CS 3710: Visual Recognition
Classification and Detection

Adriana Kovashka
Department of Computer Science

January 13, 2015

Plan for Today

- Visual recognition basics part 2: Classification and detection
- Adriana's research
- Next time: First student presentation

Classification vs Detection

- Classification
  - Given an image or an image region, determine which of *N* categories it represents
- Detection
  - Determine where in the image a category is to be found

Classification

Machine Learning Problems

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

James Hays

The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

Svetlana Lazebnik

The machine learning framework

$$y = f(\mathbf{x})$$

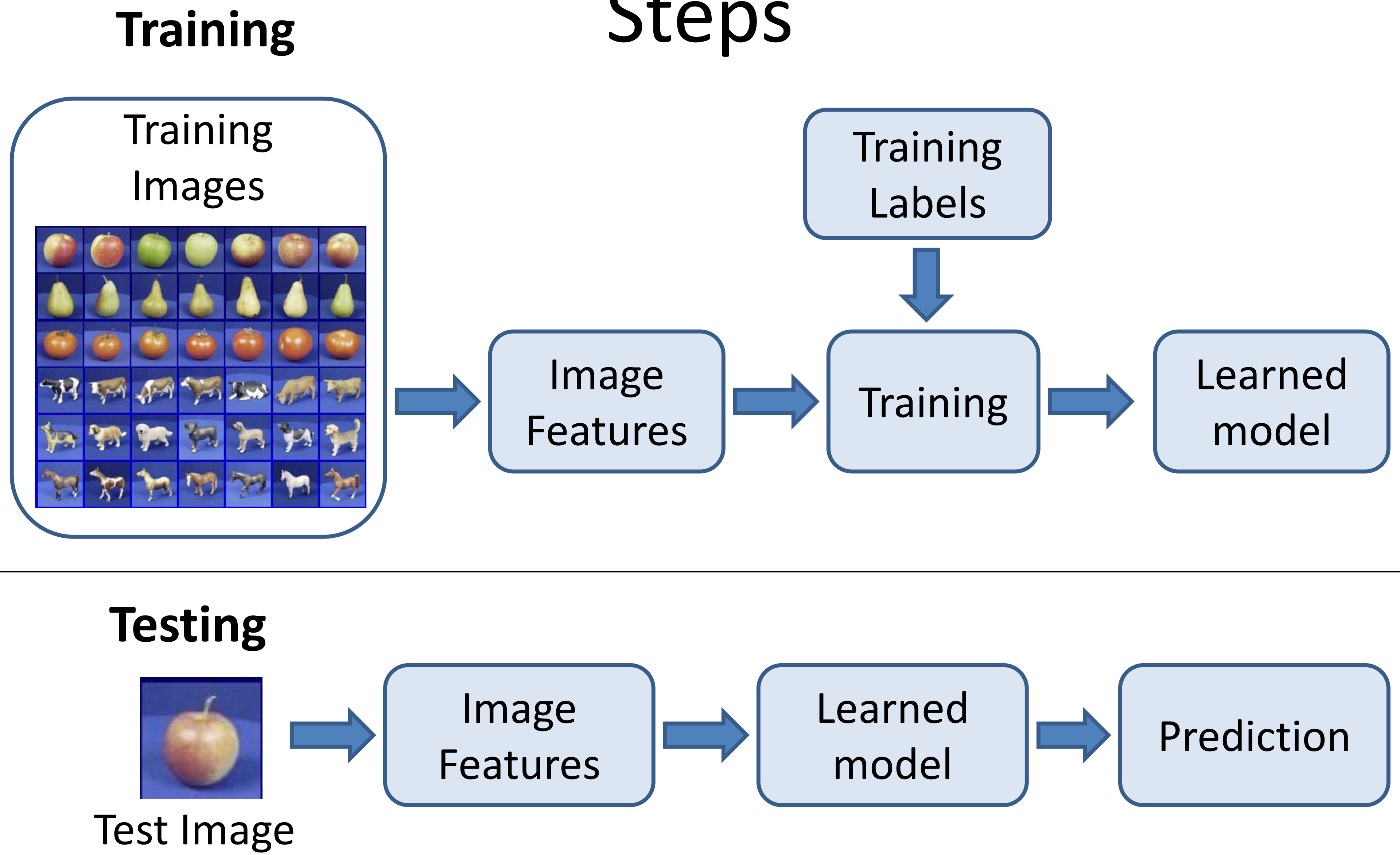
output — prediction function — image features

