Classifiers and Object Representation

Computer Vision

CS 543 / ECE 549

University of Illinois

Derek Hoiem

Today's class: classifiers and objects

- More about classifiers
- Object categories and representation

Image Categorization

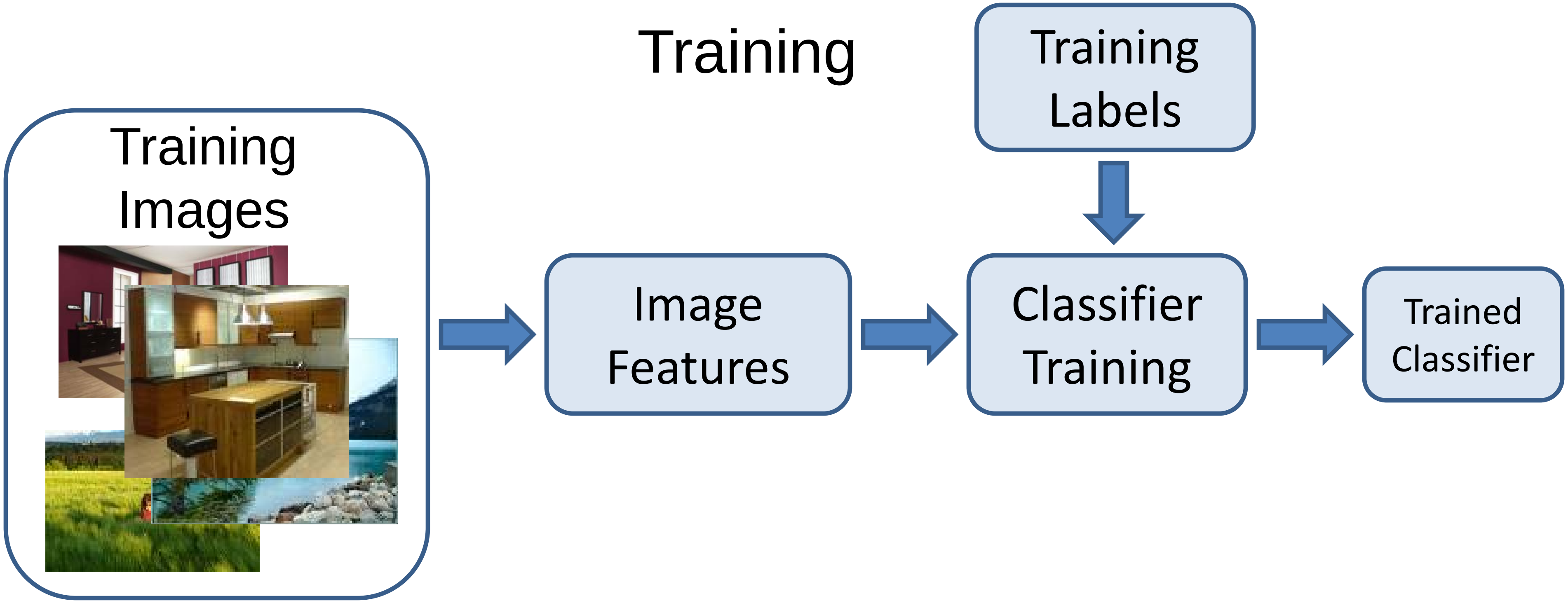
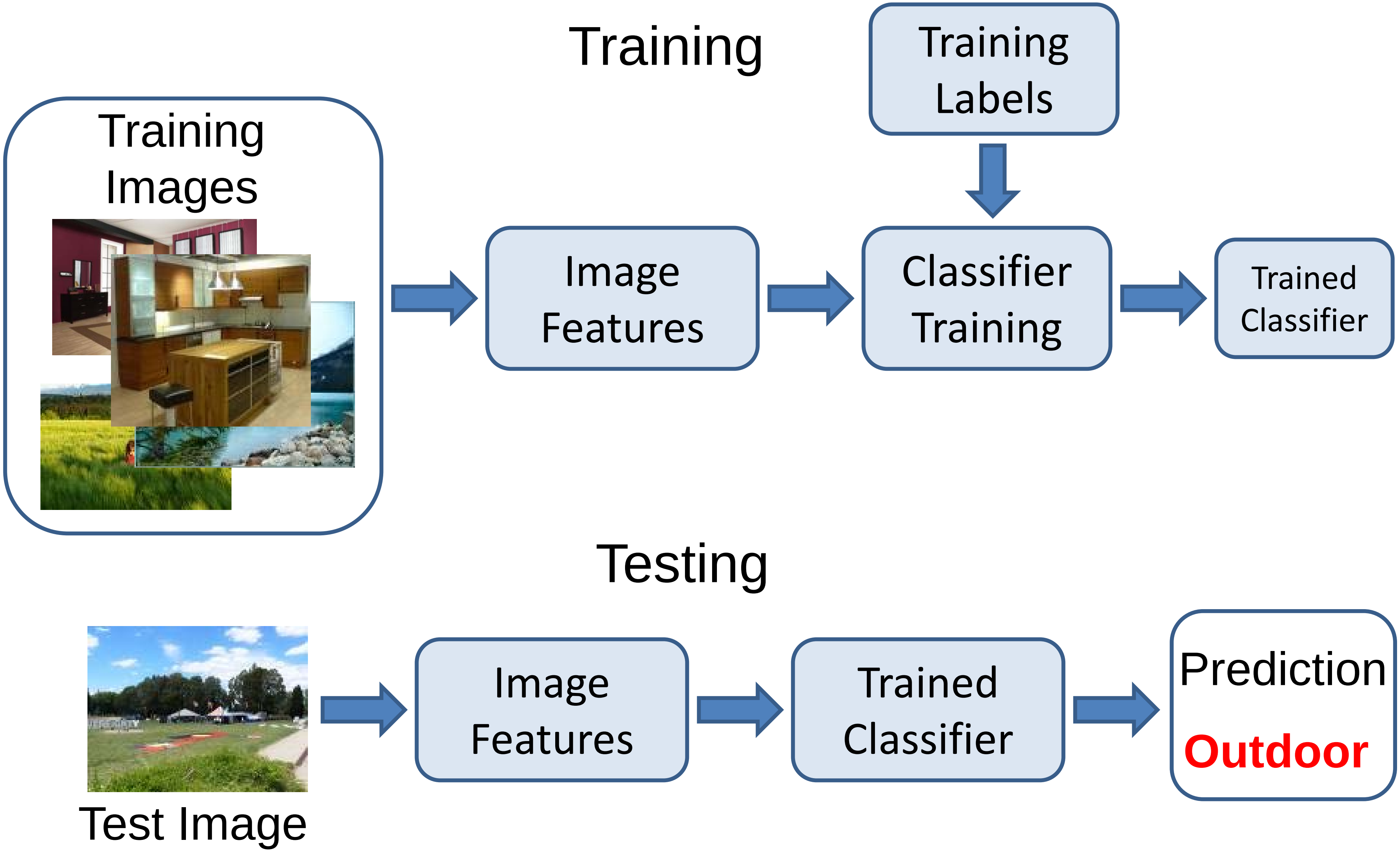


Image Categorization



Remember…

- No classifier is inherently better than any other: you need to make assumptions to generalize

- Three kinds of error
  – Inherent: unavoidable
  – Bias: due to over-simplifications
  – Variance: due to inability to perfectly estimate parameters from limited data

How to reduce variance?

- Choose a simpler classifier
- Regularize the parameters
- Get more training data

Very brief tour of some classifiers

- **SVM**
- Neural networks
- **Naïve Bayes**
- Bayesian network
- **Logistic regression**
- Randomized Forests
- **Boosted Decision Trees**
- **K-nearest neighbor**
- RBMs
- Etc.

