Sistemas Inteligentes

Introducción ML

José Eduardo Ochoa Luna
Dr. Ciencias - Universidade de São Paulo

Maestría C.C. Universidad Católica San Pablo
Sistemas Inteligentes

11 de octubre 2018

Biblio and Resources

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning, Springer, 2017.

Biblio and Resources

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning, Springer, 2017.
- Christopher M. Bishop, Pattern Recognition and Machine Learning. 2007.

Biblio and Resources

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning, Springer, 2017.
- Christopher M. Bishop, Pattern Recognition and Machine Learning. 2007.
- Tom Mitchell, Machine Learning. 1997.

Biblio and Resources

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning, Springer, 2017.
- Christopher M. Bishop, Pattern Recognition and Machine Learning. 2007.
- Tom Mitchell, Machine Learning. 1997.
- Judea Pearl, Probabilistic Reasoning in Intelligent Systems: networks of plausible inference. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1988.

Biblio and Resources

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning, Springer, 2017.
- Christopher M. Bishop, Pattern Recognition and Machine Learning. 2007.
- Tom Mitchell, Machine Learning. 1997.
- Judea Pearl, Probabilistic Reasoning in Intelligent Systems: networks of plausible inference. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1988.
- Daphne Koller, Nir Friedman, Probabilistic Graphical Models. MIT Press. 2009.

Biblio and Resources

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning, Springer, 2017.
- Christopher M. Bishop, Pattern Recognition and Machine Learning. 2007.
- Tom Mitchell, Machine Learning. 1997.
- Judea Pearl, Probabilistic Reasoning in Intelligent Systems: networks of plausible inference. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1988.
- Daphne Koller, Nir Friedman, Probabilistic Graphical Models. MIT Press. 2009.
- Andrew Ng ML handouts

Tools

- Python, Notebooks

Tools

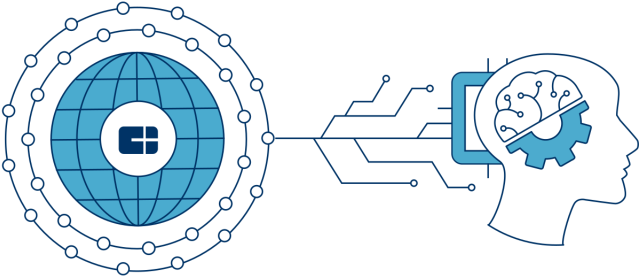
- Python, Notebooks
- SamIam, Libra

Tools

- Python, Notebooks
- SamIam, Libra
- TensorFlow

AI - Machine Learning

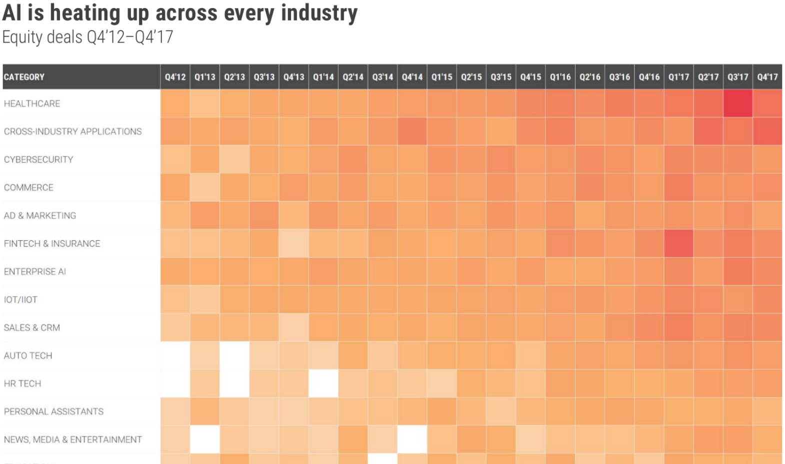
13 Artificial Intelligence trends reshaping industries and economies: CB-Insights 2018



AI - Machine Learning

2. AI for X is ? everywhere

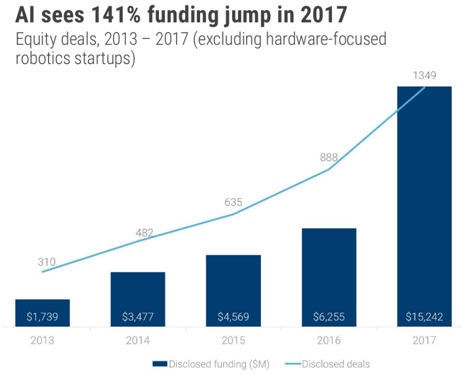
Artificial Intelligence is everywhere. Or more exactly, machine learning is everywhere.



ML Trends

10. The machine learning hype will die

Machine learning will soon become the new normal.



Machine Learning

- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed

Machine Learning

- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed
- Mitchell (1997): Machine Learning is the study of computer algorithms that improve automatically through experience.

