Science at I.S.T. Austria
Machine Learning and Computer Vision

Christoph Lampert

I.S.T. Austria

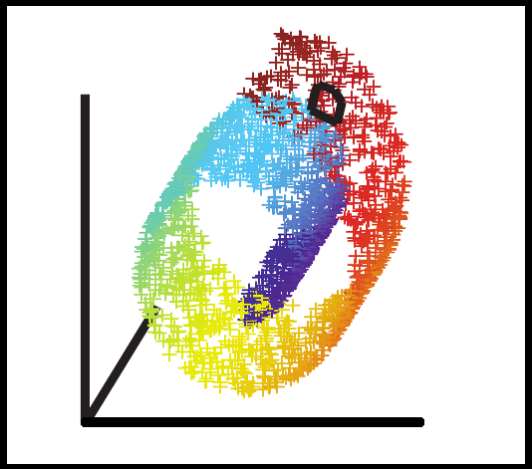
October 18th, 2010

Machine Learning

$=$

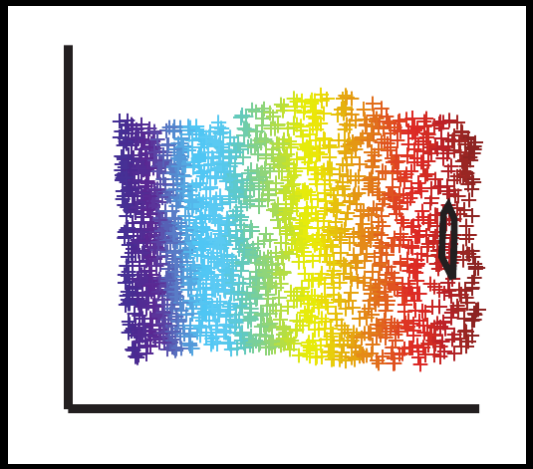
The science of automatic systems that draw conclusions from empirical data.

Dimensionality Reduction: Detect Regularities in Data



measurements

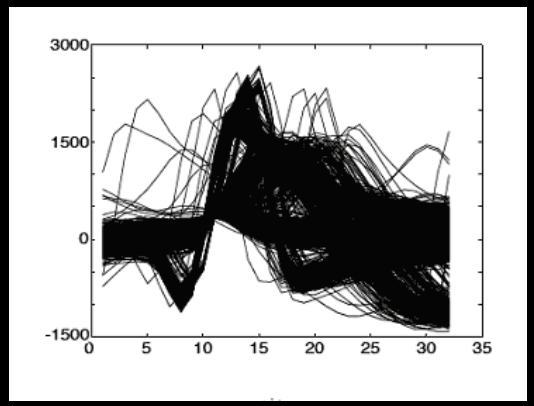
PCA/LDA/...



lower-dimensional representation

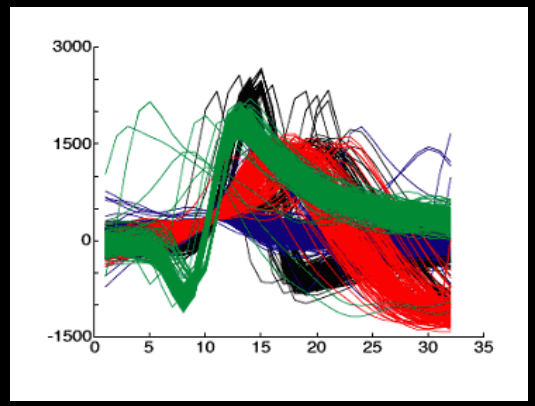
image source: [Roweis and Lawrence, *Science* 2000]

Cluster Analysis: Detect Subgroups in Data



spike measurements

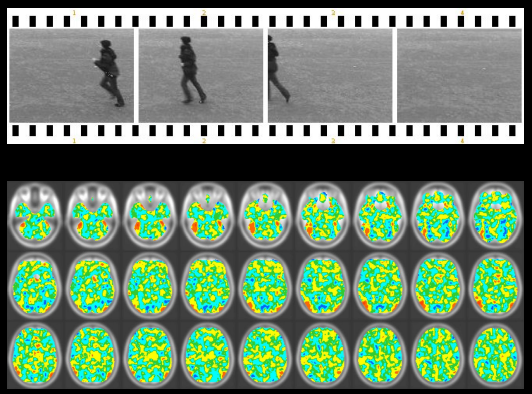
Cluster Analysis

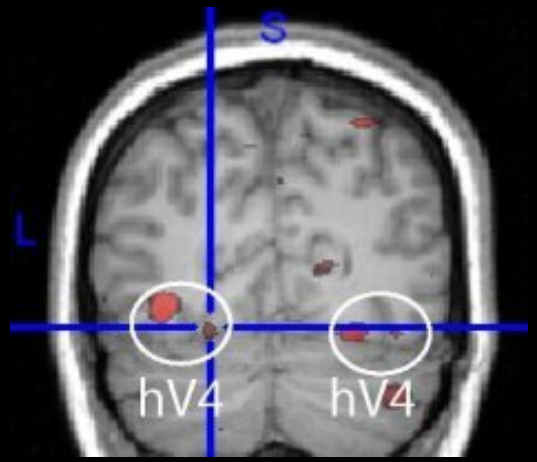


subgroups

image source: [Gibson, Markovic, *The Syndicated* 2007]

Correlation Analysis: Suppress High-Variance Noise





stimulus/response Correlation Analysis relevant signal

image source: [Hardoon et al., ICONIP 2007], [Blaschko et al., Neurocomputing 2009]

Supervised Learning: *Teaching* a computer to solve a task

Phase 1: Training

training data →  ← correct output

model

Phase 2: Prediction

new data →  → predicted output

trained model

Supervised Learning: Regression

Phase 1: Training

passport photos

regression model

19	48	62

age at that time

Phase 2: Prediction

new photo

trained model

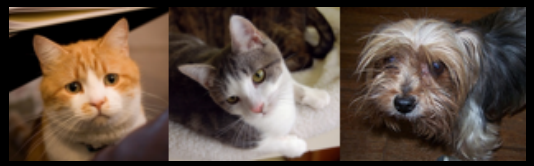
30

predicted age

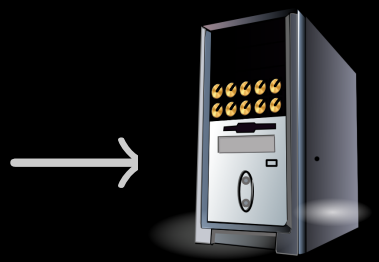
image source: [FG-Net Aging Database]

Supervised Learning: Classification

Phase 1: Training



animal photos



classification model

cat	cat	dog

correct label

Phase 2: Prediction

new photo

trained model

cat

predicted label

image source: [Everingham et al. 2008]

Machine Learning Research

=

Design and analysis of models and algorithms
that allow computers to learn from data.

Perceptron (1950s–)

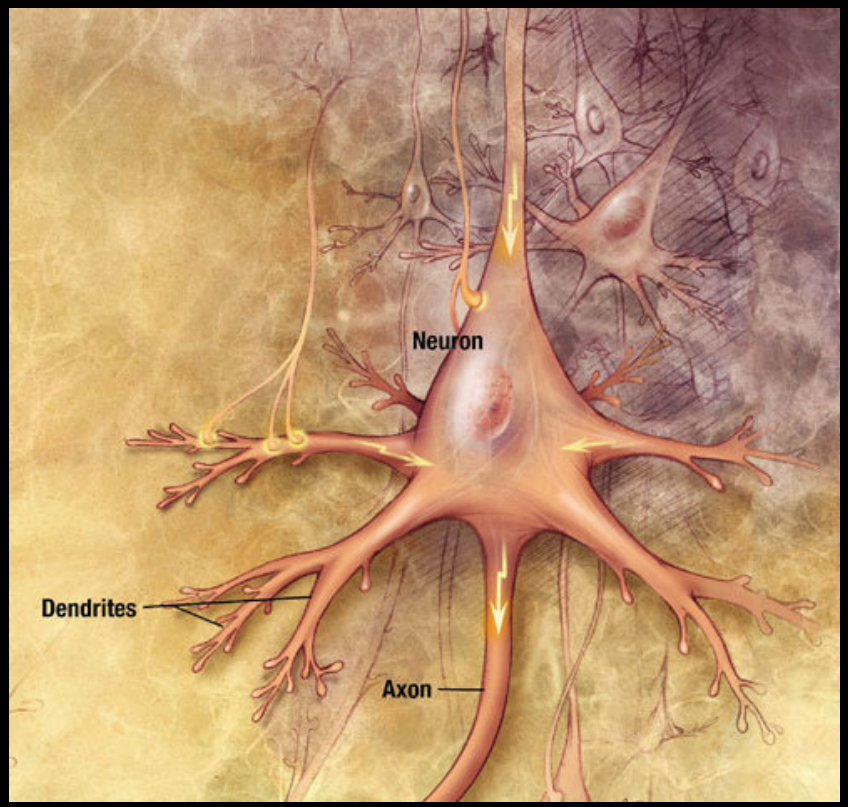
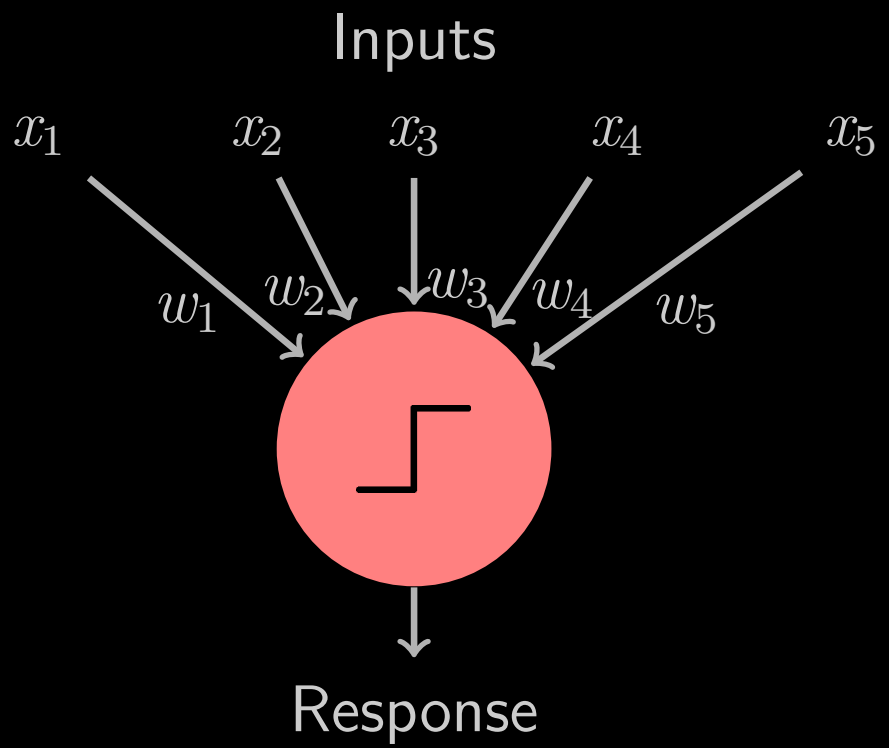


Image source: U.S. National Institutes of Health



"artificial neuron": limited capacity, but easy to train

[McCulloch, Pitts. *Bulletin of Mathematical Biophysics*, 1943], [Rosenblatt, *Psychological Review*, 1958]

Multi-Layer Perceptron (1970s–)

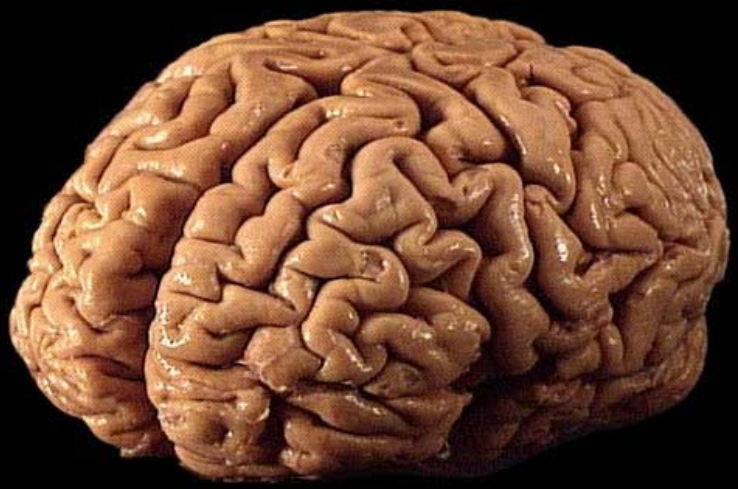
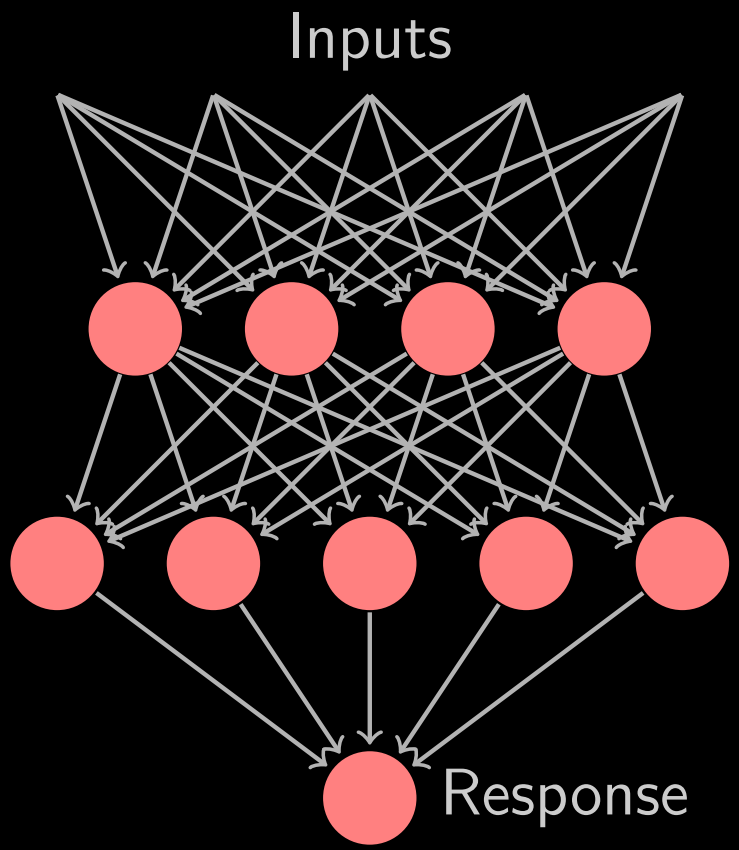