Generative vs. Discriminative Classifiers

Generative Models

- Represent both the data and the labels
- Often, makes use of conditional independence and priors
- Examples
  - Naïve Bayes classifier
  - Bayesian network

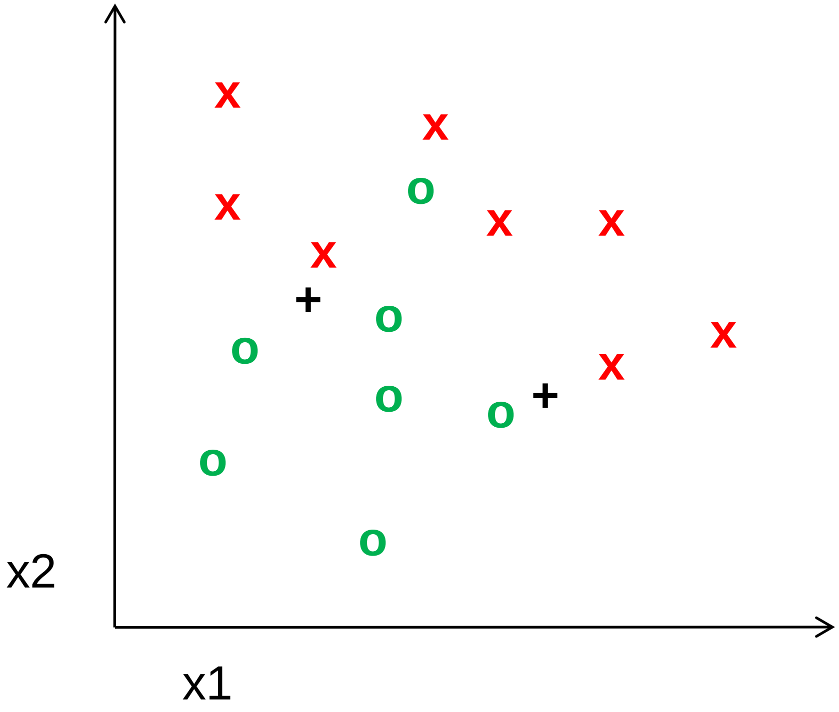
- Models of data may apply to future prediction problems

Discriminative Models

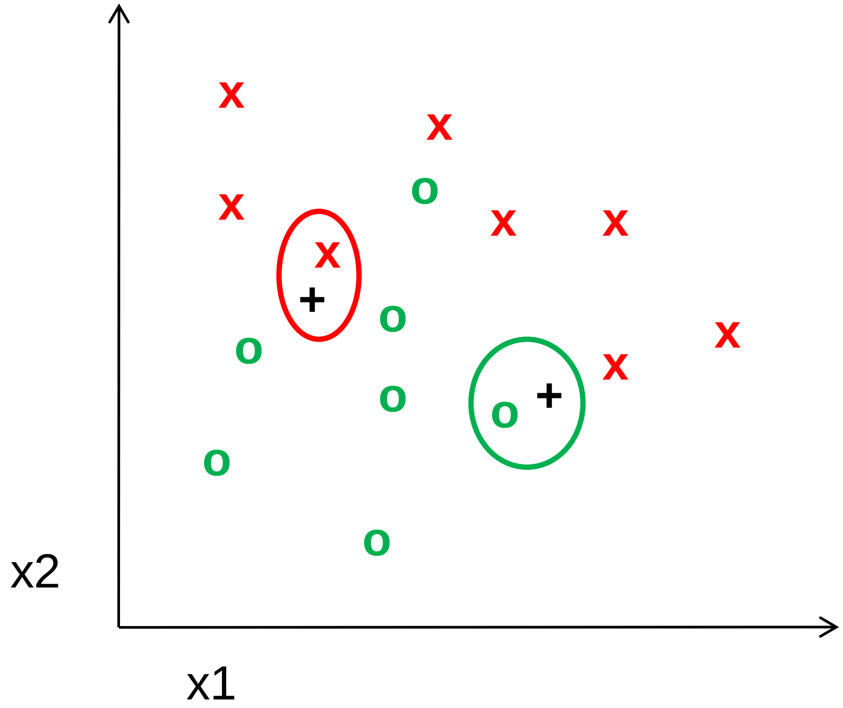
- Learn to directly predict the labels from the data
- Often, assume a simple boundary (e.g., linear)
- Examples
  - Logistic regression
  - SVM
  - Boosted decision trees

- Often easier to predict a label from the data than to model the data

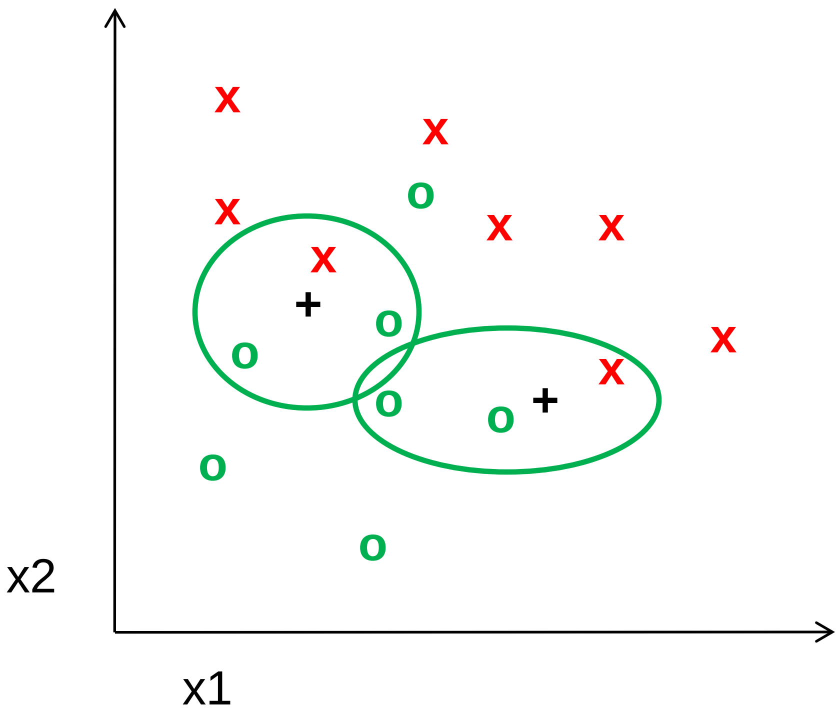
K-nearest neighbor



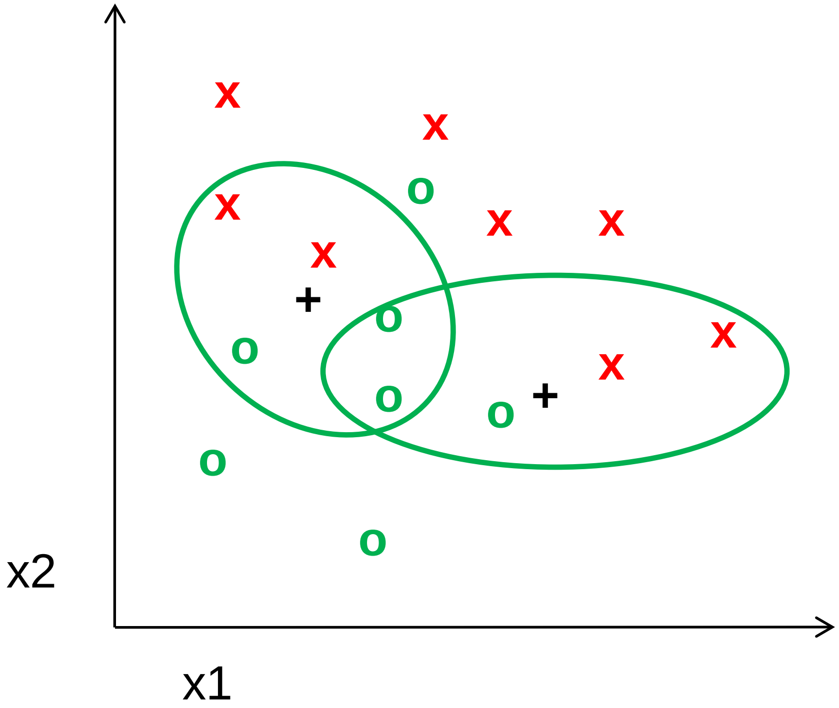
1-nearest neighbor



3-nearest neighbor



5-nearest neighbor

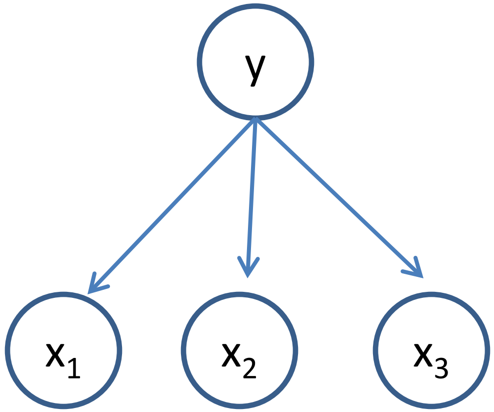


Using K-NN

- Simple, a good one to try first

- With infinite examples, 1-NN provably has error that is at most twice Bayes optimal error