Applications

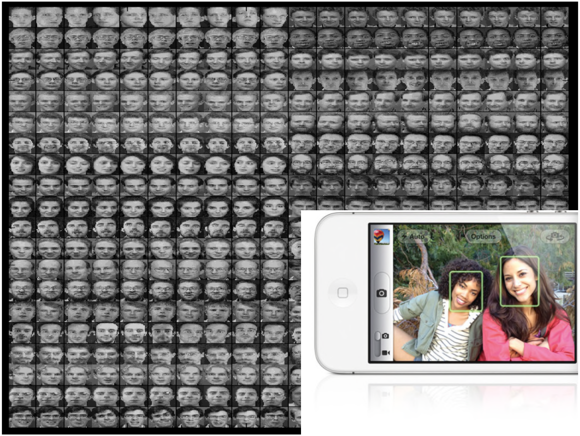
Machine Learning Applications (Nando de Freitas)

Applications

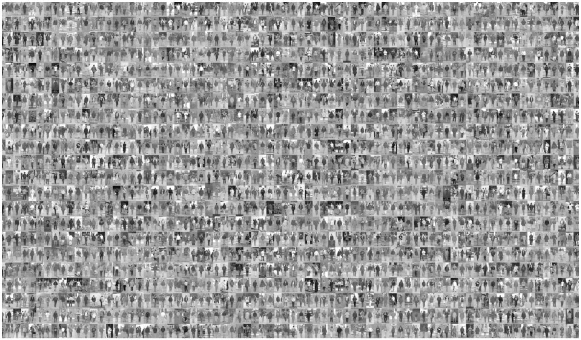
Application: Invariant recognition in natural images

[Thomas Serre 2012]

Applications



Applications



Millions of labeled examples are used to build real-world applications, such as pedestrian detection

[Tomas Serre]

Applications

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

Automatic Image Caption Generation
Sample taken from Andrej Karpathy, Li Fei-Fei

Applications

Machines that learn to recognise what they see and hear are at the heart of Apple, Google, Amazon, Facebook, Netflix, Microsoft, etc.

Applications

Review sentiment and summarization

amazon.com reviews

review us on yelp

Google Reviews

tripadvisor®

WORLD'S MOST TRUSTED TRAVEL ADVICE™

my reading was similar to everyones. she told me she was going to take her time and not rush me out of there. i was there not even 8 minutes she told me i was pregnant then she changed her mind and said i had a miscarriage. im 17 years old i told her she was wrong she then went on and said "i see you and your brother fight alot just know he loves you" i dont even have a brother.

she then told my friend she was going to get stabbed

Was this review helpful? Yes 2 Problem with this review?

Ask taydube about Fatima's Psychic Studio

Paul Bettany did a great role as the tortured father whose favorite little girl dies tragically of disease. For that, he deserves all the credit.
However, the movie was mostly about exactly that, keeping the adventures of Darwin as he gathered data for his theories as incomplete stories told to children and skipping completely the disputes regarding his ideas.
Two things bothered me terribly: the soundtrack, with its whiny sound, practically shoving sadness down the throat of the viewer, and the movie trailer, showing some beautiful sceneries, the theological musings of him and his wife and the enthusiasm of his best friends as they prepare for a battle against blind faith, thus misrepresenting the movie completely.
To put it bluntly, if one were to remove the scenes of the movie trailer from the movie, the result would be a non descript family drama about a little child dying and the hardships of her parents as a result.
Clearly, not what I expected from a movie about Darwin, albeit the movie was beautifully interpreted.

[Kotzias, Denil, Blunsom & NdF, 2014]

US Elections

BUSINESS INSIDER

POLITICS

An artificial intelligence system that correctly predicted the last 3 elections says Trump will win

Pamela Engel

Oct. 28, 2016, 8:24 PM 152,066

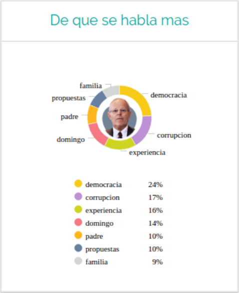
Peruvian Elections

Parlakuy.com
Parlakuy-insights.com

Elecciones 2016.

Análisis de la percepción de los candidatos presidenciales.

Predicción de resultados de acuerdo a tendencias.

