Introduction to Machine Learning 1

Nov., 2018

D. Ratner

SLAC National Accelerator Laboratory

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction

What is machine learning?

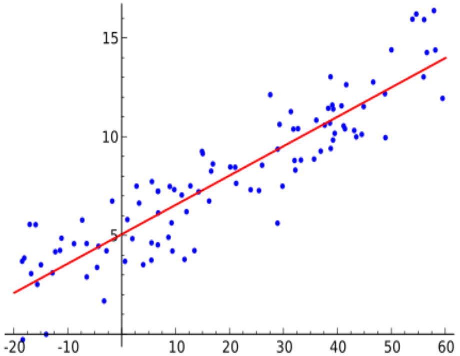
Arthur Samuel (1959): Ability to learn without being explicitly programmed

Tom Mitchell (1998): Computer program learns from experience E with respect to task T if its performance, P, improves after experience E.

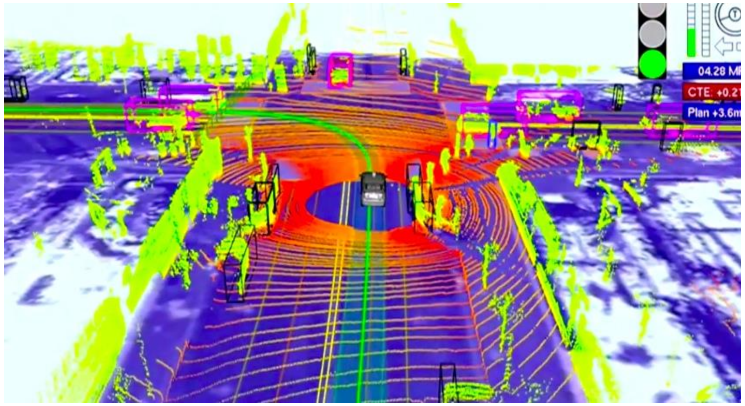
When is machine learning successful?

Tasks which humans can learn, but have trouble explaining how

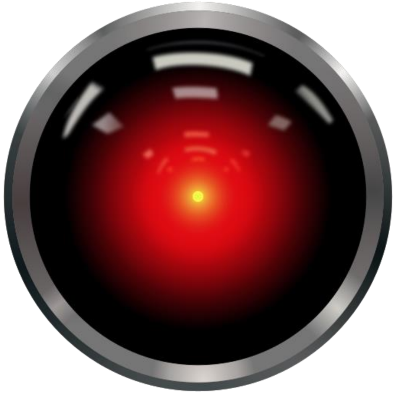
Regression



Neural networks



Sentient computers



Topics

Supervised learning (examples with labels):

ML framework/terminology

Regression vs. classification

Parameteric vs. non-parametric models

Unsupervised learning (examples, no labels):

Clustering, anomaly/breakout detection, generation

Reinforcement learning (examples, partial labels):

Control, games, optimization

Goal from Lecture 1: Learn terminology and framework of ML

Goal from Lecture 2: See examples of ML in accelerator physics

Material drawn from: Stanford CS 229, EE103
Michael Nielsen, Neural Networks and Deep Learning

Least Squares Regression

Start from a simple problem: can we predict house price?

- "Training set" consists of *m* examples
- Each example has *n* attributes (**x**) and one label (y)

Our goal: given a new example, x', can we predict its label, y'?

Hypothesis:

example *i*

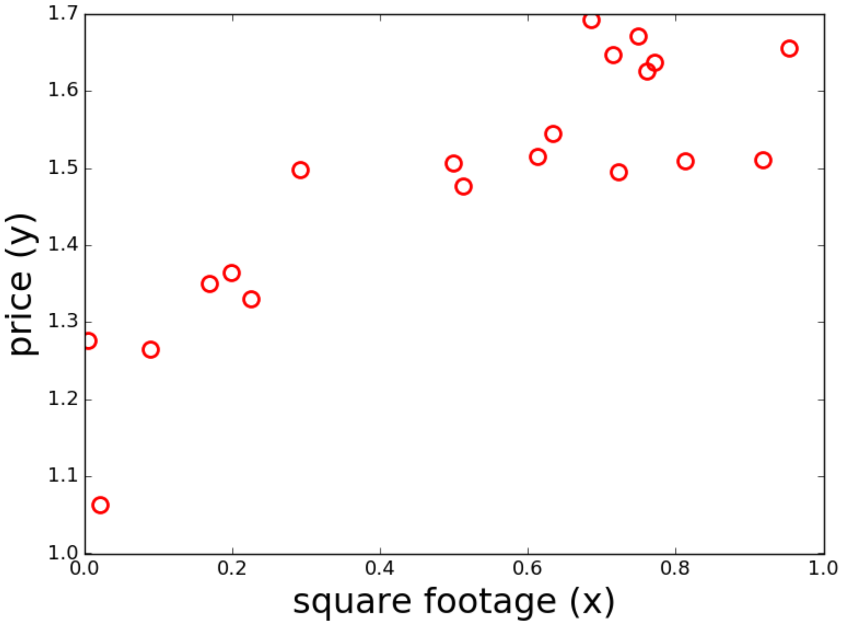
sum over *n* attributes

$$h_\theta(x^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)} = \theta^T x^{(i)}$$

guess for y

"Parameters/weights"

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x_1^{(i)}$$



Supervised learning: Parametric models

SLAC

Least Squares Regression

The core of machine learning: how do we learn best θ given data x,y?

Need a metric for “best”: Cost/Loss function

$$h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

Examples: mean square error (MSE), absolute error, etc.

MSE:

# of examples

groundtruth=label

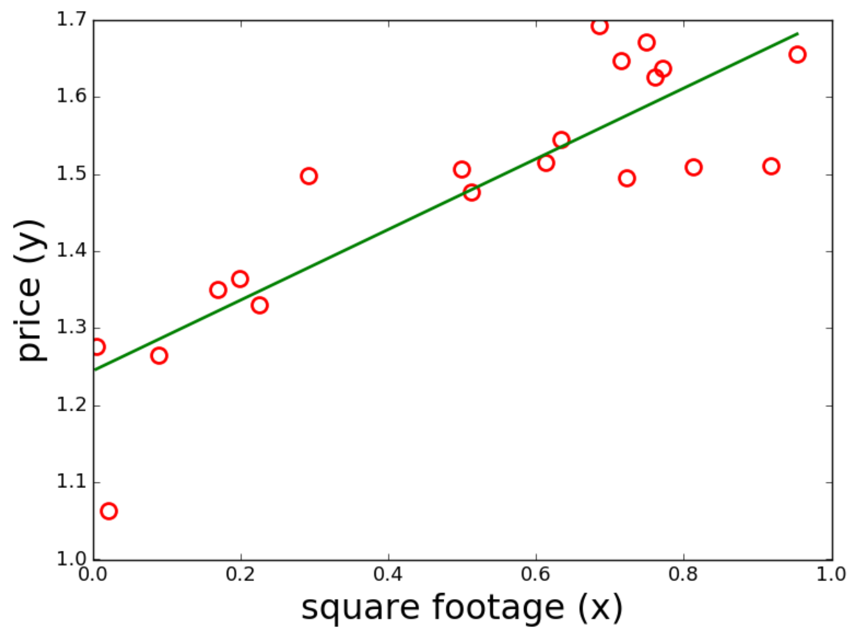
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y})$$

n+1 x 1 m x n+1 m x 1

Optimal θ:

$$\theta = (X^TX)^{-1}X^T\vec{y}.$$



SLAC

Least Squares Regression

The core of machine learning: how do we learn best θ given data X,y?

Need a metric for “best”: Cost/Loss function

$$h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

Examples: mean square error (MSE), absolute error, etc.

MSE:

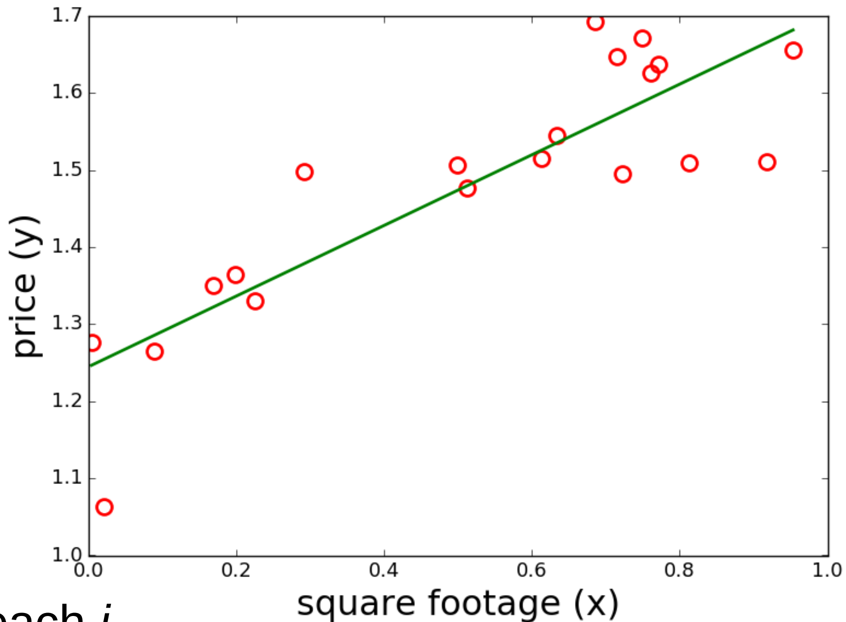
# of examples

groundtruth

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

$$\theta_j := \theta_j - \alpha \underbrace{\frac{\partial}{\partial \theta_j} J(\theta)}_{\sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}}$$