Naïve Bayes



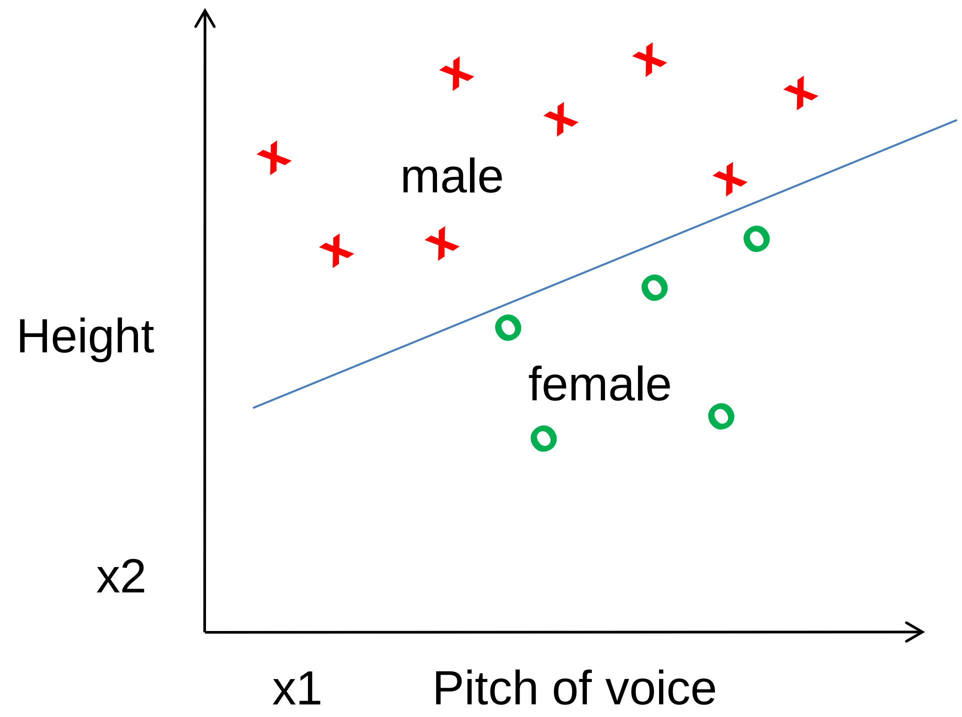
Using Naïve Bayes

- Simple thing to try for categorical data

- Very fast to train/test

Classifiers: Logistic Regression

Maximize likelihood of label given data, assuming a log-linear model



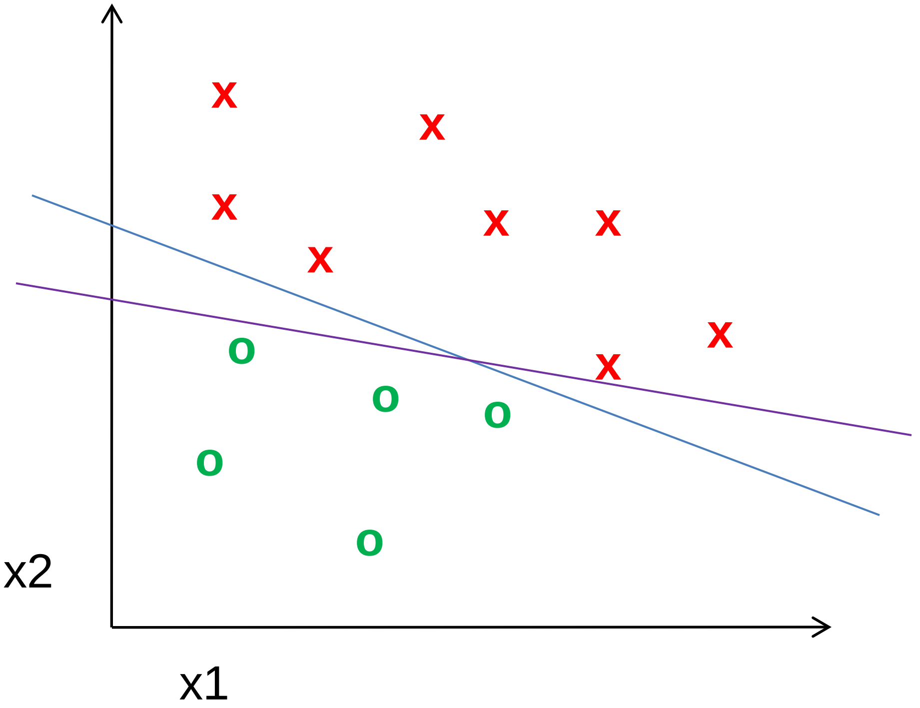
$$\log \frac{P(x_1, x_2 \mid y=1)}{P(x_1, x_2 \mid y=-1)} = \mathbf{w}^T \mathbf{x}$$

$$P(y=1 \mid x_1, x_2) = 1/\left(1+\exp\left(-\mathbf{w}^T \mathbf{x}\right)\right)$$

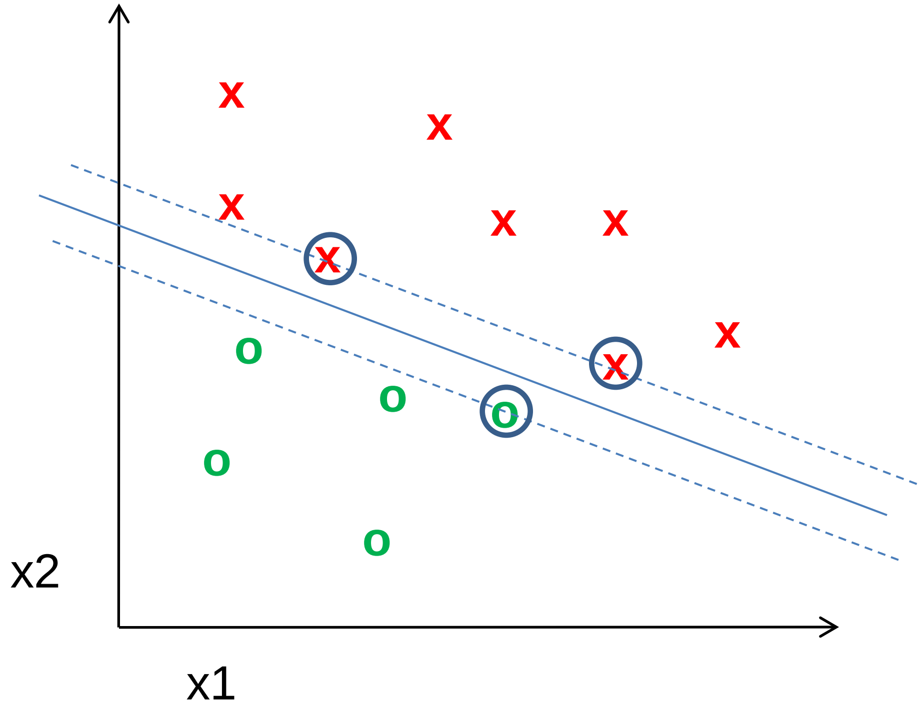
Using Logistic Regression

- Quick, simple classifier (try it first)
- Outputs a probabilistic label confidence
- Use L2 or L1 regularization
  - L1 does feature selection and is robust to irrelevant features but slower to train