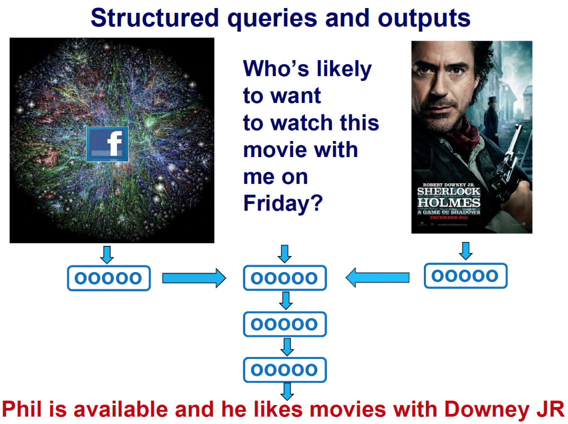


Women and Mother - Facebook

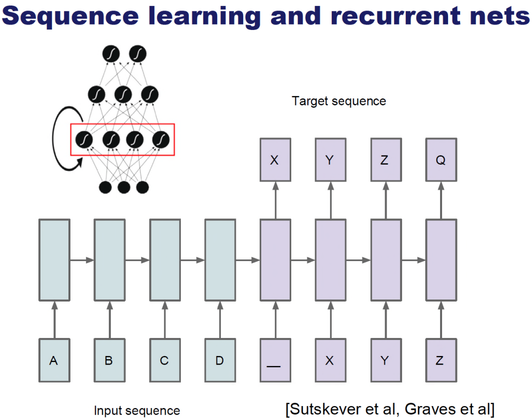
Figura: Positive

Figura: Negative

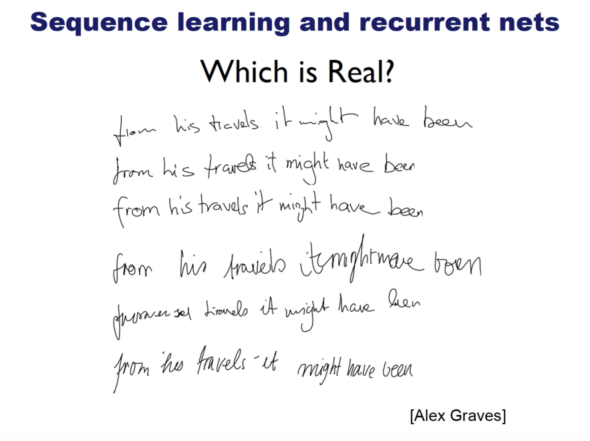
Applications



Applications



Applications

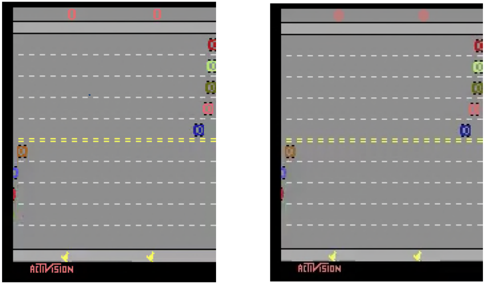


Applications

Instant Visual Translation
Example of instant visual translation, taken from the Google Blog.

Applications

Imitation learning for Atari



[Dejan Markovikj, Miroslav Bogdanovic, Misha Denil, NdF 2014]

ML Applications

ML deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)

ML Applications

ML deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)
- Imputing missing data (e.g. Netflix recommendations)

ML Applications

ML deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)
- Imputing missing data (e.g. Netflix recommendations)
- Detecting anomalies (e.g. Security, fraud, virus mutations)

ML Applications

ML deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)
- Imputing missing data (e.g. Netflix recommendations)
- Detecting anomalies (e.g. Security, fraud, virus mutations)
- Classifying (e.g. Credit risk assessment, cancer diagnosis)

ML Applications

ML deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)
- Imputing missing data (e.g. Netflix recommendations)
- Detecting anomalies (e.g. Security, fraud, virus mutations)
- Classifying (e.g. Credit risk assessment, cancer diagnosis)
- Summarizing (e.g. News, social media sentiment)

ML Applications

ML deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)
- Imputing missing data (e.g. Netflix recommendations)
- Detecting anomalies (e.g. Security, fraud, virus mutations)
- Classifying (e.g. Credit risk assessment, cancer diagnosis)
- Summarizing (e.g. News, social media sentiment)
- Decision making (e.g. AI, robotics)

When to apply ML

- Human expertise is absent (e.g. Navigation on Mars)

When to apply ML

- Human expertise is absent (e.g. Navigation on Mars)
- Humans are unable to explain their expertise (e.g. Speech recognition, vision, language)

When to apply ML

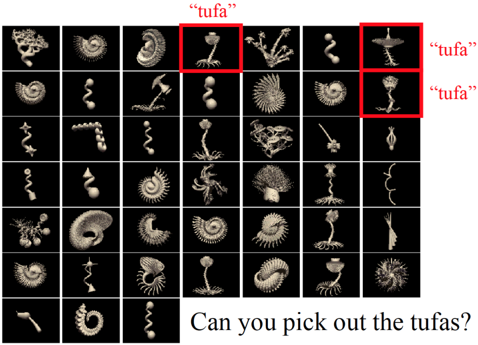
- Human expertise is absent (e.g. Navigation on Mars)
- Humans are unable to explain their expertise (e.g. Speech recognition, vision, language)
- Solution changes with time (e.g. Tracking, temperature control, preferences)

When to apply ML

- Human expertise is absent (e.g. Navigation on Mars)
- Humans are unable to explain their expertise (e.g. Speech recognition, vision, language)
- Solution changes with time (e.g. Tracking, temperature control, preferences)
- The problem size is to vast for our limited reasoning capabilities (e.g. calculating webpage ranks)

ML Challenge

Challenge: One-shot learning

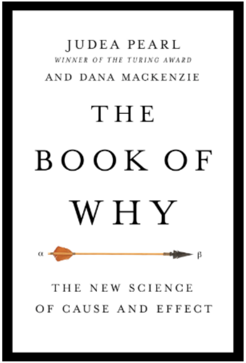


Can you pick out the tufas?

