google-ai-algorithm-masters-ancient-game-of-go-1.19234

nature

http://www.nature.com/news/the-go-files-champion-preps-for-1-million-machine-match-1.19541

nature International weekly journal of science

NATURE | BLOG

The Go Files: champion preps for $1 million machine match

***Nature* reports from a battle of man vs computer over the Go board.**

Tanguy Chouard

08 March 2016

SEOUL, SOUTH KOREA

Google DeepMind

Demis Hassabis (left), the CEO of Google DeepMind, and Lee Sedol (right), one of the world's top Go players, at the press conference in Seoul.

nature

https://actualidad.rt.com/ciencias/201602-inteligencia-artificial-alphago-google-gana-leyenda-go

9/03/2016

La inteligencia artificial hace historia: AlphaGo, de Google, derrotó al campeón del juego chino go

Publicado: 9 mar 2016 14:35 GMT

En un hito de la inteligencia artificial, la unidad DeepMind consiguió una victoria en la primera partida ante el surcoreano Lee Se-dol. "Estoy muy sorprendido, no esperaba perder", reconoció el jugador.

AFP PHOTO / GOOLGLE DEEPMIND

Deep Learning
Retos en los Juegos “inteligentes”

DeepMind AlphaGo vs Lee Sedol

Lee Sedol played a five game match against Google DeepMind's AlphaGo computer program in March 2016.

AlphaGo won the match:

AlphaGo 4 – Lee Sedol 1

After defeating Fan Hui 2p with a 5–0 score, Google DeepMind chose to challenge Lee Sedol because of his record as the best Go player in the world over the last decade.

The games will be even, with $1 million USD in prize money for the winner.

The match will take place from March 9 to 15, 2016, in Seoul, Korea. You can join us online to watch and discuss the match. Here's the schedule:

Game 1: Mar 9 — AlphaGo wins by resignation — Video
Game 2: Mar 10 — AlphaGo wins by resignation — Video
Game 3: Mar 12 — AlphaGo wins by resignation — Video
Game 4: Mar 13 — Lee Sedol wins by resignation! — Video
Game 5: Mar 15. — AlphaGo wins by resignation — Video

https://gogameguru.com/tag/deepmind-alphago-lee-sedol/

AlphaGo defeats Lee Sedol 4–1 in Google DeepMind Challenge Match

DeepMind's groundbreaking artificial intelligence, AlphaGo, defeated Lee Sedol 9p in the final game of the Google DeepMind Challenge Match on March 15, 2016, winning the five game match with a 4–1 score.

Demis Hassabis and the AlphaGo team receive the signed match Go board from Lee Sedol.

https://gogameguru.com/alphago-defeats-lee-sedol-4-1/

IMAGENET (ILSRVC): Microsoft Wins ImageNet Using Extremely Deep Neural Networks

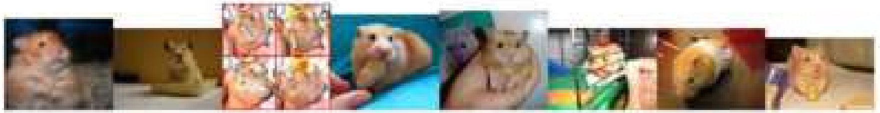
Microsoft Wins ImageNet Using Extremely Deep Neural Networks

Written by Mike James

Tuesday, 15 December 2015

While just about everyone else is forming foundations and institutes to further AI, some researchers are actually getting on with doing it. This year's ImageNet competition has been won by Microsoft, which comes as something of a surprise.

IMAGENET



It is a surprise because overall it is Google that makes the most noise about AI and in the popular mind at least Google is miles ahead of the competition. In truth all of the big companies engaged in the race to bring AI to the masses are really just fine tuning the same basic approach to the problem - the Deep Neural Network.

Microsoft's network was really deep at 150 layers (extremely deep neural network). To do this the team had to overcome a fundamental problem inherent in training deep neural networks. As the network gets deeper training becomes more difficult so you encounter a seemingly paradoxical situation that adding layers makes the performance worse.
The solution proposed is called **deep residual learning**.

http://www.image-net.org/challenges/LSVRC/

http://www.i-programmer.info/news/105-artificial-intelligence/9266-microsoft-wins-imagenet-using-extremely-deep-neural-networks.html

IMAGENET (ILSRVC 2015): Microsoft Wins ImageNet Using Extremely Deep Neural Networks

Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

(Submitted on 10 Dec 2015)

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers---8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.
The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

Comments: Tech report
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: arXiv:1512.03385 [cs.CV]
(or **arXiv:1512.03385v1** [cs.CV] for this version)

http://arxiv.org/abs/1512.03385

Deep Learning
Retos en la "pintura"

Cornell University Library

arXiv.org > cs > arXiv:1508.06576

Search or Article-id

Computer Science > Computer Vision and Pattern Recognition

A Neural Algorithm of Artistic Style

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

(Submitted on 26 Aug 2015 (v1), last revised 2 Sep 2015 (this version, v2))

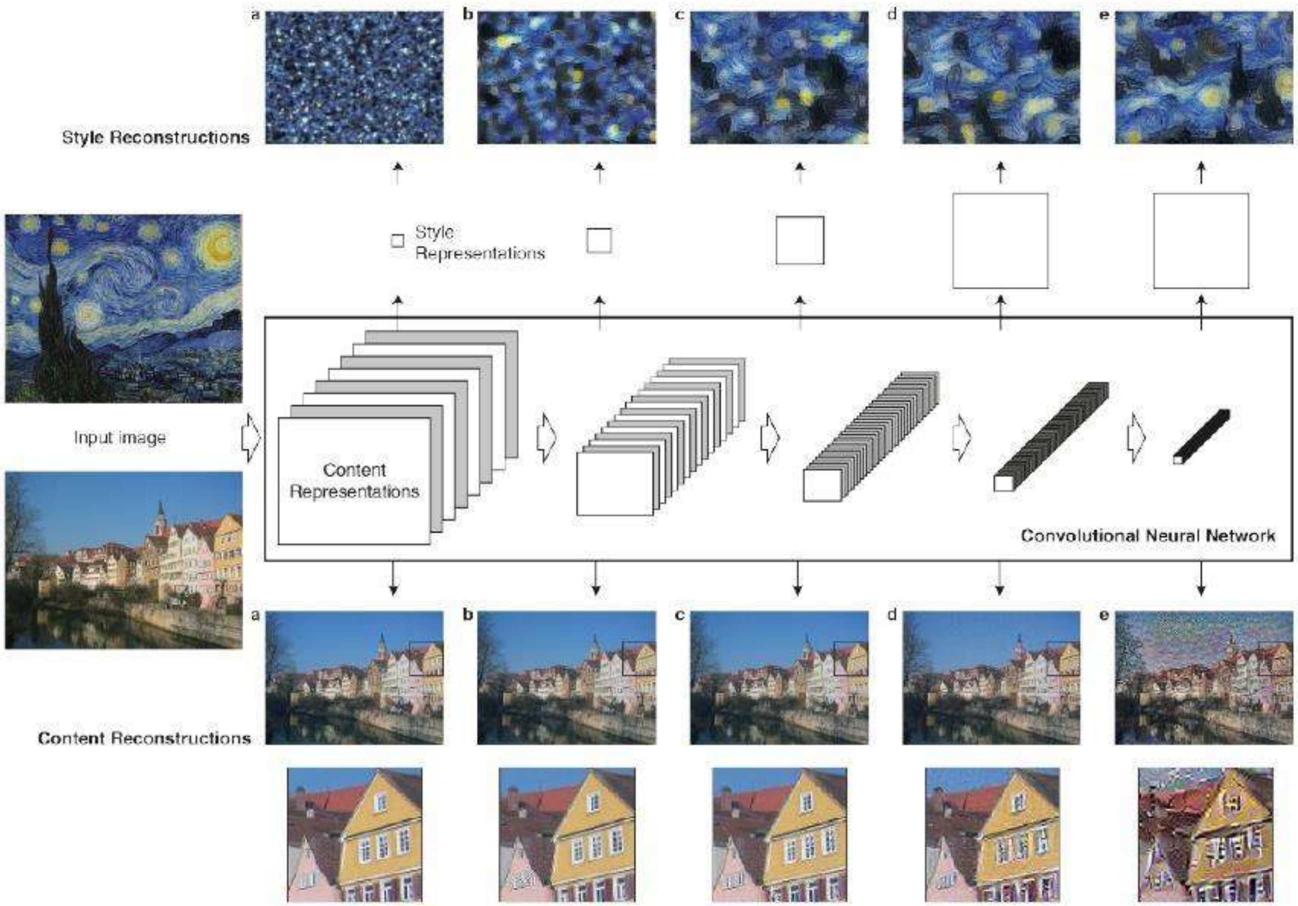
In fine art, especially painting, humans have mastered the skill to create unique visual experiences through composing a complex interplay between the content and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities. However, in other key areas of visual perception such as object and face recognition near-human performance was recently demonstrated by a class of biologically inspired vision models called Deep Neural Networks. Here we introduce an artificial system based on a Deep Neural Network that creates artistic images of high perceptual quality. The system uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images. Moreover, in light of the striking similarities between performance-optimised artificial neural networks and biological vision, our work offers a path forward to an algorithmic understanding of how humans create and perceive artistic imagery.

Subjects: **Computer Vision and Pattern Recognition (cs.CV)**; Neural and Evolutionary Computing (cs.NE); Neurons and Cognition (q-bio.NC)

Cite as: **arXiv:1508.06576 [cs.CV]**
(or **arXiv:1508.06576v2 [cs.CV]** for this version)

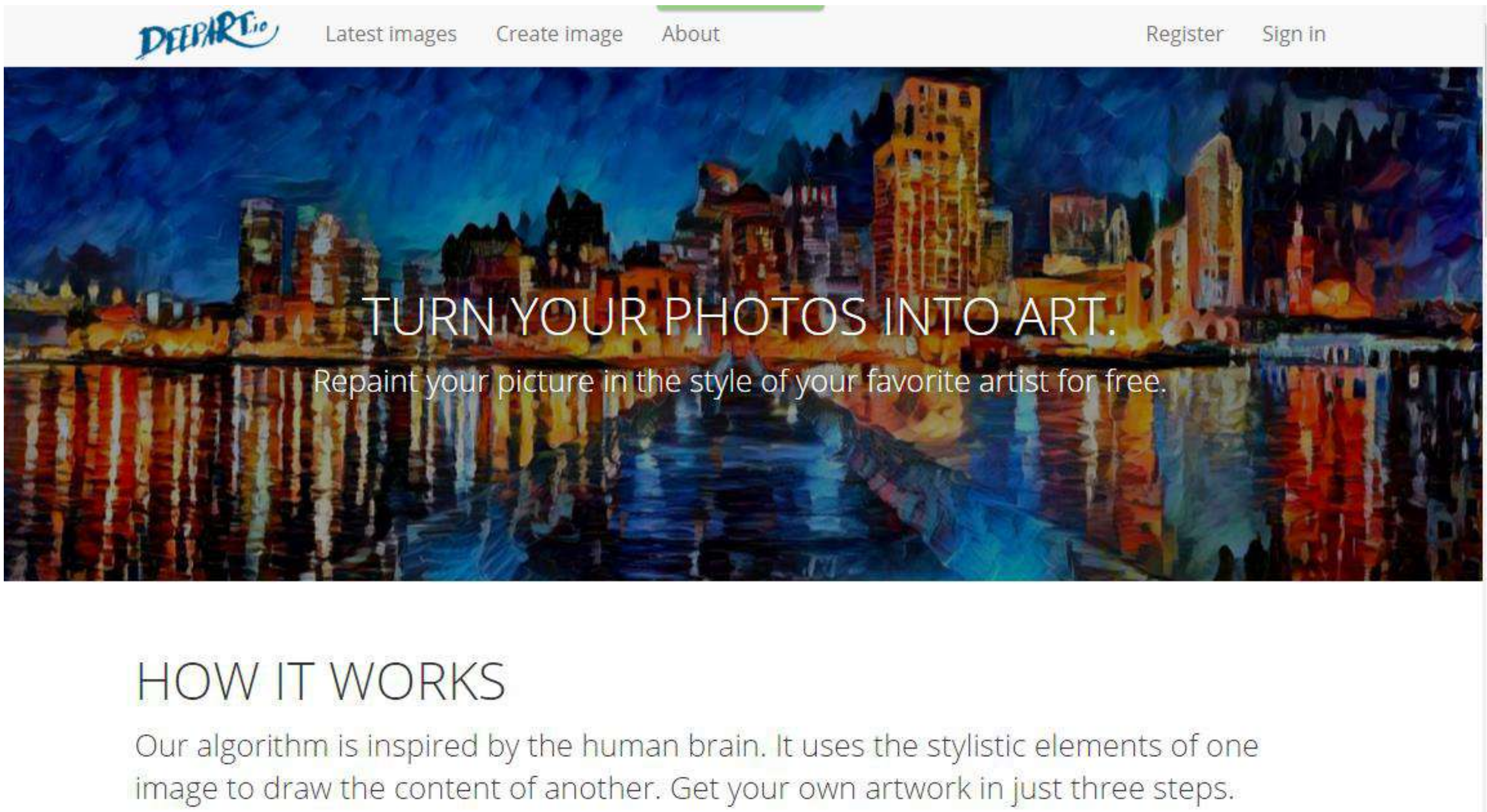
http://arxiv.org/abs/1508.06576

Deep Learning
Retos en la "pintura"



Credits: http://arxiv.org/abs/1508.06576

Deep Learning
Retos en la "pintura"



1 Upload photo

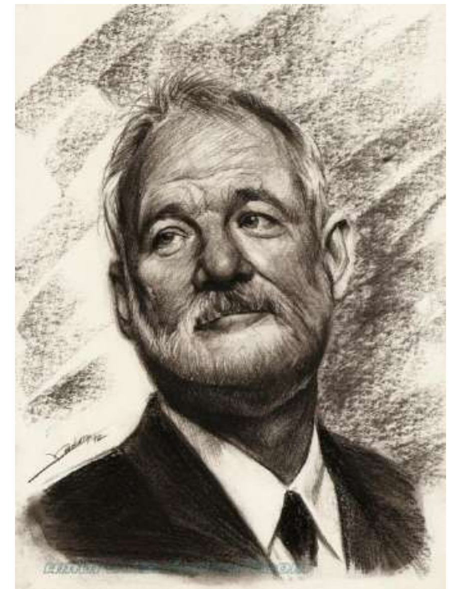
2 Choose style

3 Submit

http://www.deepart.io/

Deep Learning
Retos en la "pintura"
Ejemplos del resultado de DeepART






http://www.deepart.io/

DEEP LEARNING

Modelo de CNN utilizado y descripción de la metodología

Cornell University Library

arXiv.org > cs > arXiv:1409.1556

Search or Article-id

Computer Science > Computer Vision and Pattern Recognition

Very Deep Convolutional Networks for Large-Scale Image Recognition

Karen Simonyan, Andrew Zisserman

(Submitted on 4 Sep 2014 (v1), last revised 10 Apr 2015 (this version, v6))

http://arxiv.org/abs/1409.1556
http://arxiv.org/abs/1508.06576

Cornell University Library

arXiv.org > cs > arXiv:1508.06576

Computer Science > Computer Vision and Pattern Recognition

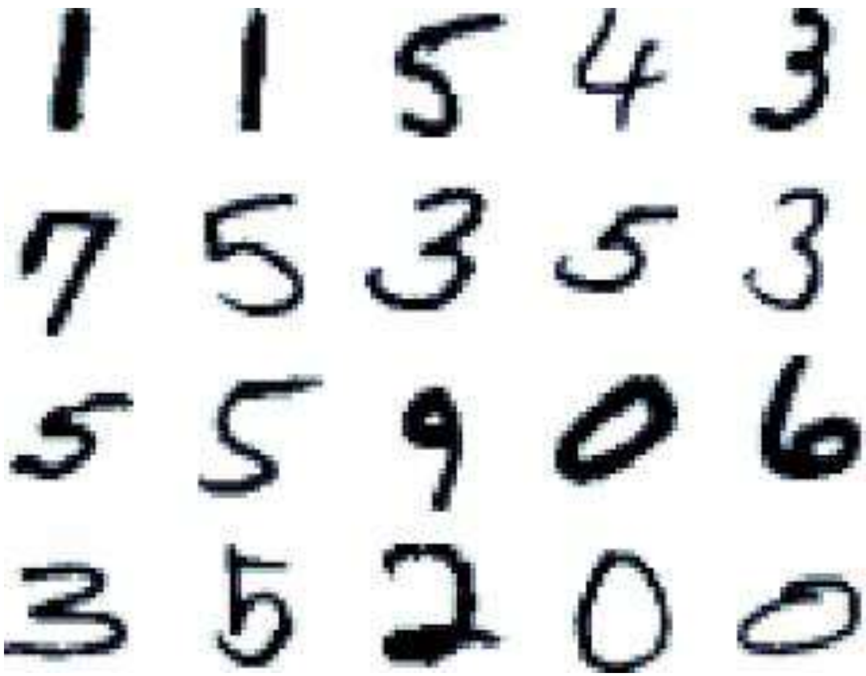
A Neural Algorithm of Artistic Style

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

(Submitted on 26 Aug 2015 (v1), last revised 2 Sep 2015 (this version, v2))

Deep Learning
Digit Recognizer Kaggle

Caso estudio: Digit Recognizer Kaggle (A. Herrera-Poyatos)



Andrés Herrera Poyatos

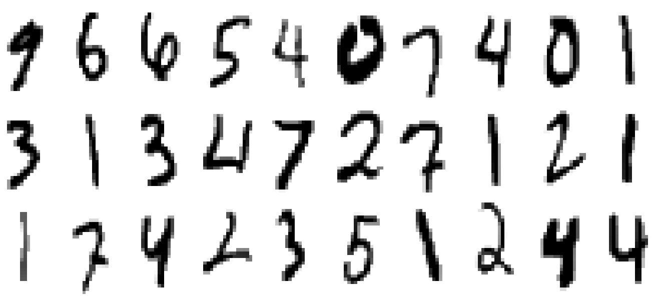
Repositorio en GitHub con el código:

https://github.com/andreshp/Kaggle

Deep Learning

Caso estudio: Digit Recognizer Kaggle

- Desarrollar un reconocedor de dígitos es uno de los **problemas clásicos** de la ciencia de datos.
- Sirve de **benchmark** para probar los nuevos algoritmos. ¡Ningún un humano acierta el 100%!
- **Aplicación práctica:** detección de matrículas, conversión de escritura a mano en texto …



9	**6**	**6**	**6**	**4**	**0**	**7**	**4**	**0**	**1**
3	**1**	**3**	**4**	**7**	**2**	**7**	**1**	**2**	**1**
1	**7**	**4**	**2**	**3**	**5**	**1**	**2**	**4**	**4**

Créditos: A. Herrera-Poyatos

Deep Learning

Caso estudio: Digit Recognizer Kaggle

- **Kaggle** mantiene una competición pública:

http://www.kaggle.com/c/digit-recognizer

- Datos a analizar: MNIST DATA (60.000 instances)

http://yann.lecun.com/exdb/mnist/

- Rodrigo Benenson has compiled an informative summary page

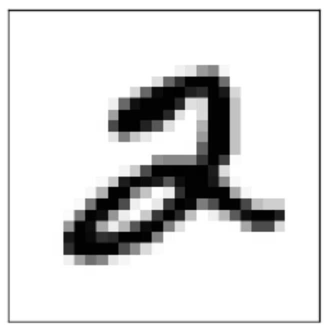
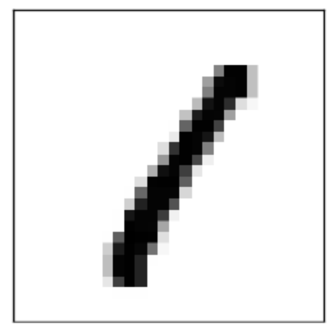
http://rodrigob.github.io/are_we_there_yet/build/classificatio n_datasets_results.html

Créditos: A. Herrera-Poyatos

Deep Learning

Caso estudio: Digit Recognizer Kaggle

- Data Set:
  - Training Set: 42.000 Imágenes
  - Test Set: 28.000 Imágenes
- Imagen:
  - 10 clases: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - 28x28 píxeles
  - Ejemplo:

- Puntuación para la clasificación general en Kaggle: índice de acierto sobre un 25% del Test Set.