Learning rate



“Stochastic gradient descent”: update θ after each *i*

Supervised learning: Hyper-parameter choice

Least Squares Regression

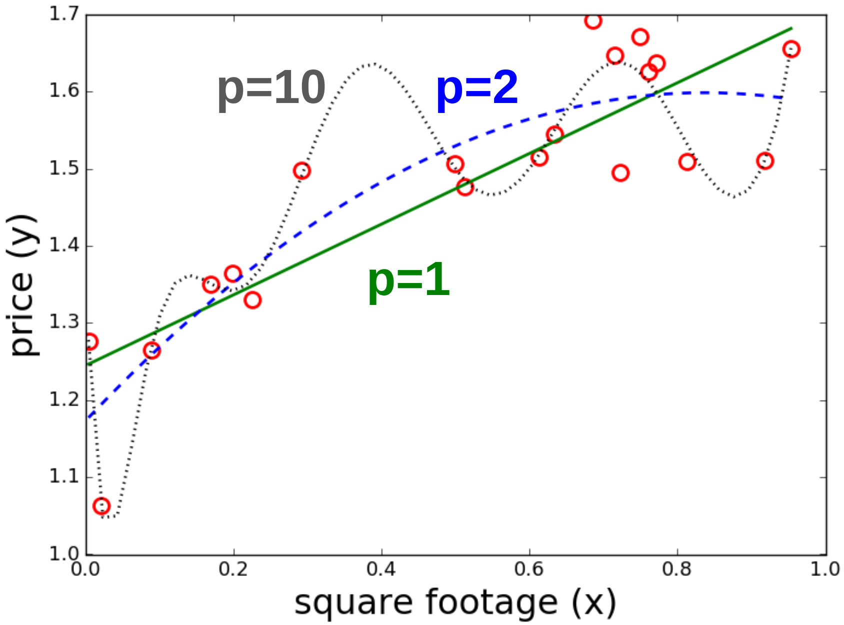
“Hyper-parameters”: how do we choose model itself?

e.g. pick model architecture, cost function, learning rate, etc.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$
$$= \sum_{k=0}^{p} \theta_k x^k$$

$$x \to \phi(x) = [x, x^2, \ldots]$$

attributes   features



Least Squares Regression

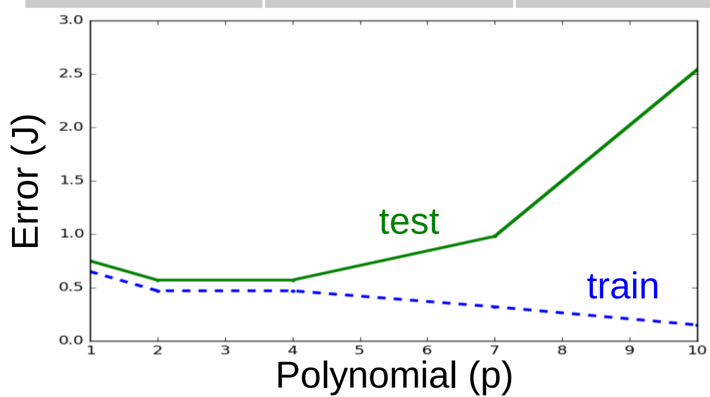
"Hyper-parameters": how do we choose model itself?

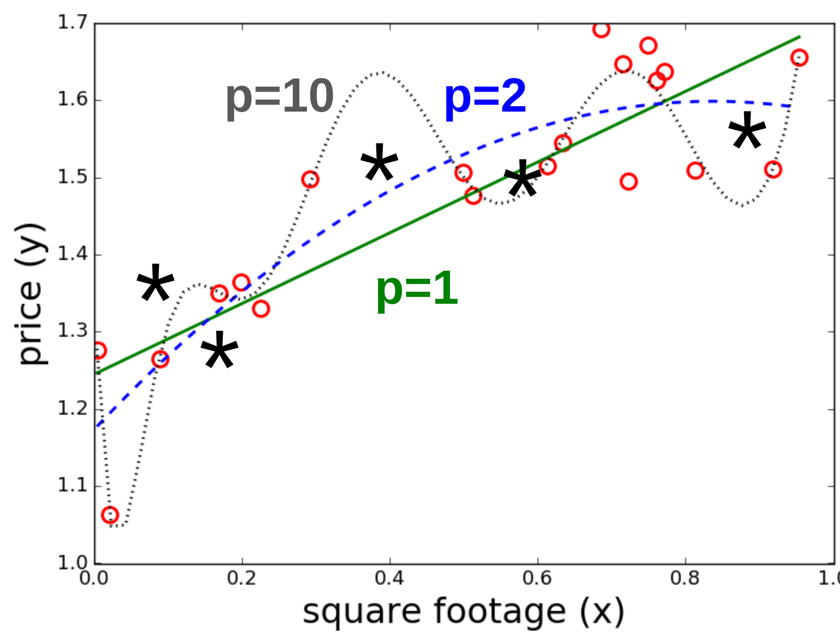
e.g. pick model architecture, cost function, learning rate, etc.