Josh Tenenbaum

AI Challenge

Explainable AI

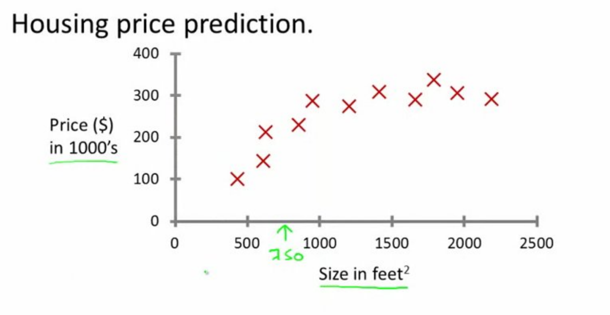


Learning

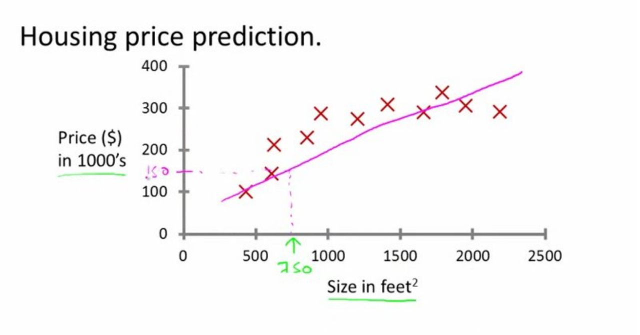
Supervised and Unsupervised Learning

Aprendizaje Supervisado

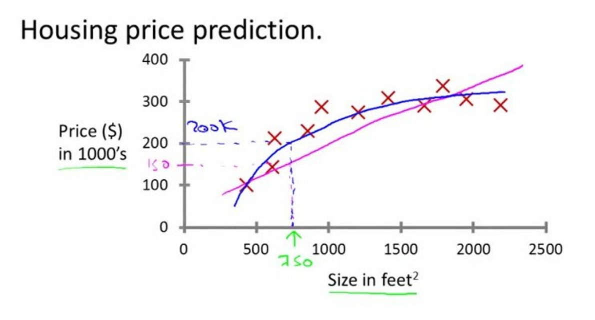
Aprendizaje Supervisado



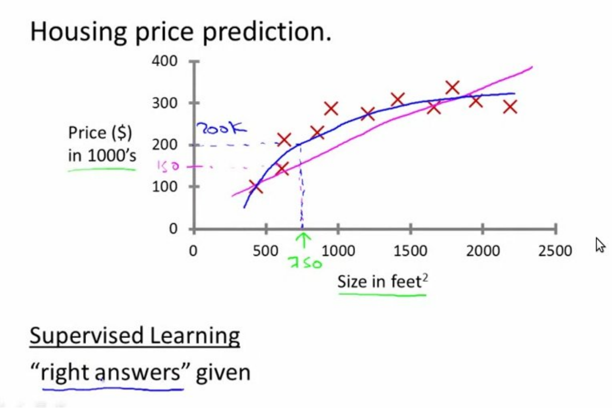
Aprendizaje Supervisado



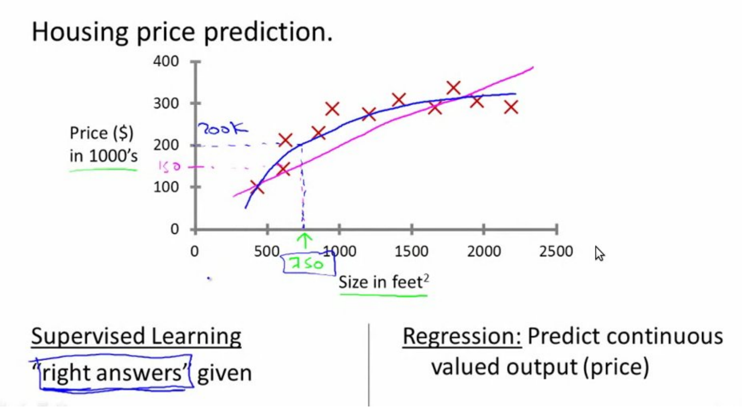
Aprendizaje Supervisado



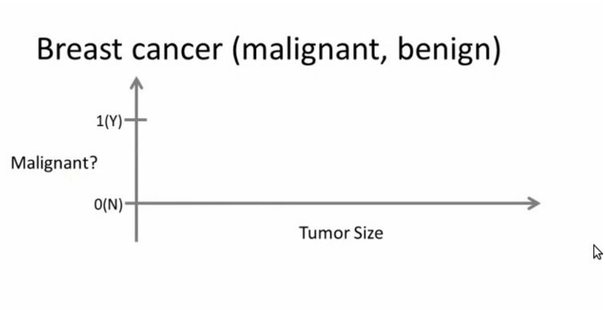
Aprendizaje Supervisado



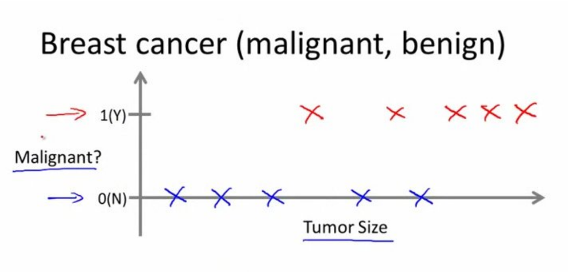
Aprendizaje Supervisado



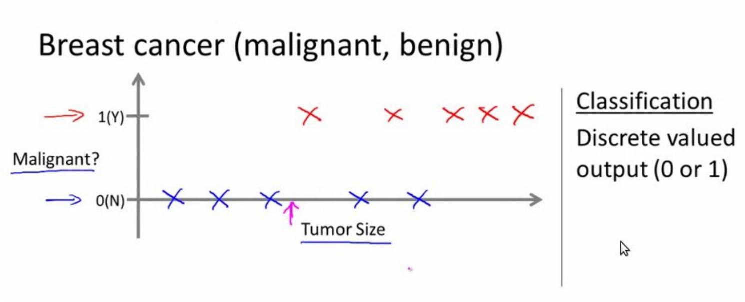
Aprendizaje Supervisado



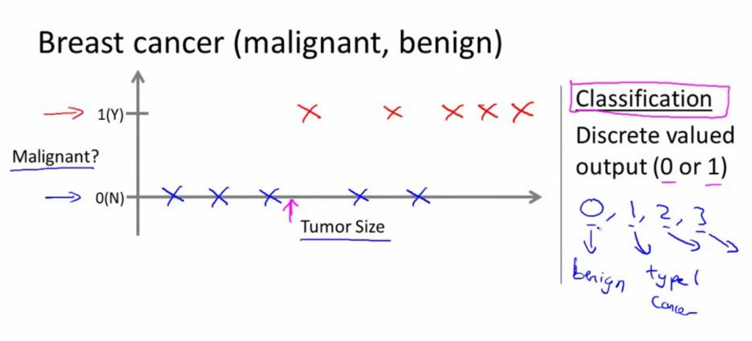
Aprendizaje Supervisado



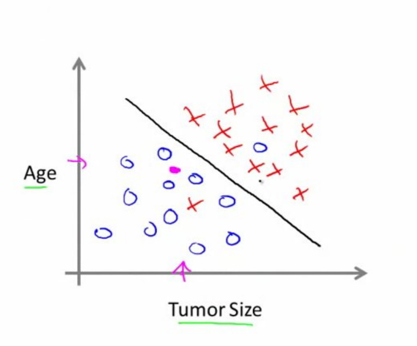
Aprendizaje Supervisado



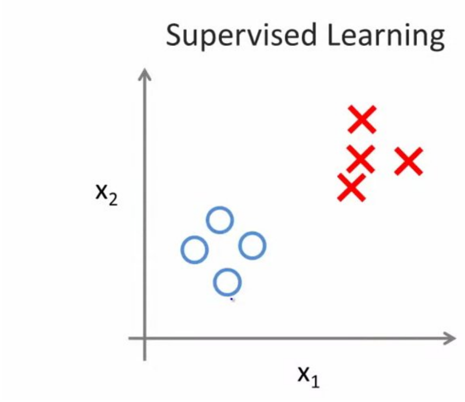
Aprendizaje Supervisado



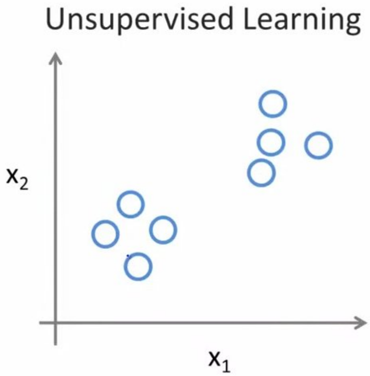
Aprendizaje Supervisado



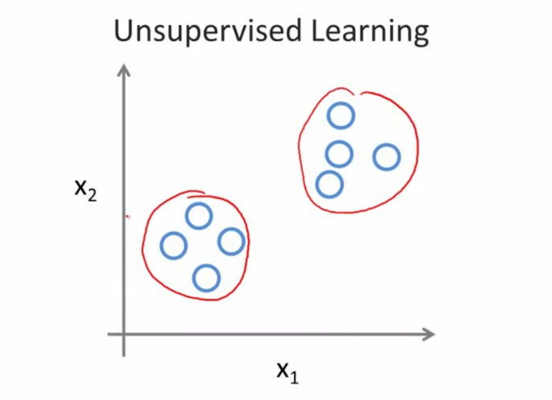
Aprendizaje No Supervisado



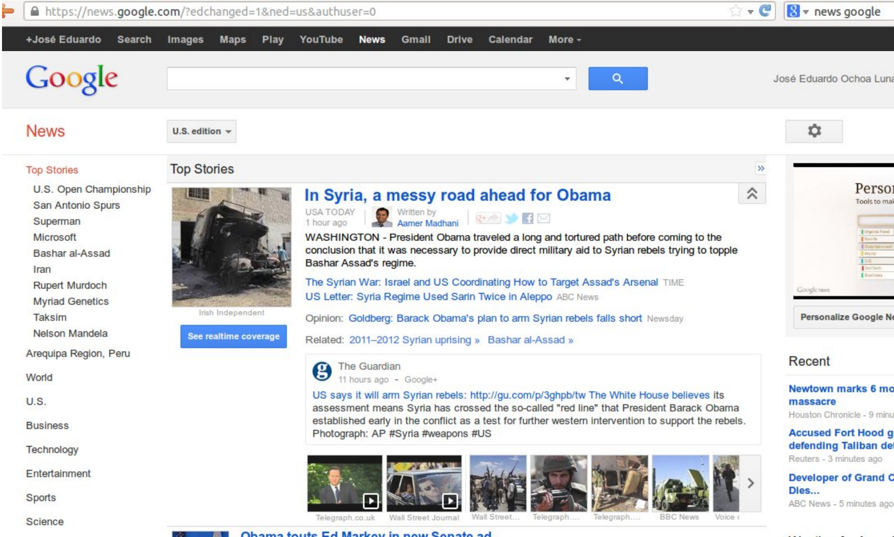
Aprendizaje No Supervisado



Aprendizaje No Supervisado

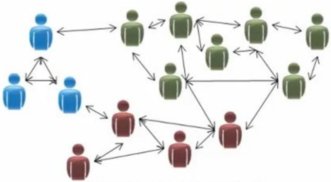


Aprendizaje No Supervisado



Aprendizaje No Supervisado

Organize computing clusters



Social network analysis

Market segmentation

Astronomical data analysis

Linear Regression

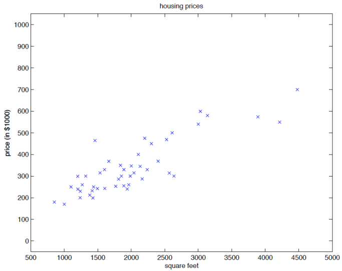
Linear Regression

Example