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set

- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Svetlana Lazebnik

Steps

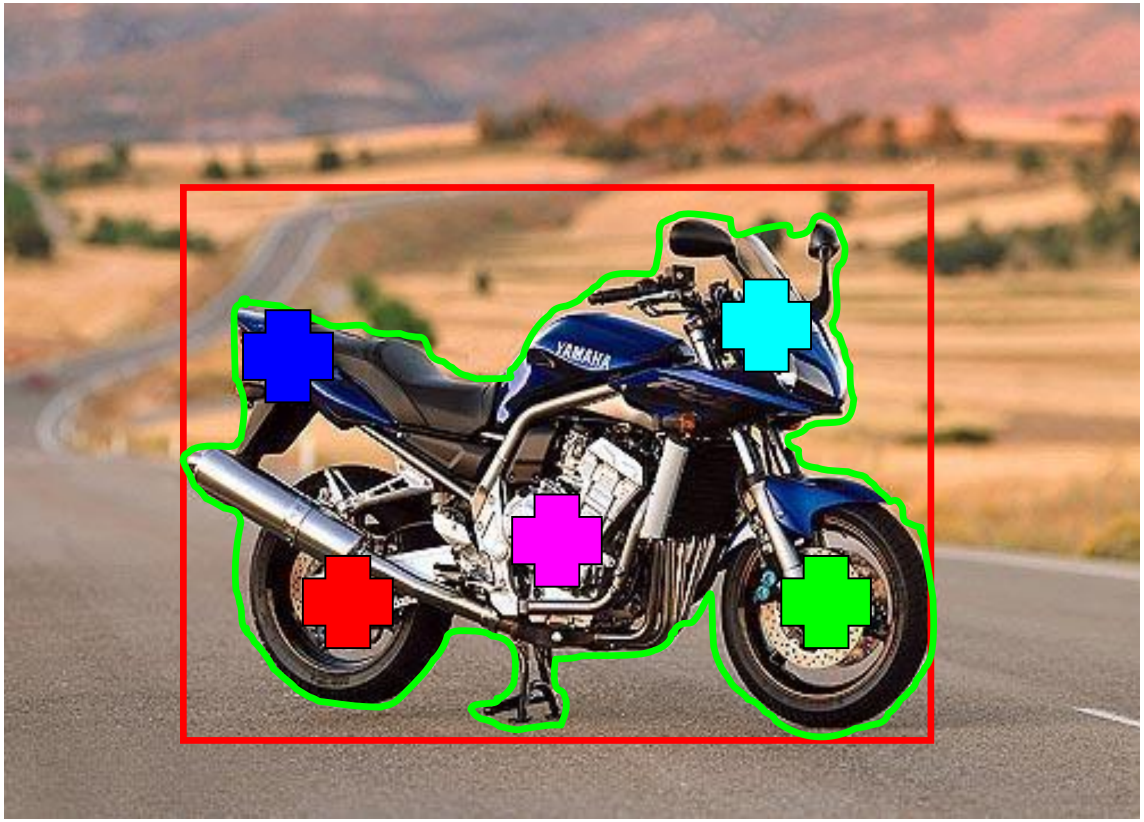


Derek Hoiem, Svetlana Lazebnik

Recognition task and supervision

- Images in the training set must be annotated with the “correct answer” that the model is expected to produce

“Contains a motorbike”



Svetlana Lazebnik

Generalization

- How well does a learned model generalize from the data it was trained on to a new test set?

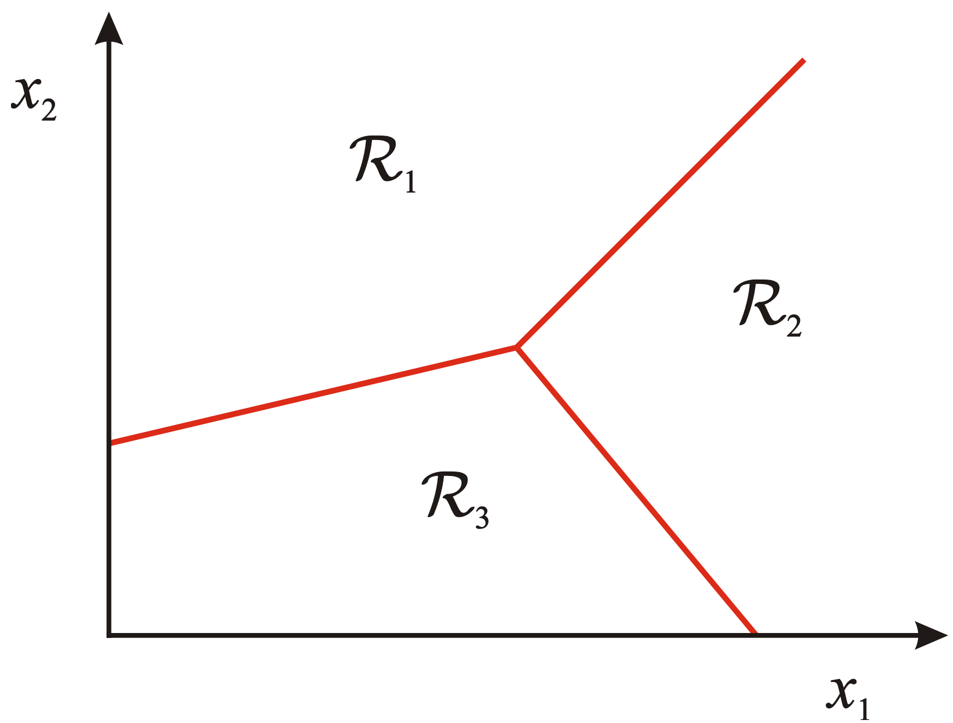
Training set (labels known)