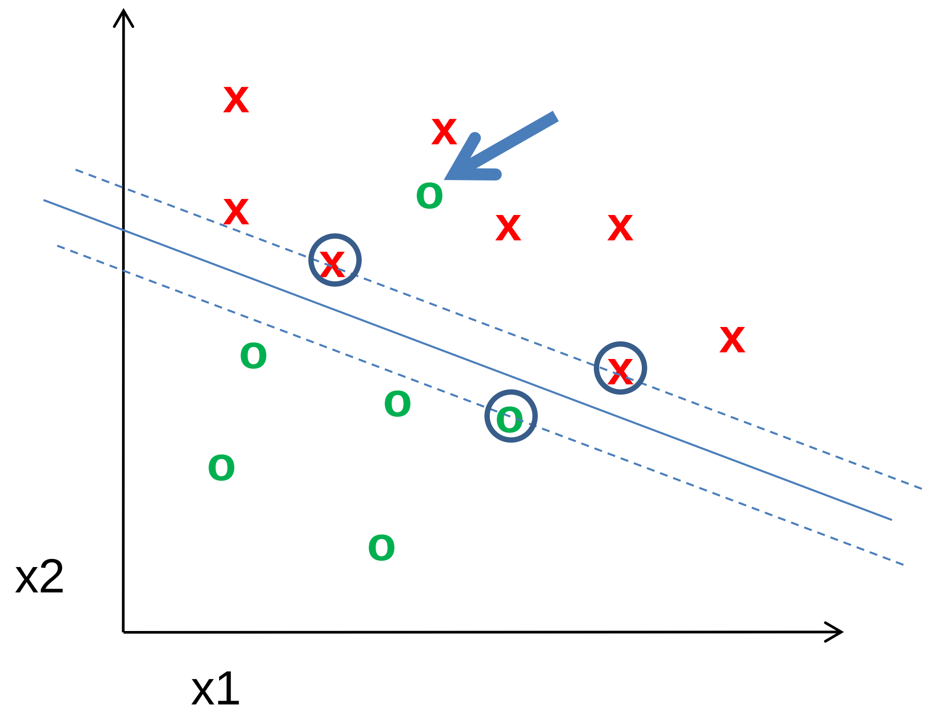
Classifiers: Linear SVM



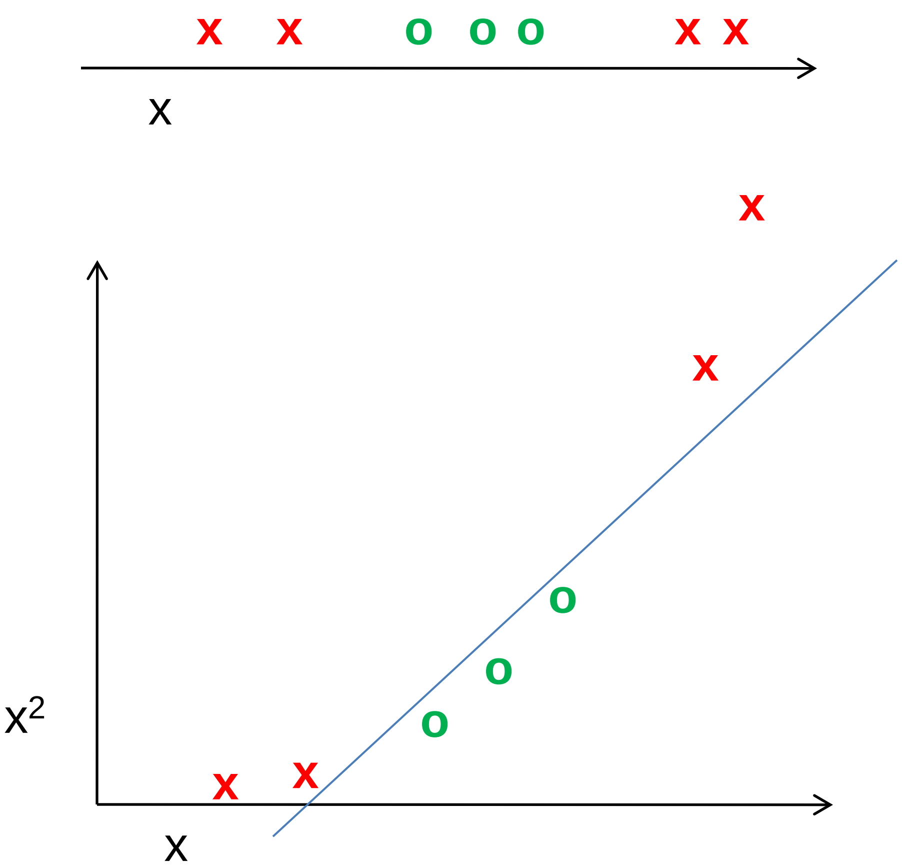
Classifiers: Linear SVM



Classifiers: Linear SVM



Classifiers: Kernelized SVM

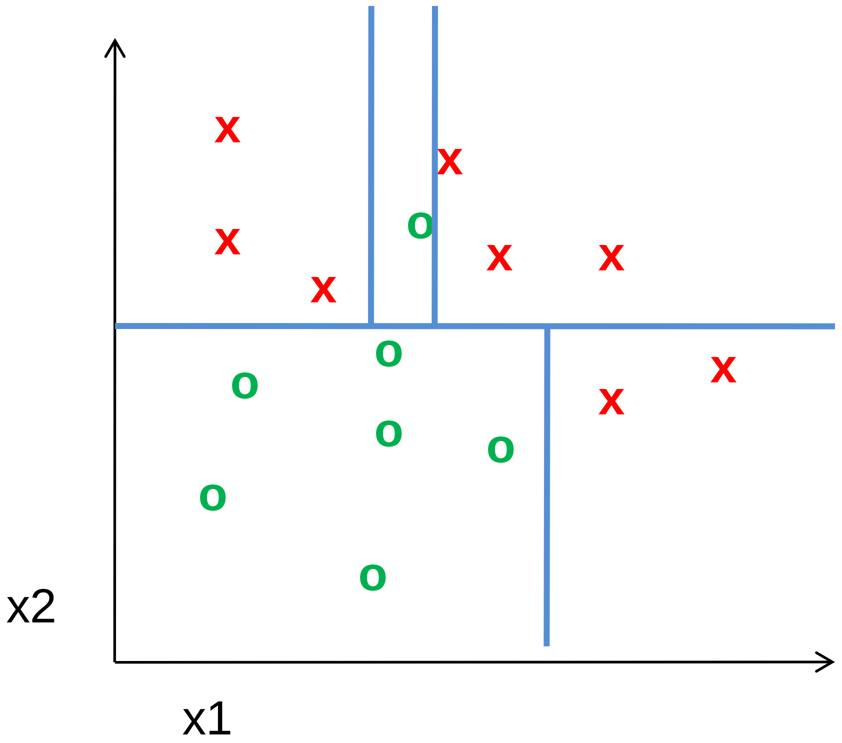


Using SVMs

- Good general purpose classifier
  - Generalization depends on margin, so works well with many weak features
  - No feature selection
  - Usually requires some parameter tuning

- Choosing kernel
  - Linear: fast training/testing – start here
  - RBF: related to neural networks, nearest neighbor
  - Chi-squared, histogram intersection: good for histograms (but slower, esp. chi-squared)
  - Can learn a kernel function