We have a dataset giving the living areas and prices of 47 houses from Portland, Oregon:

Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Data



Given data like this, how can we learn to predict the prices of other houses in Portland, as a function of the size of their living areas?

Notation

- $x^{(i)}$ denotes the input variables (living area), also called input features

Notation

- $x^{(i)}$ denotes the input variables (living area), also called input features
- $y^{(i)}$ denotes the output or target variable we are trying to predict (price)

Notation

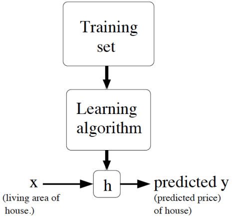
- $x^{(i)}$ denotes the input variables (living area), also called input features
- $y^{(i)}$ denotes the output or target variable we are trying to predict (price)
- A pair $(x^{(i)}, y^{(i)})$ is called a training example. A list of m training examples is called a training set

Notation

- $x^{(i)}$ denotes the input variables (living area), also called input features
- $y^{(i)}$ denotes the output or target variable we are trying to predict (price)
- A pair $(x^{(i)}, y^{(i)})$ is called a training example. A list of m training examples is called a training set
- X denotes the space of input values and Y the space of output values.

The problem

Goal: given a training set, to learn a function $h : X \to Y$ so that $h(x)$ is a good predictor for the corresponding value of y. h is called a hypothesis



Linear Regression

X's are two dimensional vectors. $x_1^{(i)}$ is the living area of the i-th house in the training set. $x_2^{(i)}$ is its number of bedrooms.

Living area (feet^2)	#bedrooms	Price (1000\$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

How to represent functions / hypotheses h in a computer?

Linear Regression II

We decide to approximate y as a linear function of x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

- θ_i's are the parameters (weights), parameterizing the space of linear functions mapping from X to Y.

Linear Regression II

We decide to approximate y as a linear function of x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

- θ_i's are the parameters (weights), parameterizing the space of linear functions mapping from X to Y.
- Let $x_0 = 1$ (the intercept term), so that

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Linear Regression II

We decide to approximate y as a linear function of x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

- θ_i's are the parameters (weights), parameterizing the space of linear functions mapping from X to Y.
- Let $x_0 = 1$ (the intercept term), so that

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

- θ and x are vectors, n is the number of input variables

Cost function

- How do we pick the parameters θ?