Test set (labels unknown)

Svetlana Lazebnik

Classification

- Assign input vector to one of two or more classes
- Any decision rule divides the input space into *decision regions* separated by *decision boundaries*



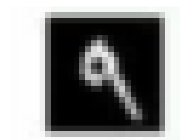
Svetlana Lazebnik

Supervised classification

- Given a collection of *labeled* examples, come up with a function that will predict the labels of new examples.

"four"

"nine"



?

Training examples

Novel input

- How good is some function that we come up with to do the classification?
- Depends on
  - Mistakes made
  - Cost associated with the mistakes

Kristen Grauman

Supervised classification

- Given a collection of *labeled* examples, come up with a function that will predict the labels of new examples.

- Consider the two-class (binary) decision problem
  - L(4→9): Loss of classifying a 4 as a 9
  - L(9→4): Loss of classifying a 9 as a 4

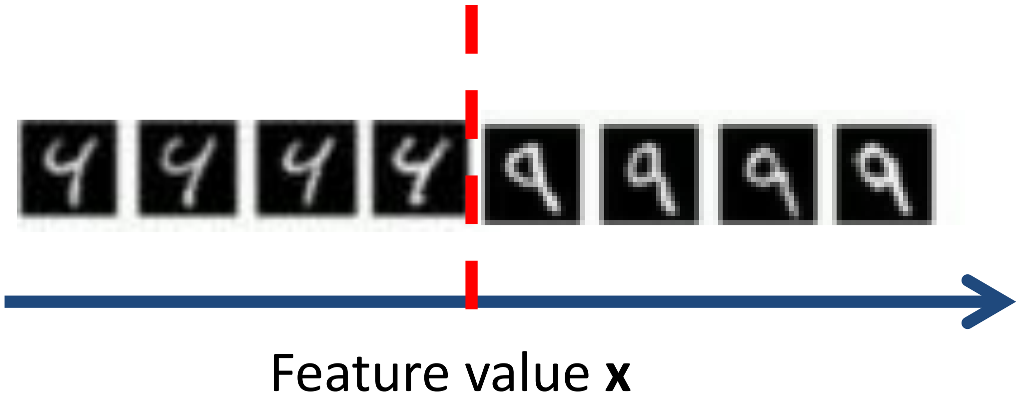
- **Risk** of a classifier *s* is expected loss:

$$R(s) = \Pr(4 \to 9 \mid \text{using } s) L(4 \to 9) + \Pr(9 \to 4 \mid \text{using } s) L(9 \to 4)$$

- We want to choose a classifier so as to minimize this total risk

Kristen Grauman

Supervised classification



Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

If we choose class "four" at boundary, expected loss is:

$$= P(\text{class is } 9 \mid \mathbf{x})\, L(9 \rightarrow 4) + \cancel{P(\text{class is } 4 \mid \mathbf{x}) L(4 \rightarrow 4)}$$

If we choose class "nine" at boundary, expected loss is:

$$= P(\text{class is } 4 \mid \mathbf{x})\, L(4 \rightarrow 9)$$

So, best decision boundary is at point **x** where

$$P(\text{class is } 9 \mid \mathbf{x})\, L(9 \rightarrow 4) = \mathrm{P}(\text{class is } 4 \mid \mathbf{x}) L(4 \rightarrow 9)$$

Kristen Grauman

Supervised classification

Feature value **x**

Optimal classifier will minimize total risk.