Classifiers: Decision Trees



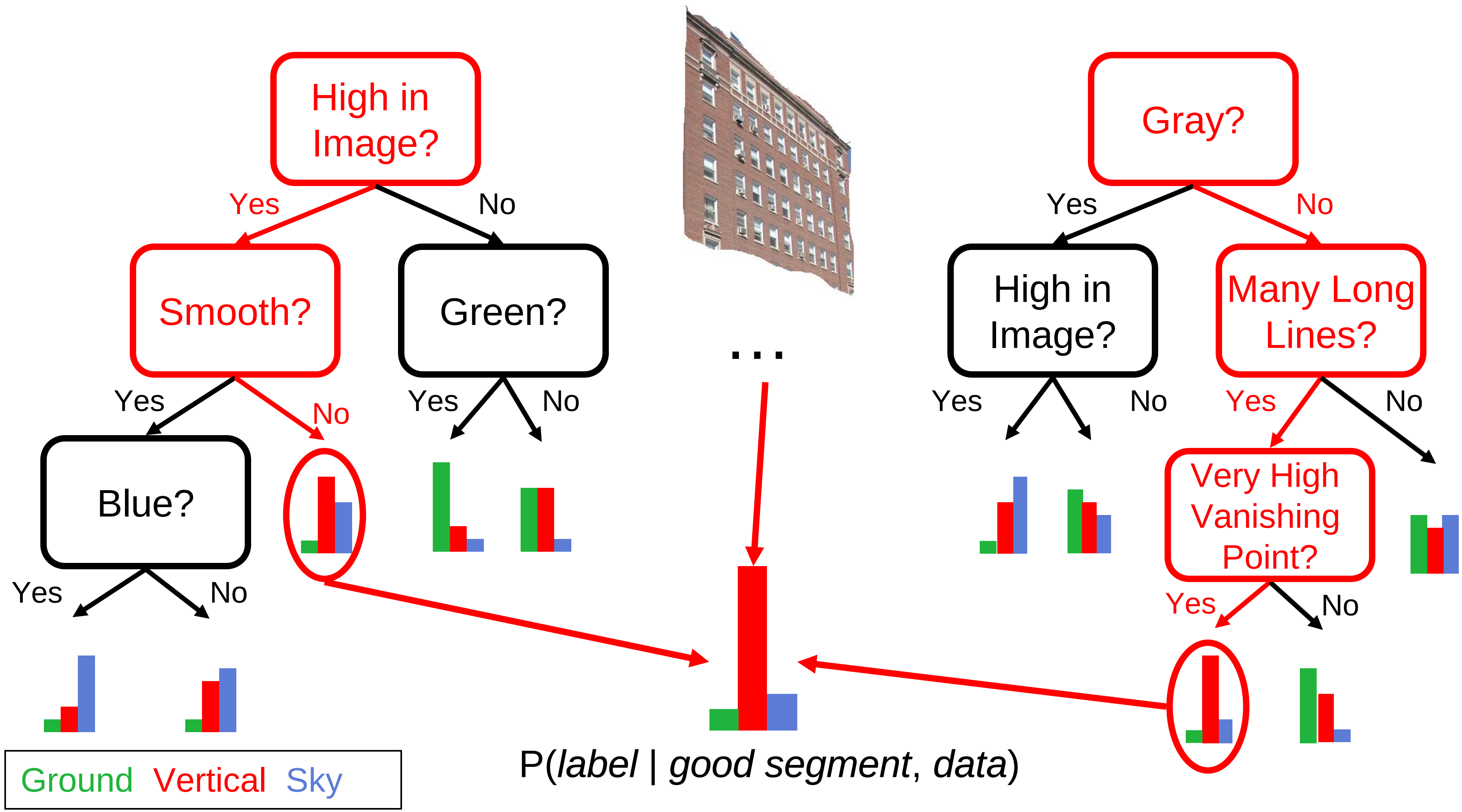
Ensemble Methods: Boosting

Discrete AdaBoost(Freund & Schapire 1996*b*)

1. Start with weights $w_i = 1/N$, $i = 1, \dots, N$.
2. Repeat for $m = 1, 2, \dots, M$:
   (a) Fit the classifier $f_m(x) \in \{-1, 1\}$ using weights w_i on the training data.
   (b) Compute $\text{err}_m = E_w[1_{(y \neq f_m(x))}]$, $c_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (c) Set $w_i \leftarrow w_i \exp[c_m \cdot 1_{(y_i \neq f_m(x_i))}]$, $i = 1, 2, \dots N$, and renormalize so that $\sum_i w_i = 1$.
3. Output the classifier $\text{sign}[\sum_{m=1}^{M} c_m f_m(x)]$

figure from Friedman et al. 2000

Boosted Decision Trees



[Collins et al. 2002]

Using Boosted Decision Trees

- Flexible: can deal with both continuous and categorical variables
- How to control bias/variance trade-off
  – Size of trees
  – Number of trees
- Boosting trees often works best with a small number of well-designed features
- Boosting "stubs" can give a fast classifier

Two ways to think about classifiers

1. What is the objective? What are the parameters? How are the parameters learned? How is the learning regularized? How is inference performed?

Ideals for a classification algorithm

- Objective function: encodes the right loss for the problem
- Parameterization: takes advantage of the structure of the problem
- Regularization: good priors on the parameters
- Training algorithm: can find parameters that maximize objective on training set
- Inference algorithm: can solve for labels that maximize objective function for a test example

Two ways to think about classifiers

1. What is the objective? What are the parameters? How are the parameters learned? How is the learning regularized? How is inference performed?

2. How is the data modeled? How is similarity defined? What is the shape of the boundary?

Comparison

assuming x in {0 1}

	Learning Objective	Training	Inference
Naïve Bayes	$\text{maximize} \sum_i \left[\begin{array}{l} \sum_j \log P(x_{ij} \mid y_i; \theta_j) \\ + \log P(y_i; \theta_0) \end{array} \right]$	$\theta_{kj} = \dfrac{\sum_i \delta(x_{ij} = 1 \wedge y_i = k) + r}{\sum_i \delta(y_i = k) + Kr}$	$\boldsymbol{\theta}_1^T \mathbf{x} + \boldsymbol{\theta}_0^T (1 - \mathbf{x}) > 0$ where $\theta_{1j} = \log \dfrac{P(x_j = 1 \mid y = 1)}{P(x_j = 1 \mid y = 0)}$, $\theta_{0j} = \log \dfrac{P(x_j = 0 \mid y = 1)}{P(x_j = 0 \mid y = 0)}$
Logistic Regression	$\text{maximize} \sum_i \log(P(y_i \mid \mathbf{x}, \boldsymbol{\theta})) + \lambda \lVert \boldsymbol{\theta} \rVert$ where $P(y_i \mid \mathbf{x}, \boldsymbol{\theta}) = 1/(1 + \exp(-y_i \boldsymbol{\theta}^T \mathbf{x}))$	Gradient ascent	$\boldsymbol{\theta}^T \mathbf{x} > 0$
Linear SVM	$\text{minimize } \lambda \sum_i \xi_i + \frac{1}{2} \lVert \boldsymbol{\theta} \rVert$ such that $y_i \boldsymbol{\theta}^T \mathbf{x} \geq 1 - \xi_i \quad \forall i$	Linear programming	$\boldsymbol{\theta}^T \mathbf{x} > 0$
Kernelized SVM	complicated to write	Quadratic programming	$\sum_i y_i \alpha_i K(\hat{\mathbf{x}}_i, \mathbf{x}) > 0$
Nearest Neighbor	most similar features → same label	Record data	y_i where $i = \underset{i}{\operatorname{argmin}}\ K(\hat{\mathbf{x}}_i, \mathbf{x})$

What to remember about classifiers

- No free lunch: machine learning algorithms are tools, not dogmas
- Try simple classifiers first
- Better to have smart features and simple classifiers than simple features and smart classifiers
- Use increasingly powerful classifiers with more training data (bias-variance tradeoff)

Some Machine Learning References

- General
  - Tom Mitchell, *Machine Learning*, McGraw Hill, 1997
  - Christopher Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995
- Adaboost
  - Friedman, Hastie, and Tibshirani, “Additive logistic regression: a statistical view of boosting”, Annals of Statistics, 2000
- SVMs
  - http://www.support-vector.net/icml-tutorial.pdf

What do we want classifiers to predict?

How should we represent objects?

What do we want to know about this object?

Recognition: describe, predict, or interact with the object based on visual cues

Can I **poke with it**?

Can I **put stuff in it**?

Is it **alive**?

What **shape** is it?

Is it **soft**?

Does it have a **tail**?

Will it **blend**?

Slide Credit: Torralba

The perception of function

- Direct perception (affordances): Gibson

Flat surface
Horizontal
Knee-high
…

→ Sittable upon

- Mediated perception (categorization)

Flat surface
Horizontal
Knee-high
…

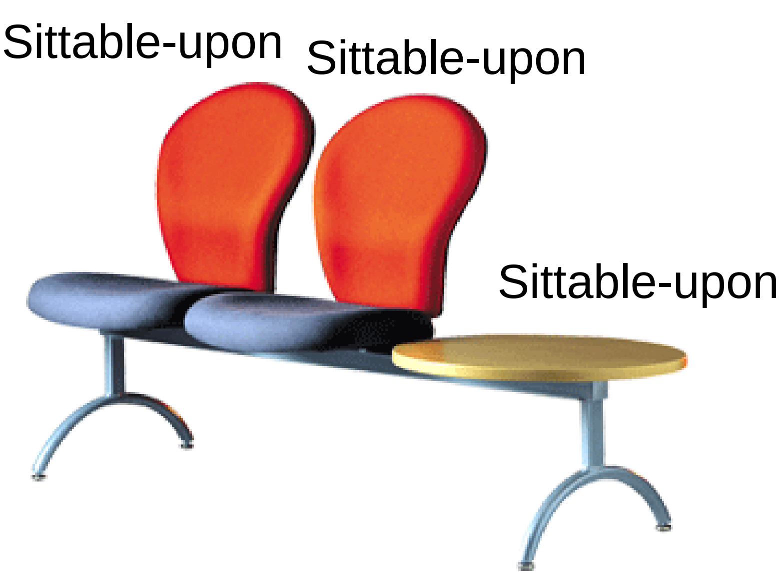
→ Chair → Sittable upon

Slide Credit: Torralba

Direct perception

Some aspects of an object's function can be perceived directly

- Functional form: Some forms clearly indicate to a function ("sittable-upon", container, cutting device, …)