Cost function

- How do we pick the parameters θ?
- make $h(x)$ close to y, at least for the training examples

Cost function

- How do we pick the parameters θ?
- make $h(x)$ close to y, at least for the training examples
- To formalize, we define a function that measures, how to close the $h(x^{(i)})$'s are to the corresponding $y^{(i)}$'s, the cost function:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Cost function

- How do we pick the parameters θ?
- make $h(x)$ close to y, at least for the training examples
- To formalize, we define a function that measures, how to close the $h(x^{(i)})$'s are to the corresponding $y^{(i)}$'s, the cost function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- least-squares cost function

Least Mean Squares Algorithm

- We want to choose θ so as to minimize $J(\theta)$

Least Mean Squares Algorithm

- We want to choose θ so as to minimize $J(\theta)$
- Let's use a search algorithm that start with some initial guess for θ and

Least Mean Squares Algorithm

- We want to choose θ so as to minimize $J(\theta)$
- Let's use a search algorithm that start with some initial guess for θ and
- repeatedly changes θ to make $J(\theta)$ smaller

Least Mean Squares Algorithm

- We want to choose θ so as to minimize $J(\theta)$
- Let's use a search algorithm that start with some initial guess for θ and
- repeatedly changes θ to make $J(\theta)$ smaller
- until hopefully we converge to a value of θ that minimizes $J(\theta)$

Gradient Descent Algorithm

- consider the gradient descent algorithm, which starts with some initial θ and performs the update:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Gradient Descent Algorithm

- consider the gradient descent algorithm, which starts with some initial θ and performs the update:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- α is called the learning rate

Gradient Descent Algorithm

- consider the gradient descent algorithm, which starts with some initial θ and performs the update:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- α is called the learning rate
- repeatedly takes a step in the direction of steepest decrease of J

Gradient Descent Algorithm

- consider the gradient descent algorithm, which starts with some initial θ and performs the update:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- α is called the learning rate
- repeatedly takes a step in the direction of steepest decrease of J
- we have to work out what is the partial derivative term

Gradient Descent Algorithm - Partial Derivatives

only one training example (x, y)

$$
\begin{aligned}
\frac{\partial}{\partial\theta_j} &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2 \cdot \frac{1}{2}(h_\theta(x) - y) \cdot \frac{\partial}{\partial\theta_j}(h_\theta(x) - y) \\
&= (h_\theta(x) - y) \cdot \frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n} \theta_i x_i - y\right) \\
&= (h_\theta(x) - y)x_j
\end{aligned}
$$

LMS update rule

For a single training example, this gives the update rule

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

The least mean squares update rule (Widrow-Hoff learning rule)

- the magnitude of the update is proportional to the error term $(y^{(i)} - h_\theta(x^{(i)}))$
- if in a training example our prediction nearly matches the actual value of $y^{(i)}$, then parameter are unchanged
- if the prediction has a large error then a larger change will be made

Batch Gradient Descent

For a training set:
Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \quad \text{for every } j$$

}

- The entire training set is considered on every step (batch gradient descent)

Batch Gradient Descent

For a training set:
Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \quad \text{for every } j$$

}

- The entire training set is considered on every step (batch gradient descent)
- This method can be susceptible to local minima in general

Batch Gradient Descent

For a training set:
Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \quad \text{for every } j$$

}

- The entire training set is considered on every step (batch gradient descent)
- This method can be susceptible to local minima in general
- The optimization problem for linear regression has only one global optima (J is a convex function)

Batch Gradient Descent

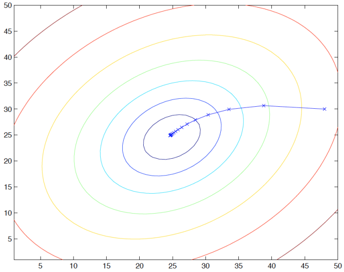
For a training set:
Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \quad \text{for every } j$$

}

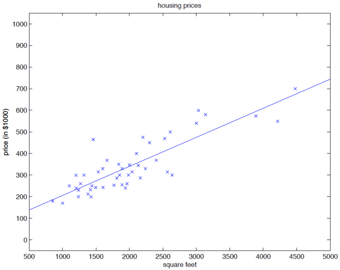
- The entire training set is considered on every step (batch gradient descent)
- This method can be susceptible to local minima in general
- The optimization problem for linear regression has only one global optima (J is a convex function)
- gradient descent always converge (assuming α is not too large)

Gradient Descent convergence



Ellipses are the contours of a quadratic function. The x's mark the successive values of θ that gradient descent went through

Gradient Descent fit



predict housing price as a function of living area: $\theta_0 = 71.27$, $\theta_1 = 0.1345$

Stochastic Gradient Descent

$$
\begin{aligned}
&\text{Loop } \{ \\
&\qquad \text{for i=1 to m, } \{ \\
&\qquad\qquad \theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)} \qquad \text{(for every } j\text{).} \\
&\qquad \} \\
&\}
\end{aligned}
$$