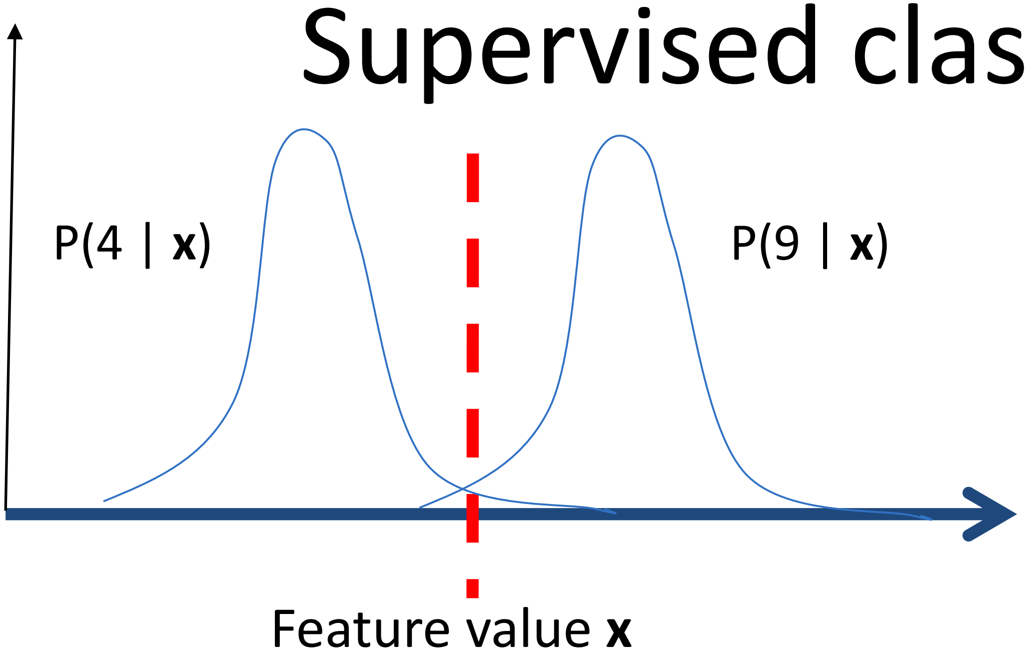
At decision boundary, either choice of label yields same expected loss.

To classify a new point, choose class with lowest expected loss; i.e., choose “four” if

$$\underbrace{P(9 \mid \mathbf{x}) L(9 \rightarrow 4)}_{\text{Loss for choosing "four"}} < \underbrace{P(4 \mid \mathbf{x}) L(4 \rightarrow 9)}_{\text{Loss for choosing "nine"}}$$

Kristen Grauman

Supervised classification



Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

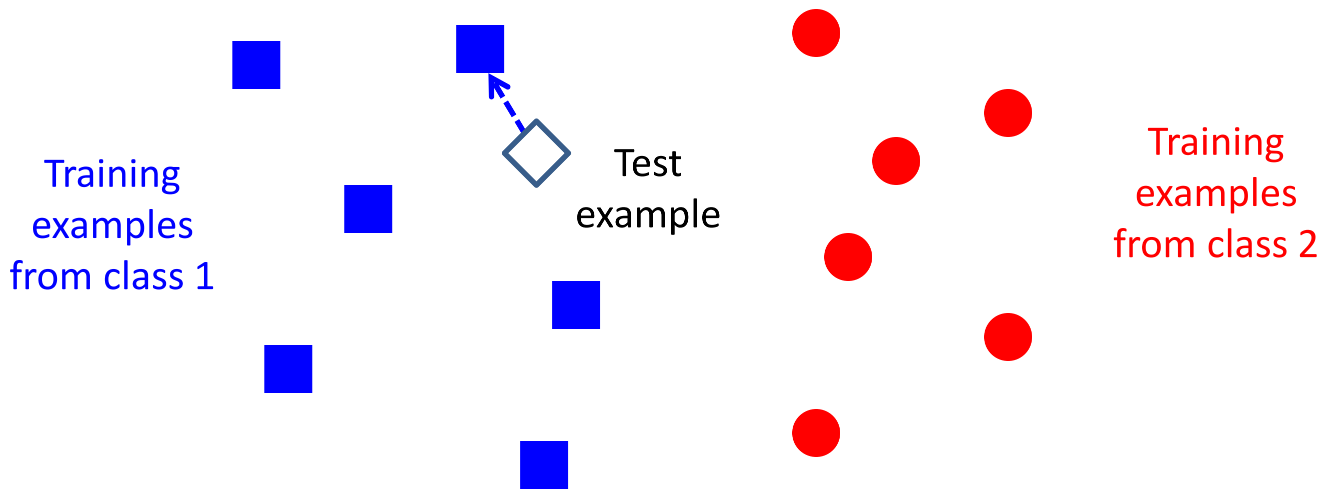
To classify a new point, choose class with lowest expected loss; i.e., choose "four" if

$$\underbrace{\boxed{P(9\,|\,\mathbf{x})}L(9 \rightarrow 4)}_{\text{Loss for choosing "four"}} < \underbrace{\boxed{P(4\,|\,\mathbf{x})}L(4 \rightarrow 9)}_{\text{Loss for choosing "nine"}}$$

How to evaluate these probabilities?