Image source: heppell.net



“artificial neural network”: high capacity, but difficult to train

[Werbos, *Harvard*, 1974], [Rumelhart, Hinton, Williams, *Nature*, 1986]

Kernel Methods (1990s–)

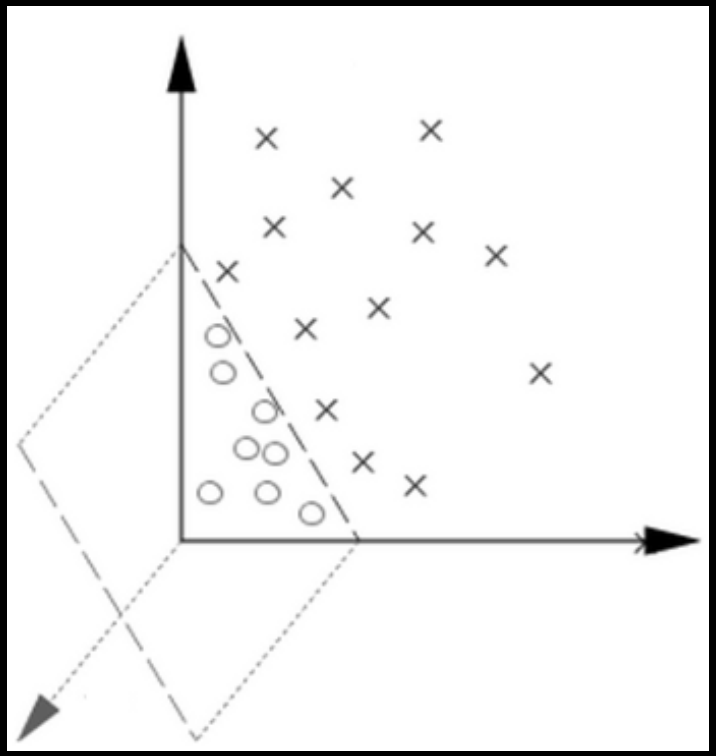
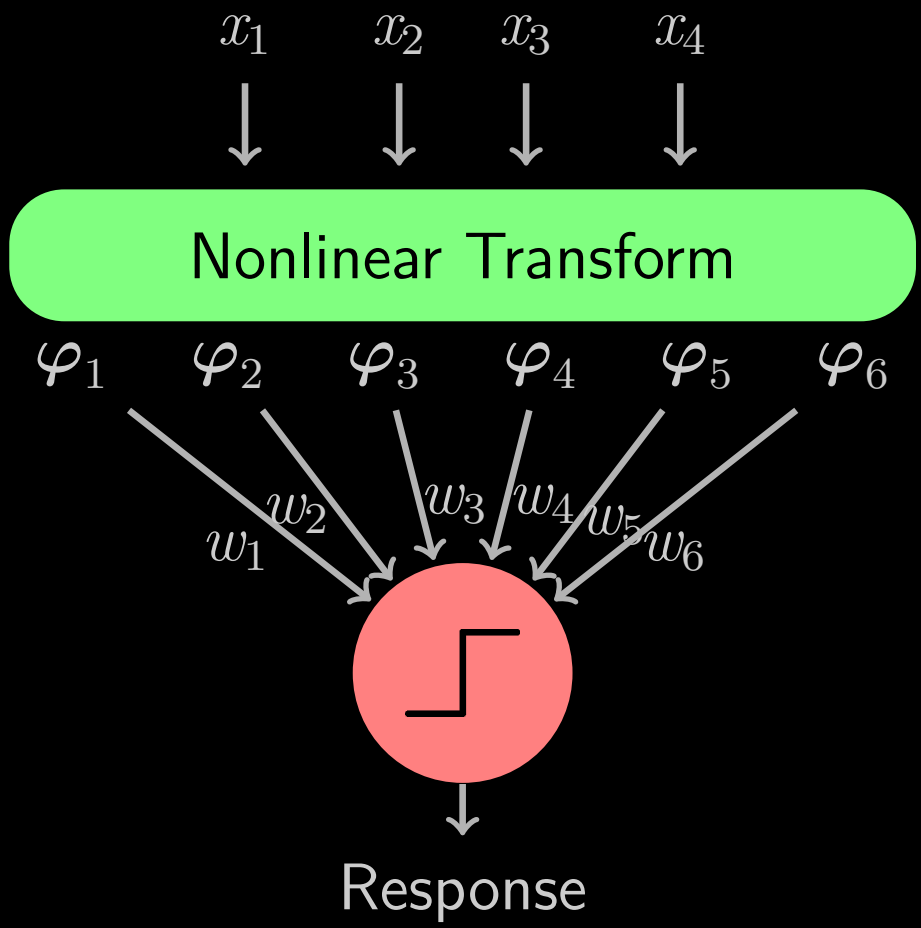


Image source: [Schölkopf, Smola, 2003]

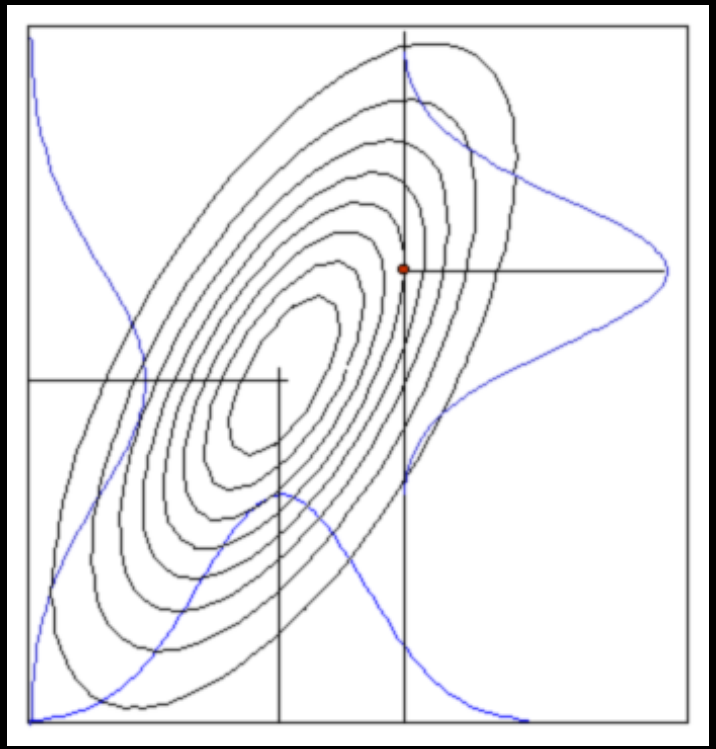


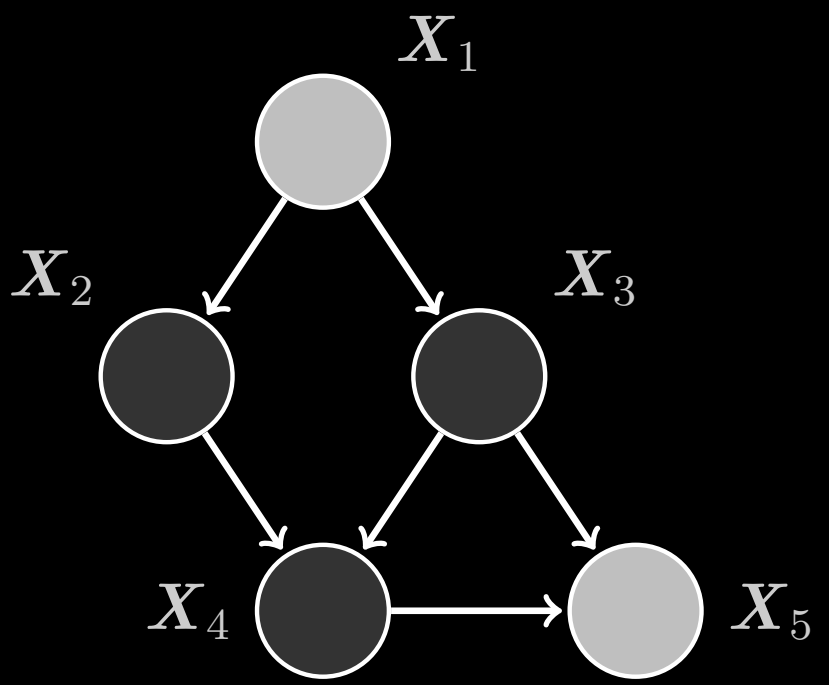
"support vector machine": high capacity, easy to train

[Boser, Guyon, Vapnik: *Computational Learning Theory*, 1992], [Cortes, Vapnik: *Machine Learning*, 1995]

Probabilistic Graphical Models (1990s–)

Image source: [Annis, StatisticalEngineering.com]





"graphical models": flexible and interpretable,
can become computationally expensive

[Pearl, 1988], [Jordan, MIT Press, 1999]

Analysis of Learning Algorithms: Statistical Learning Theory

- Total probability of making a wrong prediction

Analysis of Learning Algorithms: Statistical Learning Theory

- Total probability of making a wrong prediction
- Probablity theory to deal with uncertainty

$$p(\boldsymbol{E}_{total}) \leq p(\boldsymbol{E}_{Bayes}) + p(\boldsymbol{E}_{Model}) + p(\boldsymbol{E}_{Data}) + p(\boldsymbol{E}_{Algorithm})$$

Analysis of Learning Algorithms: Statistical Learning Theory

- Total probability of making a wrong prediction
- Probablity theory to deal with uncertainty

$$p(\boldsymbol{E}_{total}) \leq p(\boldsymbol{E}_{Bayes}) + p(\boldsymbol{E}_{Model}) + p(\boldsymbol{E}_{Data}) + p(\boldsymbol{E}_{Algorithm})$$

- Bayes error: problem-inherent uncertainty

Analysis of Learning Algorithms: Statistical Learning Theory

- Total probability of making a wrong prediction
- Probablity theory to deal with uncertainty

$$p(E_{total}) \leq p(E_{Bayes}) + p(E_{Model}) + p(E_{Data}) + p(E_{Algorithm})$$

- Bayes error: problem-inherent uncertainty
- Model error: limited model complexity

Analysis of Learning Algorithms: Statistical Learning Theory

- Total probability of making a wrong prediction
- Probablity theory to deal with uncertainty

$$p(E_{total}) \leq p(E_{Bayes}) + p(E_{Model}) + p(E_{Data}) + p(E_{Algorithm})$$

- Bayes error: problem-inherent uncertainty
- Model error: limited model complexity
- Data error: limited amount of training data

Analysis of Learning Algorithms: Statistical Learning Theory

- Total probability of making a wrong prediction
- Probablity theory to deal with uncertainty

$$p(E_{total}) \leq p(E_{Bayes}) + p(E_{Model}) + p(E_{Data}) + p(E_{Algorithm})$$

- Bayes error: problem-inherent uncertainty
- Model error: limited model complexity
- Data error: limited amount of training data
- Learning error: imperfect training algorithm

Perceptron (1950s–)

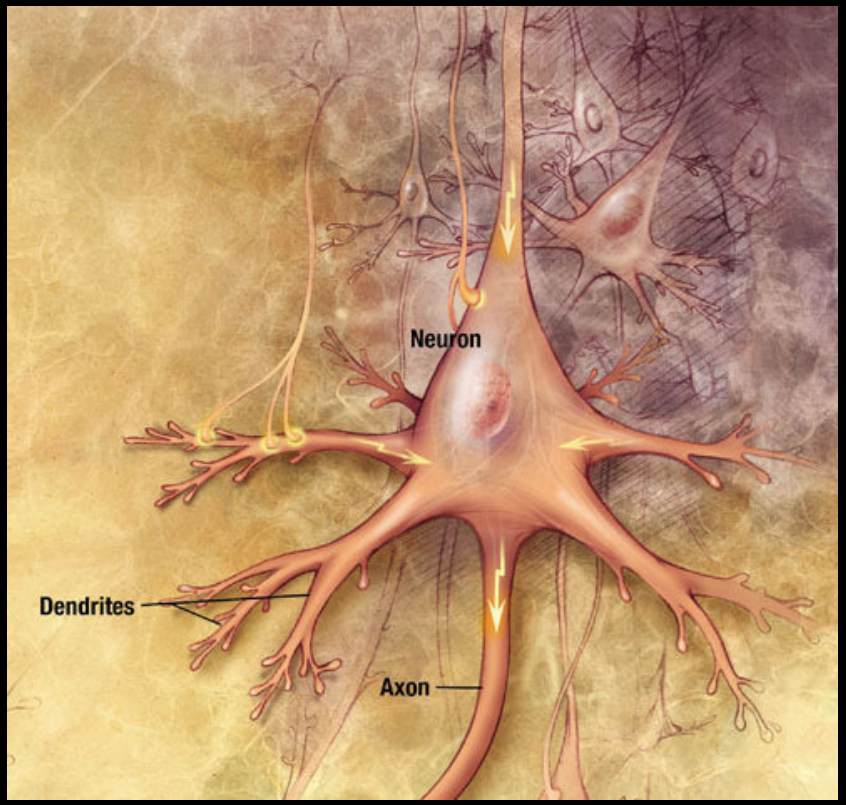
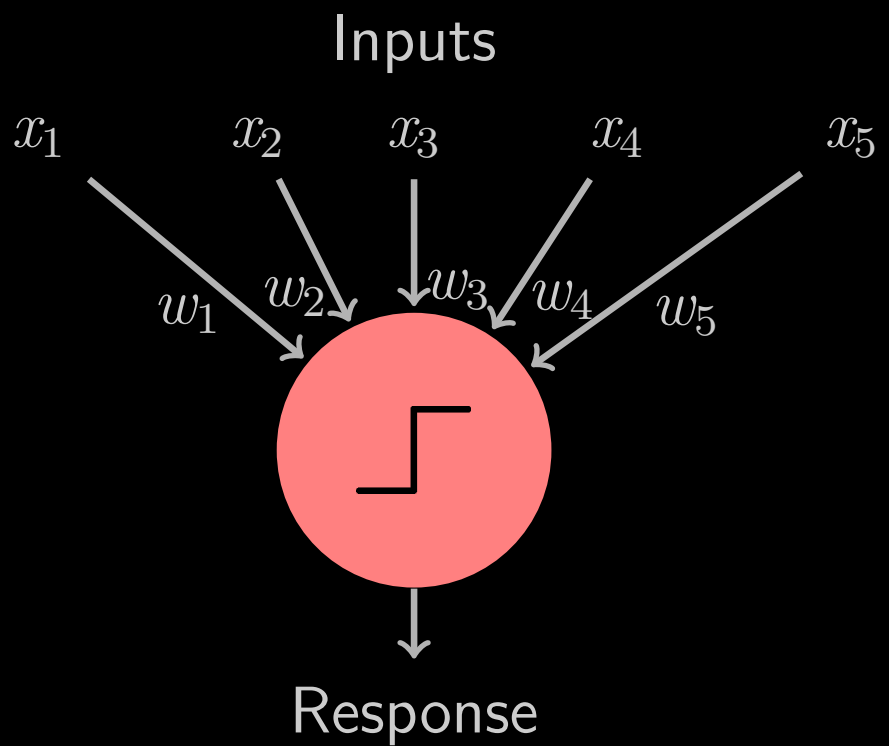