Créditos: A. Herrera-Poyatos

1. Primer paso

1. Utilizar los algoritmos más conocidos para usarlos como benchmark
   - KNN con k = 10   0.96557 en Kaggle
   - Random Forest con 1000 árboles   0.96829 en Kaggle

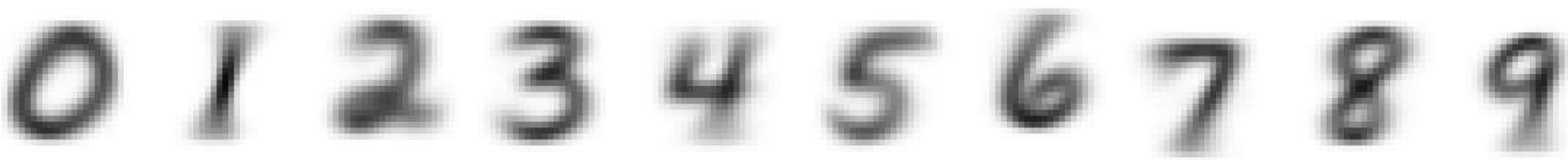
2. Optimizar los parámetros de un algoritmo sencillo
   - Cross Validation sobre KNN para encontrar el mejor valor de k.

     Solución: K=1   0.97114 en Kaggle

Créditos: A. Herrera-Poyatos

2. Visualización

- Media de todas las imágenes del training set por clases:

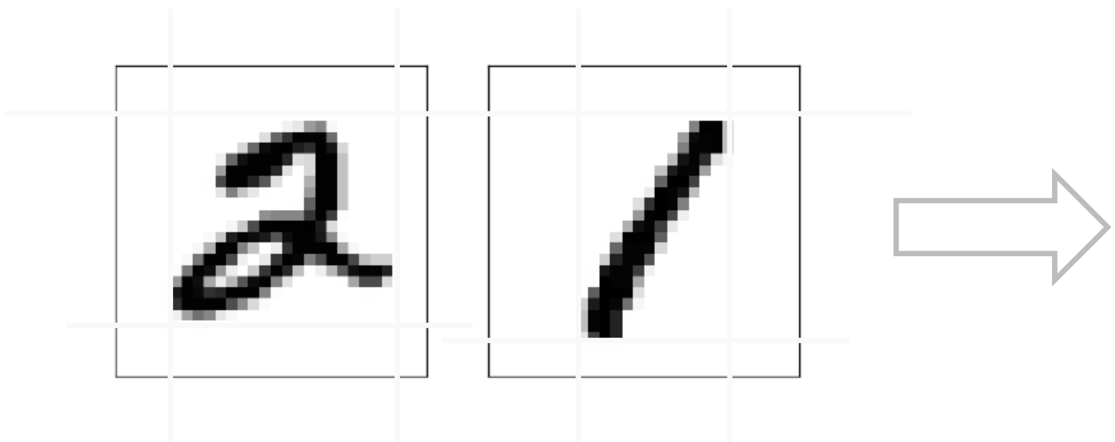


- **Observación:** Incluso las medias no están centradas (ver 6 y 7). Esto provoca problemas para clasificarlas correctamente.

  Solución: Preprocesamiento

3. Preprocesamiento

- **Idea:** Eliminar las filas y columnas de píxeles en blanco.

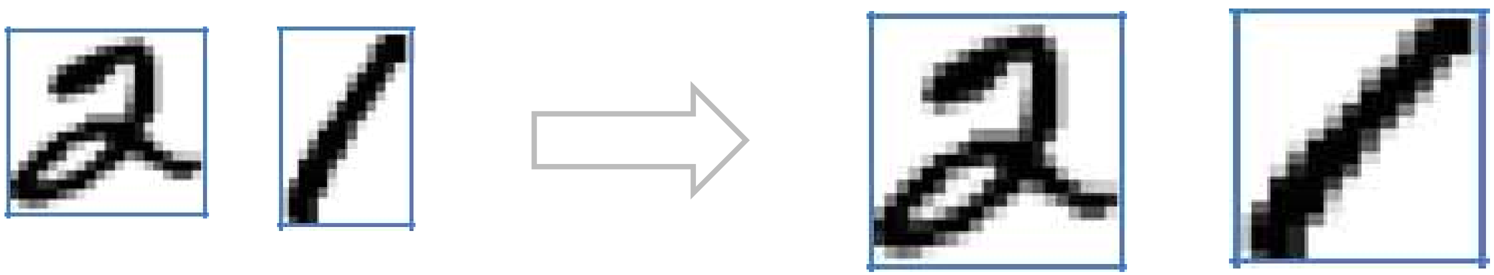


- **Problema:** Las nuevas imágenes tienen diferentes dimensiones.

Créditos: A. Herrera-Poyatos

3. Preprocesamiento

- **Solución: Redimensionar** las imágenes a 20x20 píxeles (tras el proceso anterior la imagen más grande tiene esa dimensión)



- Media de las imágenes del training set preprocesadas:

- ¡Todas están centradas!
- KNN con k=1 sobre los datos preprocesados 0.97557 en Kaggle

Créditos: A. Herrera-Poyatos

Deep Learning

Caso estudio: Digit Recognizer Kaggle

Librería que contiene Deep Learning: H2O

http://0xdata.com/

Récord del mundo en el problema MNIST sin preprocesamiento

http://0xdata.com/blog/2015/02/deep-learning-performance/

- Soporte para R, Python, Hadoop y Spark
- Se puede instalar en cualquier máquina, incluyendo un portatil, cluster de ordenadores, …
- Funcionamiento: Crea una máquina virtual con Java en la que optimiza el paralelismo de los algoritmos.

Data Science in H_2O

- Cox Proportional Hazards Model
- Deep Learning
- Generalized Linear Model
- Gradient Boosted Regression and Classification
- K-Means
- Naive Bayes
- Principal Components Analysis
- Random Forest
- Summary
- Data Science and Machine Learning
- Stochastic Gradient Descent
- References

Créditos: A. Herrera-Poyatos

Deep Learning

Caso estudio: Digit Recognizer Kaggle

http://cran.r-project.org/web/packages/h2o/index.html

h2o: R Interface for H2O

R scripting functionality for H2O, the open source math engine for big data that computes parallel distributed machine learning algorithms such as generalized linear models, gradient boosting machines, random forests, and neural networks (deep learning) within various cluster environments.

Version:	3.8.1.3
Depends:	R (≥ 2.13.0), methods, statmod, stats
Imports:	graphics, RCurl, jsonlite, tools, utils
Suggests:	devtools, roxygen2, testthat, ggplot2, mlbench
Published:	2016-03-15
Author:	Spencer Aiello, Tom Kraljevic and Petr Maj, with contributions from the H2O.ai team
Maintainer:	Tom Kraljevic <tomk at 0xdata.com>
License:	Apache License (== 2.0)
URL:	http://www.h2o.ai
NeedsCompilation:	no
SystemRequirements:	Java (>= 1.7)
Materials:	NEWS
In views:	HighPerformanceComputing, MachineLearning
CRAN checks:	h2o results

Créditos: A. Herrera-Poyatos

4. Deep Learning sobre MNIST DATA preprocesados

```
######################################################################
# Author: Andrés Herrera Poyatos
# Date:   March, 2015
# Kaggle Digit Recognizer - Deep Learning
######################################################################

#------------------------- GETTING DATA --------------------------#

working_directory <- "/home/andreshp/ComputerScience/MachineLearning/Kaggle/DigitRecognizer/R"
train <- read.csv(paste(working_directory, "/csv/train_preprocesed.csv", sep=""), header=TRUE)
test <-  read.csv(paste(working_directory, "/csv/test_preprocesed.csv", sep=""),  header=TRUE)

#---------------------- STARTING H2O LIBRARY ----------------------#

# Library h2o for deep learning
library(h2o)

# Start a local cluster with 4GB RAM
localH2O = h2o.init(ip = "localhost", port = 54321, startH2O = TRUE, Xmx = '4g')

# Convert the data to the h2o format:
train_h2o <- as.h2o(localH2O, train)
test_h2o  <- as.h2o(localH2O, test)
```

Andrés Herrera Poyatos
Repositorio en GitHub con el código:
https://github.com/andreshp/Kaggle

Créditos: A. Herrera-Poyatos

4. Deep Learning sobre MNIST DATA preprocesados

hidden=C(1024,1024,2048)

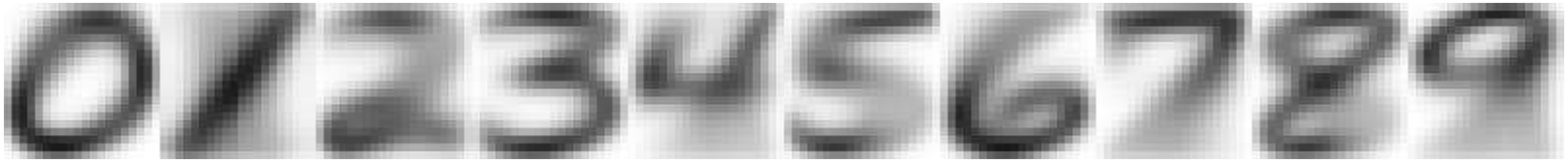
```
model <- h2o.deeplearning( x = 2:ncol(train),                        # column numbers for predictors
                           y = 1,                                     # column number for label
                           data = train_h2o,                          # data in H2O format
                           activation = "RectifierWithDropout",       # Rectifier With Dropout
                           balance_classes = TRUE,
                           hidden=c(1024,1024,2048),
                           epochs = 200,                              # max. no. of epochs
                           adaptive_rate = FALSE,
                           rate=0.01,                                 # rate
                           rate_annealing = 1.0e-6,                   # annealing rate
                           rate_decay = 1.0,
                           momentum_start = 0.5,
                           momentum_ramp = 42000*12,
                           momentum_stable = 0.99,
                           input_dropout_ratio = 0.2,
                           l1 = 1.0e-5, l2 = 0.0,
                           max_w2 = 15.0,
                           initial_weight_distribution = "Normal",
                           initial_weight_scale = 0.01,
                           nesterov_accelerated_gradient = T,
                           loss = "CrossEntropy",
                           fast_mode = T,
                           diagnostics = T,
                           ignore_const_cols = T,
                          force_load_balance = T)
```

Andrés Herrera Poyatos
Repositorio en GitHub con el código:
https://github.com/andreshp/Kaggle

Créditos: A. Herrera-Poyatos

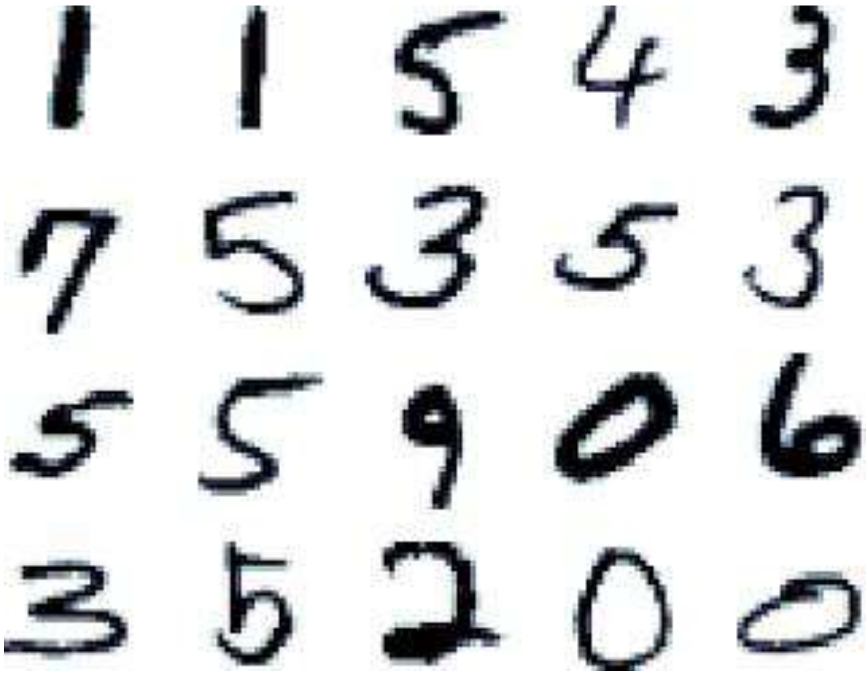
4. Deep Learning sobre MNIST DATA preprocesados

- Tiempo de Ejecución: 2.5 horas de cómputo con un Procesador Intel i5 a 2.5 GHz.
- Resultados conseguidos:
  - Deep Learning → 0.98229 en Kaggle
  - Preprocesamiento + Deep Learning → 0.98729 en Kaggle
- ¡El primer resultado era 0.96557!



Créditos: A. Herrera-Poyatos

Deep Learning
Digit Recognizer and Convolutional NN
A complete description
http://neuralnetworksanddeeplearning.com/chap6.html



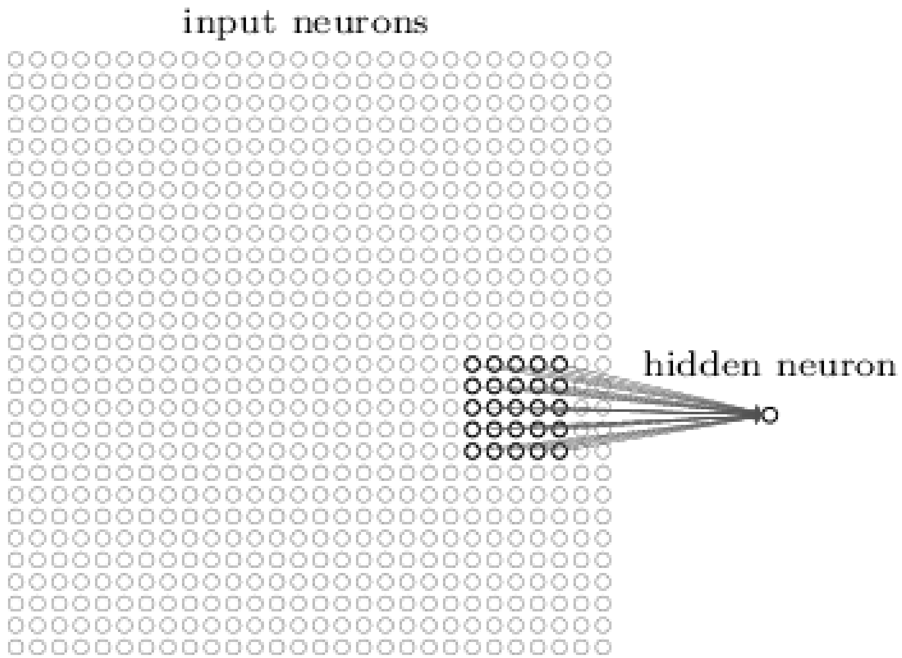
By Michael Nielsen / Jan 2016

Deep Learning
Digit Recognizer and Convolutional NN

http://neuralnetworksanddeeplearning.com/chap6.html

Convolutional neural networks use three basic ideas: *local receptive fields, shared weights,* and *pooling*. Let's look at each of these ideas in turn.