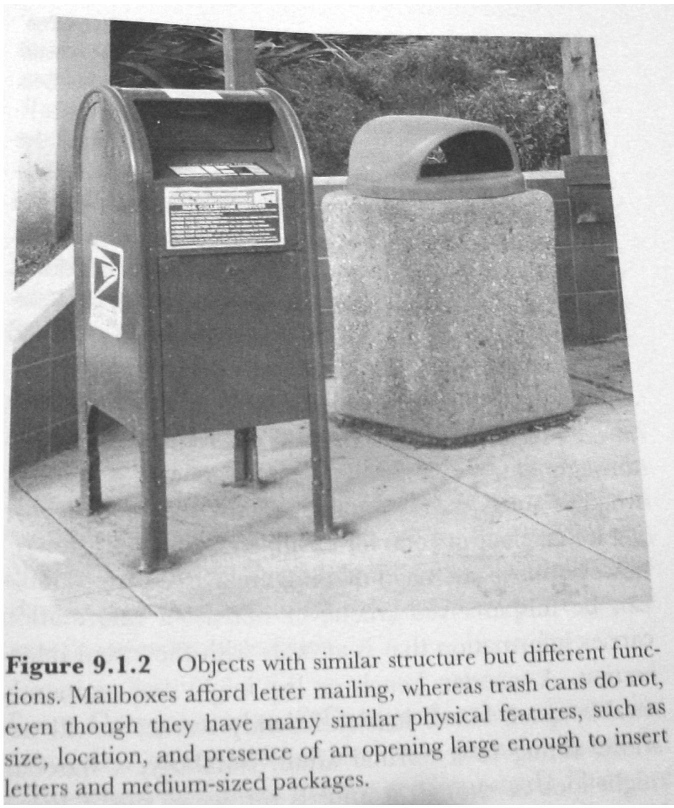
Slide Credit: Torralba

Direct perception

- Some aspects of an object function can be perceived directly
  - Observer relativity: Function is observer dependent

Limitations of Direct Perception

Objects of similar structure might have very different functions



Not all functions seem to be available from direct visual information only.

Slide Credit: Torralba

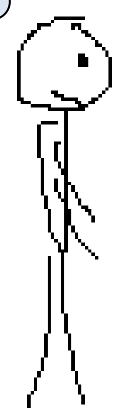
Limitations of Direct Perception

Propulsion system

Strong protective surface

Something that looks like a door

Sure, I can travel to space on this object



Visual appearance might be a very weak cue to function

Why do we care about categories?

From an object's category, we can make predictions about its behavior in the future, beyond of what is immediately perceived.

Slide Credit: Torralba

How do you define a category?

Prototype or Sum of Exemplars ?

Prototype Model

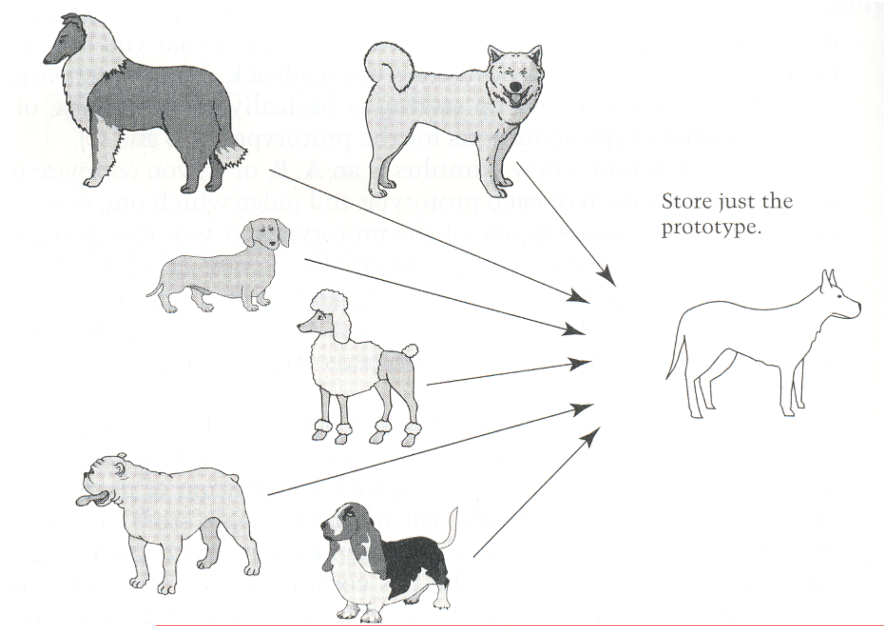


Figure 7.3. *Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.*

Category judgments are made
by comparing a new exemplar
to the prototype.

Exemplars Model

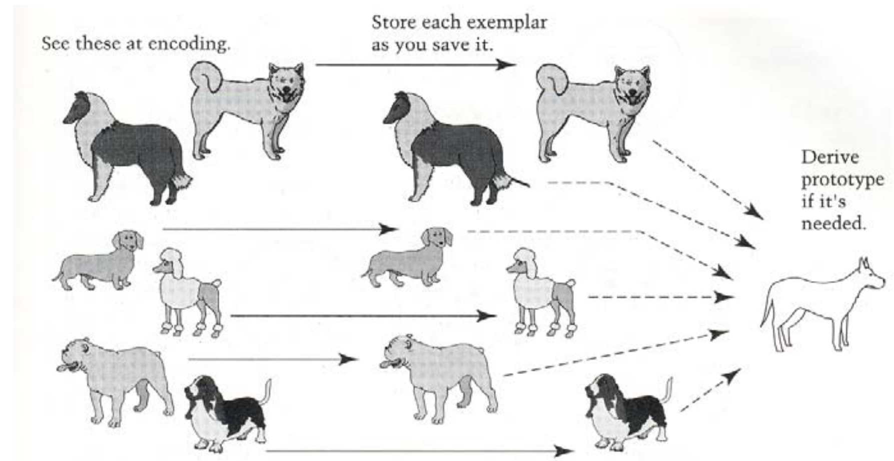
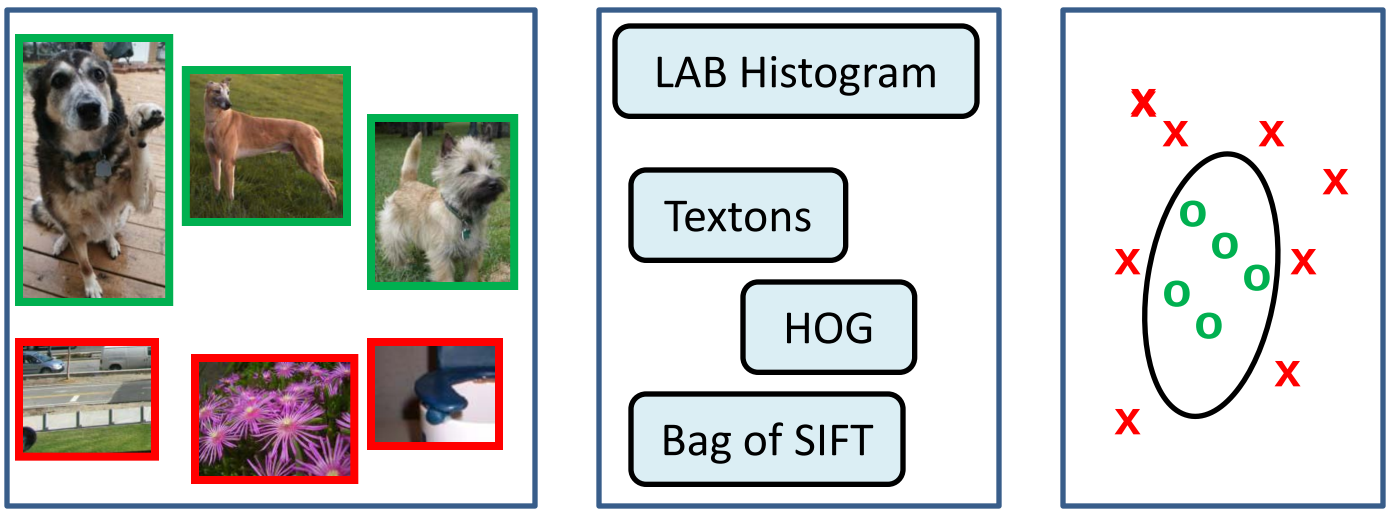


Figure 7.4. *Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.*

Category judgments are made
by comparing a new exemplar
to all the old exemplars of a category
or to the exemplar that is the most
appropriate

How do you define a category?

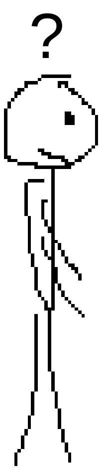
In computer vision:



Examples + Image Features + Classifier = Object Definition

Which level of categorization is the right one?