Image source: U.S. National Institutes of Health



$$p(E_{total}) \leq p(E_{Bayes}) + \underbrace{p(E_{Model})}_{\text{large}\odot} + p(E_{Data}) + p(E_{Algorithm})$$

Multi-Layer Perceptron (1970s–)

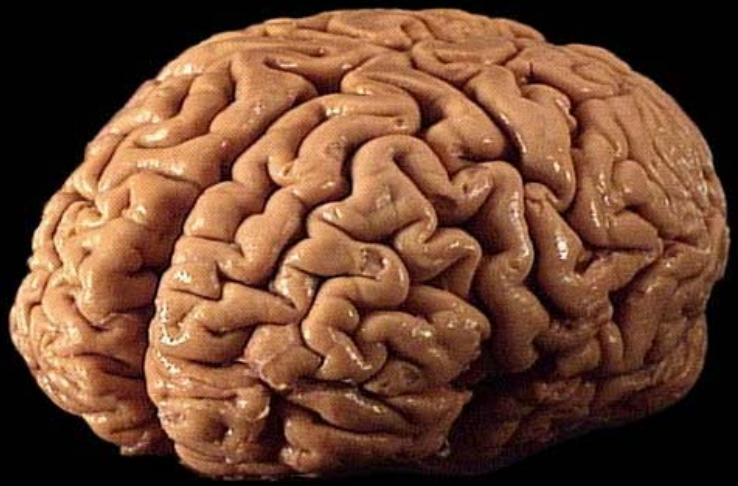
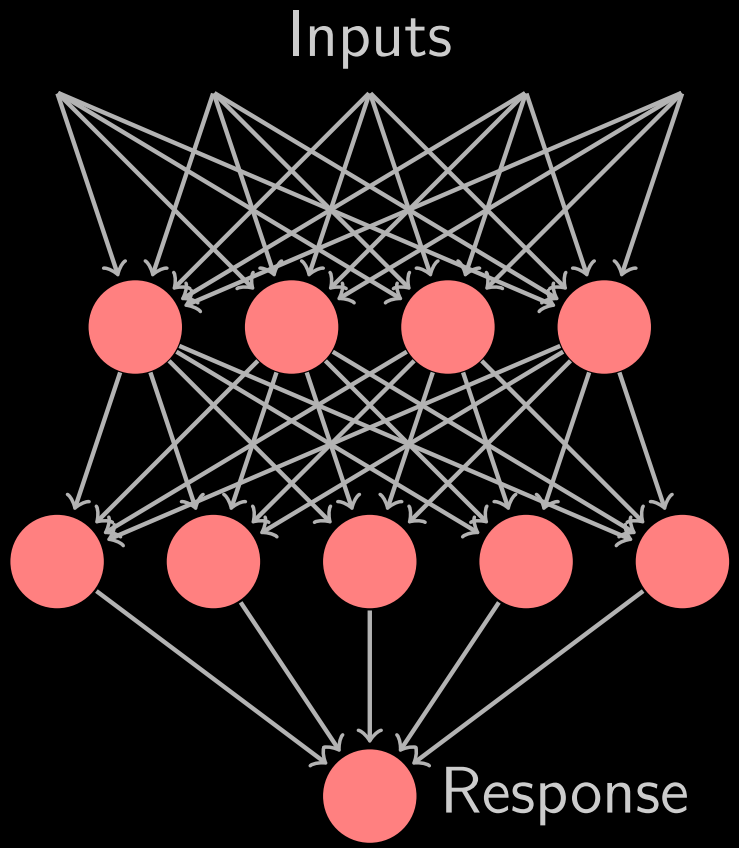


Image source: heppell.net



$$p(E_{total}) \leq p(E_{Bayes}) + \underbrace{p(E_{Model})}_{\text{small}\odot} + p(E_{Data}) + \underbrace{p(E_{Algorithm})}_{\text{large}\odot}$$

Kernel Methods (1990s–)

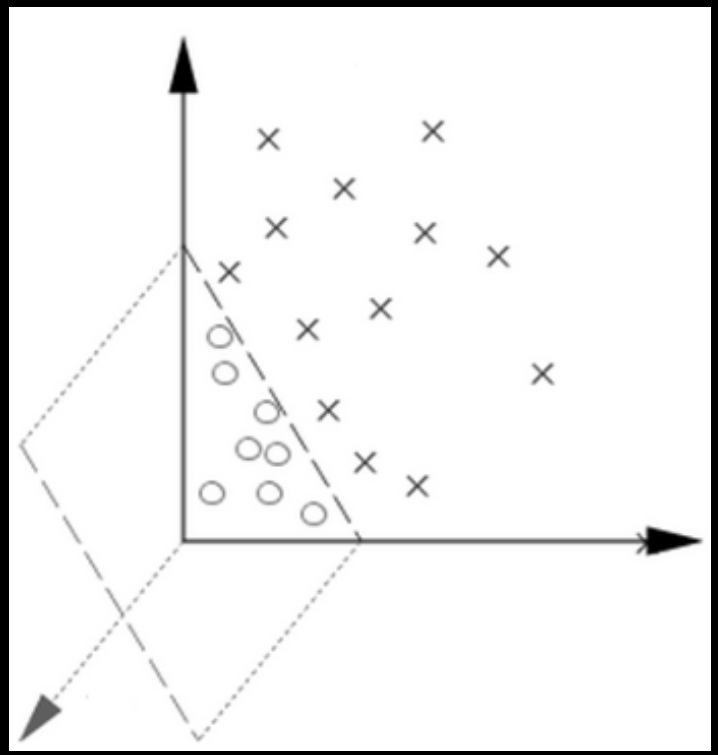
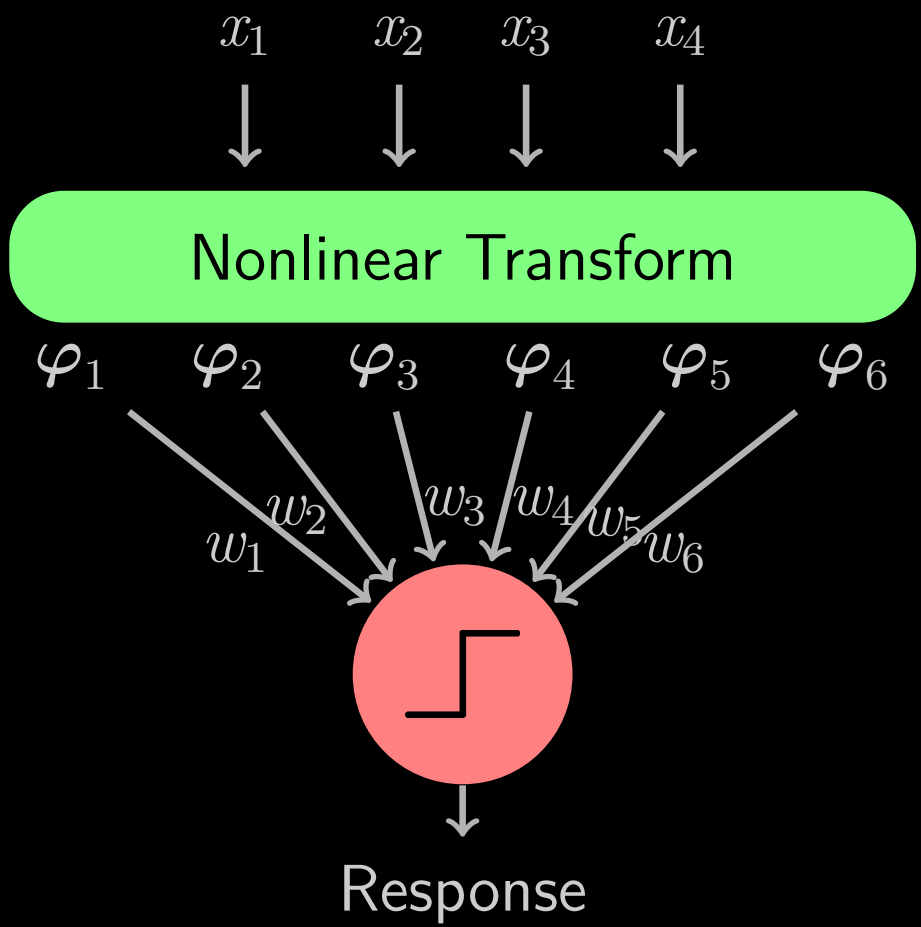


Image source: [Schölkopf, Smola, 2003]



$$p(E_{total}) \leq p(E_{Bayes}) + \underbrace{p(E_{Model})}_{\text{small}\odot} + \underbrace{p(E_{Data})}_{?} + \underbrace{p(E_{Algorithm})}_{\text{small}\odot}$$

Analysis of Learning Algorithms: Statistical Learning Theory

- Probability of making a wrong prediction – in the future

[Cristianini, Shawe-Taylor. Cambridge University Press, 2000]

Analysis of Learning Algorithms: Statistical Learning Theory

- Probability of making a wrong prediction – in the future
- Can we relate this to the accuracy on the *training set* $\mathcal{D}$:

For any $\delta > 0$, it holds with probability at least δ:

$$p(E_{total}) \leq \frac{1}{\ell\gamma} \sum_{i=1}^{\ell} \xi_i + \frac{4}{\ell\gamma}\sqrt{\mathrm{tr}(K)} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

- $\gamma = \frac{1}{||w||}$: "margin", ℓ: number of training examples

[Cristianini, Shawe-Taylor. Cambridge University Press, 2000]

Analysis of Learning Algorithms: Statistical Learning Theory

- Probability of making a wrong prediction – in the future
- Can we relate this to the accuracy on the *training set* $\mathcal{D}$:

For any $\delta > 0$, it holds with probability at least δ:

$$p(E_{total}) \leq \frac{1}{\ell\gamma} \sum_{i=1}^{\ell} \xi_i + \frac{4}{\ell\gamma}\sqrt{\mathrm{tr}(K)} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

- $\gamma = \frac{1}{||w||}$: "margin", ℓ: number of training examples
- Slack variables: $\approx$ error on training set

[Cristianini, Shawe-Taylor. Cambridge University Press, 2000]

Analysis of Learning Algorithms: Statistical Learning Theory

- Probability of making a wrong prediction – in the future
- Can we relate this to the accuracy on the *training set* $\mathcal{D}$:

For any $\delta > 0$, it holds with probability at least δ:

$$p(E_{total}) \leq \frac{1}{\ell\gamma} \sum_{i=1}^{\ell} \xi_i + \frac{4}{\ell\gamma}\sqrt{\mathrm{tr}(K)} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

- $\gamma = \frac{1}{||w||}$: "margin", ℓ: number of training examples
- Slack variables: $\approx$ error on training set
- Kernel trace: $\approx$ model complexity

[Cristianini, Shawe-Taylor. Cambridge University Press, 2000]

Analysis of Learning Algorithms: Statistical Learning Theory

- Probability of making a wrong prediction – in the future
- Can we relate this to the accuracy on the *training set* $\mathcal{D}$:

For any $\delta > 0$, it holds with probability at least δ:

$$p(E_{total}) \leq \frac{1}{\ell\gamma} \sum_{i=1}^{\ell} \xi_i + \frac{4}{\ell\gamma}\sqrt{\mathrm{tr}(K)} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

- $\gamma = \frac{1}{||w||}$: "margin", ℓ: number of training examples
- Slack variables: $\approx$ error on training set
- Kernel trace: $\approx$ model complexity
- Remaining uncertainty: because training data could be atypical

[Cristianini, Shawe-Taylor. Cambridge University Press, 2000]

Application Areas for Machine Learning

Bioinformatics
- SNP Discovery
- Molecule Structure Prediction

High-Energy Physics
- Rare-Event Detection

Medical Imaging
- Image Reconstruction
- Tissue Segmentation

Neuroscience
- Spike Sorting
- Brain Computer Interfaces

Natural Language Processing
- Speech Recognition
- Spam Filtering

Robotics
- Non-Linear Control
- Inverse Dynamics

Finance
- Fraud Detection

Computer Vision
- Character Recognition
- Image Understanding

Application Areas for Machine Learning

Bioinformatics
- SNP Discovery
- Molecule Structure Prediction

High-Energy Physics
- Rare-Event Detection

Medical Imaging
- Image Reconstruction
- Tissue Segmentation

Neuroscience
- Spike Sorting
- Brain Computer Interfaces

Natural Language Processing
- Speech Recognition
- Spam Filtering

Robotics
- Non-Linear Control
- Inverse Dynamics

Finance
- Fraud Detection

Computer Vision
- Character Recognition
- **Image Understanding**

Computer Vision: Long term goal

Automatic systems that can analyze and interpret visual data

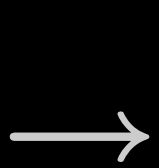
Image Understanding

‘‘Three men sit at a table in a pub, drinking beer. One of them talks while the other listen.’’