Local receptive fields: To be more precise, each neuron in the first hidden layer will be connected to a small region of the input neurons, say, for example, a 5×5 region, corresponding to 25 input pixels. So, for a particular hidden neuron, we might have connections that look like this:

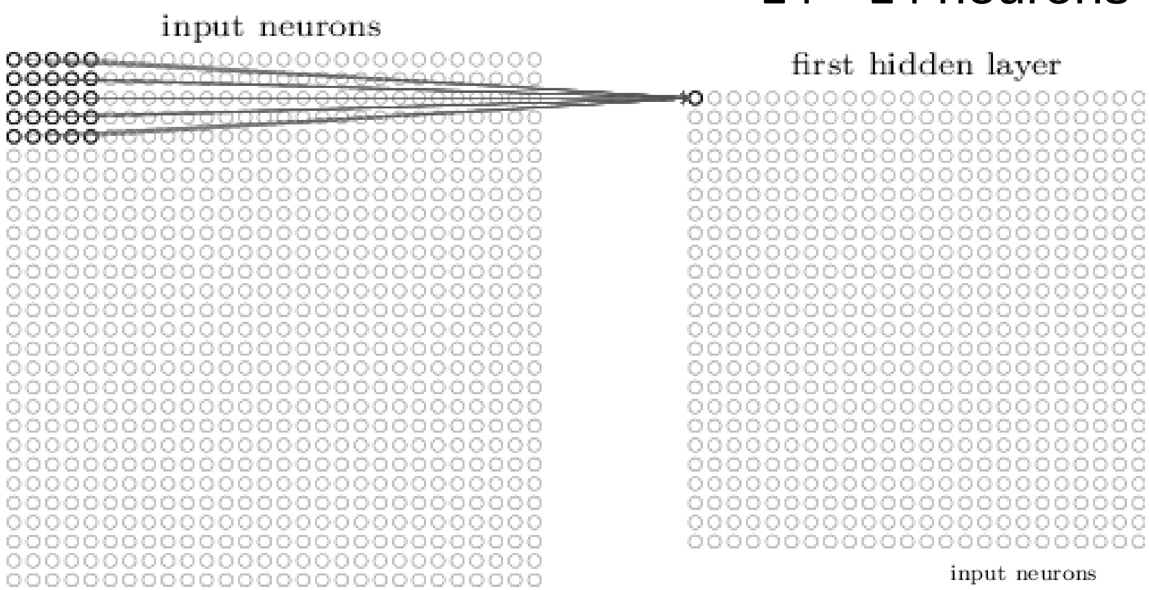


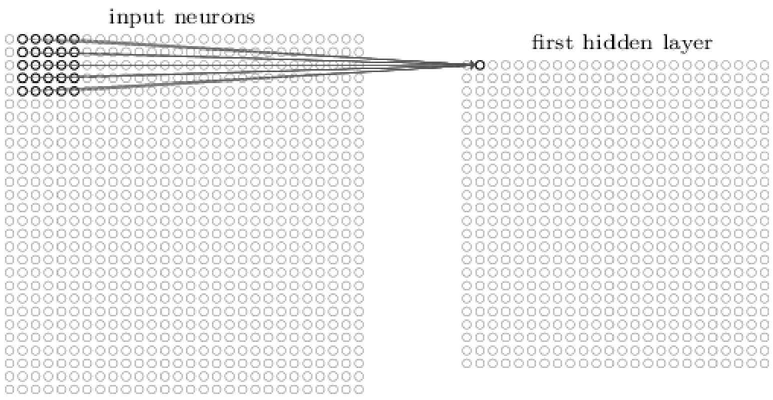
Deep Learning
Digit Recognizer and Convolutional NN

http://neuralnetworksanddeeplearning.com/chap6.html

Local receptive fields:

24 × 24 neurons



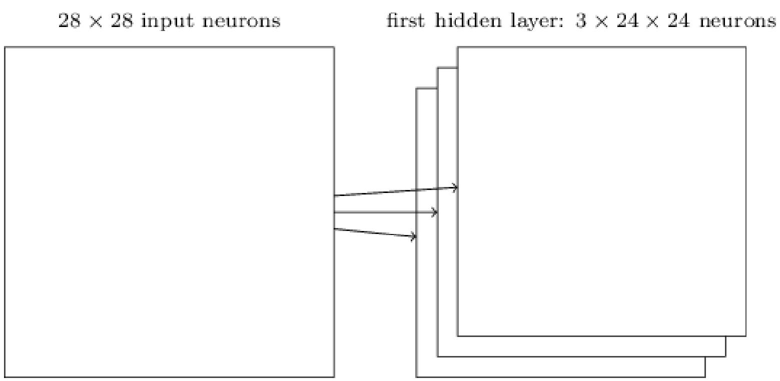


28 × 28 input image

Shared weights and biases: the *same* weights and bias for each of the 24×24 hidden neurons (sigmoide function)

$$\sigma\left(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m}\, a_{j+l,k+m}\right). \qquad (125)$$

The map from the input layer to the hidden layer a *feature map*.



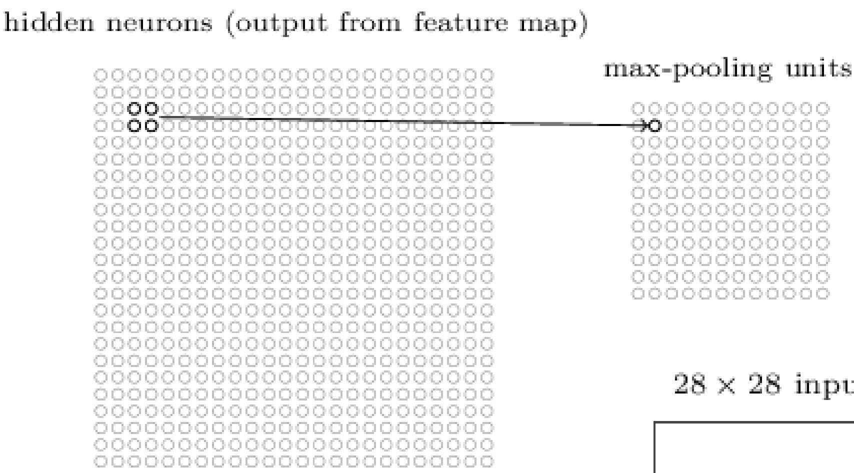
In the example shown, there are 3 feature maps.

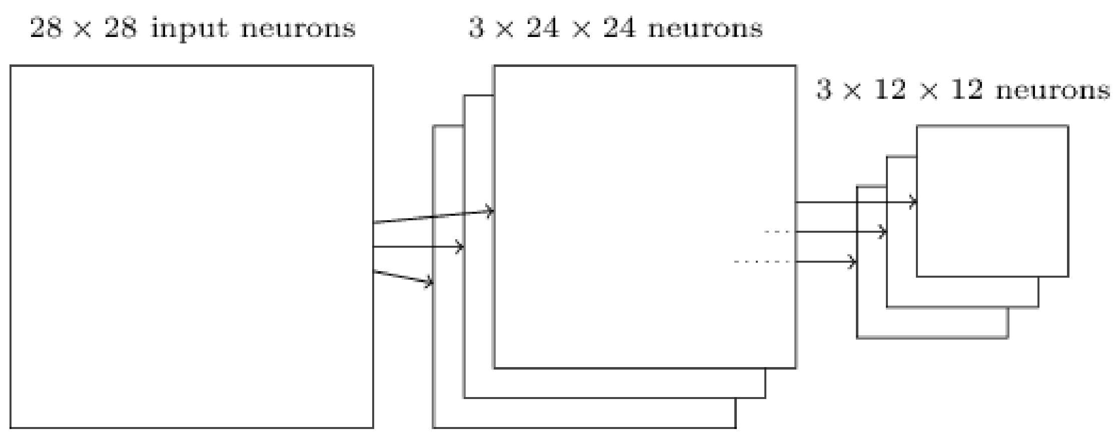
If we have 20 feature maps that's a total of 20×26=520 parameters defining the convolutional layer. By comparison, suppose we had a fully connected first layer, with 784=28×28 input neurons, 30 hidden neurons, 23,550 parameters.

Deep Learning
Digit Recognizer and Convolutional NN

http://neuralnetworksanddeeplearning.com/chap6.html

Pooling layers: **the pooling layers do is simplify the information in the output from the convolutional layer, one common procedure for pooling is known as *max-pooling*, in the 2x2 region input.**

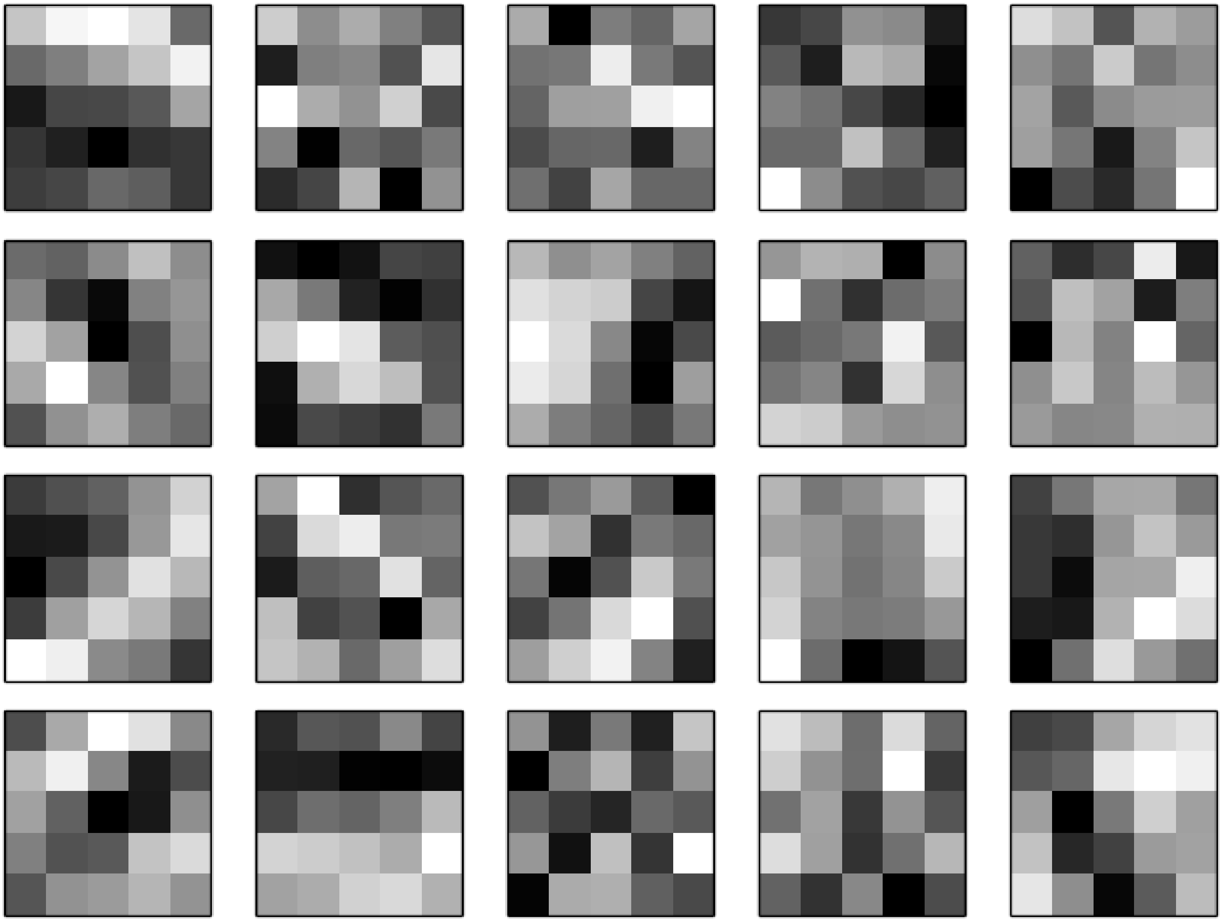




Deep Learning

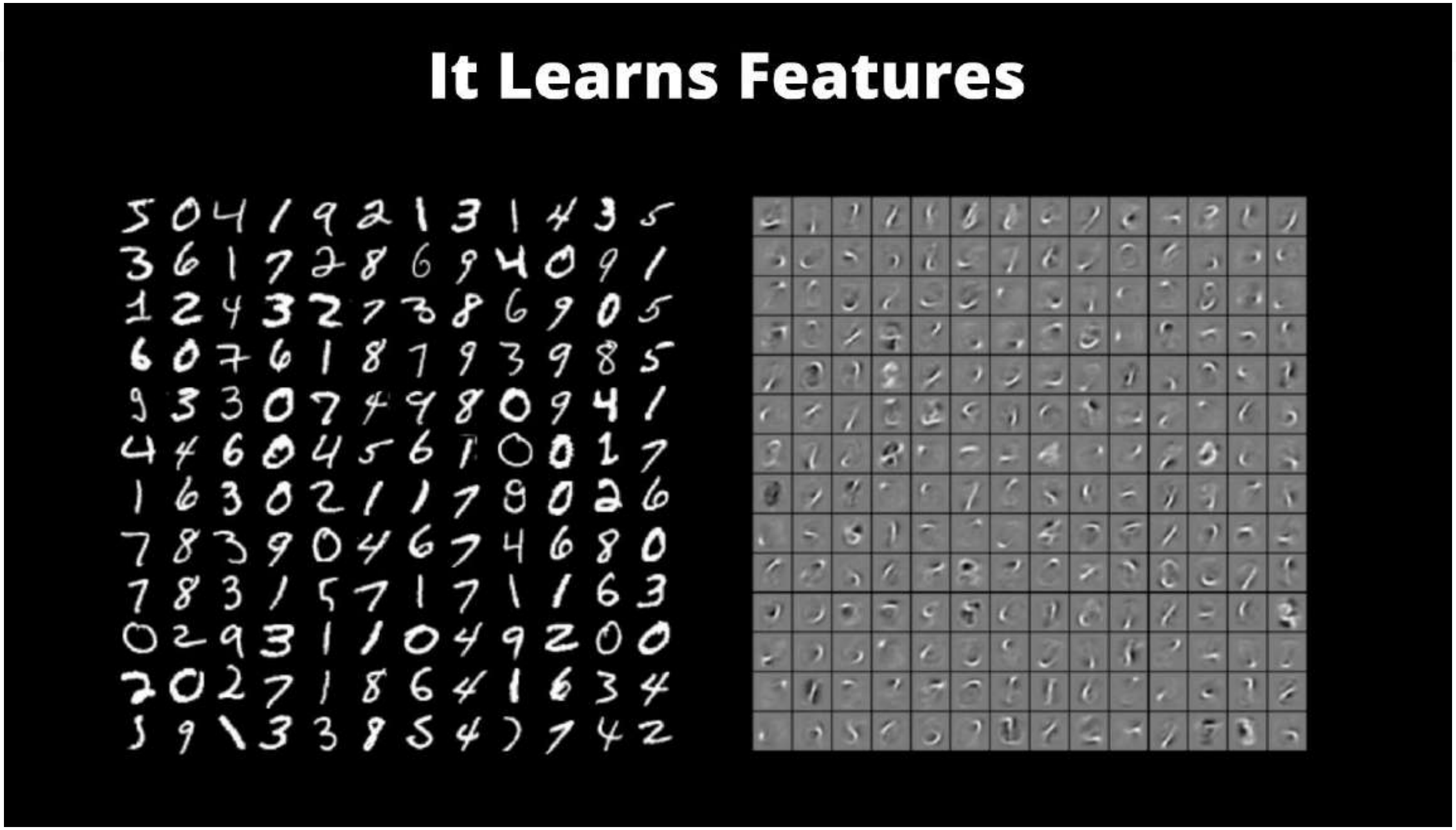
Digit Recognizer and Convolutional NN

http://neuralnetworksanddeeplearning.com/chap6.html



The 20 images correspond to 20 different feature maps

Deep Learning
Digit Recognizer and Convolutional NN



On the left, the raw input digits. On the right, graphical representations of the learned features. In essence, the network learns to “see” lines and loops.

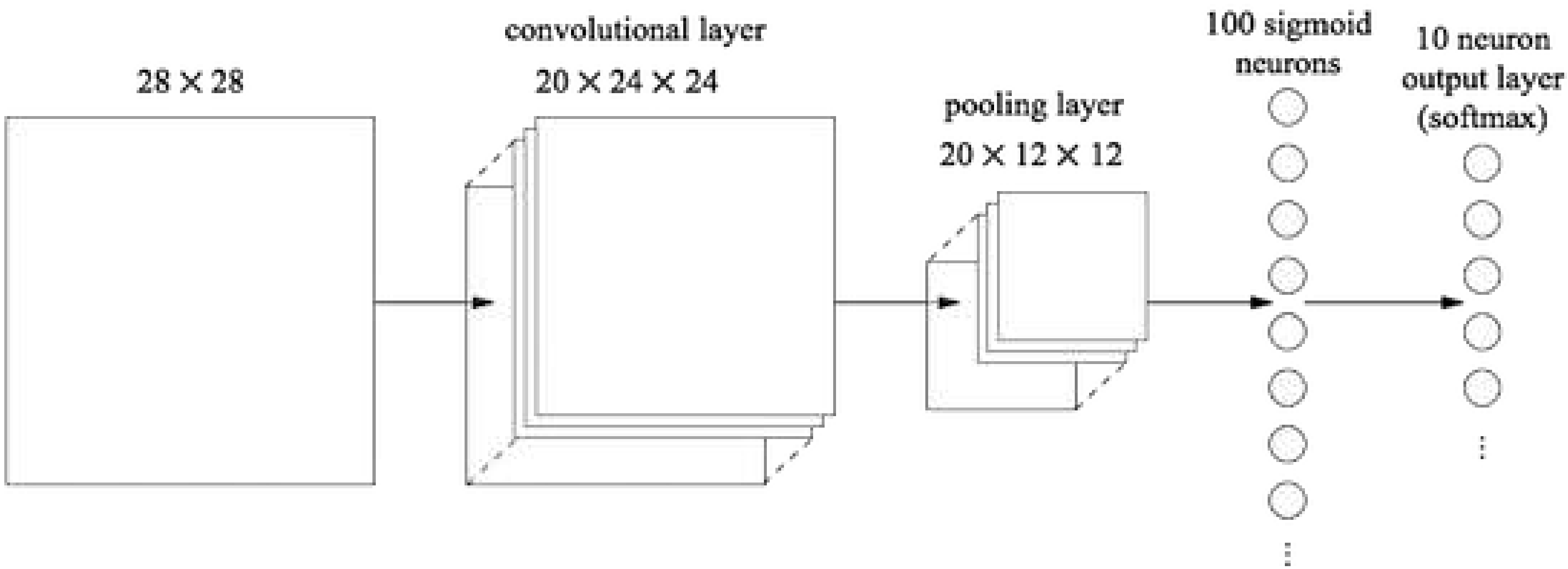
Credits: https://www.datarobot.com/blog/a-primer-on-deep-learning/

Deep Learning

Digit Recognizer and Convolutional NN

http://neuralnetworksanddeeplearning.com/chap6.html

Real experiment for DIGIT:

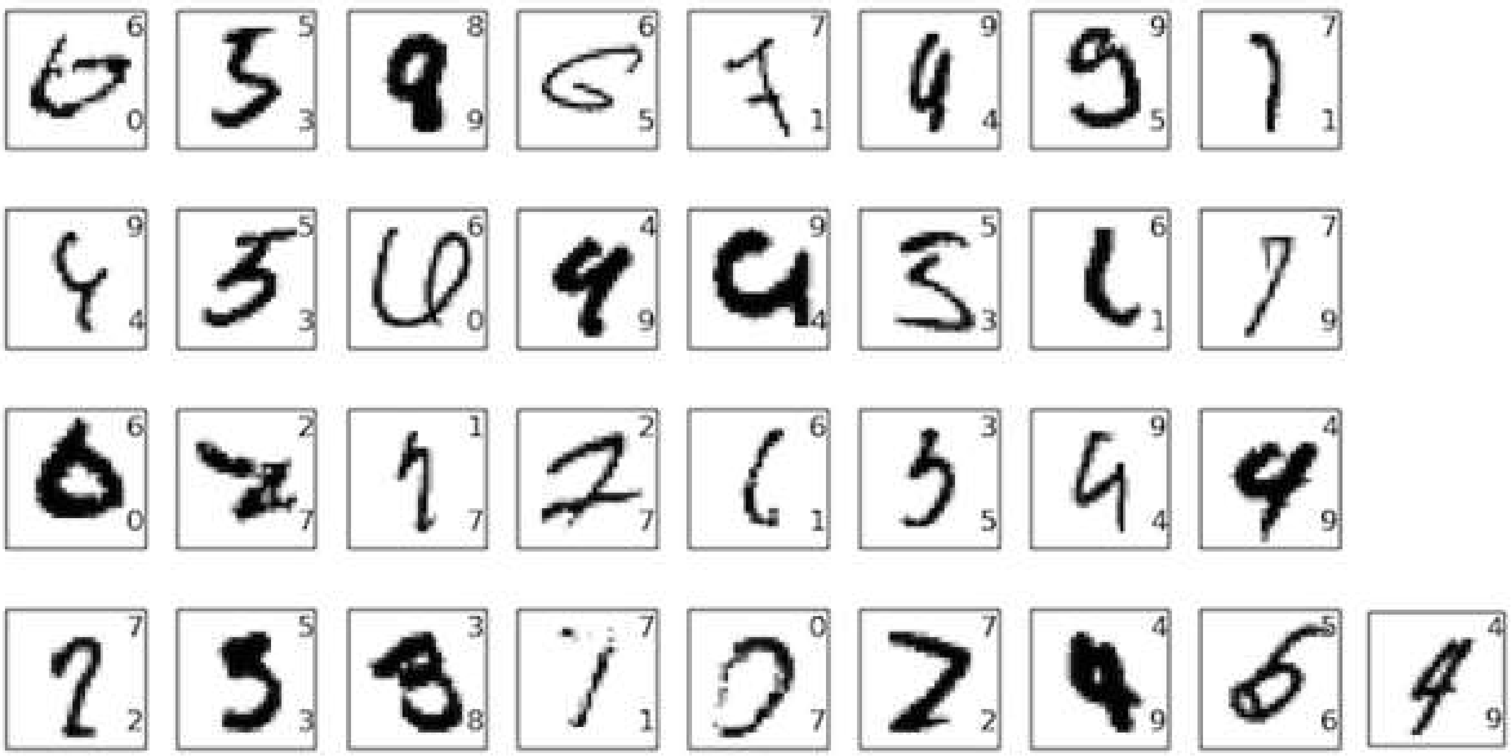


Different results, and preprocessing are analyzed in the chapter. Expanding the training data, to displace each training image by a single pixel, either up one pixel, down one pixel, left one pixel, or right one pixel.

Deep Learning
Digit Recognizer and Convolutional NN

http://neuralnetworksanddeeplearning.com/chap6.html

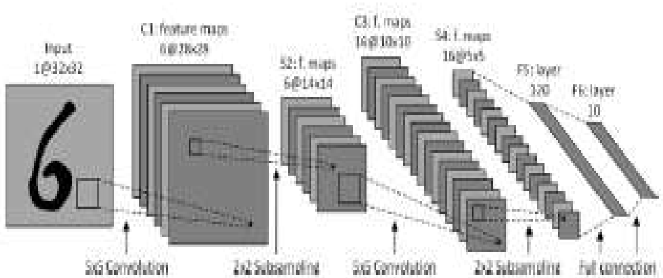
Final experiment for DIGIT (ensemble with different configurations): 99.67 percent accuracy, 33 of the 10,000 test images. **The label in the top right is the correct classification, according to the MNIST data, while in the bottom right is the label output by our ensemble of nets:**



Deep Learning: MNIST data

A snapshot of image pre-processing for convolutional neural networks: case study of MNIST
Siham Tabik, Daniel Peralta, Andrés Herrera-Poyatos, Francisco Herrera
International Journal of Computational Intelligence Systems, Vol. 10 (2017) 555–568
99.72 accuracy

Handwritting recognition. Assign a digit from 0 to 9.



The 28 handwritten digits misclassified by ensemble-5 of Network3
The digit between () represents the correct class.
The 13 digits labeled with asterisks are also misclassified by DropConnect

http://yann.lecun.com/exdb/mnist/

yann.lecun.com/exdb/mnist/

Aplicaciones Bookmarks Google Nueva pestaña Guide 2 Research Information Fusion -

THE MNIST DATABASE

of handwritten digits

Yann LeCun, Courant Institute, NYU
Corinna Cortes, Google Labs, New York
Christopher J.C. Burges, Microsoft Research, Redmond

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

Four files are available on this site:

```
train-images-idx3-ubyte.gz:  training set images (9912422 bytes)
train-labels-idx1-ubyte.gz:  training set labels (28881 bytes)
t10k-images-idx3-ubyte.gz:   test set images (1648877 bytes)
t10k-labels-idx1-ubyte.gz:   test set labels (4542 bytes)
```

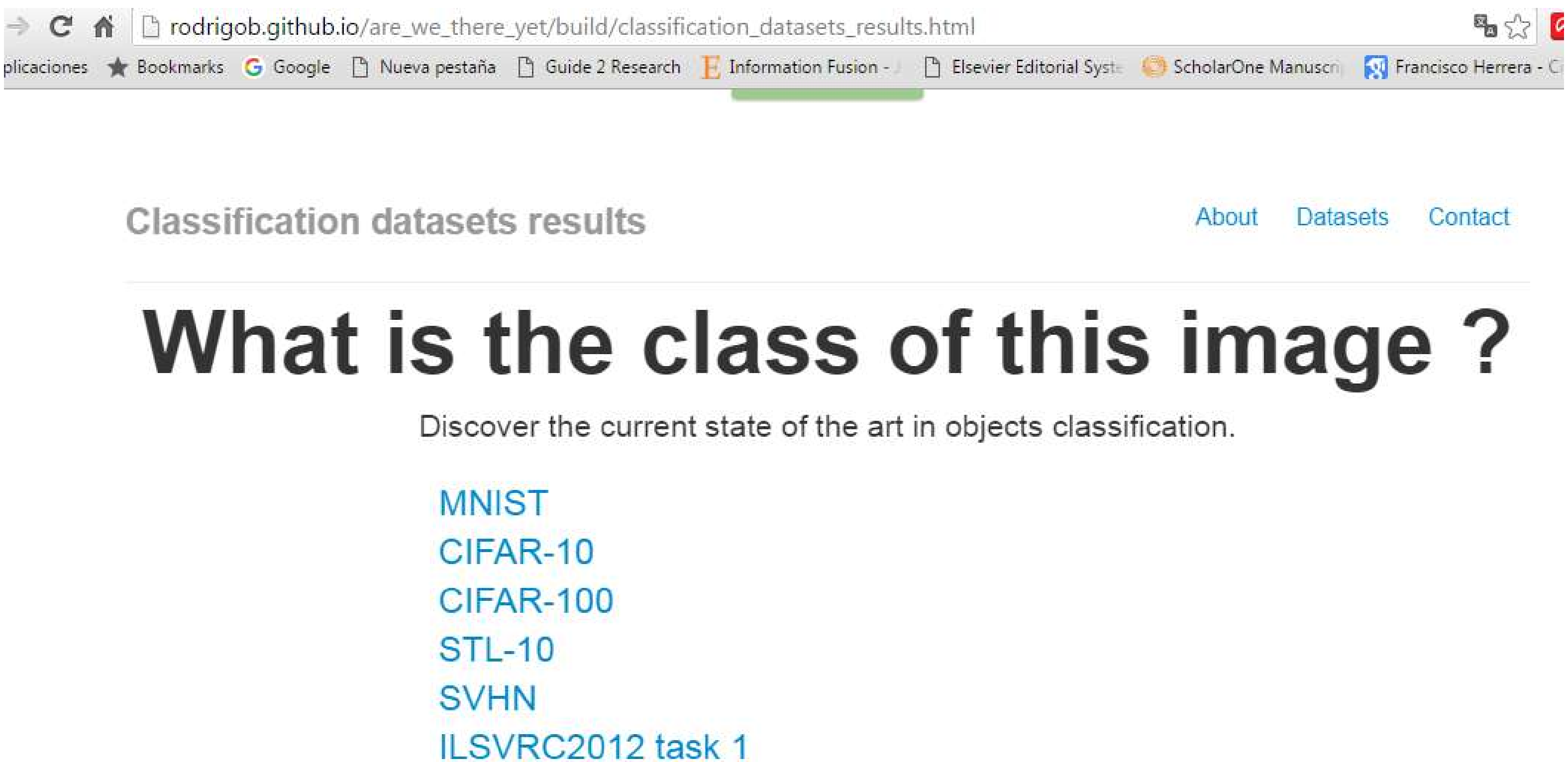
Deep Learning
Digit Recognizer

http://yann.lecun.com/exdb/mnist/

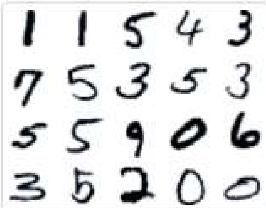
CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
K-NN with non-linear deformation (IDM)	shiftable edges	0.54	Keysers et al. IEEE PAMI 2007
K-NN with non-linear deformation (P2DHMDM)	shiftable edges	0.52	Keysers et al. IEEE PAMI 2007
Virtual SVM deg-9 poly [distortions]	none	0.8	LeCun et al. 1998
Virtual SVM, deg-9 poly, 1-pixel jittered	none	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 1-pixel jittered	deskewing	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 2-pixel jittered	deskewing	0.56	DeCoste and Scholkopf, MLJ 2002
large/deep conv. net, 1-20-40-60-80-100-120-120-10 [elastic distortions]	none	0.35	Ciresan et al. IJCAI 2011
committee of 7 conv. net, 1-20-P-40-P-150-10 [elastic distortions]	width normalization	0.27 +-0.02	Ciresan et al. ICDAR 2011
committee of 35 conv. net, 1-20-P-40-P-150-10 [elastic distortions]	width normalization	0.23	Ciresan et al. CVPR 2012

Deep Learning
Digit Recognizer

http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html



Deep Learning
Digit Recognizer



MNIST 50 results collected

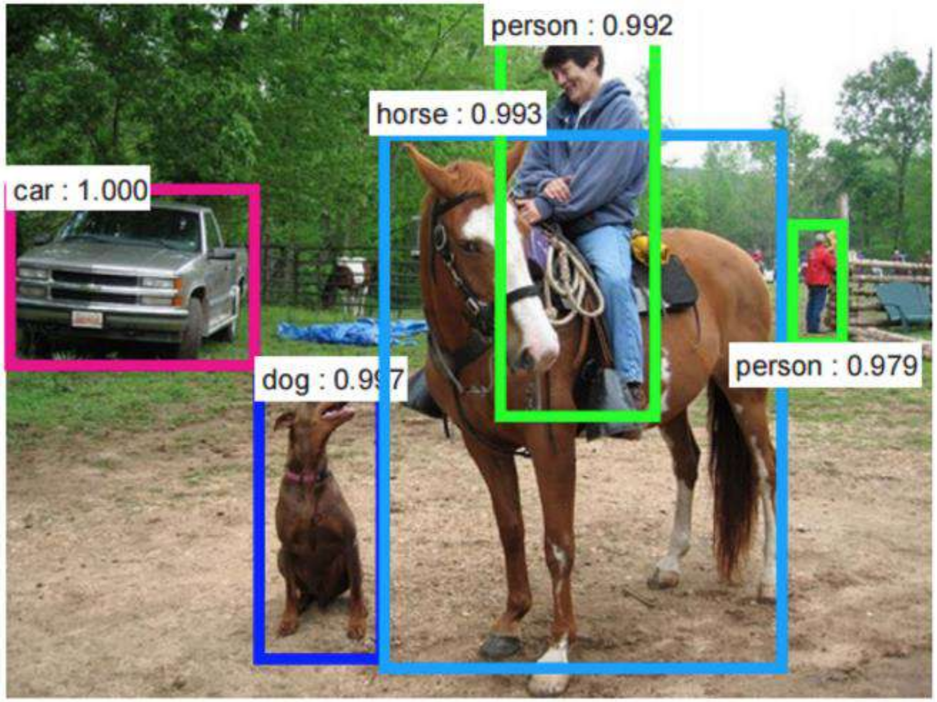
Units: error %

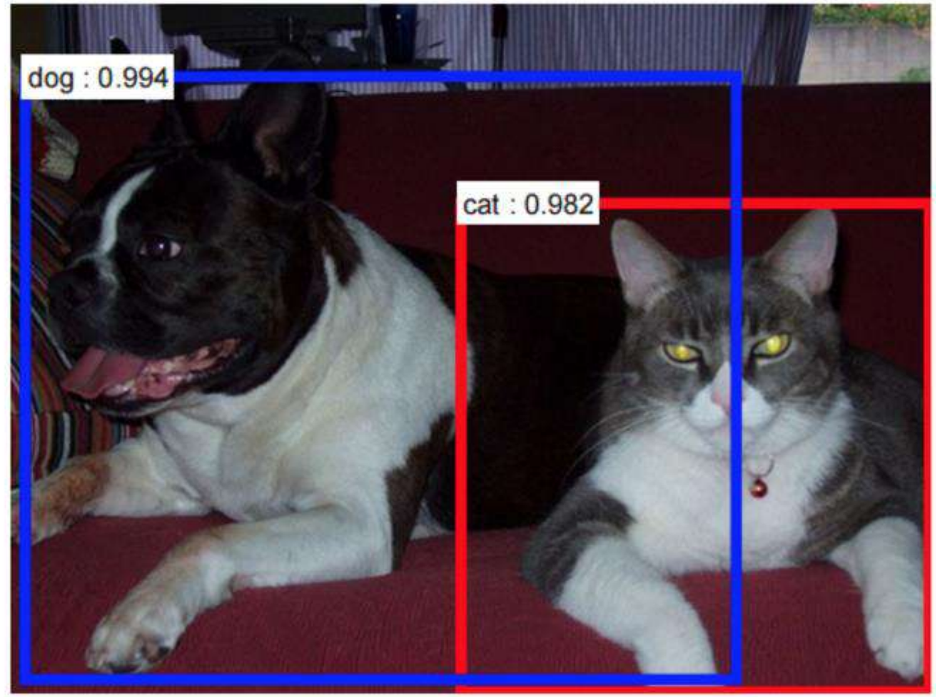
> Classify handwriten digits. Some additional results are available on the original dataset page.

Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification	CVPR 2012	
0.23%	APAC: Augmented PAttern Classification with Neural Networks	arXiv 2015	
0.24%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
0.29%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
0.31%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	
0.31%	On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units	arXiv 2015	
0.32%	Fractional Max-Pooling	arXiv 2015	Details
0.33%	Competitive Multi-scale Convolution	arXiv 2015	

Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html





In Figure 1 above, a ConvNet is able to recognize scenes and the system is able to suggest relevant tags such as 'bridge', 'railway' and 'tennis' while Figure 2 shows an example of ConvNets being used for recognizing everyday objects, humans and animals. Lately, ConvNets have been effective in several Natural Language Processing tasks (such as sentence classification) as well.

Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html

Convolution example: Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

Then, the Convolution of the 5 x 5 image and the 3 x 3 matrix (used as filter)can be computed as shown in the animation in Figure 5 below:

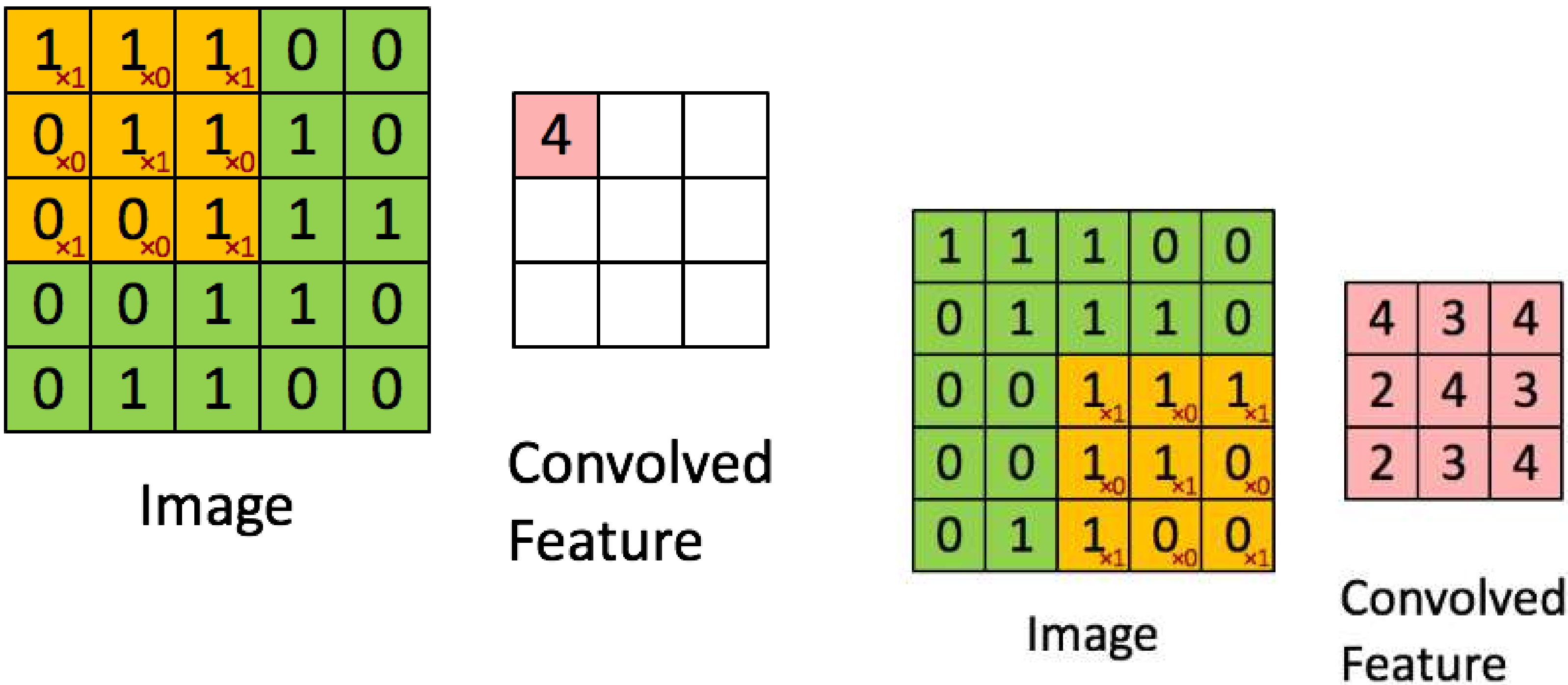
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Figure 5: The Convolution operation. The output matrix is called Convolved Feature or Feature Map. Source [7]

Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html



http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

Figure 5: The Convolution operation. The output matrix is called Convolved Feature or Feature Map. Source [7]

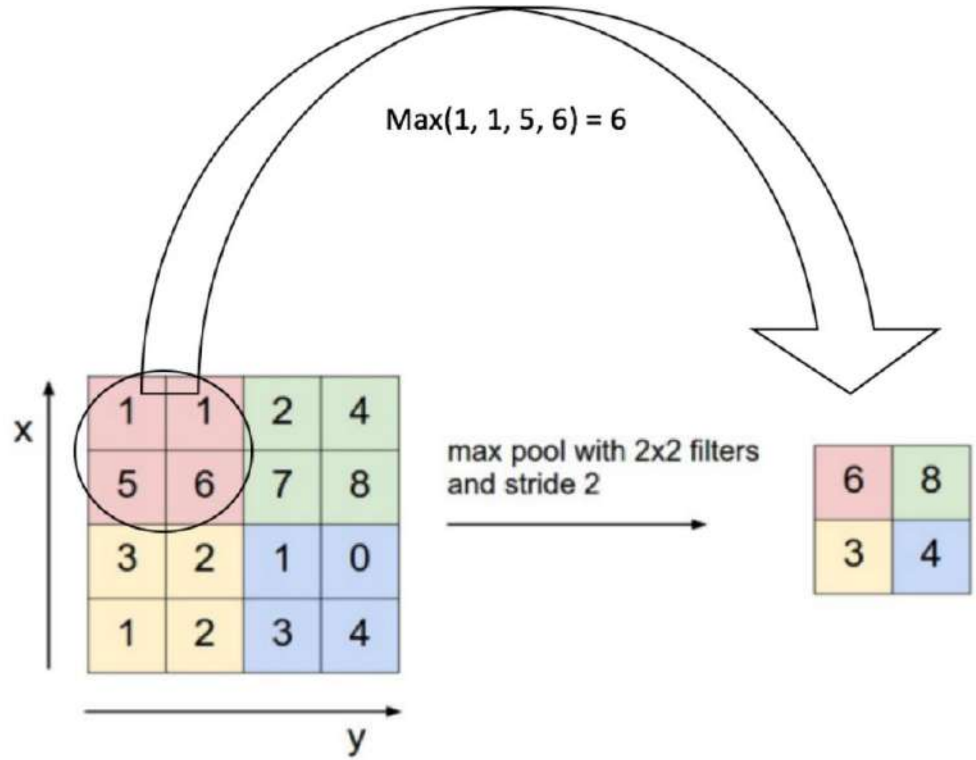
Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html

In CNN terminology, the 3×3 matrix is called a 'filter' or 'kernel' or 'feature detector' and the matrix formed by sliding the filter over the image and computing the dot product is called the 'Convolved Feature' or 'Activation Map' or the 'Feature Map'. It is important to note that filters acts as feature detectors from the original input image.

Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html

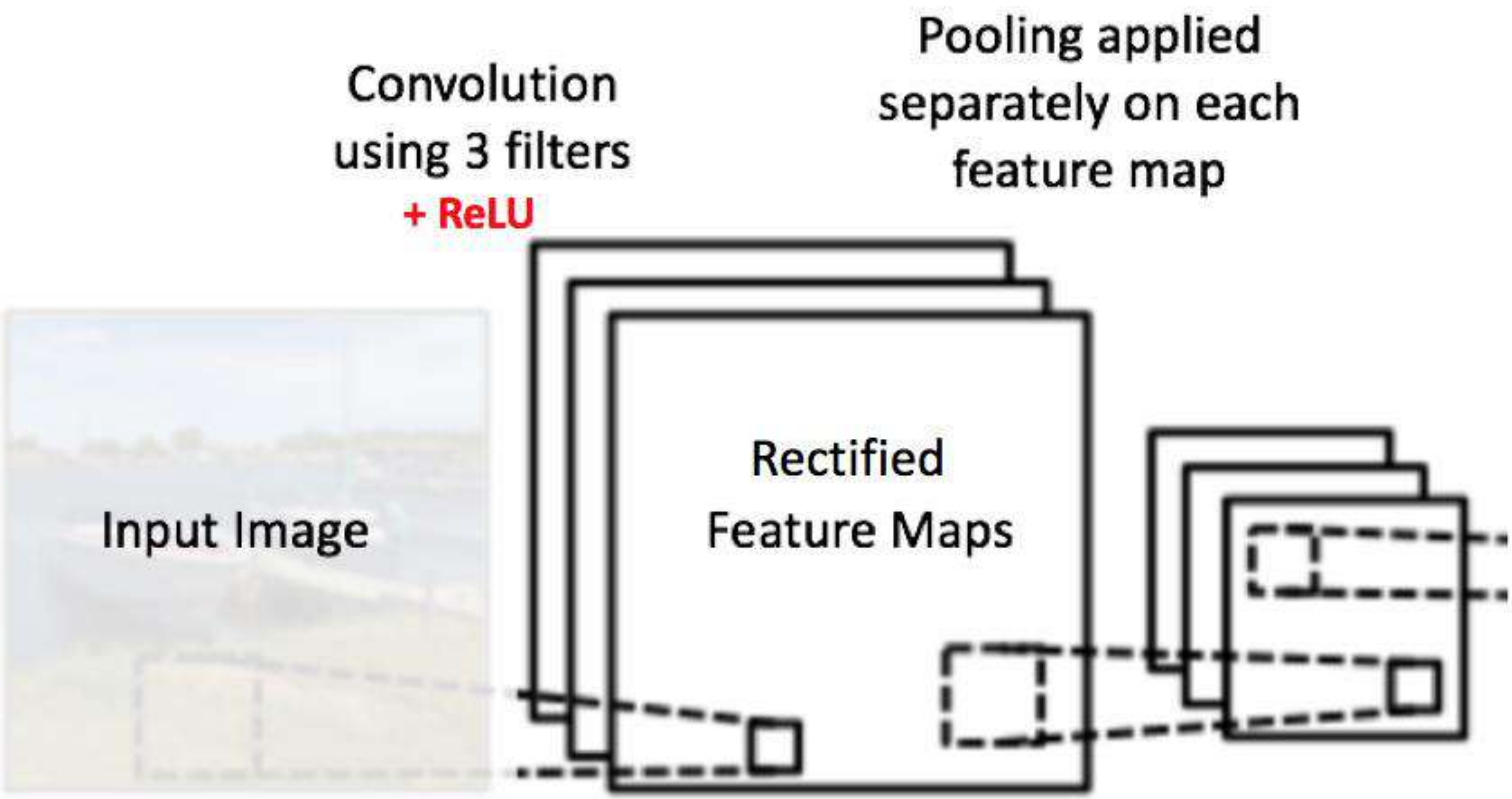


Rectified Feature Map

The Pooling Step Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc.

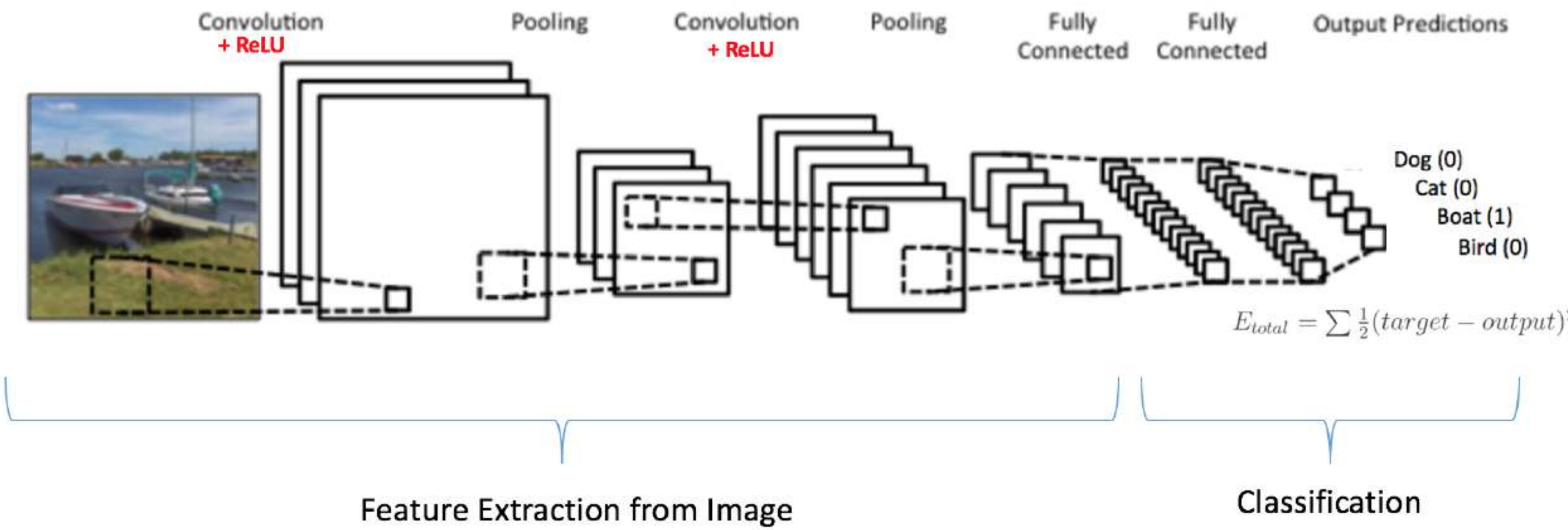
Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html



Deep Learning

http://www.kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html



A CNN automatically learns the values of its filters based on the task you want to perform.

Deep Learning

Librerías de Deep Learning

Caffe

http://caffe.berkeleyvision.org/

Caffe is a deep learning framework made with expression, speed, and modularity in mind.

It is developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors.

Google's DeepDream is based on Caffe Framework. This framework is a BSD-licensed C++ library with Python Interface.

SparkNet

https://github.com/amplab/SparkNet

Librerías de Deep Learning

TensorFlow

https://www.tensorflow.org/

TensorFlow™ is an open source software library for numerical computation using data flow graphs.

The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

Librerías de Deep Learning

Tensor Flow

https://www.tensorflow.org/

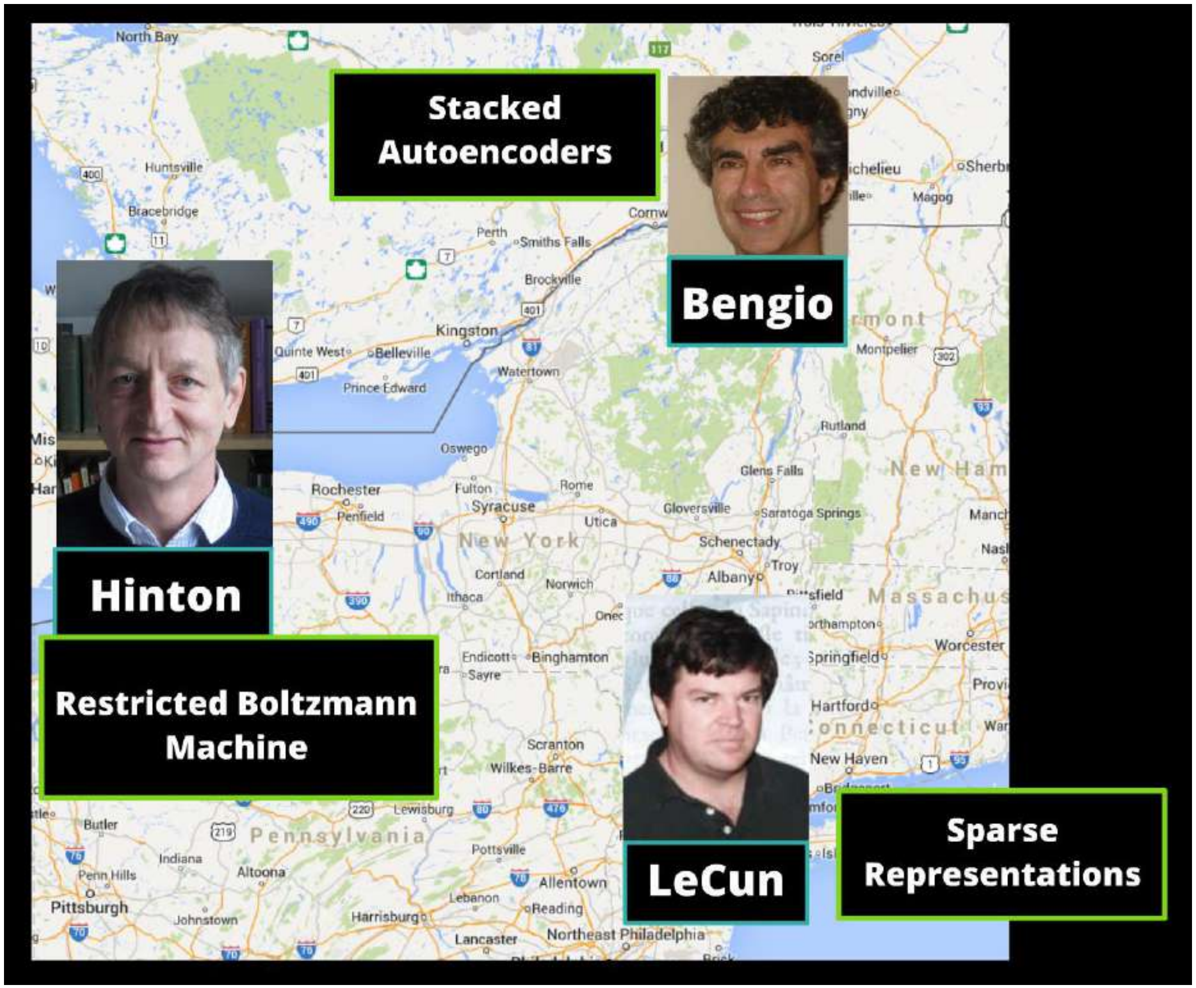
Scikit Flow: Easy Deep Learning with TensorFlow and Scikit-learn

https://github.com/tensorflow/skflow

Deep Neural Network
Convolutional NN

http://www.kdnuggets.com/2016/02/scikit-flow-easy-deep-learning-tensorflow-scikit-learn.html

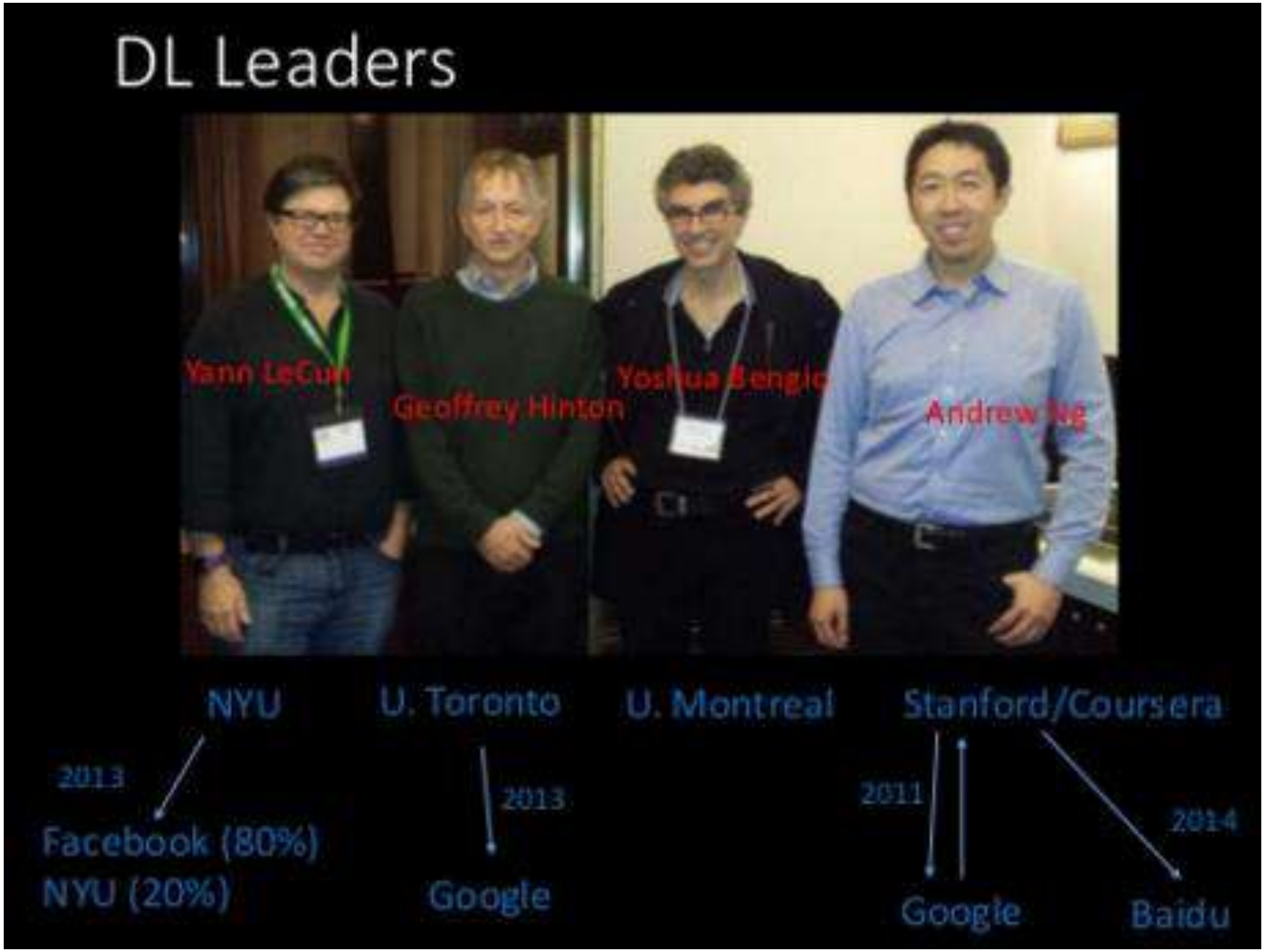
Deep Learning
Relevant researchers



The Fathers of Deep Learning

Credits: https://www.datarobot.com/blog/a-primer-on-deep-learning/

Deep Learning
Relevant researchers



Credits: http://www.slideshare.net/david.kh/promises-of-deep-learning

Deep Learning

Final Comments: Overview sobre las estructuras de representación para aprendizaje y deep learning

En este artículo de review, Bengio y coautores hacen una revisión muy interesante sobre la representación para el aprendizaje de características, fundamental para entender deep learning, analizando el estado del arte y las perspectivas futuras.