Split data into training and test (and validation) sets

Typical split: 80/20 or 80/10/10

Degree (p)	Train error	Test error
1	0.65	0.75
2	0.47	0.57
10	0.15	2.54



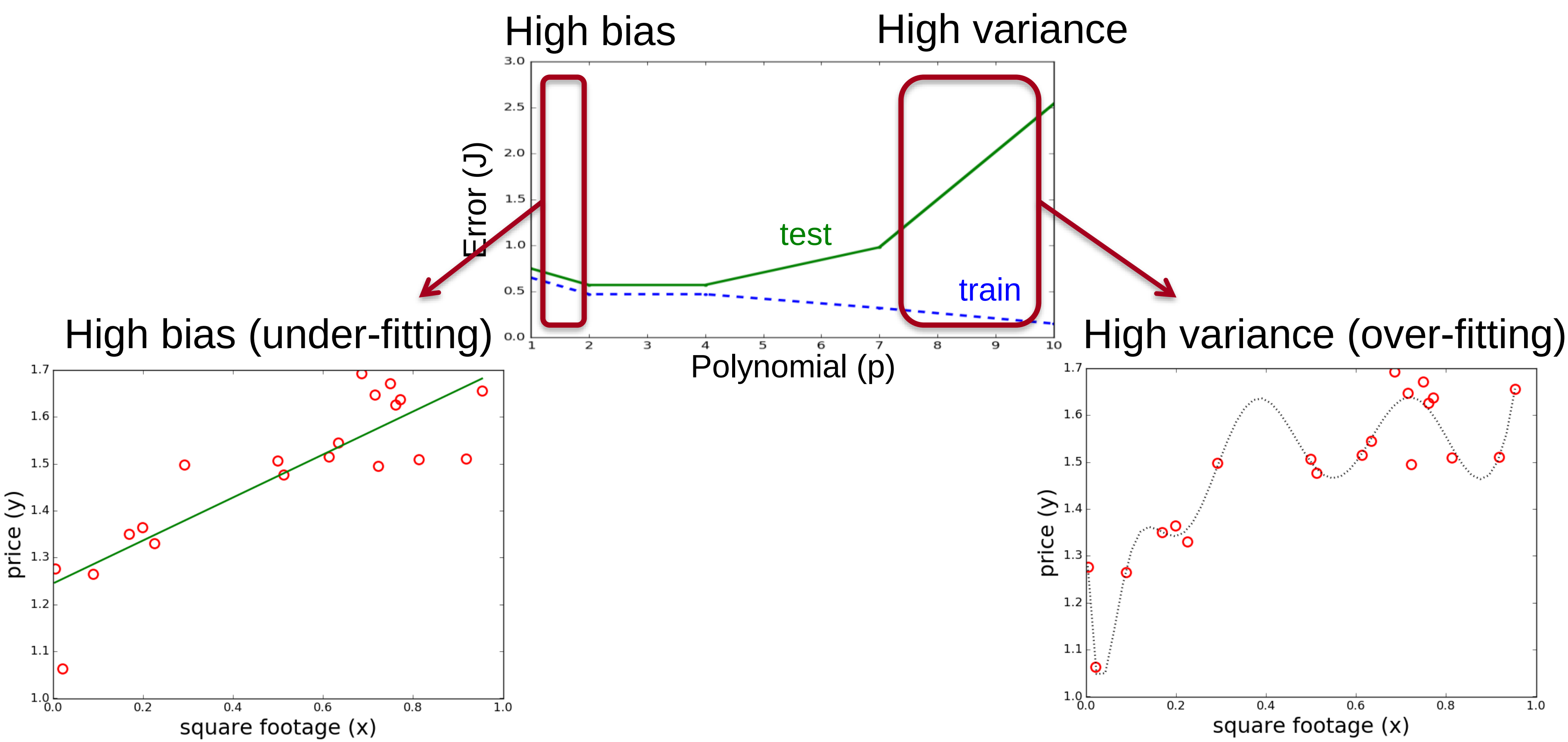


Supervised learning: Hyper-parameter choice

Bias-Variance Tradeoff

High bias ➔ Collect new attributes, create new features, more parameters

High variance ➔ Fewer features (e.g. “mutual information”), more data



Supervised learning: Hyper-parameter choice

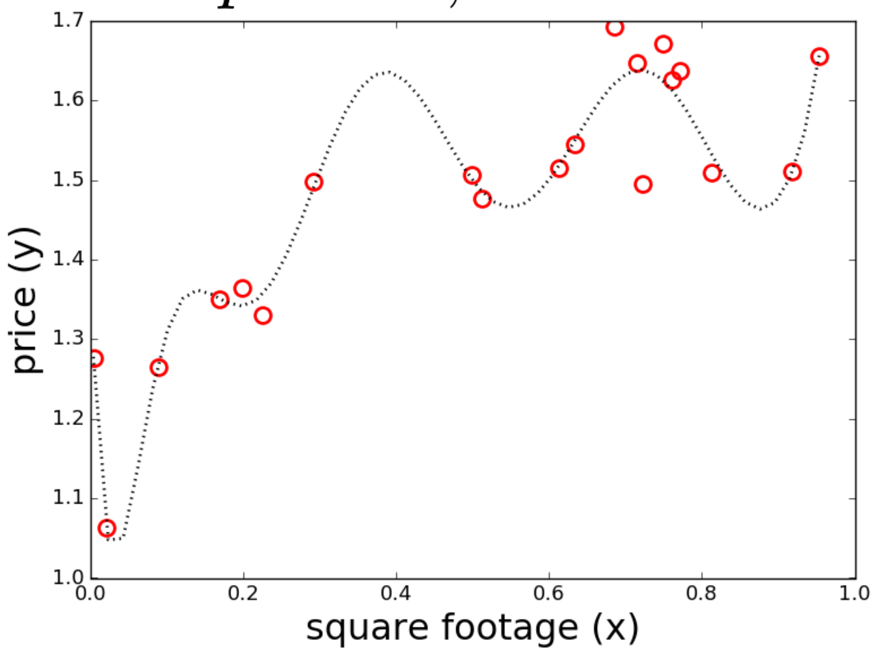
Bias-Variance Tradeoff

Regularization: modify the cost function

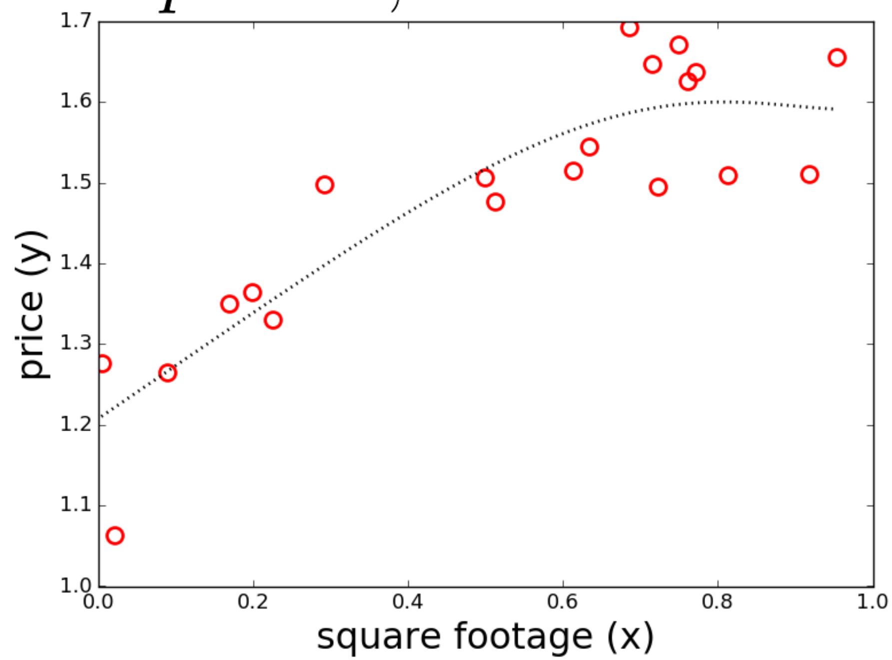
Penalizes large amplitudes of θ

$$J(\theta) = ||h_\theta(\mathbf{x}) - y||^2 + \lambda||\theta||^2$$

$p = 10, \lambda = 0$



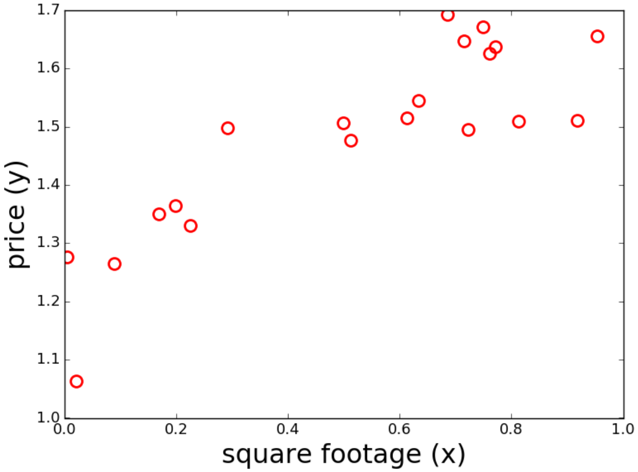
$p = 10, \lambda = 0.02$



Least Squares Regression: Probabilistic interpretation

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

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



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

"Most likely" $\theta = y^{(i)}/x^{(i)}$

Define "Likelihood":

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

SLAC

Least Squares Regression: Probabilistic interpretation

“Maximum Likelihood Estimation (MLE)”

$$\ell(\theta) = \log L(\theta)$$

“log likelihood”

$$= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$= \cancel{m \log \frac{1}{\sqrt{2\pi}\sigma}} - \frac{1}{\sigma^2} \cdot \underbrace{\frac{1}{2} \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})^2}_{\text{Least squares}}$$

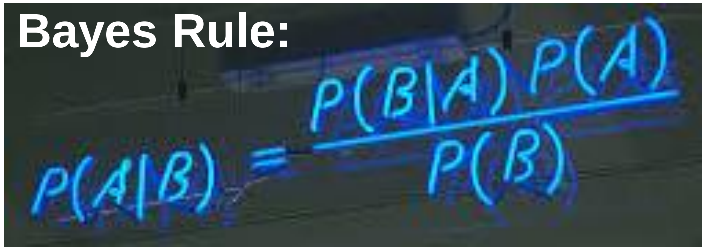
Supervised learning: Parametric models

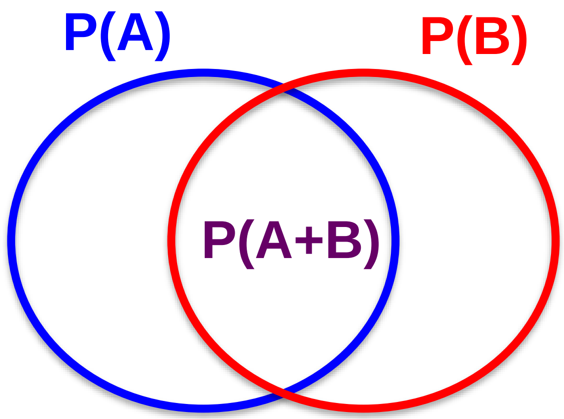
Least Squares Regression: Bayesian interpretation

	Sick (1% of pop.)	**Healthy (99% of pop.)**
Positive test	90%	10%
Negative test	10%	90%

Given positive result, what is probability of correct diagnosis?

~8%





$$P(A|B) = P(A+B)/P(B)$$

$$P(B|A) = P(A+B)/P(A)$$

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)}$$

Regularization term

"Maximum a posteriori (MAP)" $\theta_{\mathrm{MAP}} = \arg\max_{\theta} \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}, \theta)p(\theta)$

Supervised learning: Parametric models

SLAC

Logistic Regression

Classification problem: Did a house sell?

$$h_\theta(x^{(i)}) = \theta^T x^{(i)}$$

Output limited to range [0 , 1] ➔ full regression seems awkward

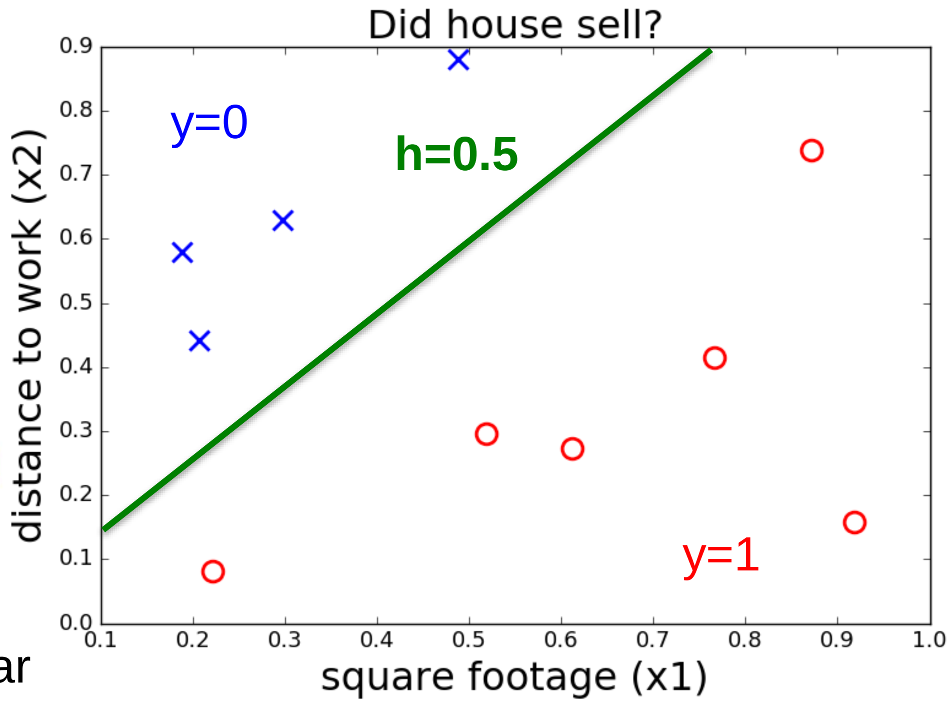
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Use MLE to derive update rule:

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

Same as OLS except now *h* is non-linear

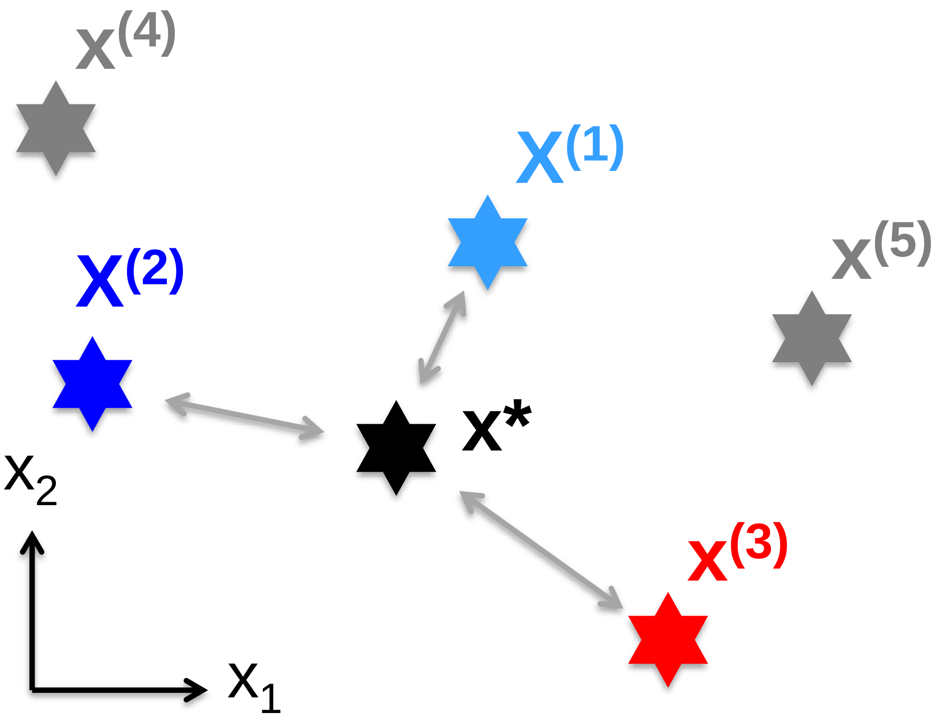


Supervised learning: Non-parametric

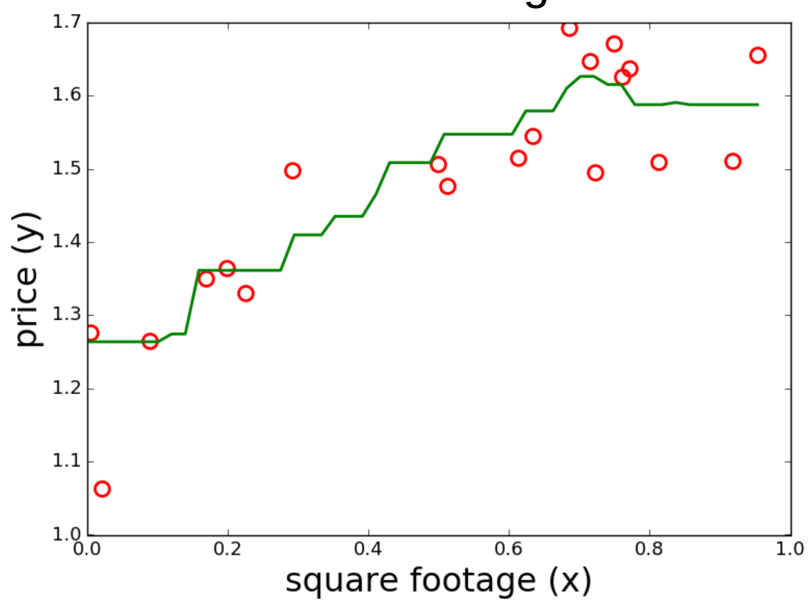
Instance-based learning

Parametric model: $h_{\theta_1,\ldots,\theta_n}(x)$

Non-parametric model: $h_{x^{(1)},\ldots,x^{(m)}}(x)$



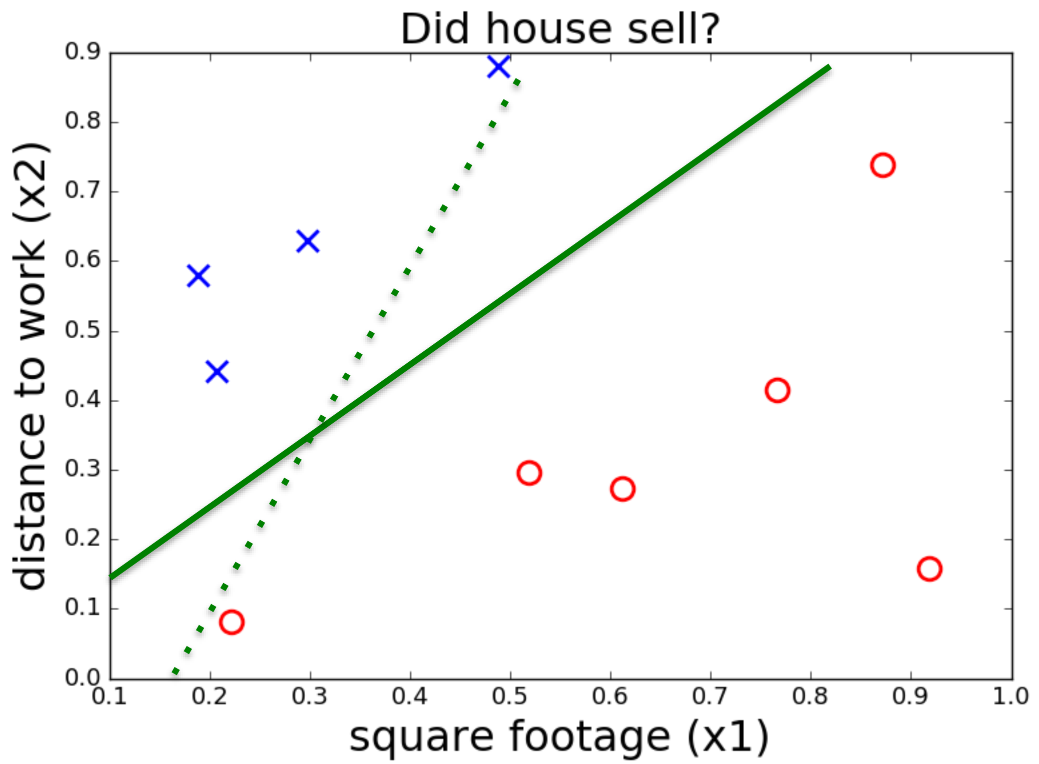
K-nearest neighbors



Optimal-margin classifier

Alternative classifier definition:
find hyperplane that divides classes $w^T x^{(i)} + b$

Optimal-margin classifier: pick line with *maximize minimum distance* from plane



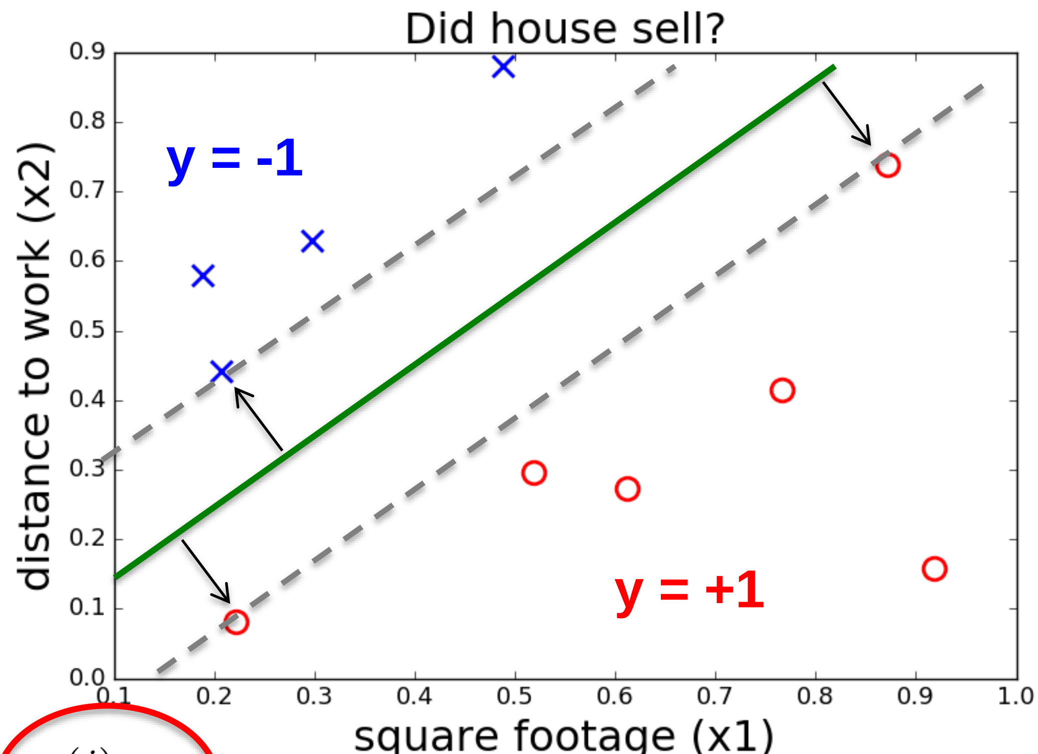
Supervised learning: Non-parametric

SLAC

Optimal-margin classifier

Alternative classifier definition:
find hyperplane that divides classes $w^T x^{(i)} + b$

Optimal-margin classifier: pick line with *maximize minimum distance* from plane



Support vector machine (SVM):

$$\min_{w,b} \quad \frac{1}{2}||w||^2$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1$$

Prediction rule:

$$h(x) = \text{sgn}(w^T x + b) = \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b$$

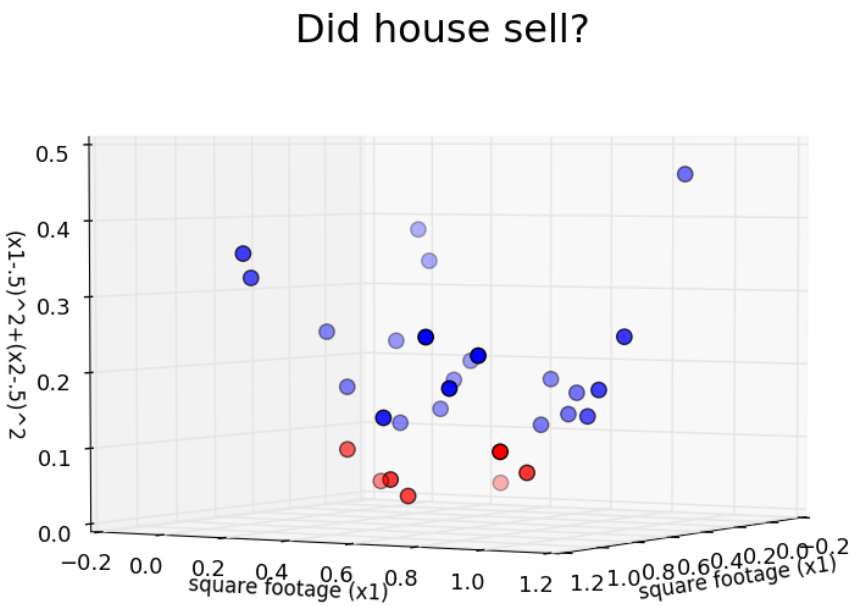
Supervised learning: Non-parametric

Support Vector Machines

What happens if classes aren't separable?