Computer Vision: Short/medium term goal

Automatic systems that can analyze certain aspects of visual data

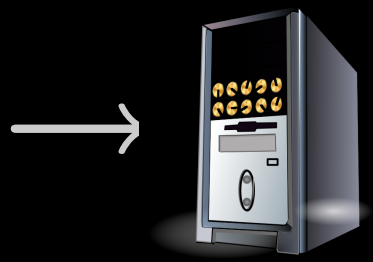
Scene Classification



- indoors
- in a pub

Computer Vision: Short/medium term goal

Automatic systems that can analyze certain aspects of visual data



Action Classification

Computer Vision: Short/medium term goal

Automatic systems that can analyze certain aspects of visual data

Object Recognition

- three persons
- one table
- three glasses

Machine Learning for Object Recognition

Machine Learning provides:

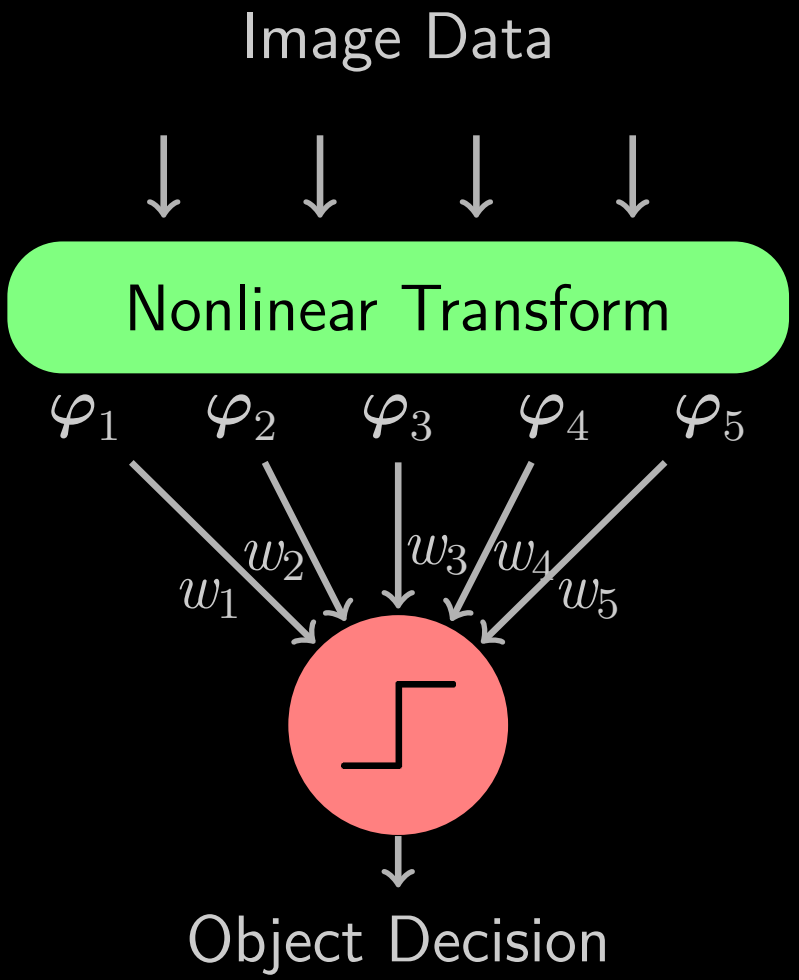
- Support vector machine

Computer Vision adds:

- Image data representations
- Nonlinear transform φ
- Image-specific inference

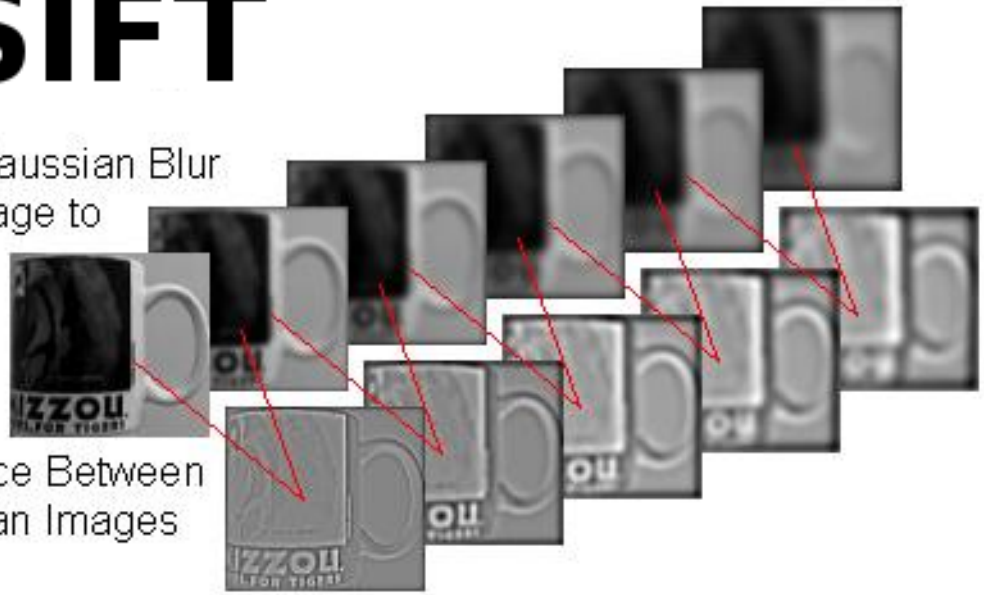
Successful φ makes use of:

- local receptive fields
- center-surround effects
- gradient orientations



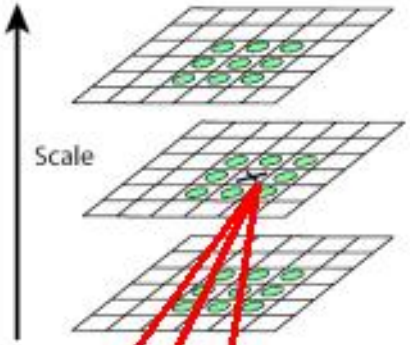
SIFT

Incrementally Gaussian Blur The Original Image to Create a Scale Space

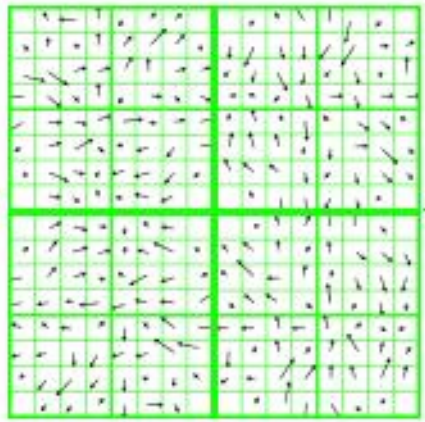


Find the Difference Between Adjacent Gaussian Images in Scale Space

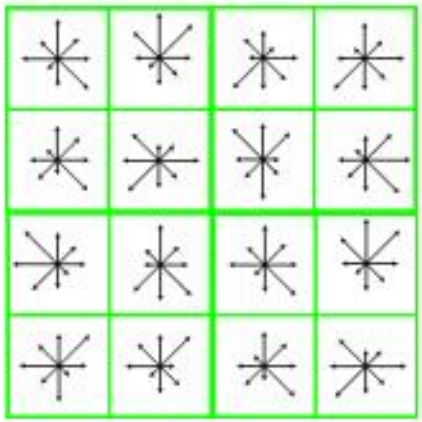
Keypoints are Pixels in Difference Images That are Larger Than or Smaller Than all 26 Neighbors



Hundreds of Keypoints are Found



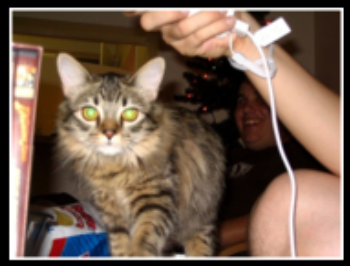
The Gradient of Pixels Around Each Keypoint is Determined At the Gaussian Scale at Which It Was Found



Sixteen Histograms are Created Using The Gradients. Using 8 Orientations, This Makes 128-D Feature Vectors.

image source: [eecs.vanderbilt.edu]

Machine Learning for Object Recognition: Training Data


cat person


eiffel tower boat


person boat


aeroplane


stairs chair


aeroplane


person plant


sheep


computer chair monitor


train


plant bus


car


person

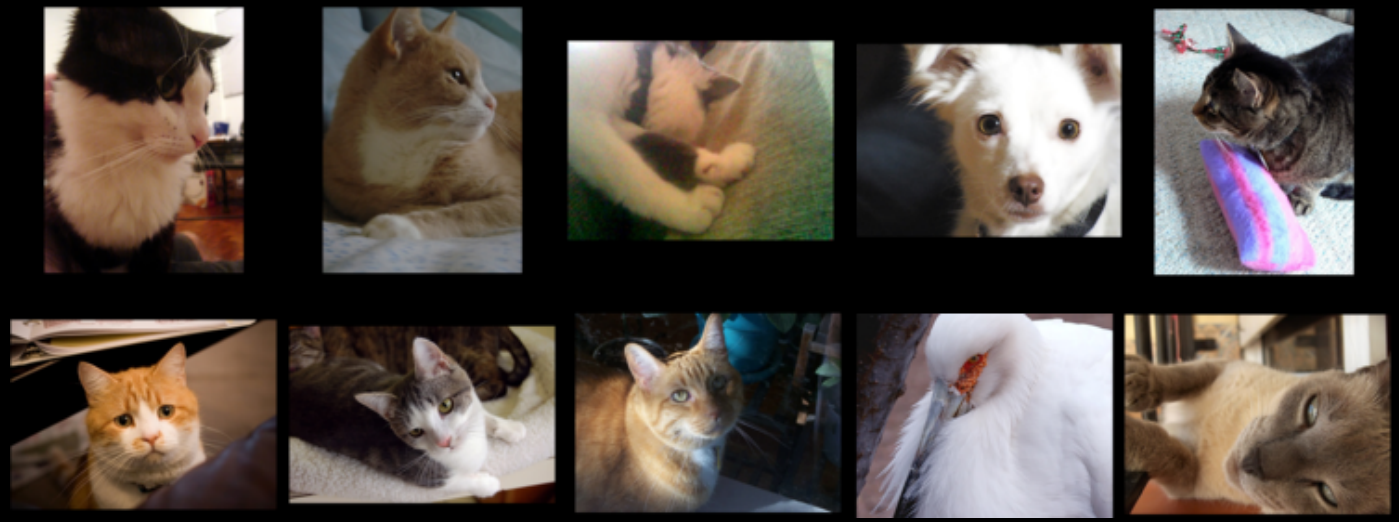

person dog


train

(5000 images in total)

Results. The computer thinks...

...these images contain **cats**:

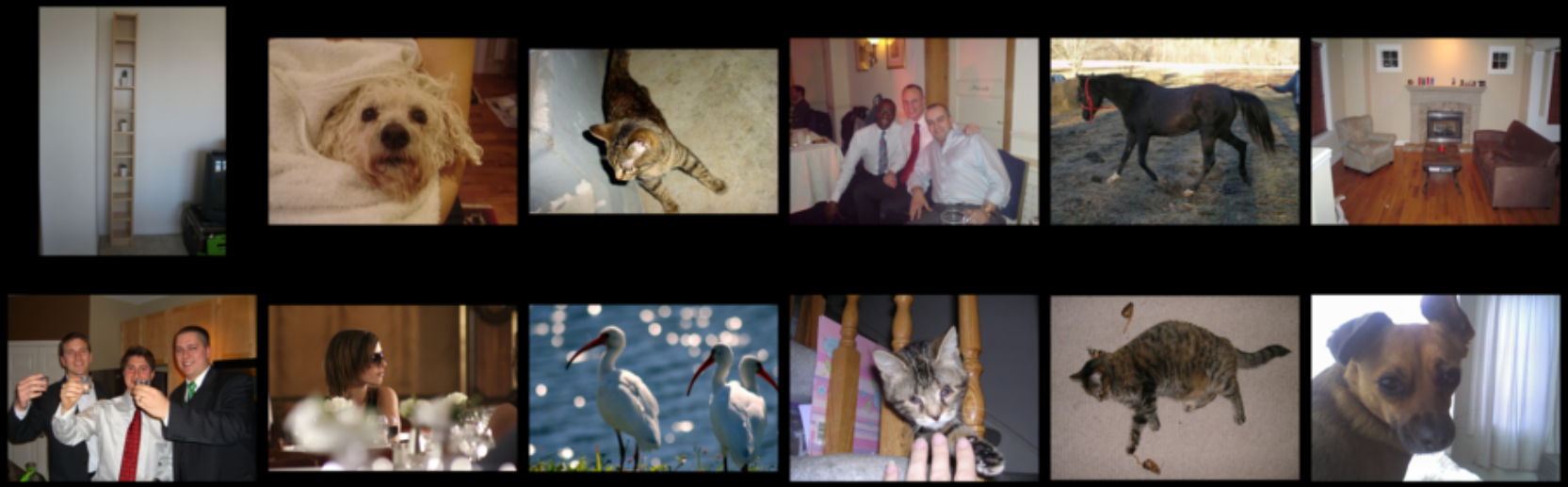


...these images contain **no cats**:

Results. The computer thinks...

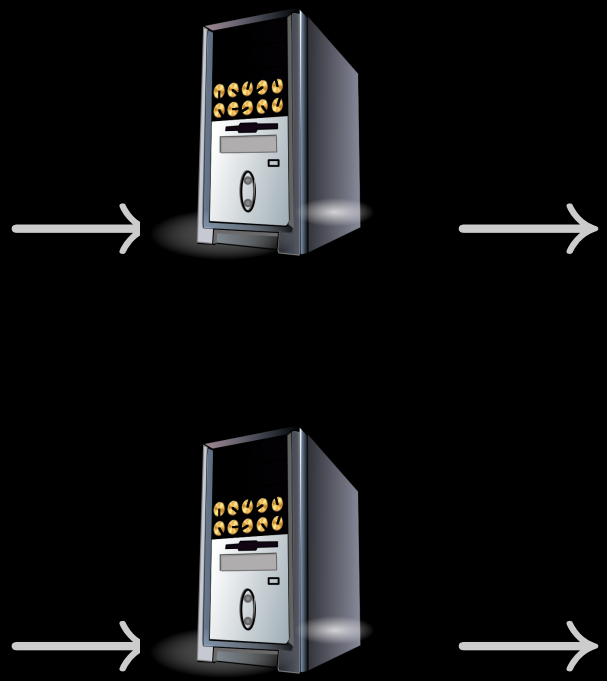
...these images contain **cars**:

...these images contain **no cars**:



Object Localization instead of only Classification

Object Localization instead of only Classification



- one person
- one car

- one person
- one car

Object Classification

Object Localization instead of only Classification

Object Localization

Object Localization brings us closer to *image interpretation*:

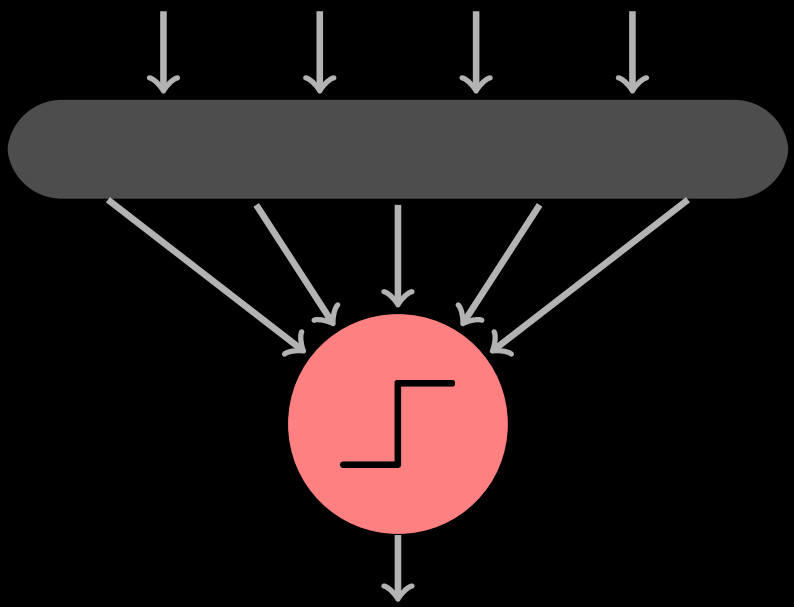
- Person inside a car. $\longrightarrow$ He's driving.
- Person outside of a car. $\longrightarrow$ He's passing by.