- we repeatedly run through the training set, and each time we encounter a training example, we update the parameters

Stochastic Gradient Descent

Loop {

for i=1 to m, {

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every } j\text{)}.$$

}

}

- we repeatedly run through the training set, and each time we encounter a training example, we update the parameters
- Batch gradient descent has to scan through the entire training set (costly if m is large)

Stochastic Gradient Descent

Loop {

for i=1 to m, {

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every } j\text{)}.$$

}

}

- we repeatedly run through the training set, and each time we encounter a training example, we update the parameters
- Batch gradient descent has to scan through the entire training set (costly if m is large)
- Often SGD gets θ close to the minimum much faster than batch GD

Probabilistic Interpretation

Let us asume that the target variables and the inputs are related via the equation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- $\epsilon^{(i)}$ is an error term that captures either unmodeled effects or random noise

Probabilistic Interpretation

Let us asume that the target variables and the inputs are related via the equation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- $\epsilon^{(i)}$ is an error term that captures either unmodeled effects or random noise
- $\epsilon^{(i)}$ are distributed IID (independently and identically distributed) according to a Gaussian distribution with mean zero and some variance σ^2, ($\epsilon^{(i)} \sim N(0, \sigma^2)$)

Probabilistic Interpretation

Let us asume that the target variables and the inputs are related via the equation

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- $\epsilon^{(i)}$ is an error term that captures either unmodeled effects or random noise
- $\epsilon^{(i)}$ are distributed IID (independently and identically distributed) according to a Gaussian distribution with mean zero and some variance σ^2, ($\epsilon^{(i)} \sim N(0, \sigma^2)$)
- density of $\epsilon^{(i)}$ is given by

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(\epsilon^{(i)})^2}{2\sigma^2}\right)$$

Probabilistic Interpretation

This implies

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- this is the distribution of $y^{(i)}$ given $x^{(i)}$ and parameterized by θ (it can also write as $y^{(i)}|x^{(i)};\theta \sim N(\theta^T x^{(i)}, \sigma^2)$)

Probabilistic Interpretation

This implies

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- this is the distribution of $y^{(i)}$ given $x^{(i)}$ and parameterized by θ (it can also write as $y^{(i)}|x^{(i)};\theta \sim N(\theta^T x^{(i)}, \sigma^2)$)
- Given X (all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?

Probabilistic Interpretation

This implies

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- this is the distribution of $y^{(i)}$ given $x^{(i)}$ and parameterized by θ (it can also write as $y^{(i)}|x^{(i)};\theta \sim N(\theta^T x^{(i)}, \sigma^2)$)
- Given X (all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?
- The probability of the data is given by $p(\vec{y}|X;\theta)$

Probabilistic Interpretation

This implies

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- this is the distribution of $y^{(i)}$ given $x^{(i)}$ and parameterized by θ (it can also write as $y^{(i)}|x^{(i)};\theta \sim N(\theta^T x^{(i)}, \sigma^2)$)
- Given X (all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?
- The probability of the data is given by $p(\vec{y}|X;\theta)$
- This quantity is typically viewed a function of $\vec{y}$ (and perhaps X), for a fixed value of θ

Probabilistic Interpretation

This implies

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

- this is the distribution of $y^{(i)}$ given $x^{(i)}$ and parameterized by θ (it can also write as $y^{(i)}|x^{(i)};\theta \sim N(\theta^T x^{(i)}, \sigma^2)$)
- Given X (all the $x^{(i)}$'s) and θ, what is the distribution of the $y^{(i)}$'s?
- The probability of the data is given by $p(\vec{y}|X;\theta)$
- This quantity is typically viewed a function of $\vec{y}$ (and perhaps X), for a fixed value of θ
- When view this as a function of θ, it is called the likelihood function

Likelihood Function

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta)$$

By the independence assumption on the $\epsilon^{(i)}$'s , this can also be written

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

We should choose θ to maximize $L(\theta)$ (maximum likelihood)

log Likelihood

Instead of maximizing $L(\theta)$, we can also maximize the log likelihood:

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2 \end{aligned}$$

hence, maximizing $l(\theta)$ gives the same answer as minimizing

$$\frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2$$

Projects

Last year

- Sentiment Analysis (Spanish)
- Machine Translation, Question Answering

This year

- Visual Question Answering (Medical domain)
- Sentiment Analysis (GANs Data Augmentation)
- Diagnosis (CNNs + Bayesian Networks)

Tarea

- Crear una cuenta en kaggle,
- Participar del siguiente desafio: https://www.kaggle.com/c/house-prices-advanced-regression-techniques
- Bajar el dataset, crear un jupyter notebook, donde se implementen las diversas etapas, get data, data exploration, prepare data, etc.
- Probar varios algoritmos, en el caso de la regresion lineal multivariada, usar una implementación propia.
- Subir el IPython al Github personal e imagen rmse obtenido en kaggle (18 octubre)