Try adding new features: e.g. $x_1^2 + x_2^2$

$$x \Rightarrow \phi(x)$$



Supervised learning: Non-parametric

SVMs and Kernels

Feature mapping: $x \Rightarrow \phi(x) \qquad \langle x, z \rangle \Rightarrow \langle \phi(x), \phi(z) \rangle$

SVM equation: $w^T x + b = \sum_{i=1}^{m} \alpha_i y^{(i)} \langle \phi(x^{(i)}), \phi(x) \rangle + b$

Define "kernel": $K(x, z) = \left\langle \phi(x), \phi(z) \right\rangle = \phi(x)^T \phi(z)$

New SVM equation: $w^T x + b = \sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b$

$$K(x, z) = (x^T z)^2 \Longleftrightarrow \phi(x) = \sum_{i,j=1}^{n} x_i x_j$$

$O(n)$ $\qquad$ $O(n^2)$

Supervised learning: Non-parametric

SVMs and Kernels

Feature mapping: $x \Rightarrow \phi(x)$ $\quad \langle x, z \rangle \Rightarrow \langle \phi(x), \phi(z) \rangle$

SVM equation: $$w^T x + b = \sum_{i=1}^{m} \alpha_i y^{(i)} \langle \phi(x^{(i)}), \phi(x) \rangle + b$$

Define "kernel": $$K(x, z) = \big\langle \phi(x), \phi(z) \big\rangle = \phi(x)^T \phi(z)$$

New SVM equation: $$w^T x + b = \sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b$$

$$K(x, z) = e^{-||x-z||^2/2\sigma^2} \Longleftrightarrow \phi(x)$$

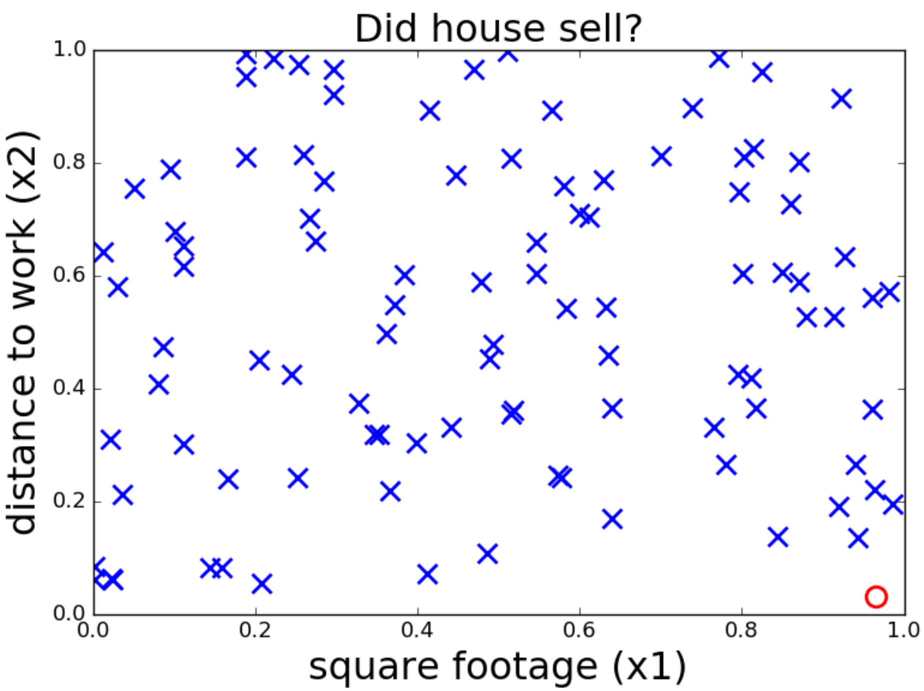
Mercer's theorem: K(x,z) is kernel iff symmetric, positive, semi-definite

$O(n)$ **"Kernel trick"** $O(\infty)$

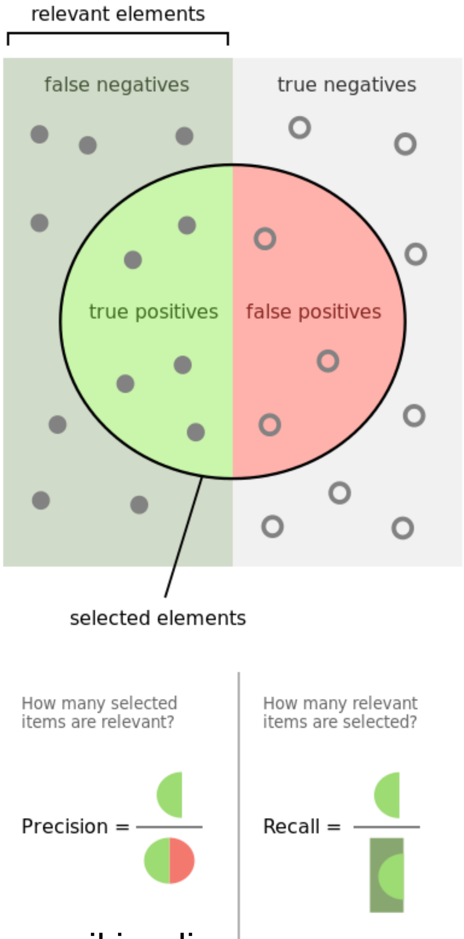
Presenting Classification Results

How do I report how well my model works?

$$h(x) = 0$$ 99% accurate!



Precision-Recall

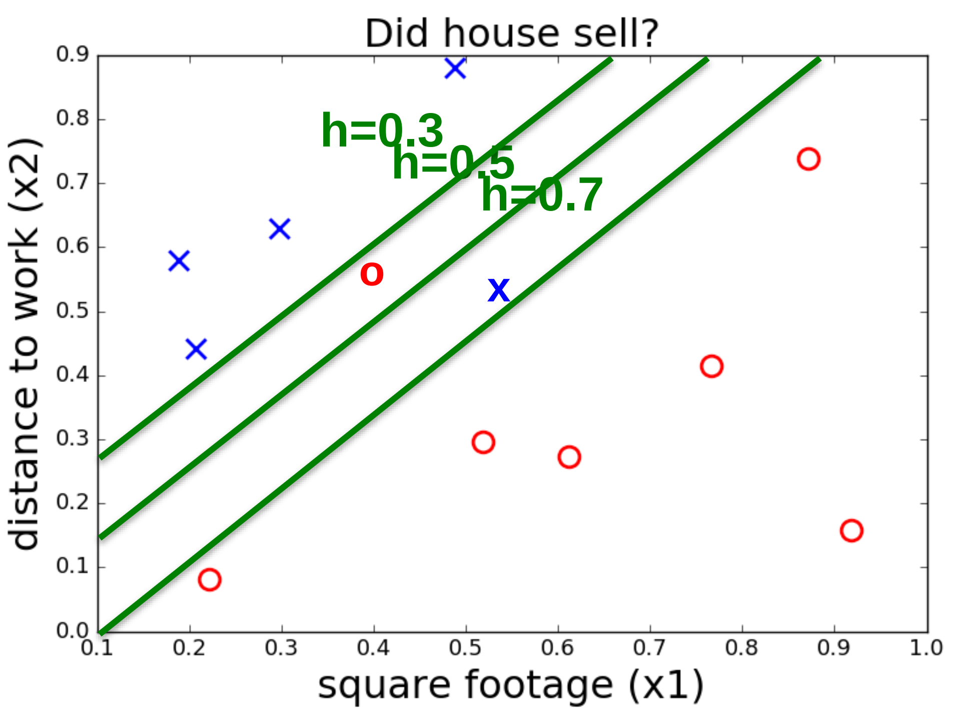


wikipedia

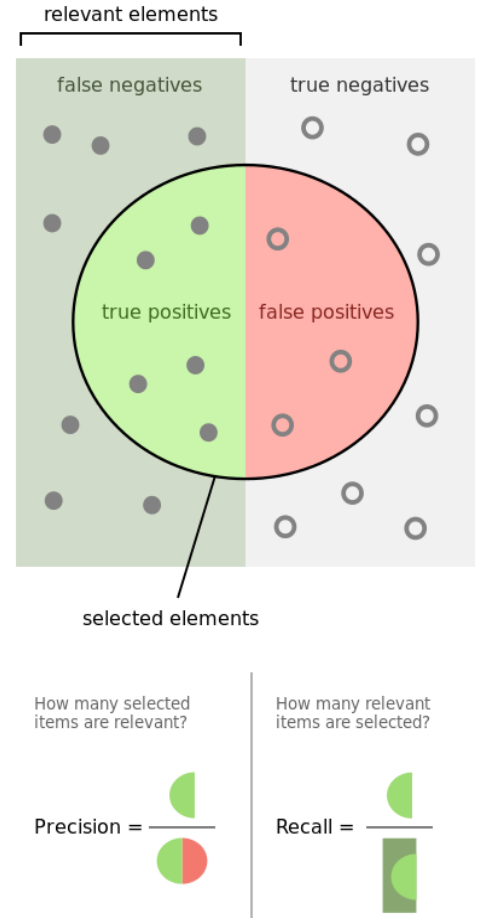
Presenting Classification Results

How do I report how well my model works?

How do I pick the threshold for classification?