Object Localization

- Localization is much harder than Classification

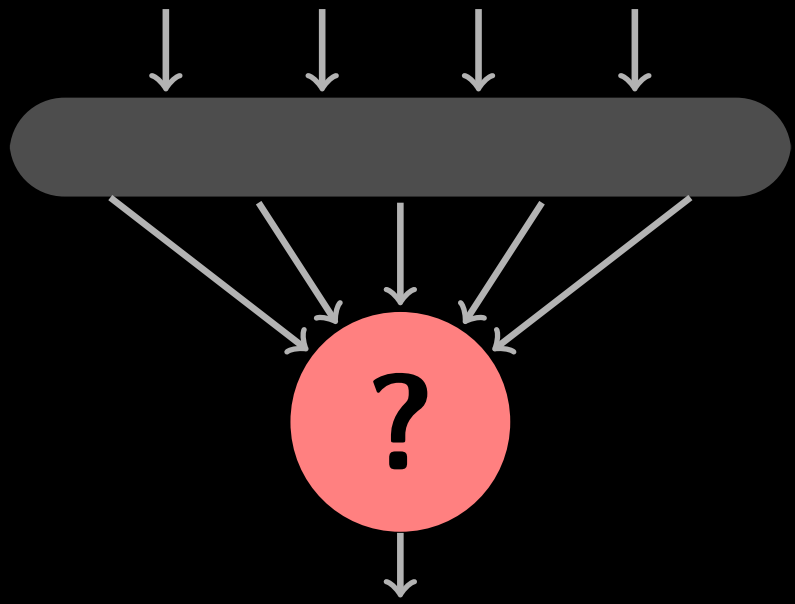
Object Localization

- Localization is much harder than Classification



`yes/no`
2 possibilities

Object Classification



[`left`, `top`, `right`, `bottom`]
millions of possibilities

Object Localization

Object Localization in a *Regression Problem*

- Training images:

$$x_1, \ldots, x_n \in \mathbb{R}^{\textit{width} \times \textit{height}} \quad =: \mathcal{X}$$

- Annotation: coordinates $y = [\textit{left}, \textit{top}, \textit{right}, \textit{bottom}]$

$$y_1, \ldots, y_n \in \mathbb{R}^4 \quad =: \mathcal{Y}$$

- Task: learn a prediction function

$$f : \mathcal{X} \rightarrow \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i.$$

[Lampert, Blaschko, Hofmann, CVPR 2008], [Blaschko, Lampert, ECCV 2008], [Lampert, Blaschko, Hofmann, PAMI 2009]

Object Localization in a *Regression Problem*

- Training images:

$$x_1, \ldots, x_n \in \mathbb{R}^{\textit{width} \times \textit{height}} \quad =: \mathcal{X}$$

- Annotation: coordinates $y = [\textit{left}, \textit{top}, \textit{right}, \textit{bottom}]$

$$y_1, \ldots, y_n \in \mathbb{R}^4 \quad =: \mathcal{Y}$$

- Task: learn a prediction function

$$f : \mathcal{X} \rightarrow \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i.$$

- **Regression between structured spaces**

[Lampert, Blaschko, Hofmann, CVPR 2008], [Blaschko, Lampert, ECCV 2008], [Lampert, Blaschko, Hofmann, PAMI 2009]

- Task: learn a prediction function, i.e. learn weight vector w:

$$f : \mathcal{X} \rightarrow \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i$$

Solve a *minimization problem* for w:

$$\min_{w \in \mathbb{R}^d, \xi \in \mathbb{R}^n} \quad \|w\|^2 + \sum_{i=1}^{n} \xi_i$$

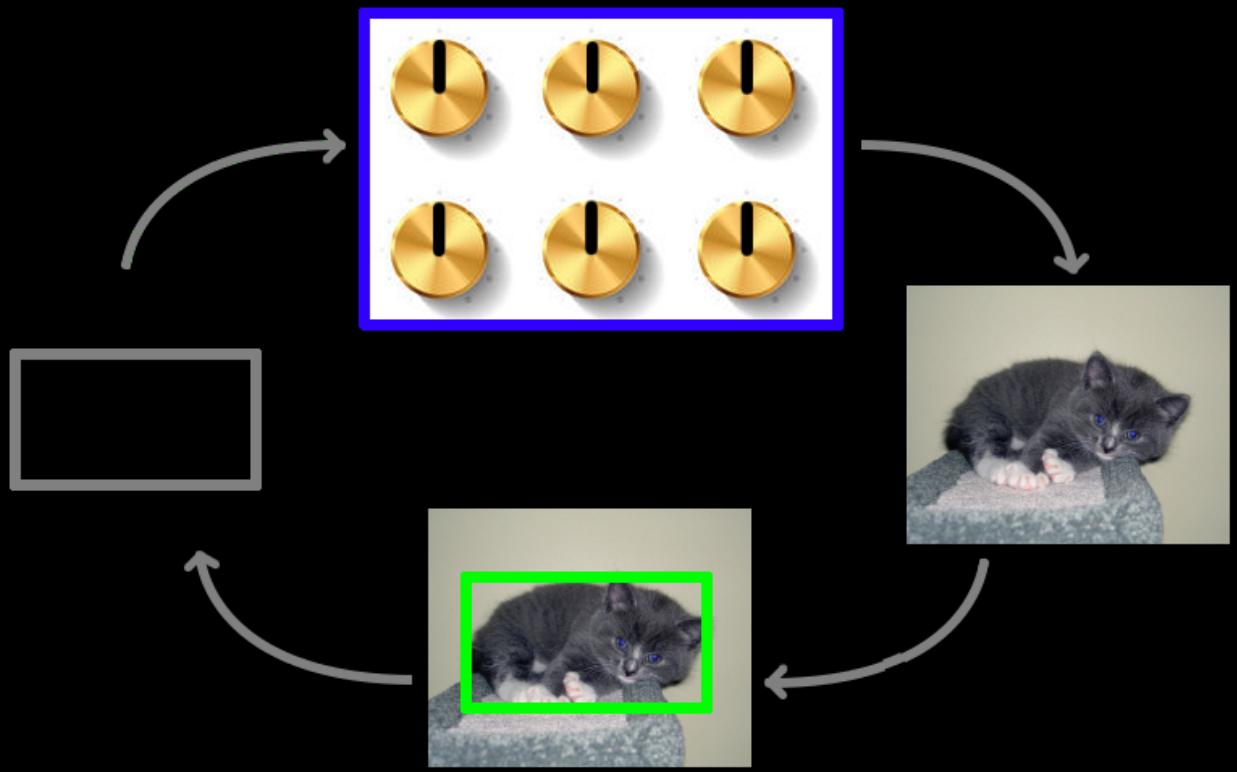
subject to the constraints

$$\langle w, \varphi(x_i, y_i) \rangle - \langle w, \varphi(x_i, y) \rangle \geq \Delta(y_i, y) - \xi_i$$

- Optimization problem is *convex*
  $\Rightarrow$ we can find the global minimum $\Rightarrow$ $p(E_{Algorithm})$ small
- w very high-dimensional (even $d = \infty$) $\Rightarrow$ $p(E_{Model})$ small

Learning by Iterative Self Correction

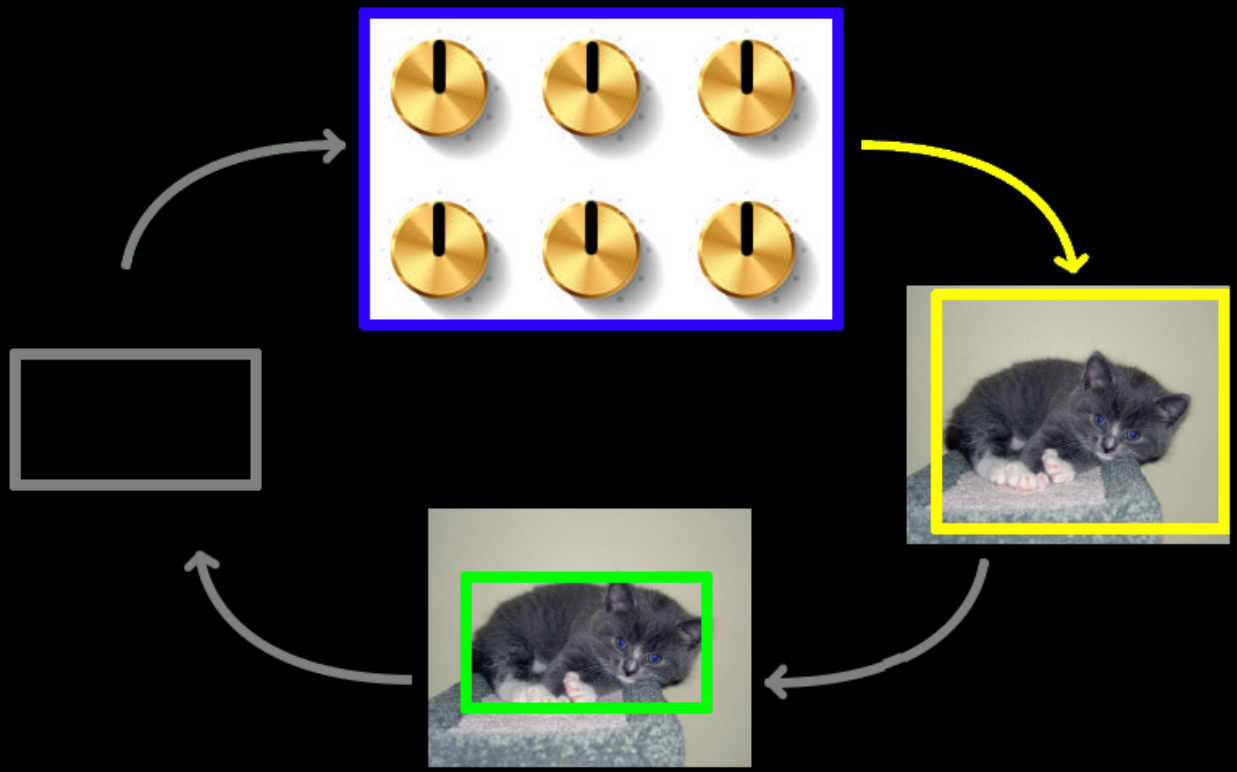
Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.