1798 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 35, NO. 8, AUGUST 2013

Representation Learning: A Review and New Perspectives

Yoshua Bengio, Aaron Courville, and Pascal Vincent

Abstract—The success of machine learning algorithms generally depends on data representation, and we hypothesize that this is because different representations can entangle and hide more or less the different explanatory factors of variation behind the data. Although specific domain knowledge can be used to help design representations, learning with generic priors can also be used, and the quest for AI is motivating the design of more powerful representation-learning algorithms implementing such priors. This paper reviews recent work in the area of unsupervised feature learning and deep learning, covering advances in probabilistic models, autoencoders, manifold learning, and deep networks. This motivates longer term unanswered questions about the appropriate objectives for learning good representations, for computing representations (i.e., inference), and the geometrical connections between representation learning, density estimation, and manifold learning.

Index Terms—Deep learning, representation learning, feature learning, unsupervised learning, Boltzmann machine, autoencoder, neural nets

Deep Learning

Readings: Recent Overview

nature International weekly journal of science

Menu ▸ Advance

archive ▸ volume 521 ▸ issue 7553 ▸ insights ▸ reviews ▸ article

NATURE | INSIGHT | REVIEW

Deep learning

Yann LeCun, Yoshua Bengio & Geoffrey Hinton

Affiliations | Corresponding author

Nature **521**, 436–444 (28 May 2015) | doi:10.1038/nature14539

Received 25 February 2015 | Accepted 01 May 2015 | Published online 27 May 2015

http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html

Deep Learning

Readings: Recent Overview

Deep learning, Yann LeCun, Yoshua Bengio & Geoffrey Hinton

Abstract

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html

Inteligencia de Negocio

TEMA 7. Modelos Avanzados de Minería de Datos

1. Clases no balanceadas/equilibradas

2. Características intrínsecas de los datos en clasificación

3. Clasificación no Estándar: Más allá del aprendizaje clásico

4. Detección de anomalías

5. Deep Learning

6. Análisis de Sentimientos

ROADMAP

1. Introduction
2. The Sentiment Analysis Problem
3. The Sentiment Analysis Process
4. Validivia Master’s Thesis

1. INTRODUCTION

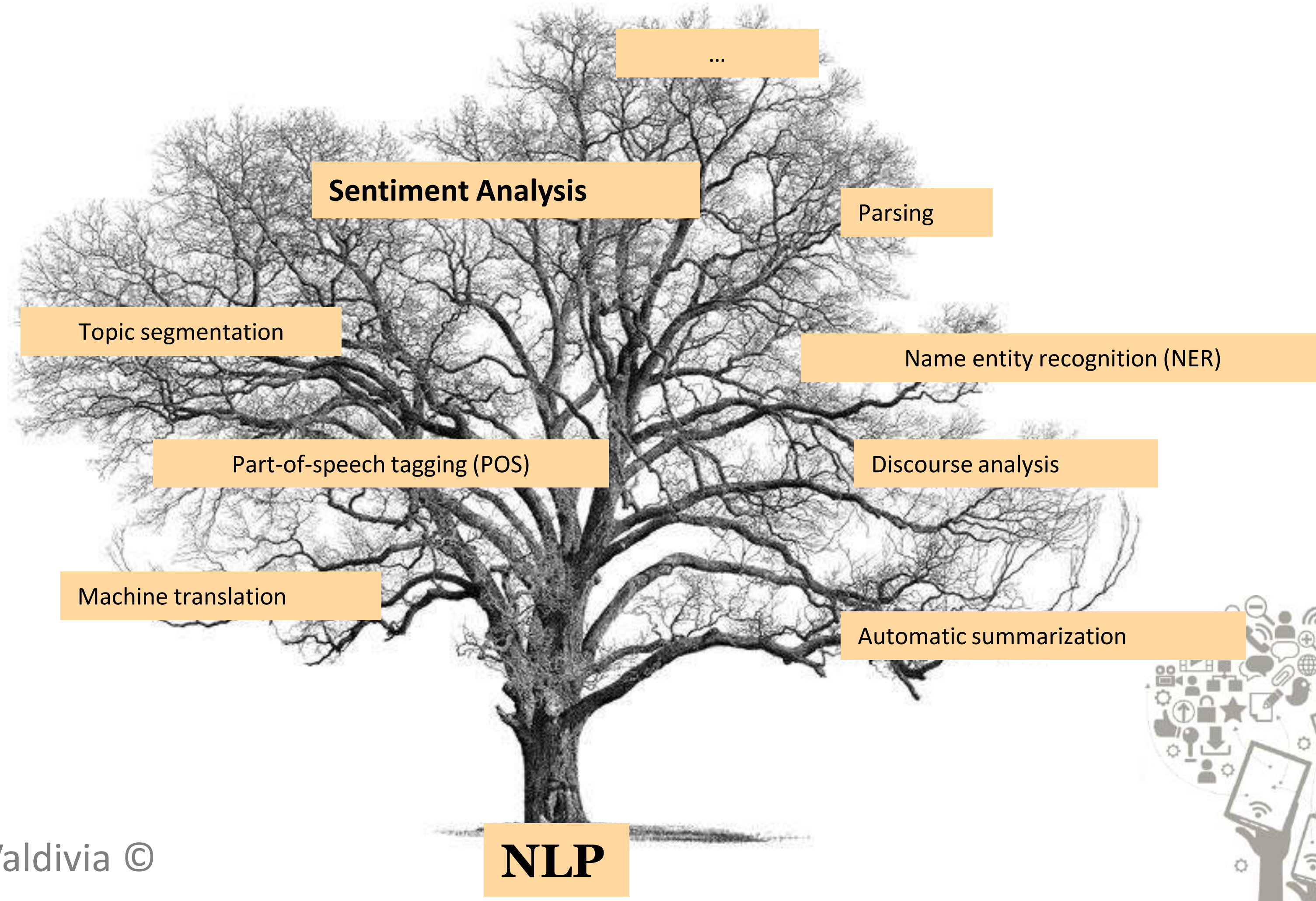
What is SA?

Sentiment Analysis (SA) is the field of knowledge that analyses people's opinions, reviews or thoughts about products, companies or experiences identifying its sentiment.

Also referred as ***Opinion Mining***.

1. INTRODUCTION

Where it comes from?



1. INTRODUCTION

Why is SA being popular?

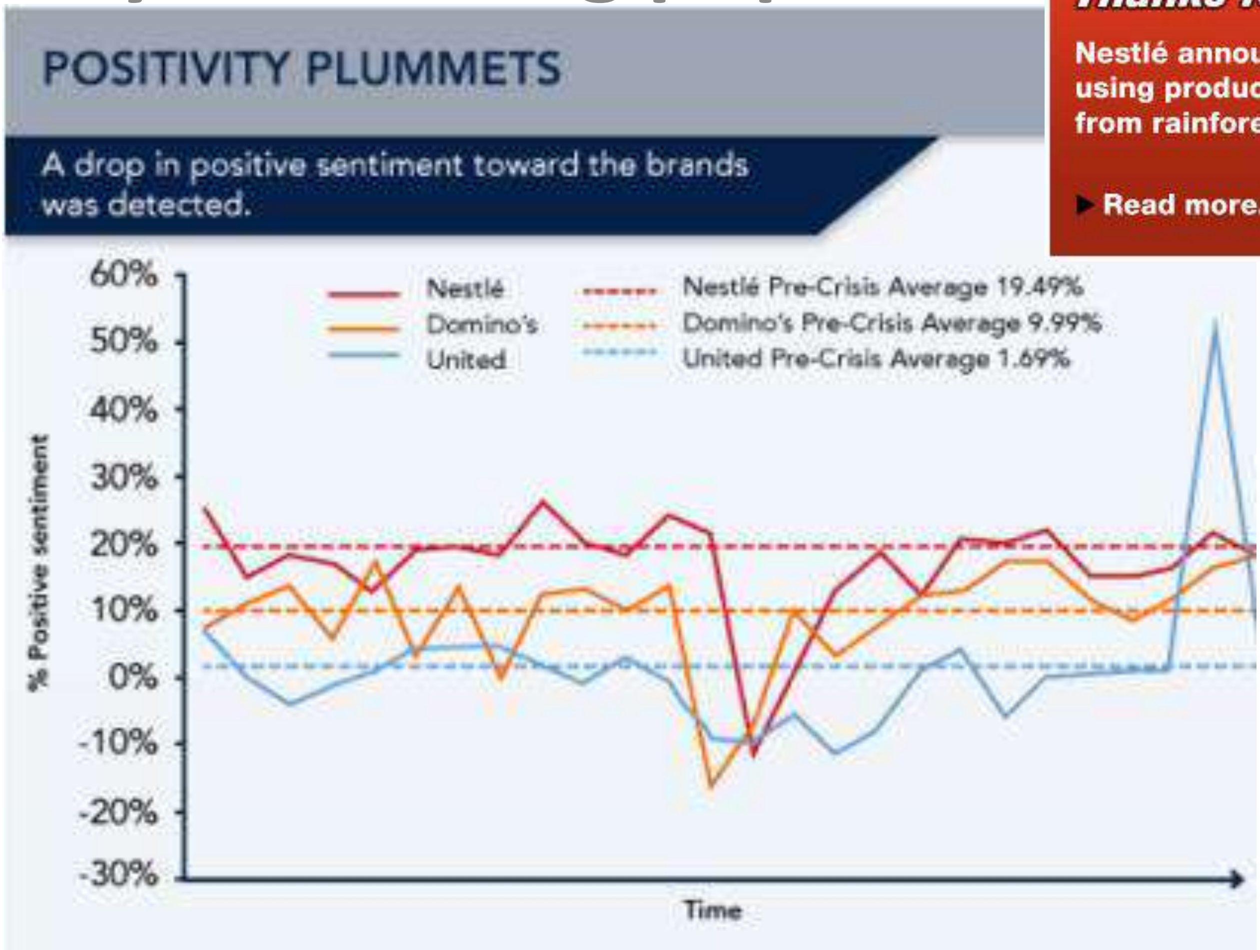
1. INTRODUCTION

Customer's satisfaction

http://www.slideshare.net/robin_allfamous/sentiment-analysis-and-applications-in-the-news-and-media-industry

1. INTRODUCTION

Why is SA being popular?



Social media sentiment is the #nofilter voice of the people.

Ana Valdivia ©

http://www.slideshare.net/robin_allfamous/sentiment-analysis-and-applications-in-the-news-and-media-industry

ROADMAP

1. Introduction
2. The Sentiment Analysis Problem
3. The Sentiment Analysis Process
4. Valdivia Master's Thesis

2. THE SENTIMENT ANALYSIS PROBLEM

What's an opinion?

Ana Valdivia ©

What’s an opinion?

“If we cannot structure a problem, we probably do not understand the problem” .
B. Liu

2. THE SENTIMENT ANALYSIS PROBLEM

What's an opinion?

Cambridge Dictionary

opinion

noun · UK /əˈpɪn.jən/ US /əˈpɪn.jən/

- **B1** [C] **a thought or belief about something or someone:**
- **B2** [U] **the thoughts or beliefs that a group of people have:**

"If we cannot structure a problem, we probably do not understand the problem" .
B. Liu

2. THE SENTIMENT ANALYSIS PROBLEM

What's an opinion?

Cambridge Dictionary

opinion

noun · UK /əˈpɪn.jən/ US /əˈpɪn.jən/

B1 [C] **a thought or belief about something or someone:**

B2 [U] **the thoughts or beliefs that a group of people have:**

"If we cannot structure a problem, we probably do not understand the problem".
B. Liu.

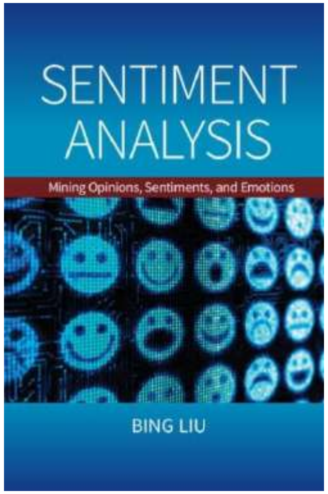
Liu's proposal:

- An *opinion* is a quintuple

  (*entity*, *aspect*, *sentiment*, *holder*, *time*)

  where
  - ***entity***: target entity (or object).
  - ***aspect***: aspect (or feature) of the entity.
  - ***sentiment***: +, -, or neu, a rating, or an emotion.
  - ***holder***: opinion holder.
  - ***time***: time when the opinion was expressed.

BOOK REMARK
B. Liu, *Sentiment analysis and opinion mining*



2. THE SENTIMENT ANALYSIS PROBLEM

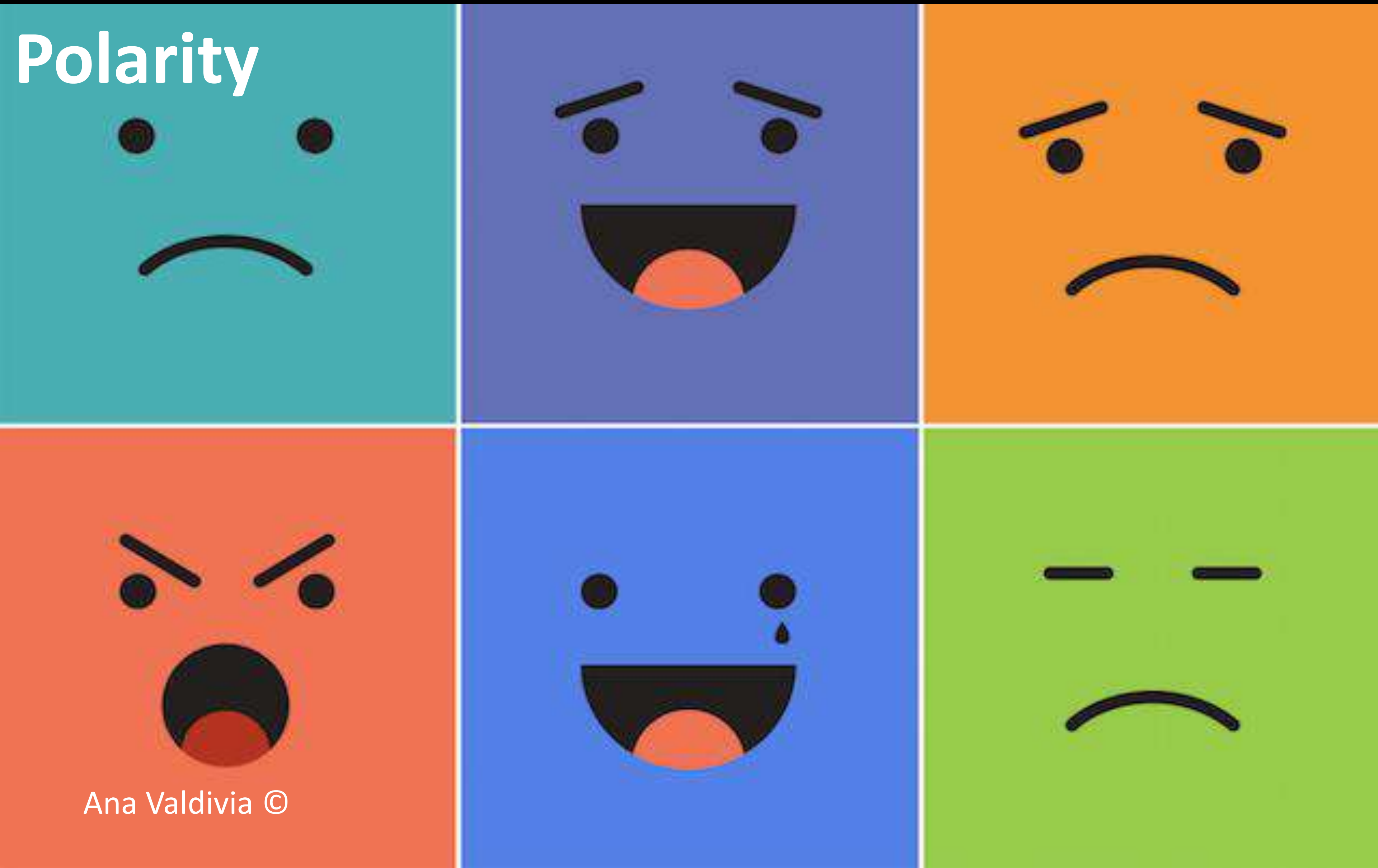
Polarity

2. THE SENTIMENT ANALYSIS PROBLEM

Polarity

2. THE SENTIMENT ANALYSIS PROBLEM

Polarity



2. THE SENTIMENT ANALYSIS PROBLEM

One example is worth a thousand words…

2. THE SENTIMENT ANALYSIS PROBLEM

One example is worth a thousand words...

Liu's proposal:

Concept	Value
entity	abaco te
opinion holder	beth r
date	2015-09-23
(aspect, sentiment)	(view, positive)
(aspect, sentiment)	(fruit juice, positive)
(aspect, sentiment)	(fruit juice, positive)
(aspect, sentiment)	(desert, positive)

"We were very tired after a loong walk. We stopped her for a rest, the first nice thing here, is the view, and the fruit juices were excellent. We felt much better after drunk it. Also the desert were very good. Thank you."