Precision-Recall

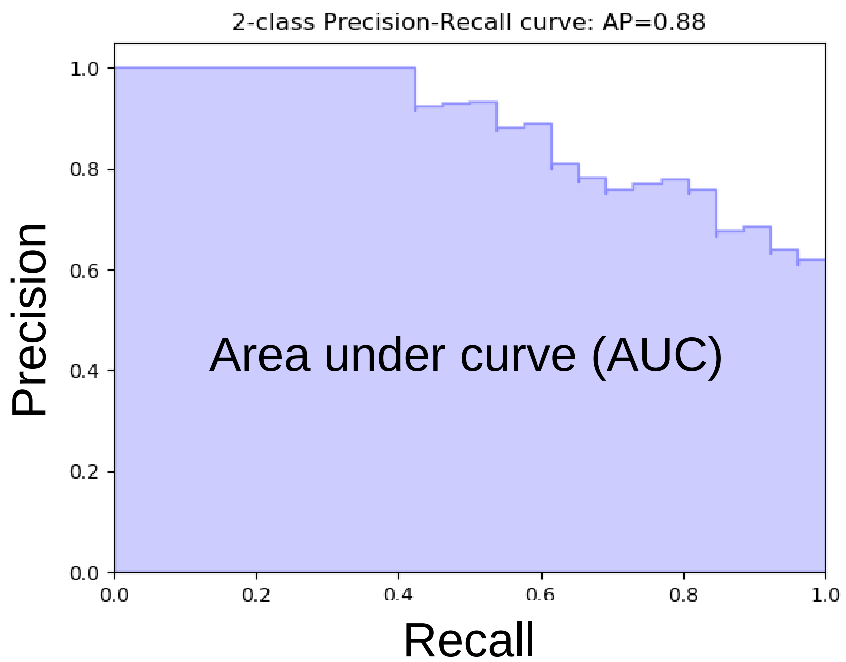


wikipedia

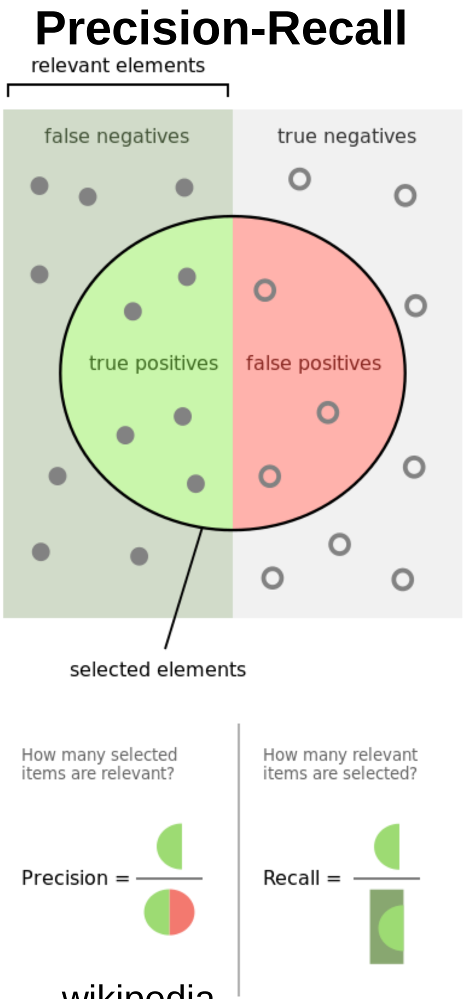
Presenting Classification Results

How do I report how well my model works?

How do I pick the threshold for classification?



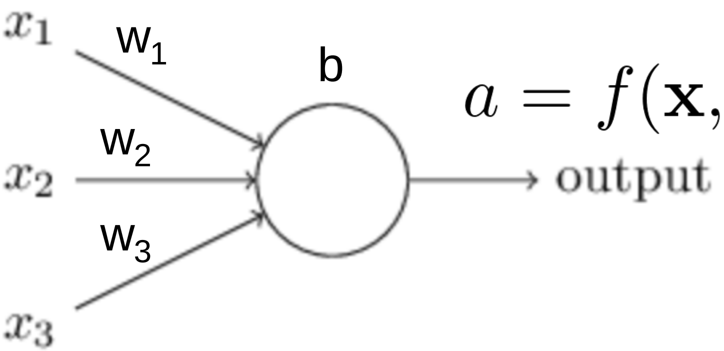
scikit-learn



wikipedia

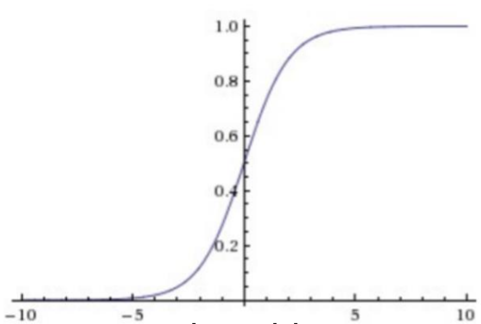
Supervised learning: Parametric models

The Perceptron

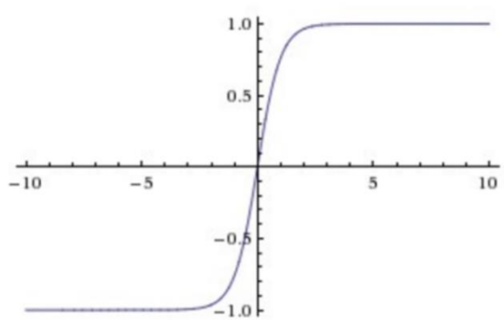


$$a = f(\mathbf{x}, \mathbf{w}, b)$$

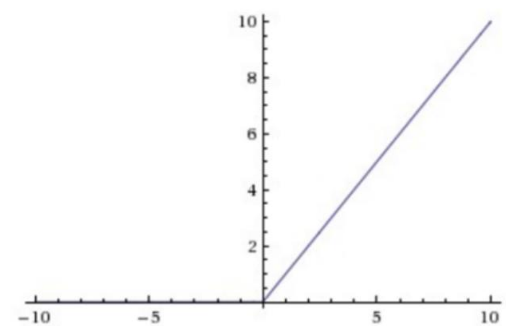
$$f(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{x} - b}}$$



Sigmoid



Tanh



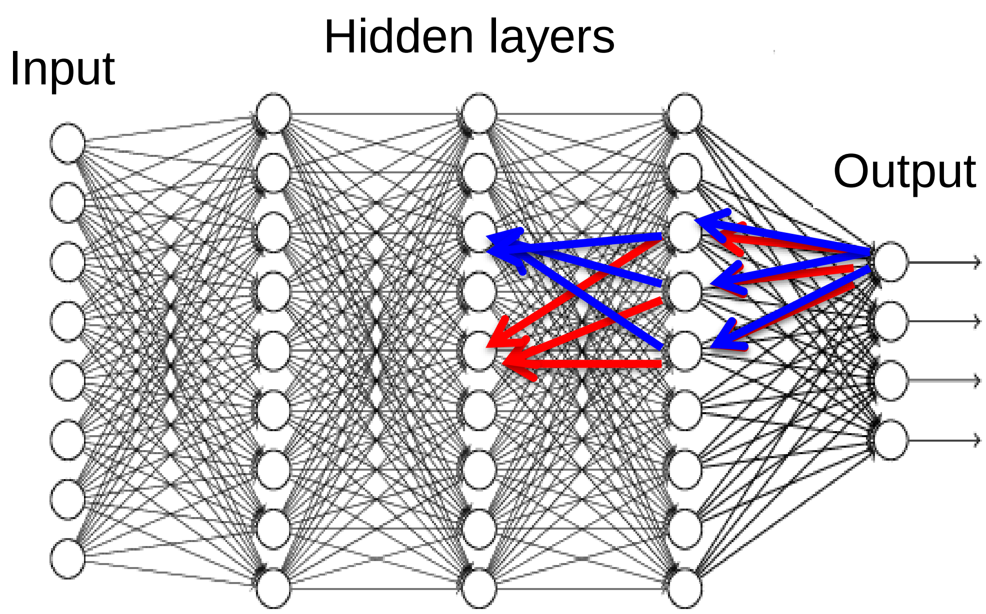
ReLU

Michael Nielsen, Neural Networks and Deep Learning, Determination Press (2015)

Supervised learning: Parametric models

SLAC

Artificial Neural Networks



Cost function, e.g. MSE

$$C_{\mathbf{w},b} = ||a_{\mathbf{w},b} - y||^2$$

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} C_{\mathbf{w},b}$$

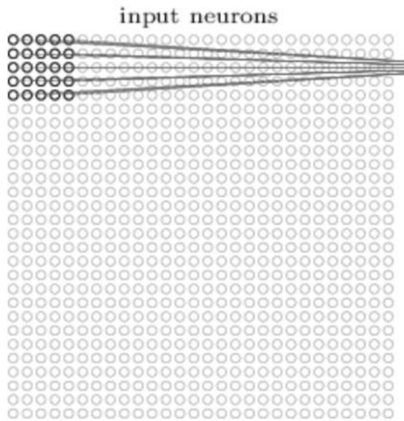
$$\frac{\partial C}{\partial w_j} \approx \frac{C(w + \epsilon e_j) - C(w)}{\epsilon}$$