[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

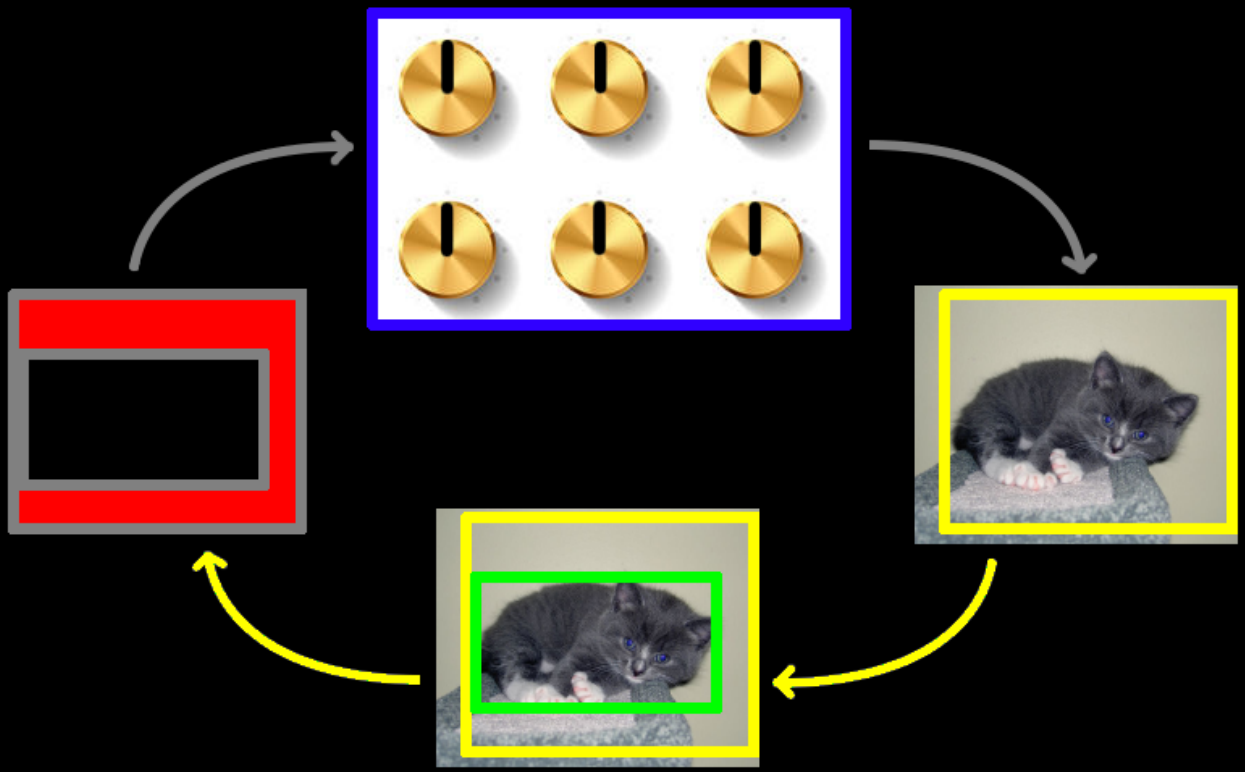


predict with initial model

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

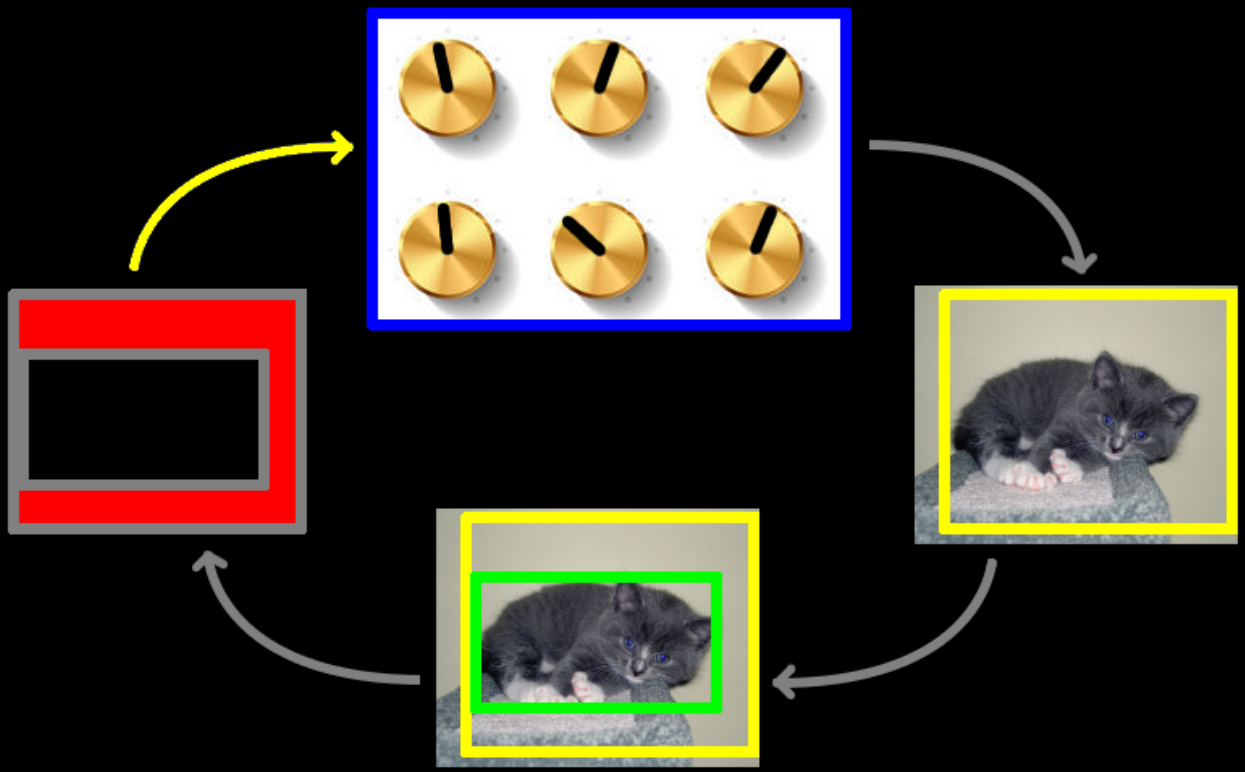


compare to training labels

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

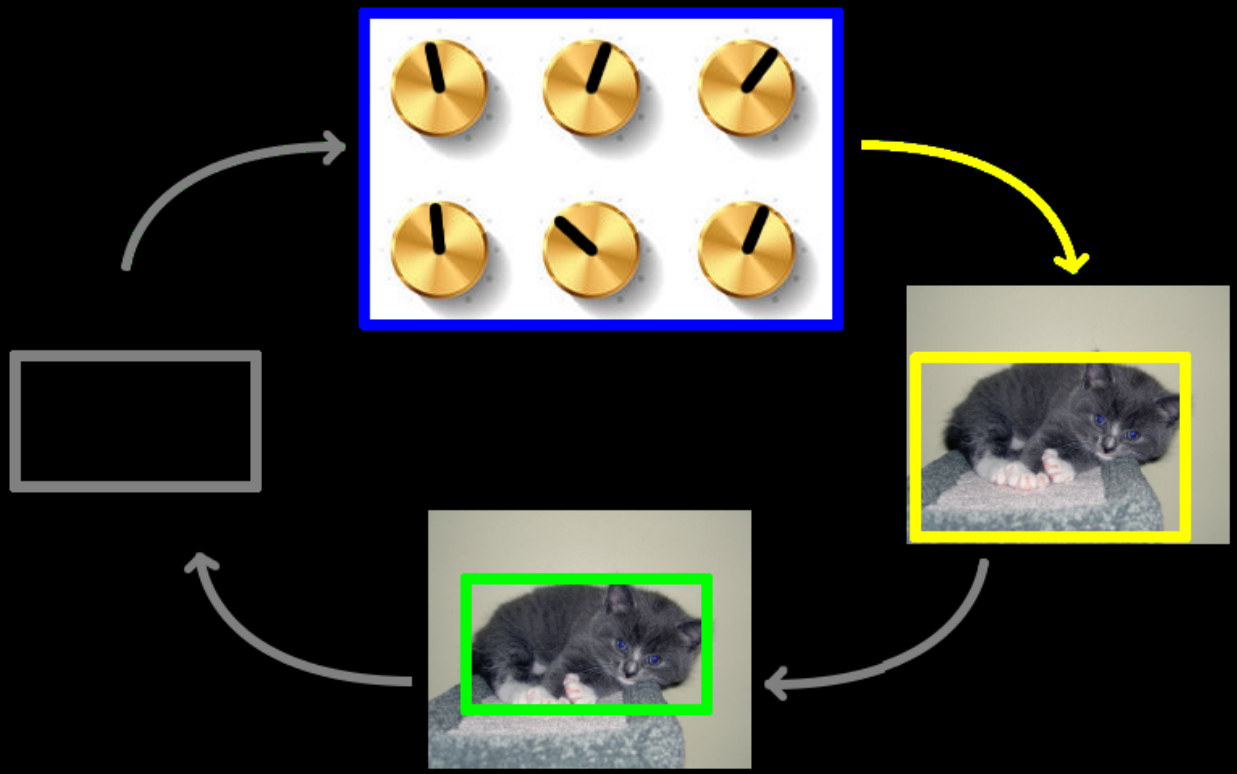


use difference to adjust weights

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \mathrm{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

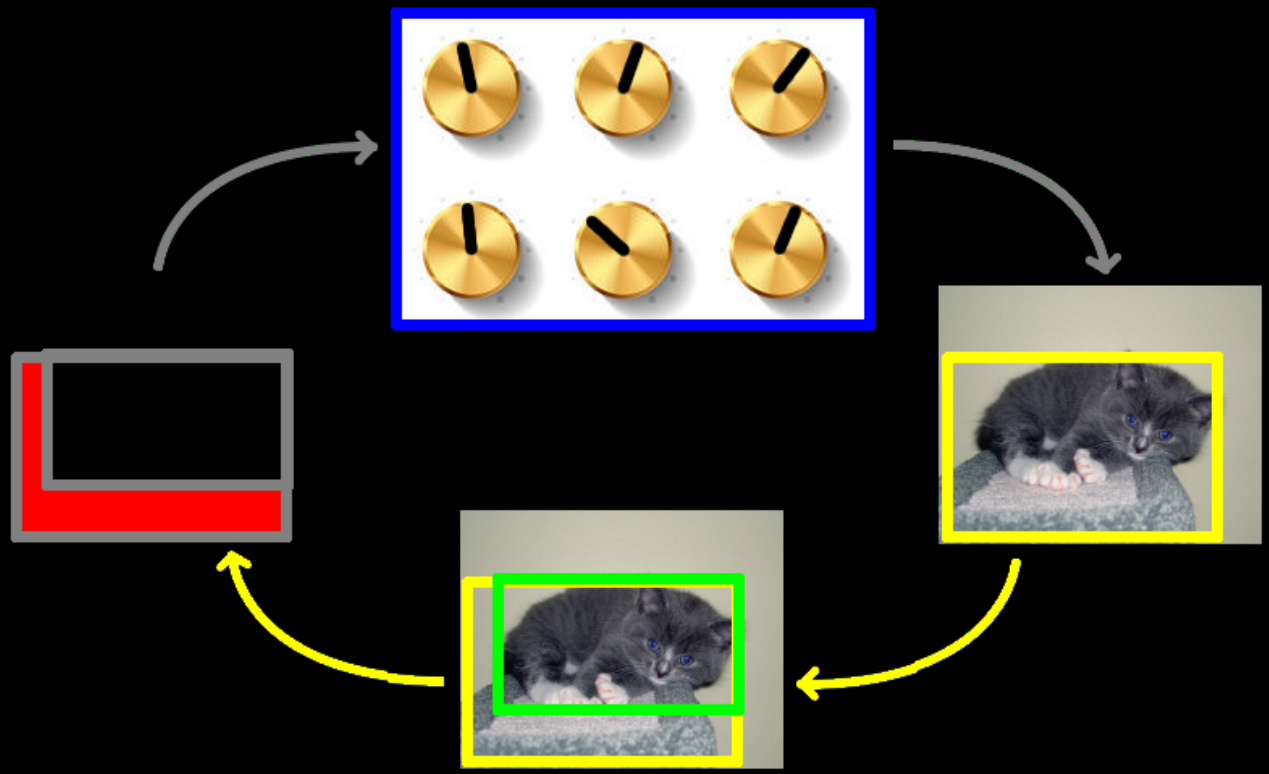


predict with current model

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

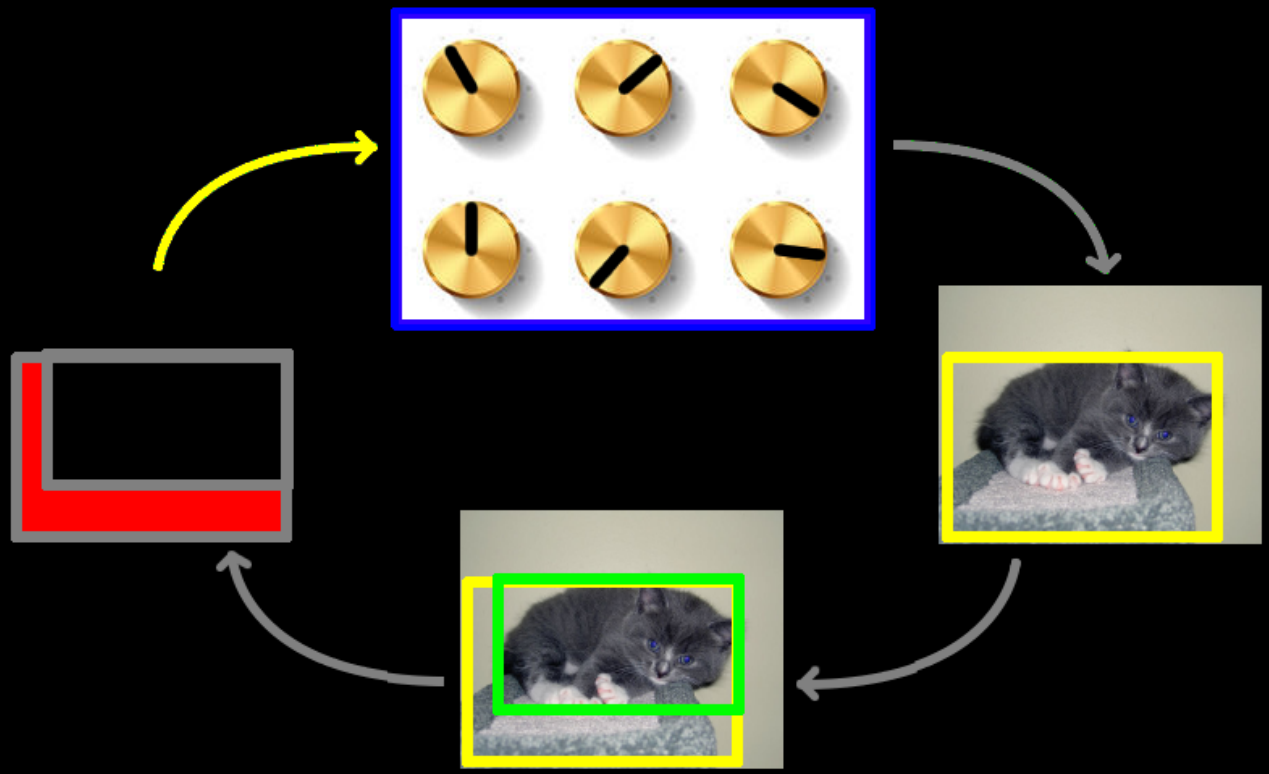


compare to training labels

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

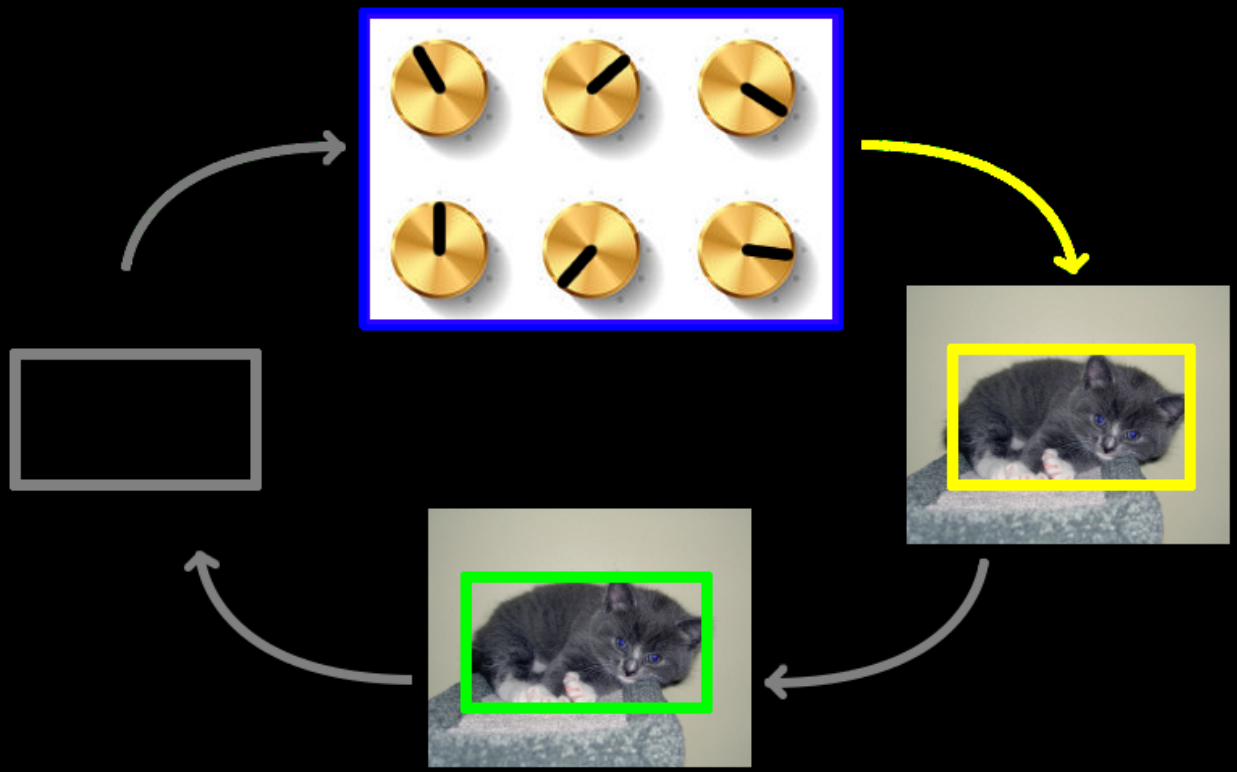


use difference to adjust weights

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \mathrm{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