Car is an object composed of:
a few doors, four wheels (not all visible at all times), a roof, front lights, windshield



Slide Credit: Torralba

Levels of Categorization

SUPERORDINATE LEVEL CATEGORIES

↑

BASIC-LEVEL CATEGORIES

↓

SUBORDINATE LEVEL CATEGORIES

Table 7.3. ***Examples of Nested Category Structures***

Superordinate Level	Basic Level	Subordinate Level	
Musical instrument	Guitar	Folk guitar	Classical guitar
	Piano	Grand piano	Upright piano
Fruit	Peach	Freestone peach	Cling peach
	Grapes	Concord grapes	Green seedless grapes
Tree	Maple	Silver maple	Sugar maple
	Birch	River birch	White birch
	Oak	White oak	Red oak

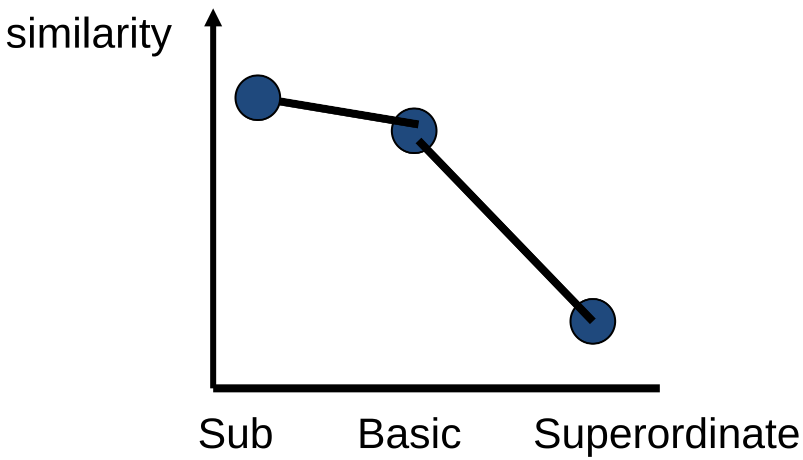
Rosch’s Levels of Categorization

Definition of Basic Level:

• **Similar shape**: Basic level categories are the highest-level category for which their members have similar shapes.

• **Similar motor interactions**: … for which people interact with its members using similar motor sequences.

• **Common attributes**: … there are a significant number of attributes in common between pairs of members.



Similarity declines only slightly going from subordinate to basic level, and then drops dramatically.

Levels of Categorization

- Rosch et al (1976) found that
  - People can tell whether an object belongs to a basic-level category faster
  - People tend to predict the basic category (e.g., “dog”) before superordinate (“animal”) or subordinate (“cocker-spaniel”) categories

- “Basic” could be different for different people (e.g., is “tree” basic, or “oak”?)

Entry-level categories
(Jolicoeur, Gluck, Kosslyn 1984)

- Typical member of a basic-level category are categorized at the expected level
- Atypical members tend to be classified at a subordinate level.

A bird

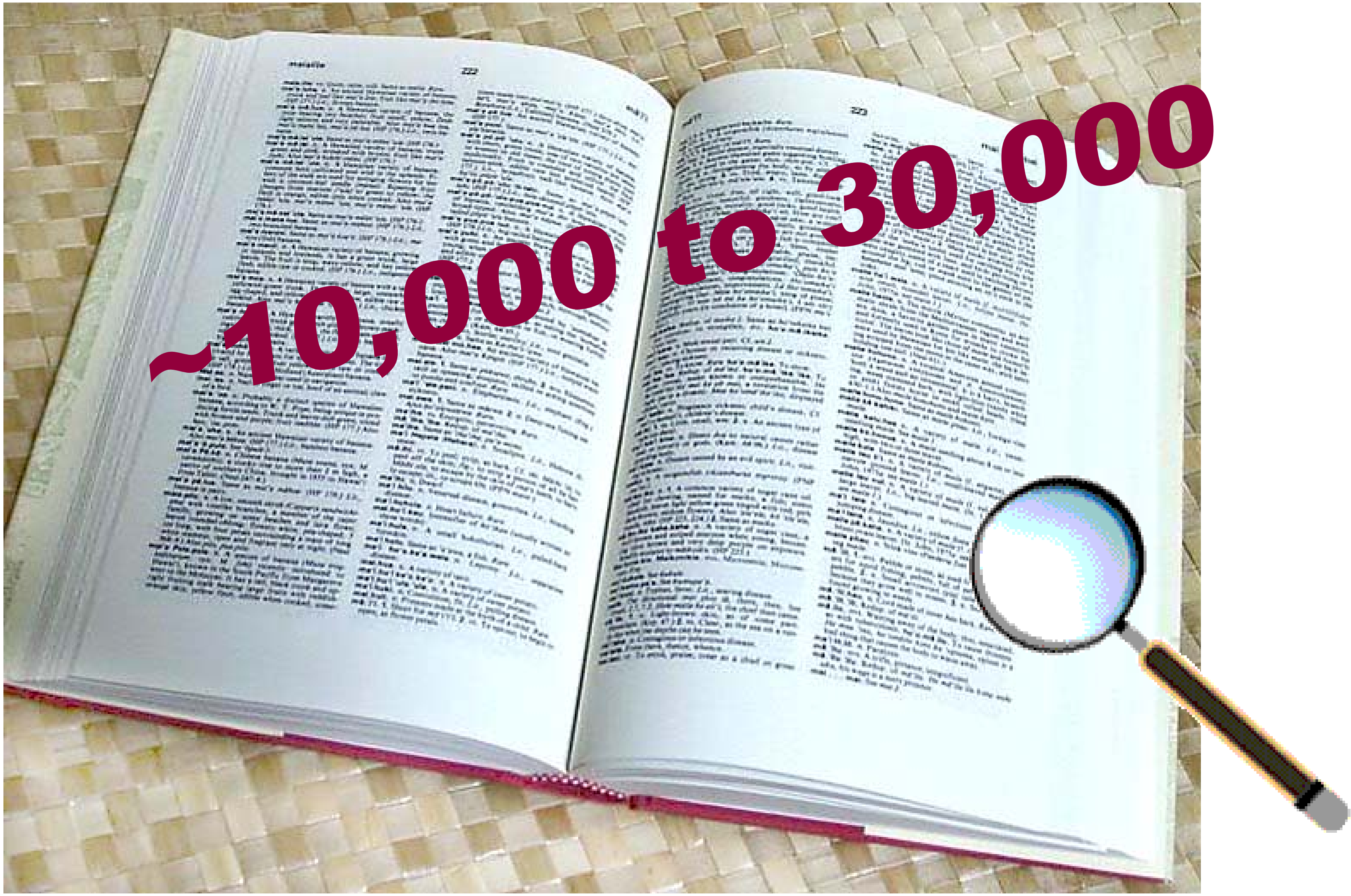

An ostrich

How many categories?

Slide by Aude Oliva

Many

How many object categories are there?



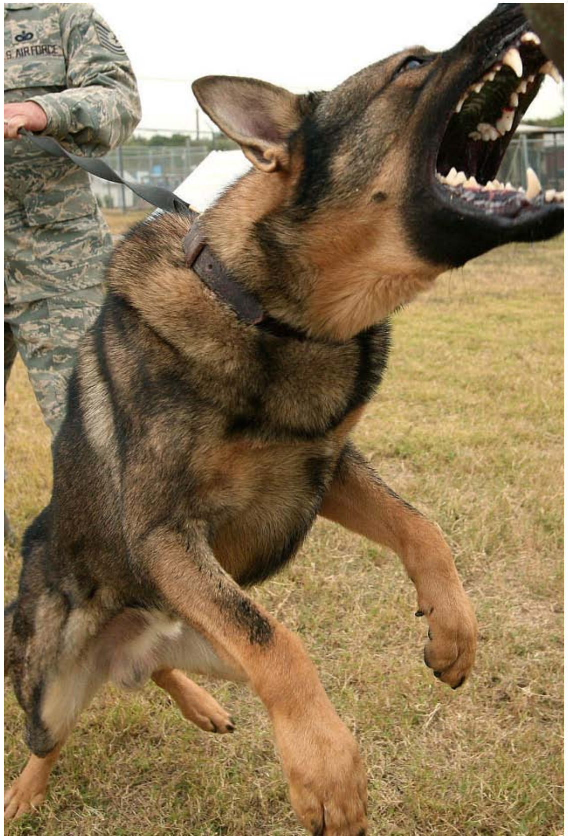
Biederman 1987

How many categories?