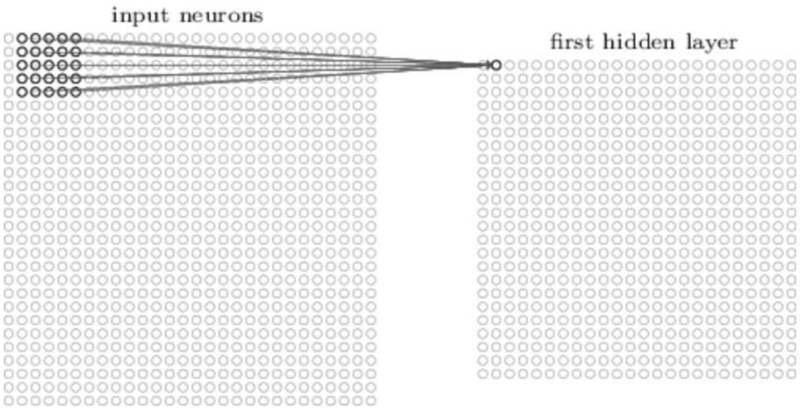
Problem: $O(n^2)$

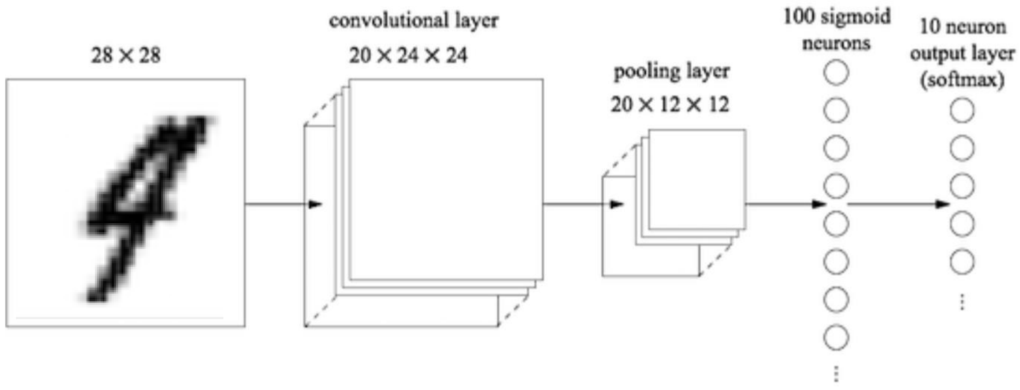
Clever idea to the rescue: Use the chain rule!
➔ Backpropagation

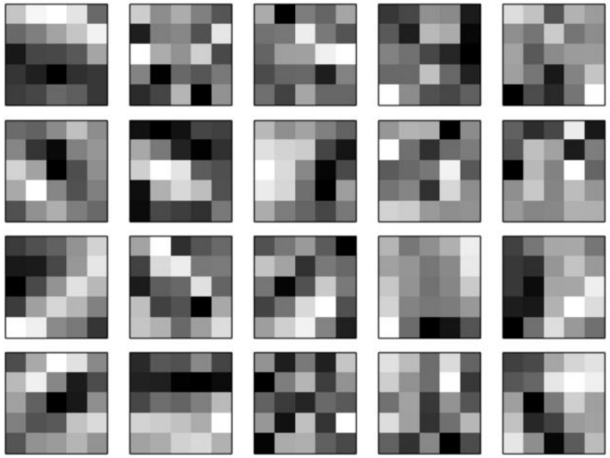
Michael Nielsen, Neural Networks and Deep Learning, Determination Press (2015)

Convolutional Neural Networks



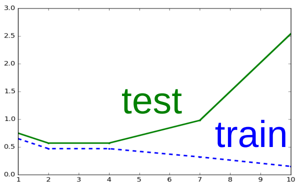




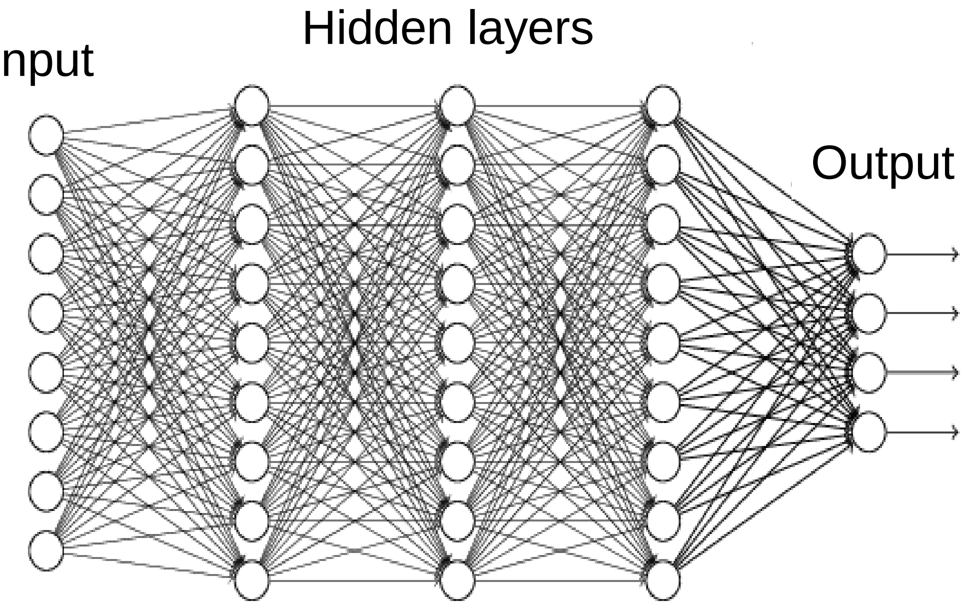


Michael Nielsen, Neural Networks and Deep Learning, Determination Press (2015)

ANNs practical tips



1. Training is slow ➔ use GPUs
2. Large models can have millions of parameters, prone to over-fitting
   ➔ Use regularization, drop-out, noise-layers, lots of data
3. Always plot training AND validation loss ➔ shows bias vs. variance
4. Not training? ➔ Try different loss functions, activations, architectures, mini-batch parameters, optimization algorithms, learning rates, data quality …



Unsupervised learning

What can be accomplished without labels?

Supervised learning: **X, y**

Unsupervised learning: **X**

What can we hope to accomplish?

1. Clustering (classification)
2. Decomposition (e.g. separating audio signals)
3. Anomaly/breakout detection (e.g. fault detection/prediction)
4. Generation (e.g. creating new examples within a class)

What can be accomplished without labels?

Clustering: Divide **x** into **k** categories

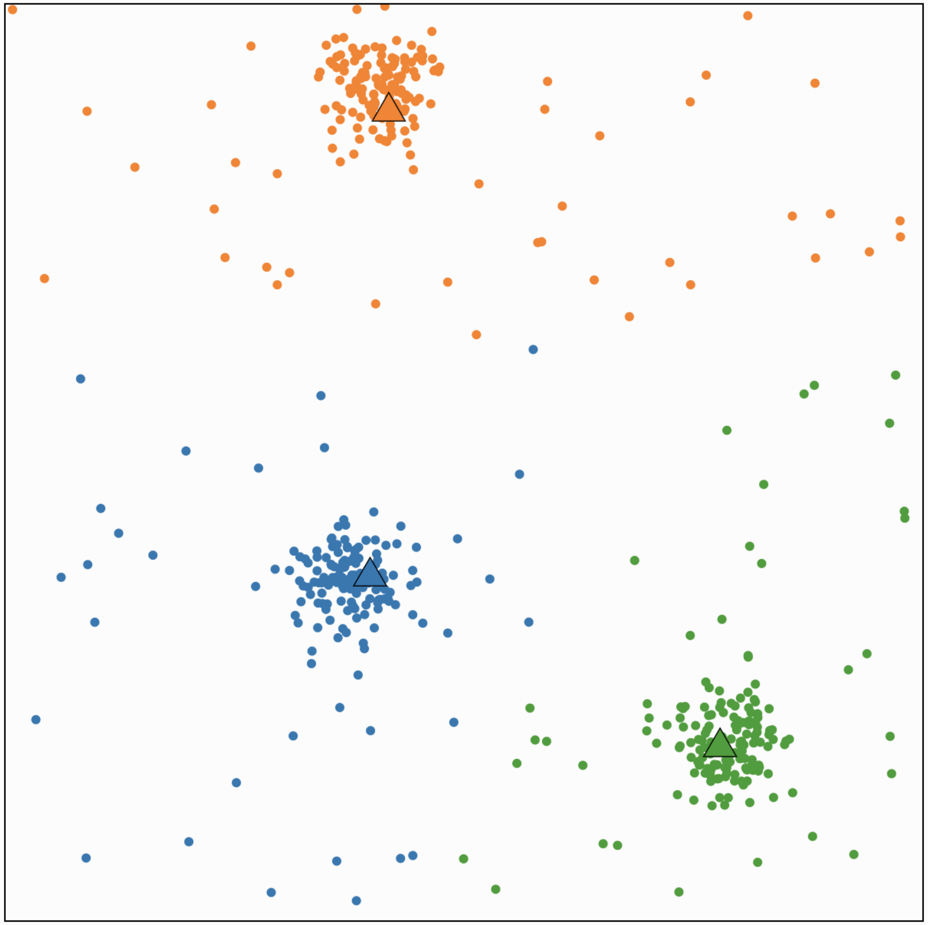
K-means algorithm:

a. Pick 'k' random centroids
b. Loop until convergence {
    1. Assign examples to nearest centroid
    2. Update centroids to mean of clusters

    }

See also: **Hierarchical clustering, DBSCAN, etc…**

K-means



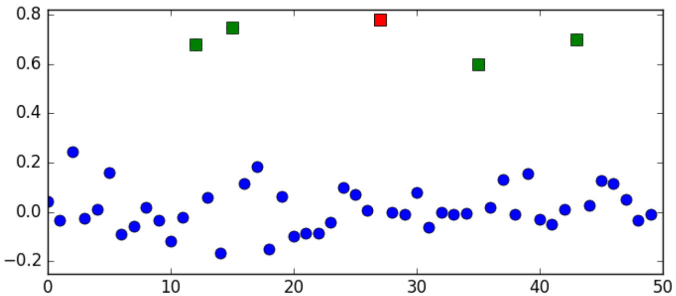
http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Unsupervised learning

SLAC

Time series data: Anomaly/Breakout/Changepoint Detection

Anomaly detection:
identify points that are statistical outliers from a distribution