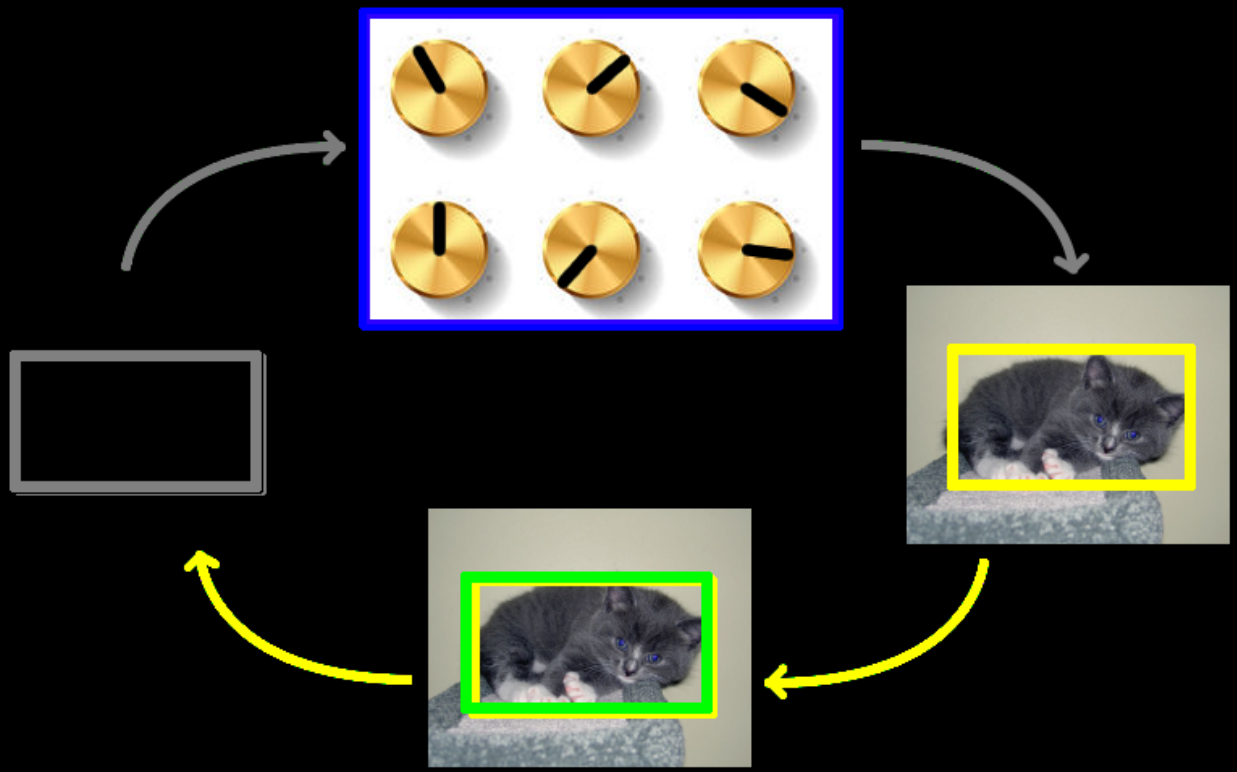


predict with current model

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

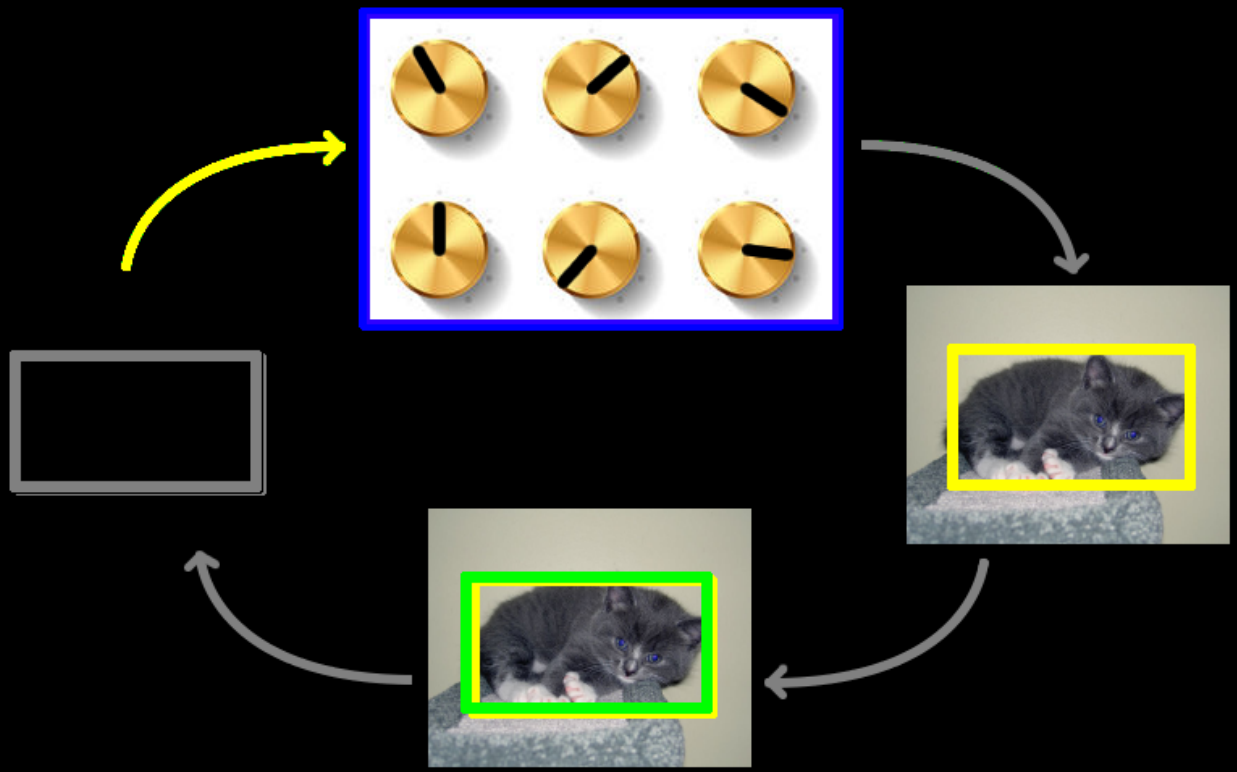


compare to training labels

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Learning by Iterative Self Correction

Learning model: generalized support vector machine
$f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$, parameter vector w.

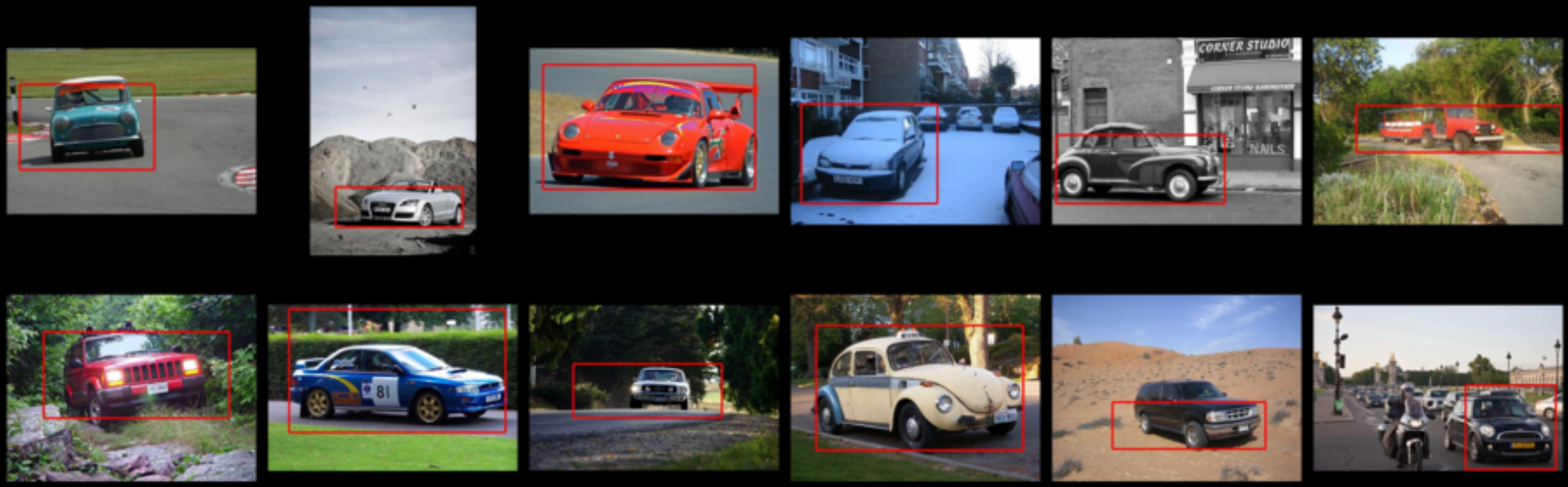


...until convergence

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

Visual Evaluation

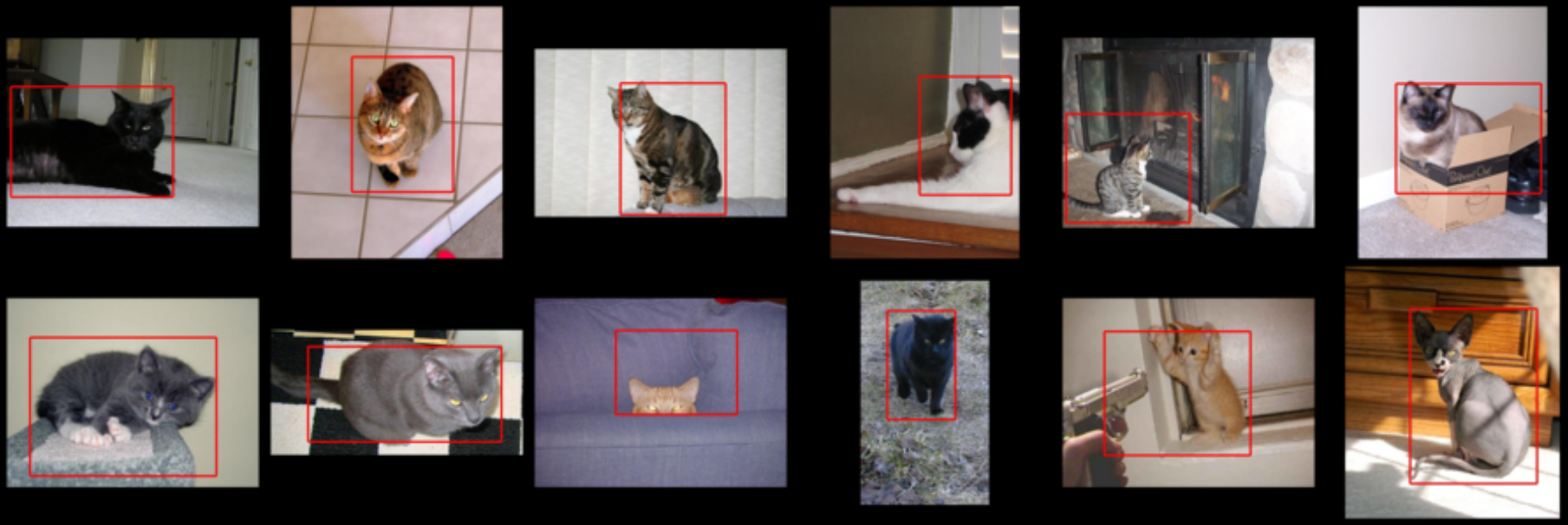
Car detections:



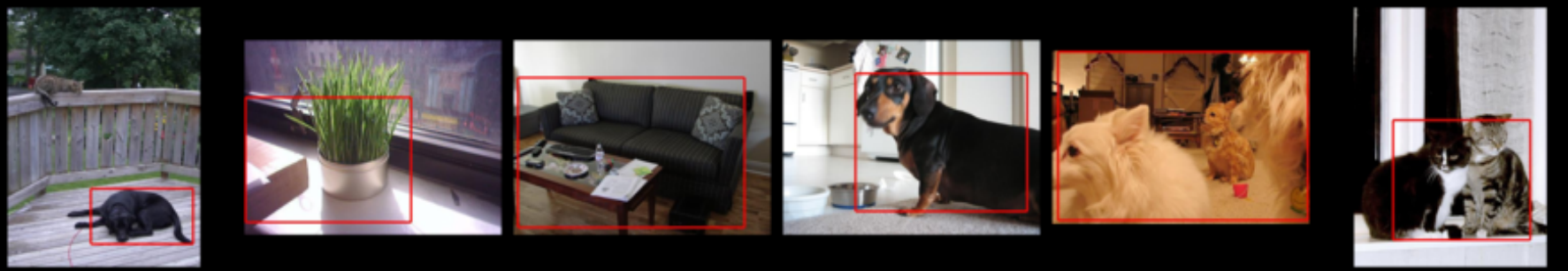
Car misdetections:

Visual Evaluation

Cat detections:



Cat misdetections:



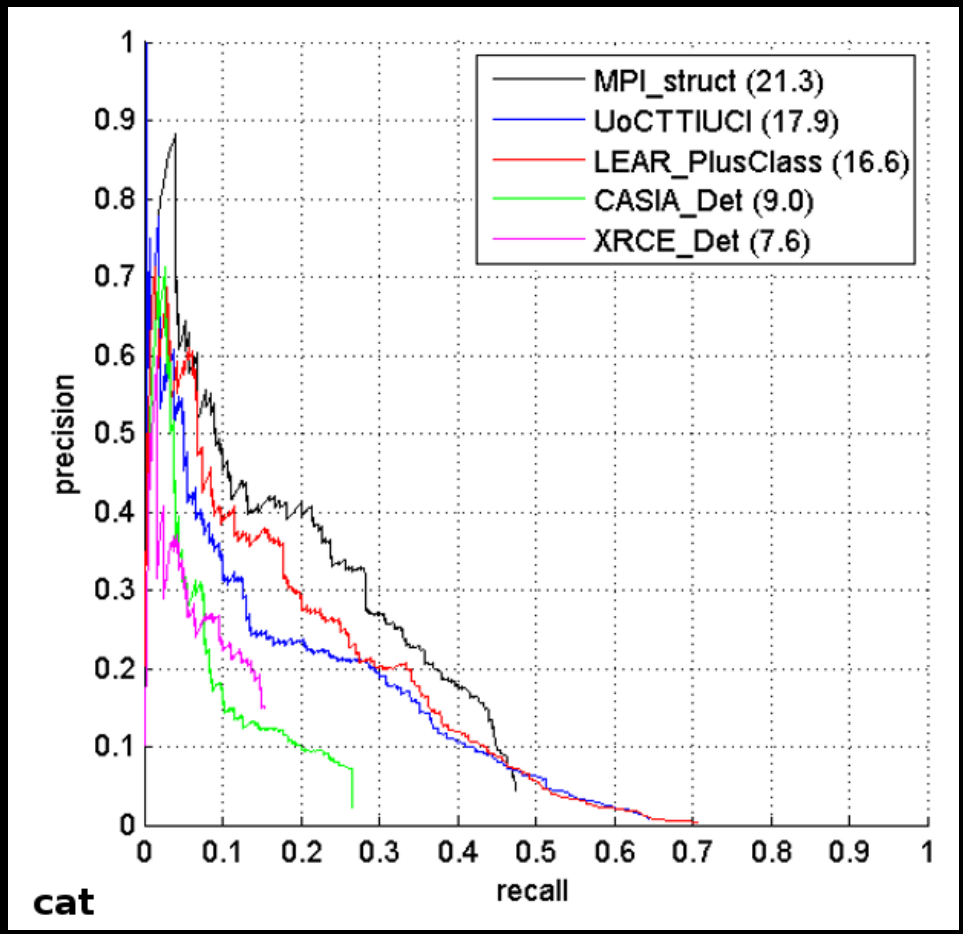
Objective Evaluation

- Example: PASCAL VOC Competition 2008
  - annual benchmark for object classification/localization
  - 10,000 images (5,000 training / 5,000 evaluation)
  - 20 object categories

Objective Evaluation

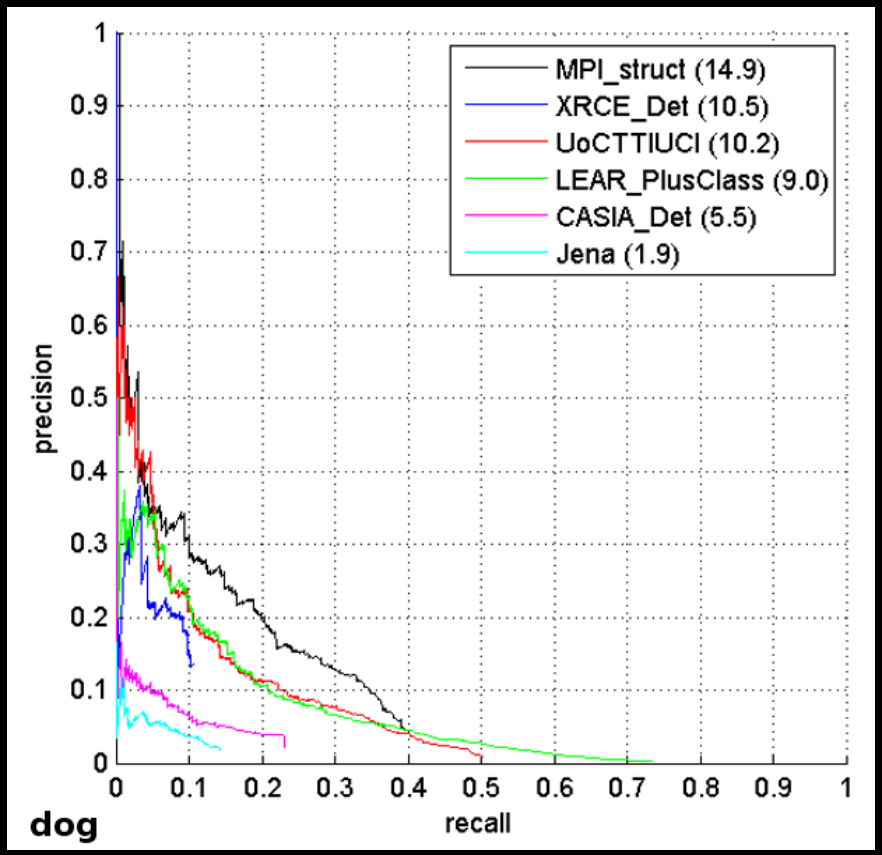
- Example: PASCAL VOC Competition 2008
  - annual benchmark for object classification/localization
  - 10,000 images (5,000 training / 5,000 evaluation)
  - 20 object categories

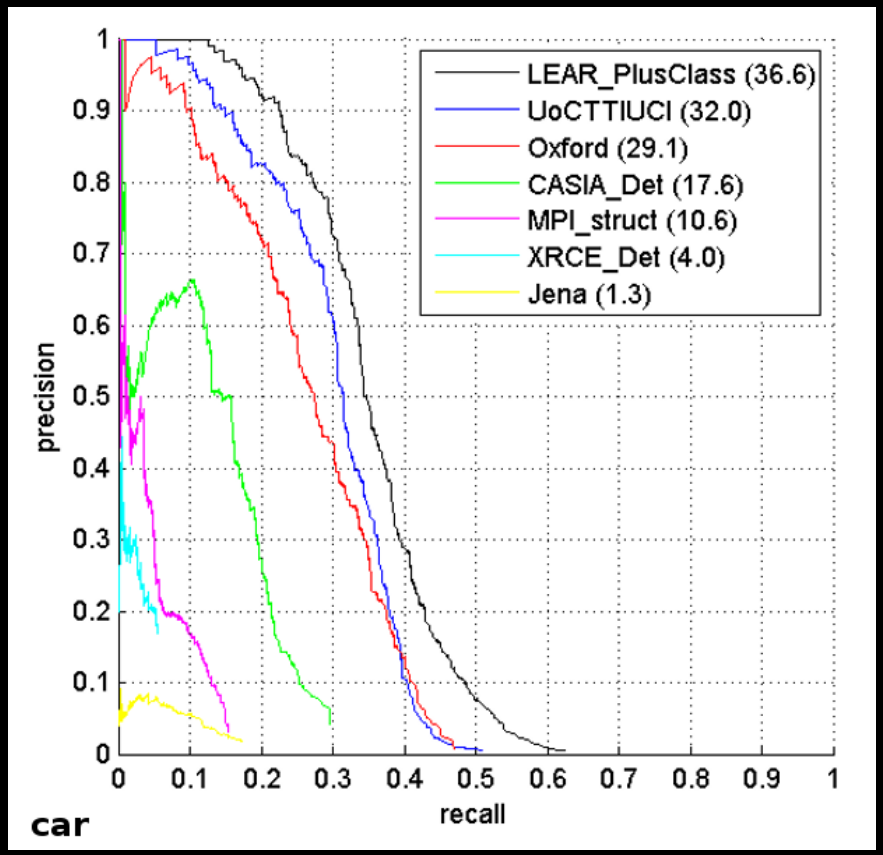
Performance in terms of precision/recall curves:



Objective Evaluation

- Example: PASCAL VOC Competition 2008
  - annual benchmark for object classification/localization
  - 10,000 images (5,000 training / 5,000 evaluation)
  - 20 object categories



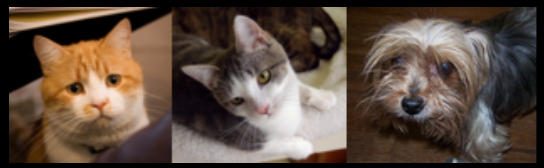


Future Challenge: Video

Learning from videos needs adaptivity:

- Two-stage learning is unrealistic.

Learning



cat cat dog

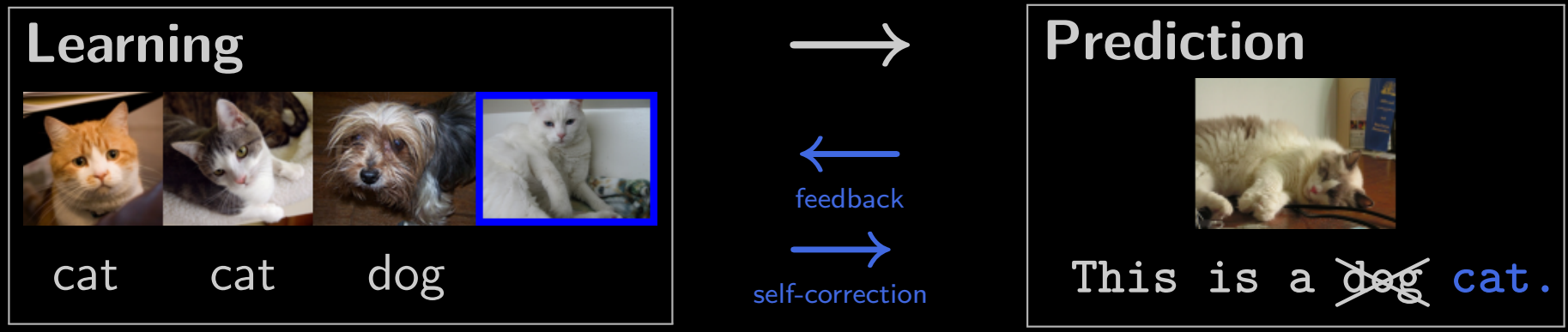
Prediction

`This is a dog.`

Future Challenge: Video

Learning from videos needs adaptivity:

- Two-stage learning is unrealistic.
- Natural learning is incremental and adaptive.



[Lampert, Peters, DAGM 2009], [Dhillon, Nowozin, Lampert, VISU@CVPR 2009]

Future Challenge: Large Scale / Fine-Grained Categories

There's tens of thousands of object categories:

- Learning one model for each of them is unrealistic.

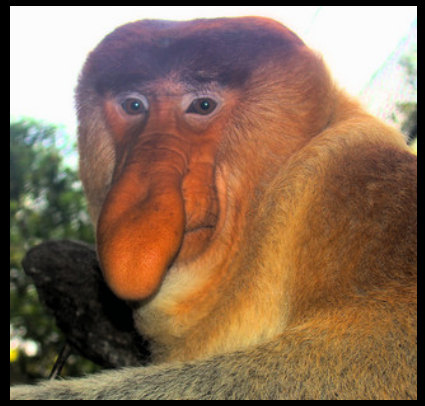
Future Challenge: Large Scale / Fine-Grained Categories

There's tens of thousands of object categories:

- Learning one model for each of them is unrealistic.

Which of these images shows an **axolotl**?





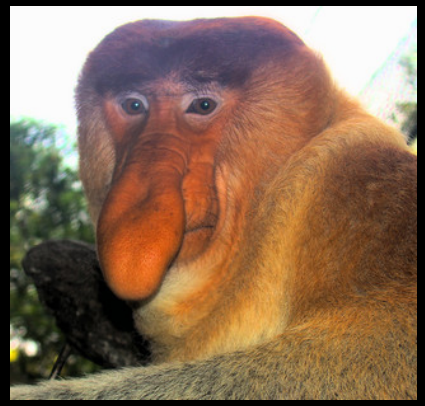

Future Challenge: Large Scale / Fine-Grained Categories

There's tens of thousands of object categories:

- Learning one model for each of them is unrealistic.

Which of these images shows an **axolotl**?






Description: **Axolotls**

- live in **water**,
- are **white**,
- have no **long fur**.

We can classify objects based on a **description**!

Live Demo...

nearly every recent digicam: face detection

Online Demo...

Microsoft Photosynth: 3D Reconstruction
`http://www.photosynth.com`

Homework... check out these demos / videos!

Microsoft Kinect: human pose tracking

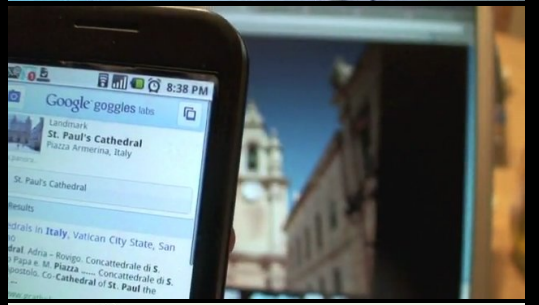
`http://www.youtube.com/watch?v=ewlJTZw3f70`

(new input device for *Xbox 360*)

Google Goggles: object recognition

`http://www.youtube.com/watch?v=WA7wwKIC24s`

(free Android app available)



City-scale 3D reconstruction

`http://www.youtube.com/watch?v=sz0UbHvEttI`

VisLab (Google, too): driver-free cars

`http://viac.vislab.it` (and others)

Summary

Machine Learning:

What are the principles of **learning from data**?
How can we build learning machines?

Computer Vision:

Automatic systems that **understand visual data**

Future Challenges:

Learning from Video / Learning with Attributes

Additional Material

- Task: learn a prediction function

$$f : \mathcal{X} \to \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i$$

Generalized support vector machine:

$$f(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$$

- Free parameters: w
- $\varphi(x, y)$ depends on y
  - similar to a collection of classifiers, one per position
  - coupled by shared weights w

[Tsochantaridis, Joachims, Hofmann, Altun, JMLR 2005], [Lampert, Blaschko, ML 2008]

- Task: learn a prediction function

$$f : \mathcal{X} \rightarrow \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i$$

Loss function: "How bad is predicting y' if y would be correct"?

$$\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Box overlap:

$$\Delta(y, y') := 1 - \frac{\mathbf{area}(y \cap y')}{\mathbf{area}(y \cup y')}$$

- $0 \leq \Delta(y, y') \leq 1$,
- $\Delta(y, y') = 0 \quad \Leftrightarrow \quad y'$ is identical to y,
- $\Delta(y, y') = 1 \quad \Leftrightarrow \quad y'$ has no overlap with y.

- Task: learn a prediction function, i.e. learn weight vector w:

$$f : \mathcal{X} \rightarrow \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i$$

Solve a *minimization problem* for w:

$$\min_{w \in \mathbb{R}^d, \xi \in \mathbb{R}^n} \quad \|w\|^2 + \sum_{i=1}^{n} \xi_i$$

subject to the constraints

$$\langle w, \varphi(x_i, y_i) \rangle - \langle w, \varphi(x_i, y) \rangle \geq \Delta(y_i, y) - \xi_i$$

- Optimization problem is *convex*
  $\Rightarrow$ we can find the global minimum $\Rightarrow$ $p(E_{Algorithm})$ small
- w very high-dimensional (even $d = \infty$) $\Rightarrow$ $p(E_{Model})$ small