- An infinite number ("the kind of person who would wear a yellow hat")… but not all are useful

Beyond categories… a property-based view of recognition

1. We want detailed information about objects



"Dog"
vs.
"Large, angry animal with pointy teeth"

2. We want to be able to infer something about unfamiliar objects

Familiar Objects

New Object

2. We want to be able to infer something about unfamiliar objects

If we can infer category names…

Familiar Objects

New Object

Cat

Horse

Dog

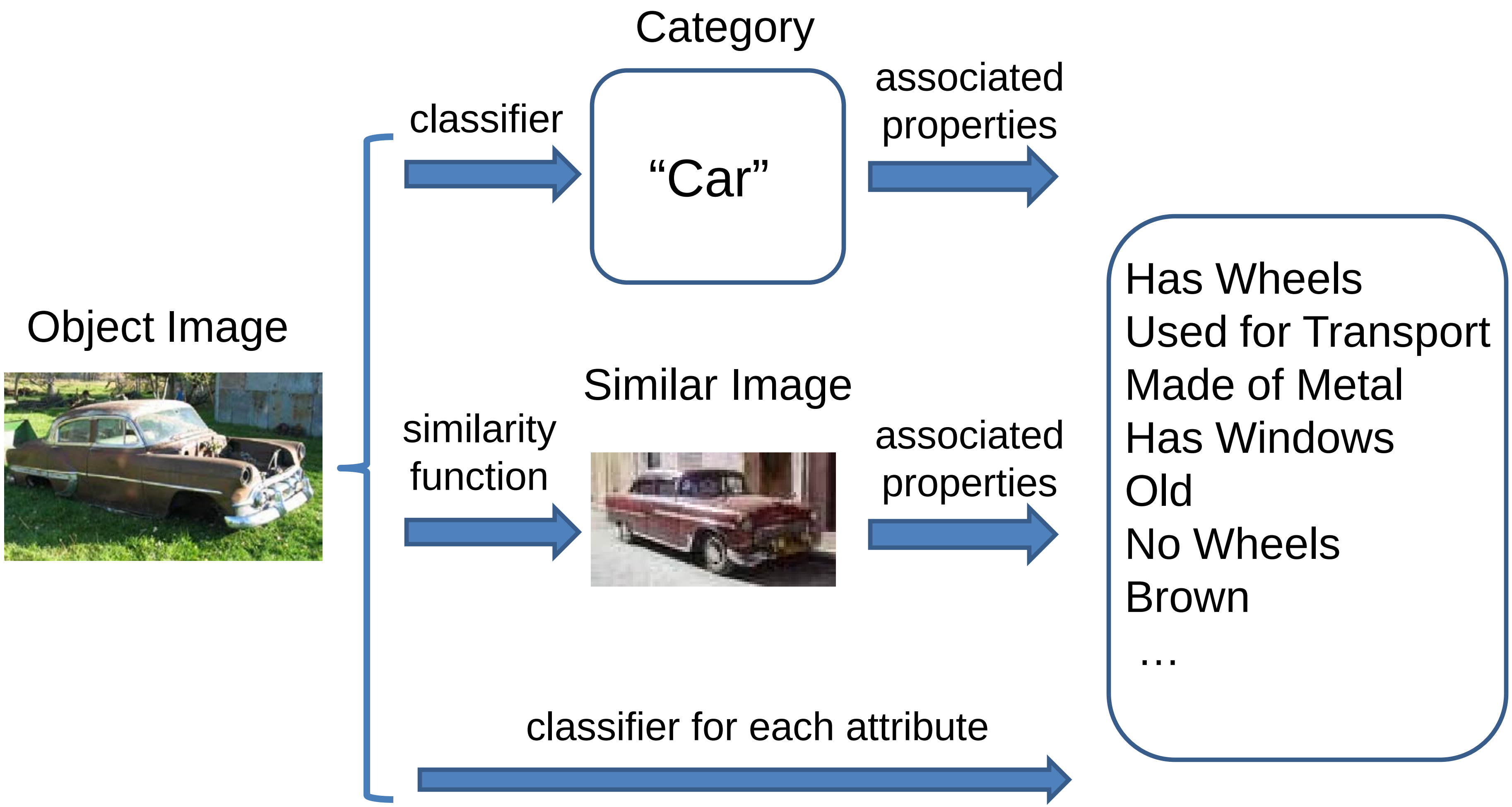
???

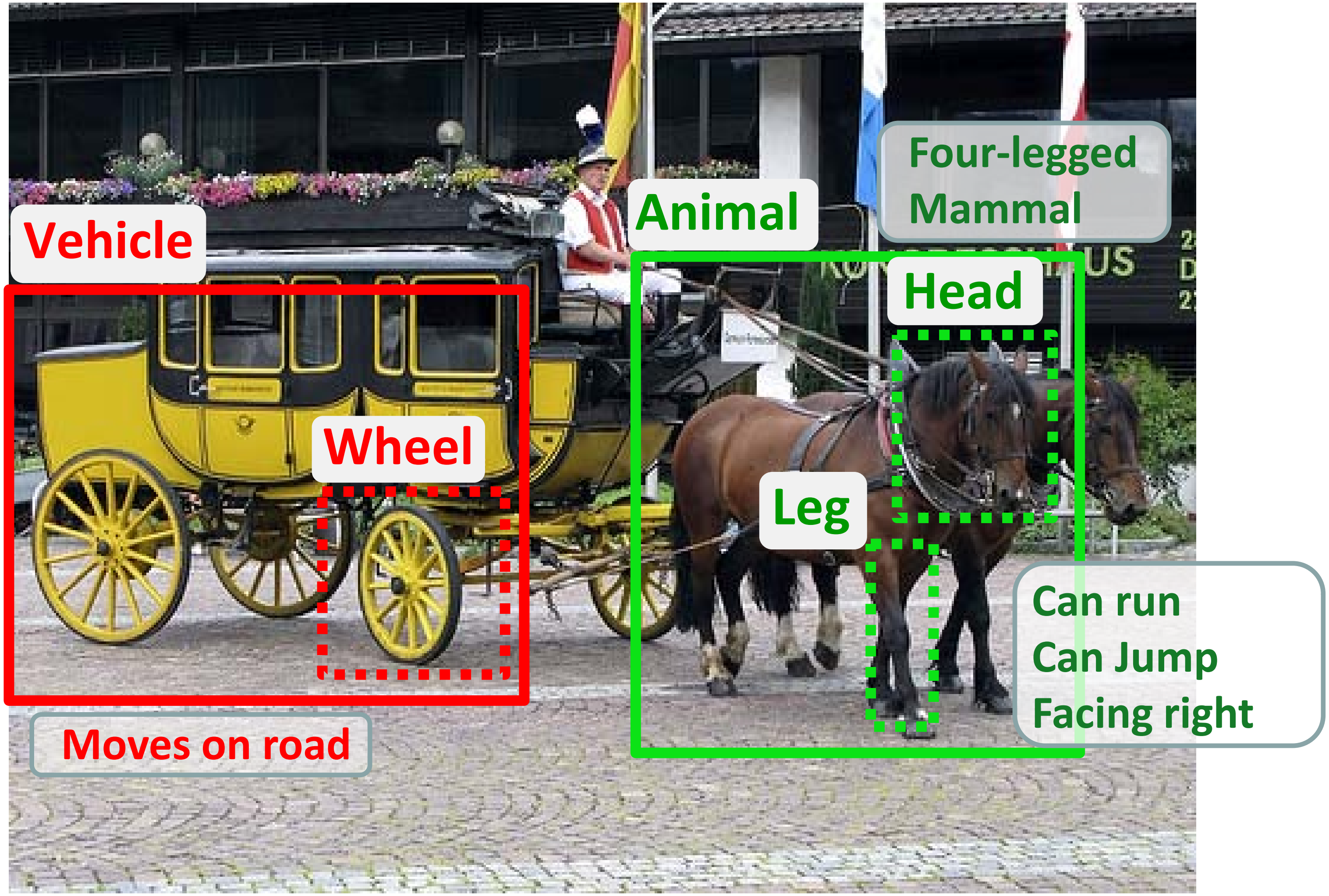
3. We want to make comparisons between objects or categories

What is unusual about this dog?

What is the difference between horses and zebras?

All three strategies are important





Farhadi Endres Hoiem 2010

Next class

- Sliding window detectors