- $(e_1, a_{11}, s_{1111}, h_1, t_1) = (\textit{abacto te, view, positive, beth r, 2015-09-23})$
- $(e_1, a_{12}, s_{1211}, h_1, t_1) = (\textit{abacto te, fruit juice, positive, beth r, 2015-09-23})$
- $(e_1, a_{12}, s_{1211}, h_1, t_1) = (\textit{abacto te, fruit juice, positive, beth r, 2015-09-23})$
- $(e_1, a_{13}, s_{1311}, h_1, t_1) = (\textit{abacto te, desert, positive, beth r, 2015-09-23})$

2. THE SENTIMENT ANALYSIS PROBLEM

Different analytic levels

- **Document level**

- **Sentence level**

- **Aspect or entity level**

2. THE SENTIMENT ANALYSIS PROBLEM

Main concerns

- **- Different types of opinions**
  Direct/indirect, comparative, explicit/implicit, …
- **- Deal with text mining**
  Grammar mistakes, emoticons, …
- **- Irony and sarcasm**
- **- Fake or spam opinions**

ROADMAP

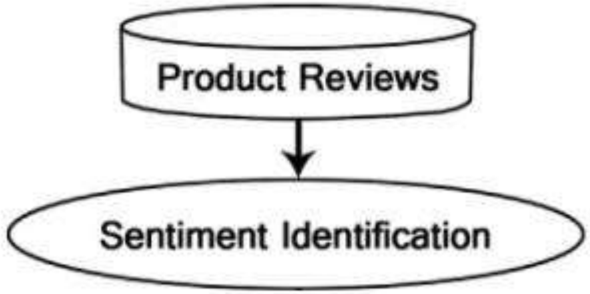
1. Introduction
2. The Sentiment Analysis Problem
3. The Sentiment Analysis Process
4. Valdivia Master's Thesis

3. THE SENTIMENT ANALYSIS PROCESS

Step by step

3. THE SENTIMENT ANALYSIS PROCESS

Step by step

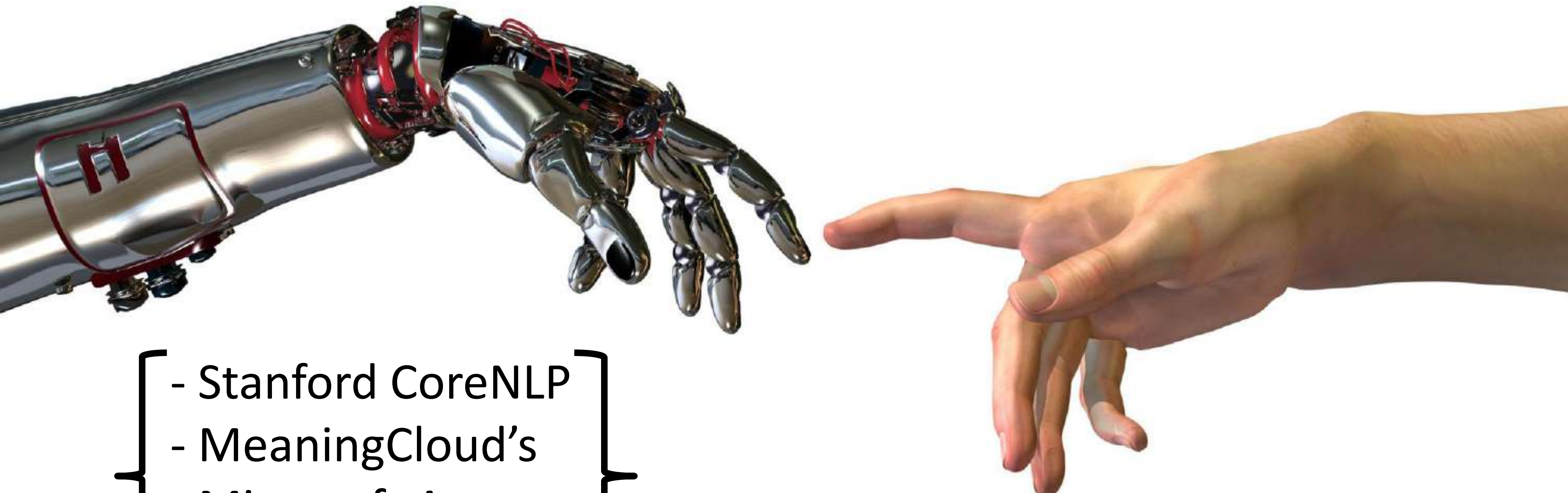


3. THE SENTIMENT ANALYSIS PROCESS

Sentiment identification

Sentiment extraction algorithms

Expert or user

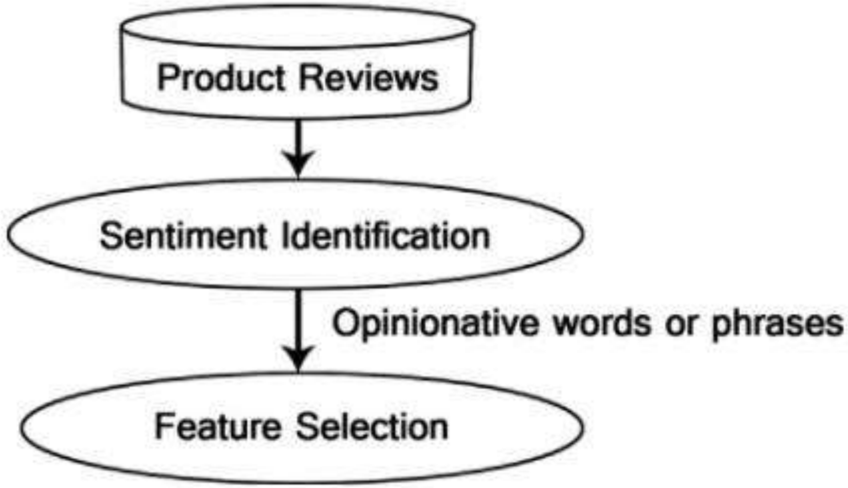


- Stanford CoreNLP
- MeaningCloud's
- Microsoft Azure
- …

Ana Valdivia ©

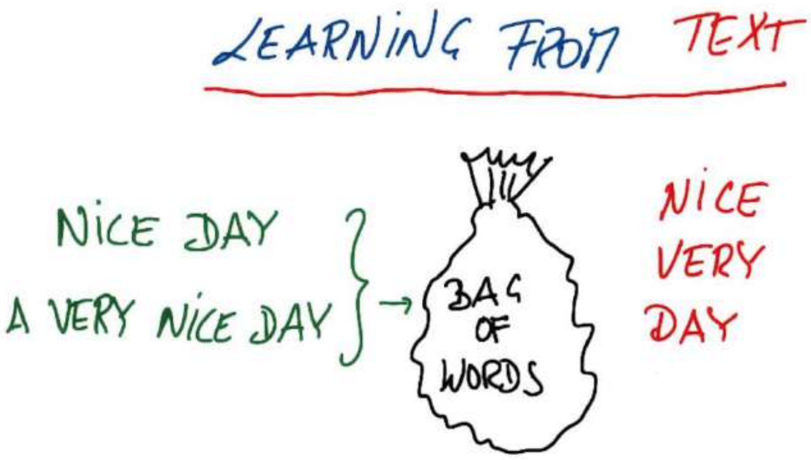
3. THE SENTIMENT ANALYSIS PROCESS

Step by step



Feature Selection

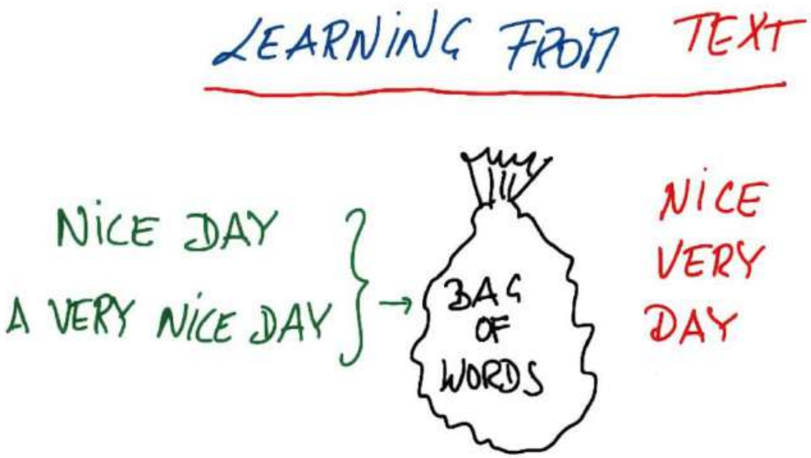
Bag of Words



3. THE SENTIMENT ANALYSIS PROCESS

Feature Selection

Bag of Words



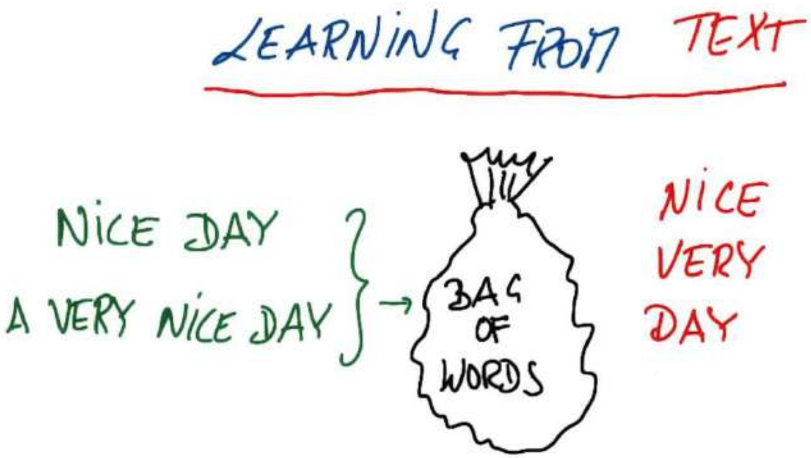
Term-Document Matrix

$$
\begin{array}{c}
 \\ D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_n
\end{array}
\begin{pmatrix}
T_1 & T_2 & \dots & T_t \\
w_{11} & w_{21} & \dots & w_{t1} \\
w_{12} & w_{22} & \dots & w_{t2} \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots \\
w_{1n} & w_{2n} & \dots & w_{tn}
\end{pmatrix}
$$

3. THE SENTIMENT ANALYSIS PROCESS

Feature Selection

Bag of Words



Term-Document Matrix

$$
\begin{array}{cc}
 & \begin{array}{cccc} T_1 & T_2 & \cdots & T_t \end{array} \\
\begin{array}{c} D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_n \end{array} &
\begin{pmatrix}
w_{11} & w_{21} & \cdots & w_{t1} \\
w_{12} & w_{22} & \cdots & w_{t2} \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots \\
w_{1n} & w_{2n} & \cdots & w_{tn}
\end{pmatrix}
\end{array}
$$

tf-idf

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

3. THE SENTIMENT ANALYSIS PROCESS

Feature Selection

Text Preprocessing

Parsing

Stemming

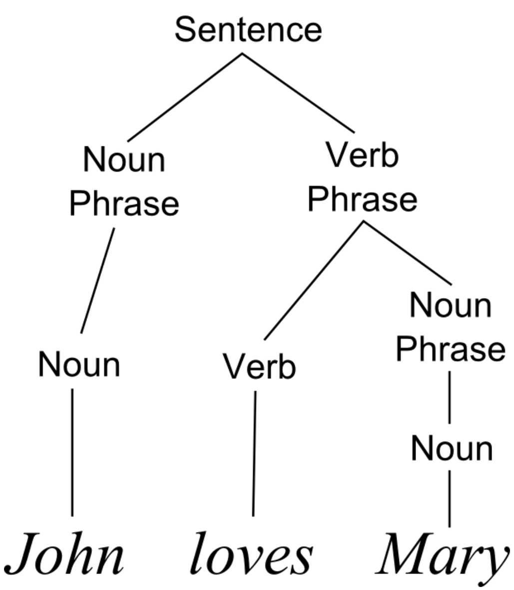
Remove STOP Words

3. THE SENTIMENT ANALYSIS PROCESS

Feature Selection

Text Preprocessing

Parsing



Stemming

{nightmare, nighttime, nocturnal, nightlife…} ⟶ *night*

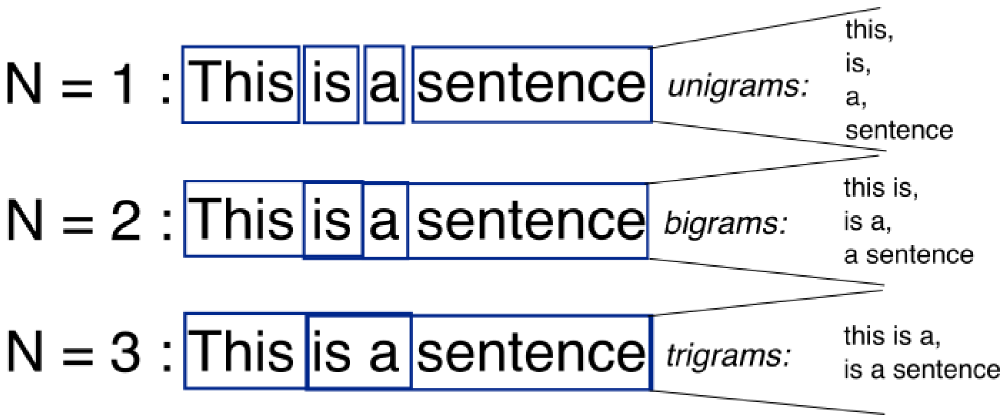
Remove STOP Words

Ana Valdivia ©

3. THE SENTIMENT ANALYSIS PROCESS

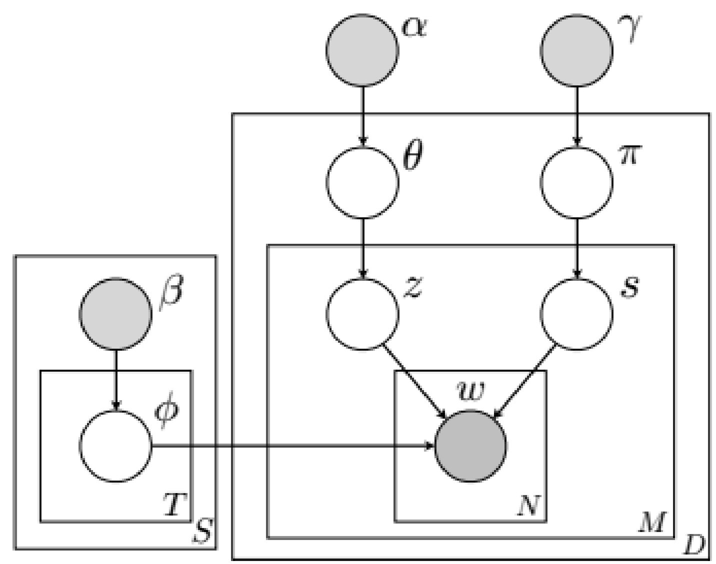
Feature Selection

N-grams



More sophisticated…

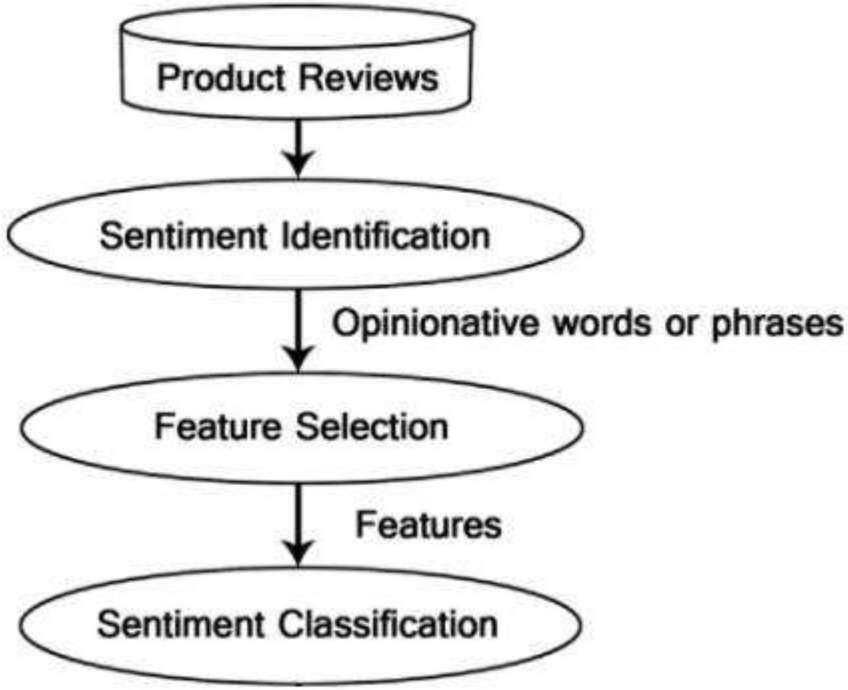
Aspect-Based Sentiment Analysis



ASUM

3. THE SENTIMENT ANALYSIS PROCESS

Step by step



Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." *Ain Shams Engineering Journal* 5.4 (2014): 1093-1113.