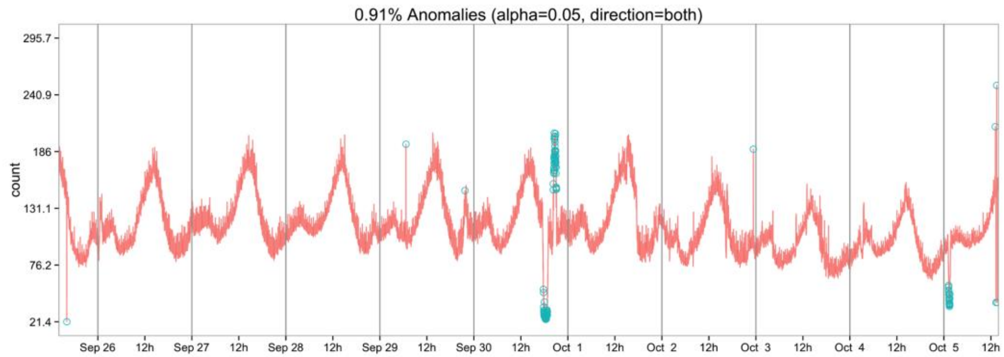


PyAstronomy: Generalized ESD (GESD)
(Available from pip install)

Breakout/Changepoint detection:
Find point in time at which distribution changed

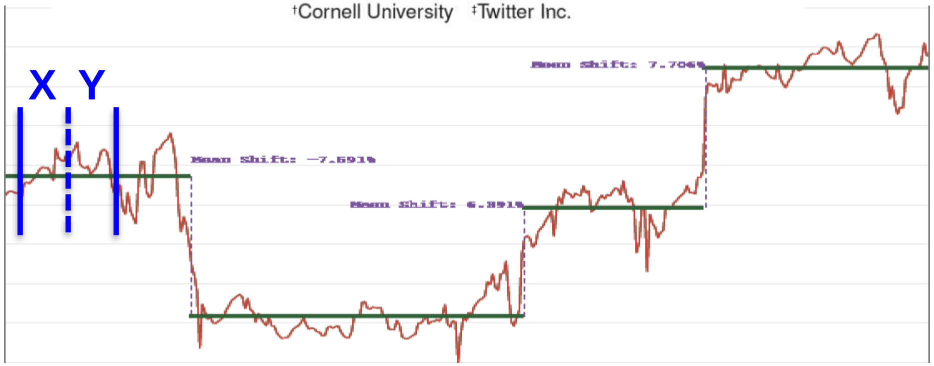
$$\mathcal{E}(X,Y) = 2E|X-Y| - E|X-X'| - E|Y-Y'|$$

twitter / **AnomalyDetection**



Leveraging Cloud Data to Mitigate User Experience from 'Breaking Bad'

Nicholas A. James[†] Arun Kejariwal[‡] David S. Matteson[†]
[†]Cornell University [‡]Twitter Inc.



Unsupervised learning

Generating new data

Unsupervised learning with neural networks:
train a model to generate new examples based on training set

Deep dreaming of dogs

If you train a network to recognize dogs…

…it will hallucinate dogs

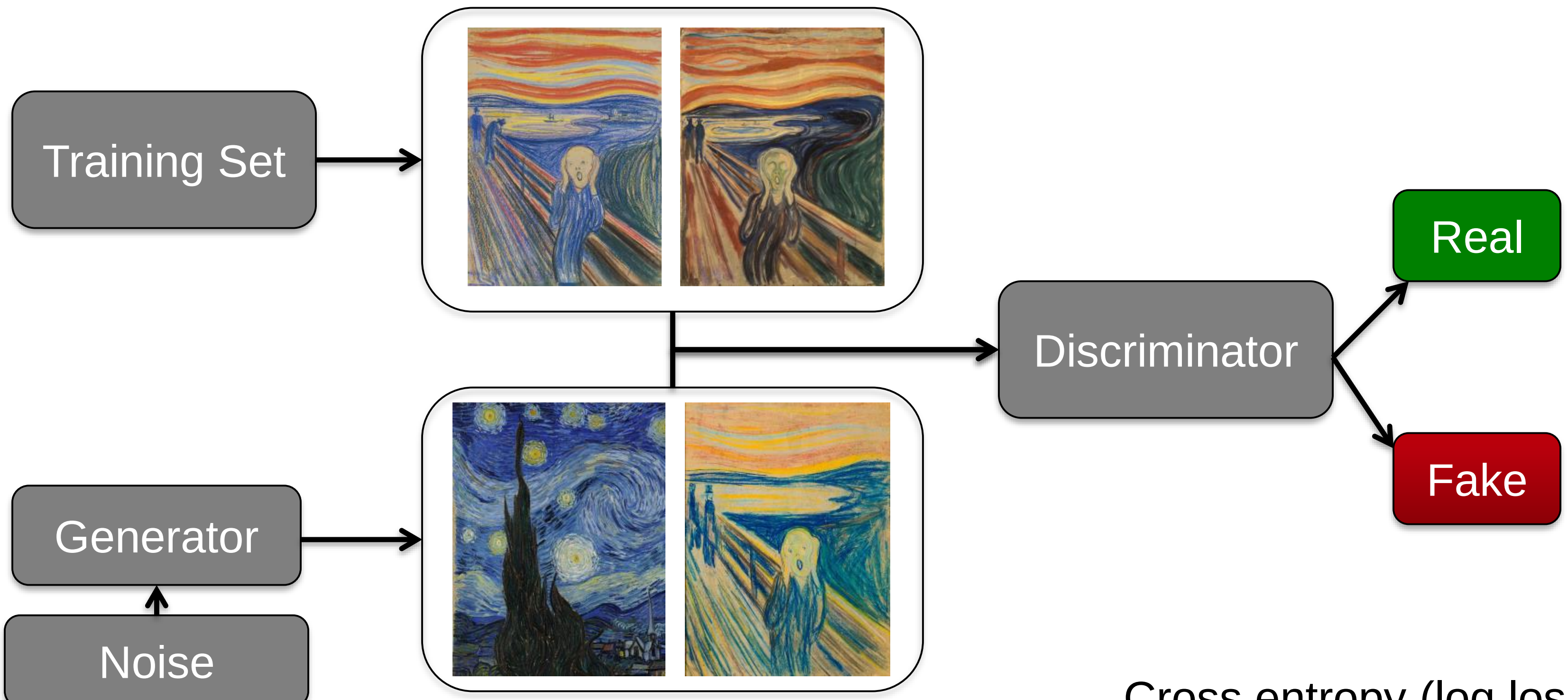
Style transfer

Gatys, et al.

Unsupervised learning

Generating new data

Generative Adversarial Network (GAN)



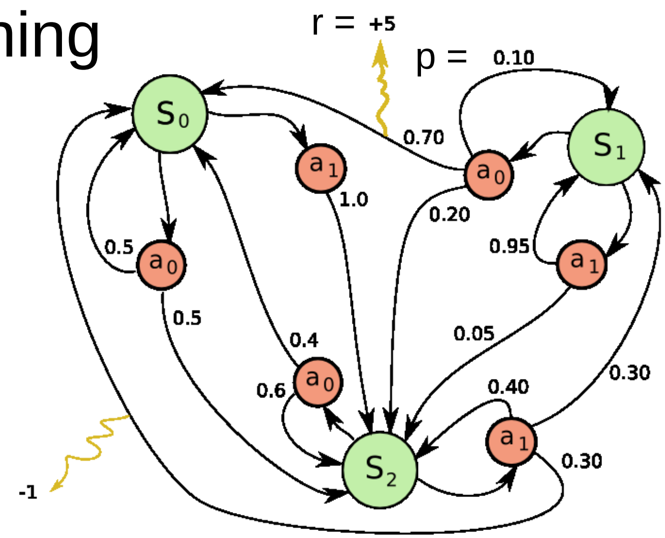
Cross entropy (log loss)

$$J^{(G)} = -J^{(D)}$$

$$J^{(D)} = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

Partial supervision

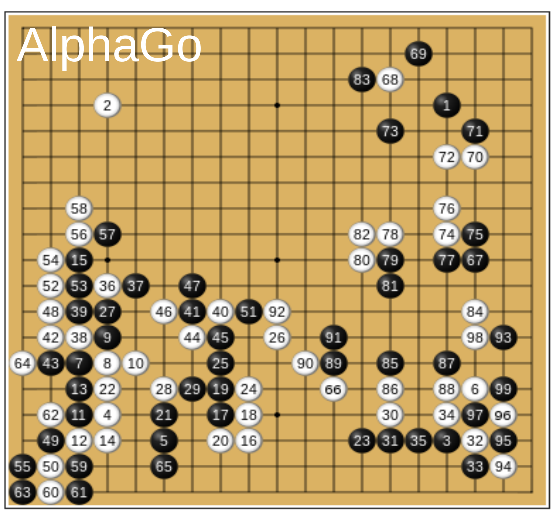
Reinforcement Learning

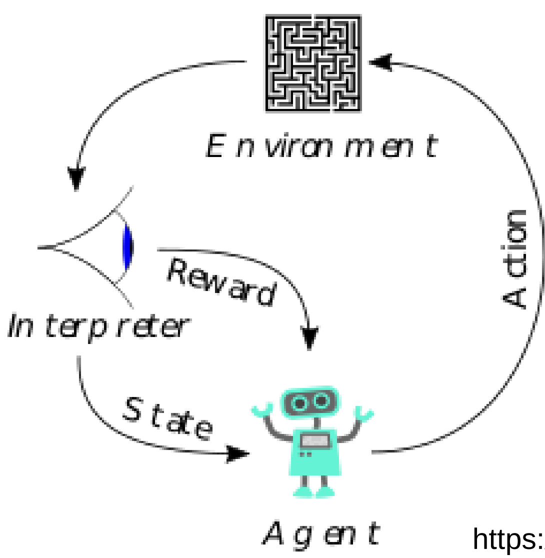


Third category: partial supervision

e.g. when playing a game, will not have a known label for every position

Goal is to find "policy":
optimal action a_s, given state s





Actions: a
States: s
Transition probability: p
Rewards: r

https://en.wikipedia.org/wiki/Reinforcement_learning