ROADMAP

1. Introduction
2. The Sentiment Analysis Problem
3. The Sentiment Analysis Process
4. Valdivia Master's Thesis

4. MY MASTER’S THESIS

Master in Data Science and Computer Engineering

Title: *Sentiment Analysis For Touristic Attractions: A Case Study On The Alhambra.*

Author: Ana Valdivia García

Advisor: Salvador García López

Department: Computer Science and Artificial Intelligence

University: University of Granada

Delivery date: 12/09/2016

Academic year: 2015/2016

4. MY MASTER'S THESIS

Objectives

1. **Study** correlation between **human** and **machine sentiment**
2. **Classify** opinions
3. **Dicover** interesting **patterns** in **negative** opinions

4. MY MASTER'S THESIS

tripadvisor®

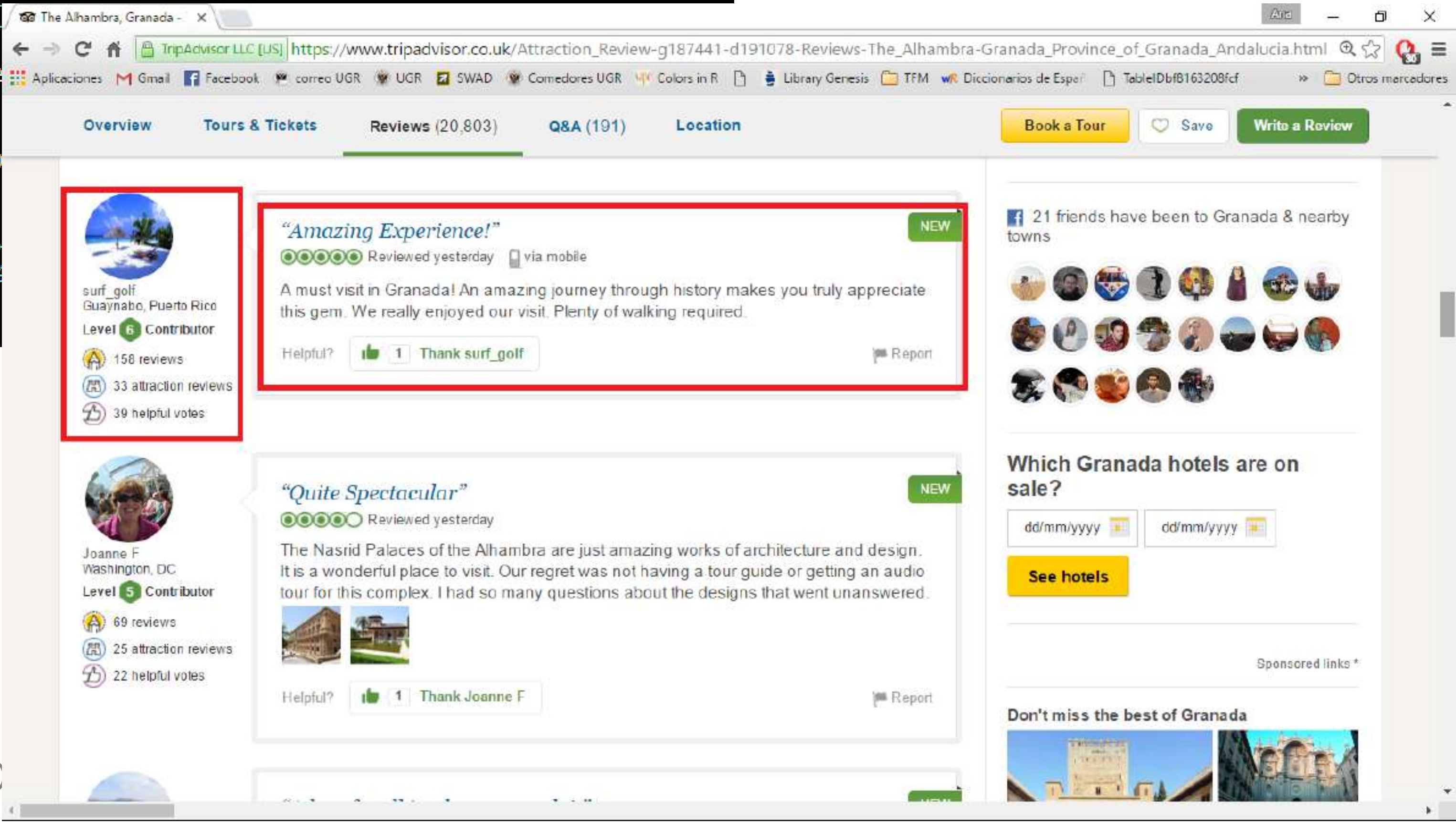
4. MY MASTER'S THESIS

```r
for(i in 1:length(users)){
  user.ch <- as.character(users[i])
  if(substr(user.ch, 27, 33) == "<div id"){
    rev <- append(rev, i, after=length(rev))
  }
  else if(substr(user.ch, 27, 44) == '<div class="avatar'){
    rev <- append(rev, i, after=length(rev))
  }
}
users <- users[

location <- use
  html_nodes(".
  html_text()
location <- rep

# opinions
users <- url %>
  read_html() %
  html_nodes(".

```

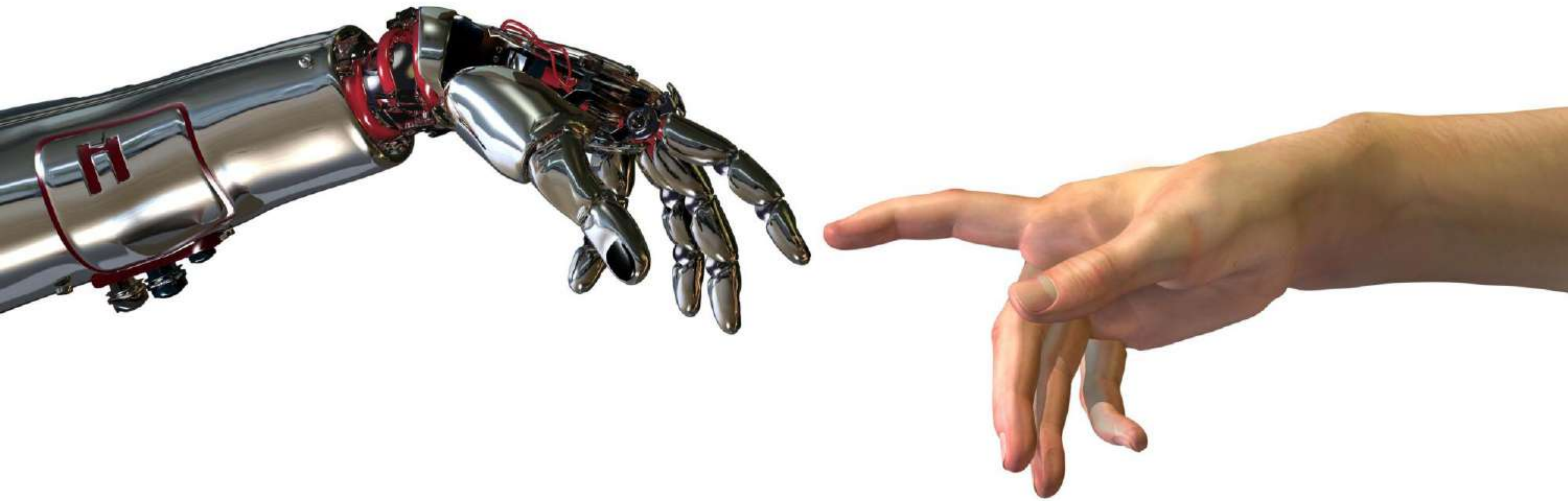


4. MY MASTER'S THESIS

Studying *correlation* between different *sentiment labels*

`SentimentCoreNLP`

`SentimentValue`



4. MY MASTER'S THESIS

Studying *correlation* between different *sentiment labels*

	SentimentCoreNLP			
SentimentValue	positive	neutral	negative	Total
positive	4,049	1,071	2,508	7,628
neutral	51	32	260	343
negative	5	6	158	169
Total	4,105	1,109	2,926	8,140

Table 4.2: Correlation between SentimentValue and SentimentCoreNLP

53.08 % of coincidence

4. MY MASTER'S THESIS

Studying *correlation* between different *sentiment labels*

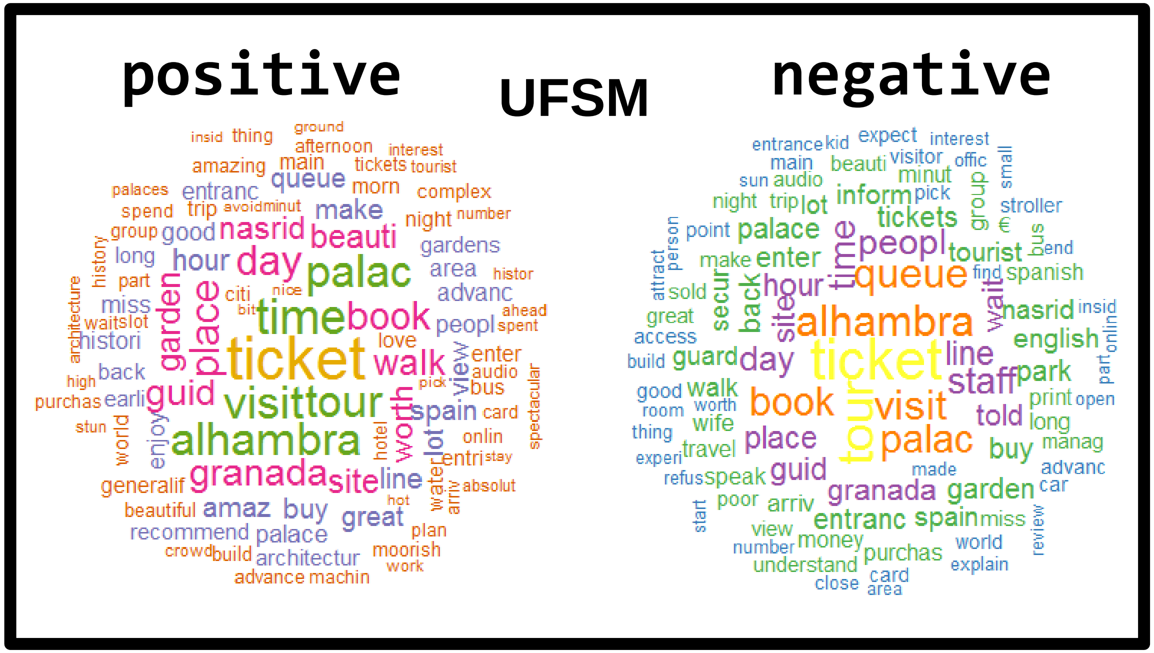
	SentimentCoreNLP			
SentimentValue	positive	neutral	negative	Total
positive	4,049	1,071	2,508	7,628
neutral	51	32	260	343
negative	5	6	158	169
Total	4,105	1,109	2,926	8,140

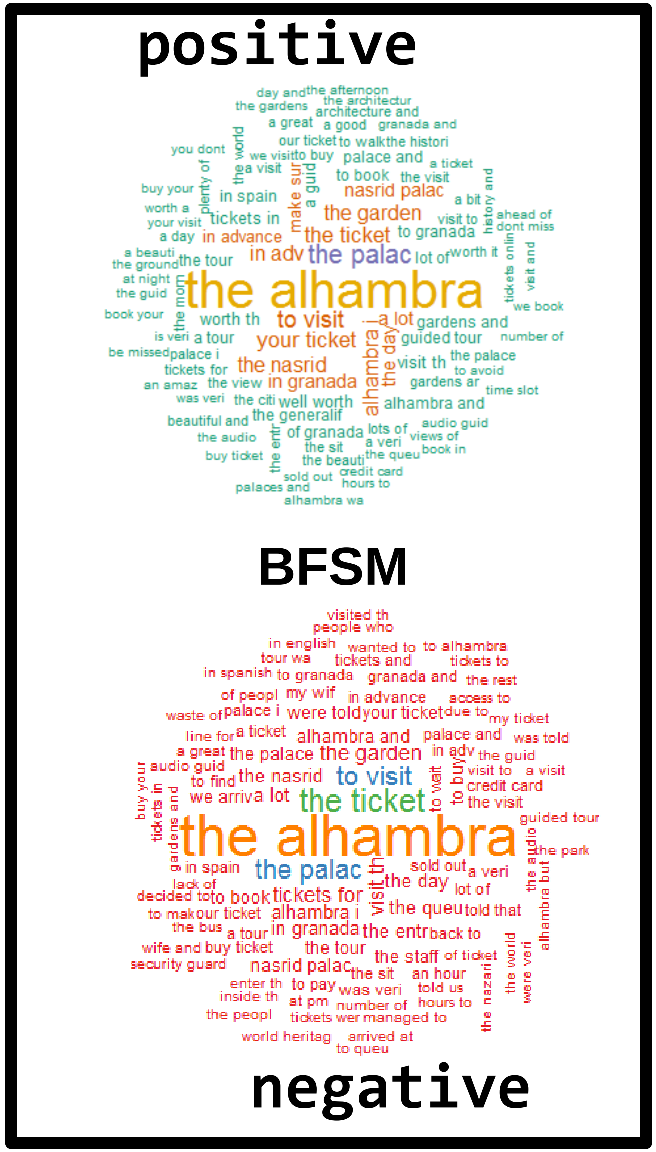
Table 4.2: Correlation between SentimentValue and SentimentCoreNLP

93.49 % of coincidence

4. MY MASTER'S THESIS

Classification problem





Ana Valdivia ©

4. MY MASTER'S THESIS

TripAdvisor Alhambra data set

DocumentTerm Matrix

Use UFSM and BFSM

Split it in three sets depending on **sentiment class label**

Split it up

Split complete set in 75% training set and 25% testing set

Preprocessing

If it is very unbalanced, apply oversampling techniques

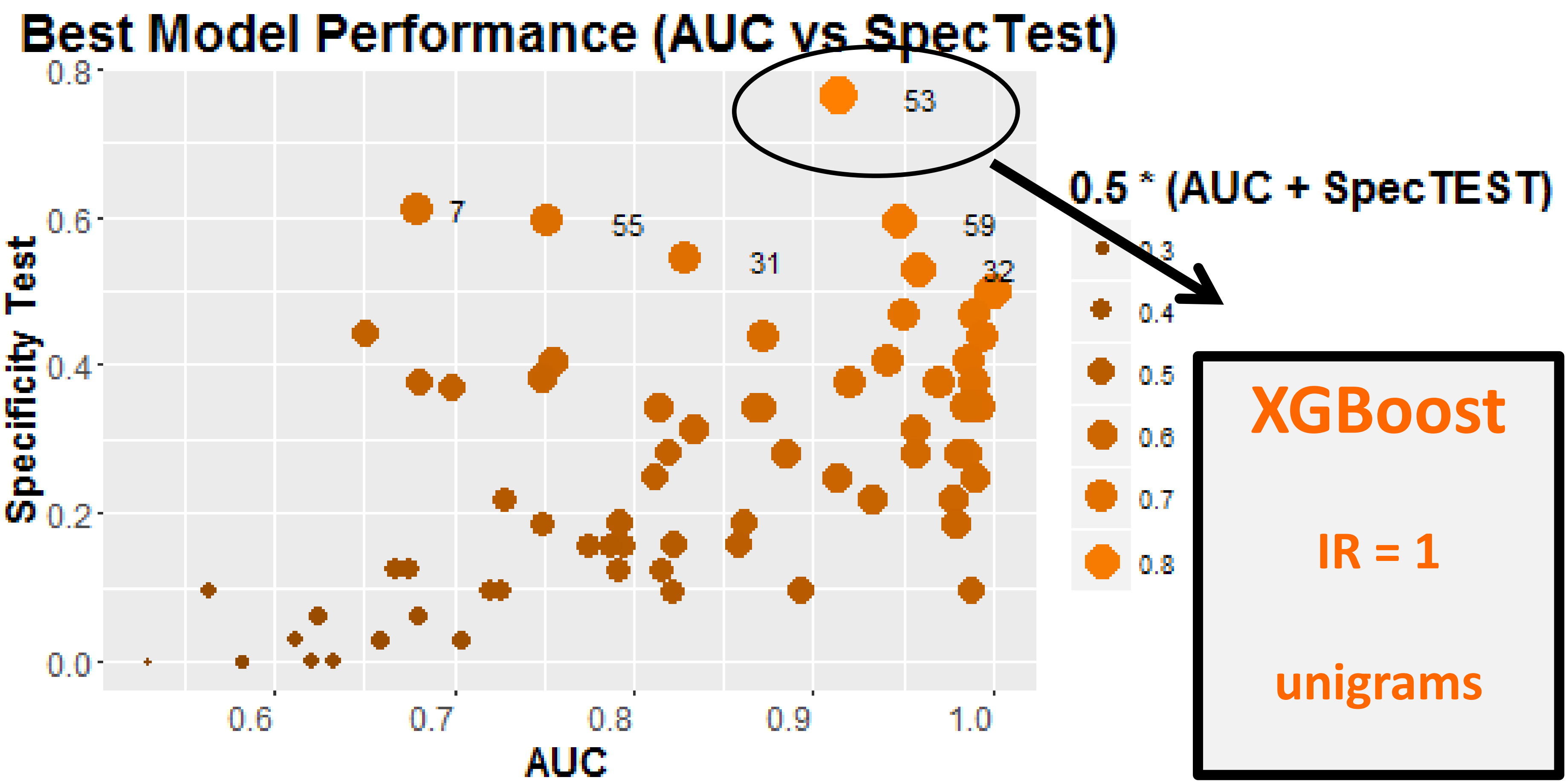
Classification algorithms

Apply different machine learning algorithms in train data set with 5cv

Evaluate Results

Check measure values and dicuss best model

4. MY MASTER'S THESIS



XGBoost

IR = 1

unigrams

4. MY MASTER'S THESIS

Subgroup Discovery

$$R : Cond \longrightarrow Target_{value}$$

negative

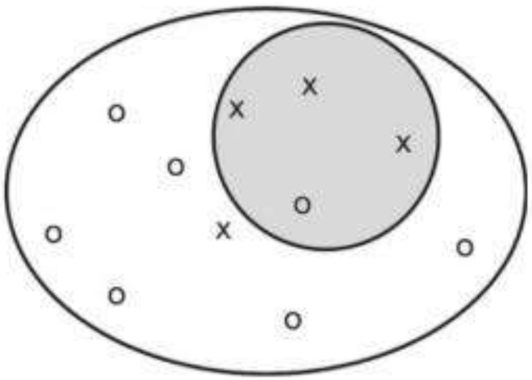


Figure 5.1: Visualization of subgroup discovery rule for $Target_{value} = \times$. Source: [Herrera et al., 2011].

SD-Map algorithm

q_{BT}	p	n	Description
17.17	0.82	22	guard=1
16.21	0.81	21	terribl=1
12.29	0.68	19	rude=1
3.85	1	4	rude=1, guard=1
2.89	1	3	terribl=1, guard=1
2.77	0.5	6	babi=1
1.85	0.5	4	babi=1, strollX=1
1.6	0.06	64	strollX=1
30.33	0.46	71	staff=1
6.7	0.88	8	attitud=1
3.85	1	4	horribl=1
3.85	1	4	attitud=1, staff=1
3.85	1	4	horribl=1, staff=1
1.92	1	2	horribl=1, attitud=1, staff=1
30.33	0.14	284	queueX=1
27.06	0.06	1303	time=1
21.55	0.19	145	queueX=1, time=1
5.21	0.29	21	wheelchair=1
4.66	0.56	9	disabl=1
2.85	0.75	4	disabl=1, wheelchair=1
8.19	0.08	208	night=1
7.2	0.08	181	night=1, light=0
1.41	0.06	69	light=1
0.99	0.07	27	light=1, night=1
0.42	0.05	42	light=1, night=0
0.03	0.29	79	speakX=1
14.13	0.17	103	english=1
6.62	0.7	10	staff=1, english=1, speakX=1
7.62	0.8	10	email=1
5.47	0.43	14	confirm=1
3.85	1	4	email=1, confirm=1

Table 5.2: Subgroups generated from feature correlation.

SUMMARY

SA is a very challenging problem

Lots of applications

New research line

INTELIGENCIA DE NEGOCIO
2020-2021

- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Minería de Datos. Ciencia de Datos
- Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
- Tema 4. Preparación de Datos
- Tema 5. Modelos de Agrupamiento o Segmentación
- Tema 6. Modelos de Asociación
- Tema 7. Modelos Avanzados de Minería de Datos
- Tema 8. Big Data