Kristen Grauman

Classifiers: Nearest neighbor

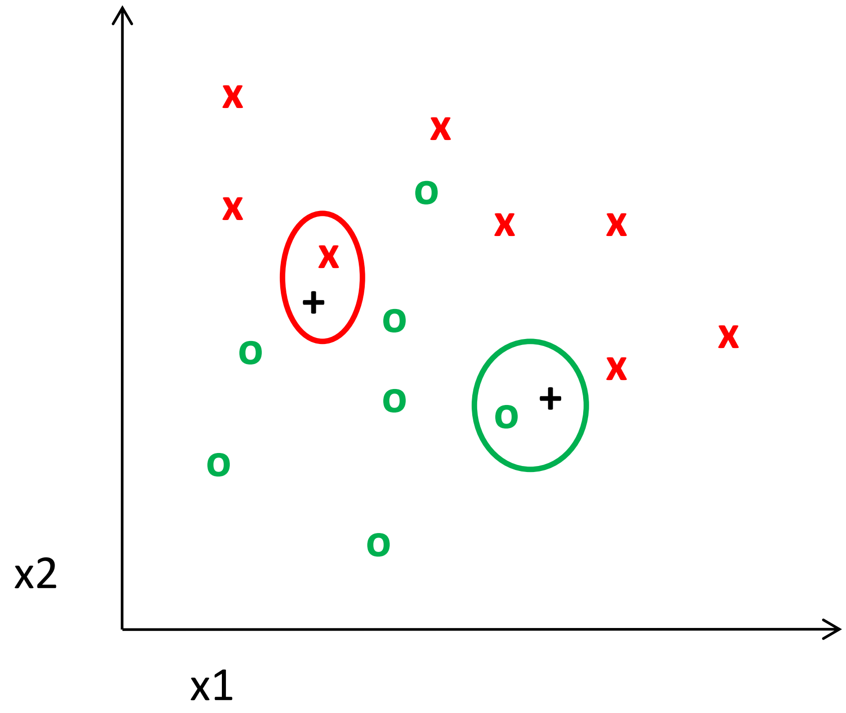


f(**x**) = label of the training example nearest to **x**

- All we need is a distance function for our inputs
- No training required!

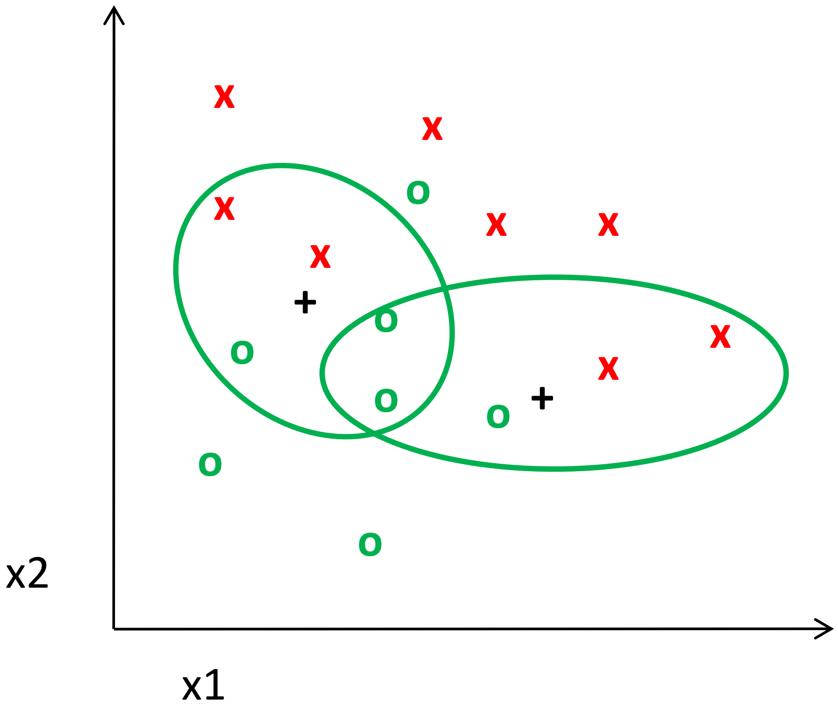
Svetlana Lazebnik

1-nearest neighbor



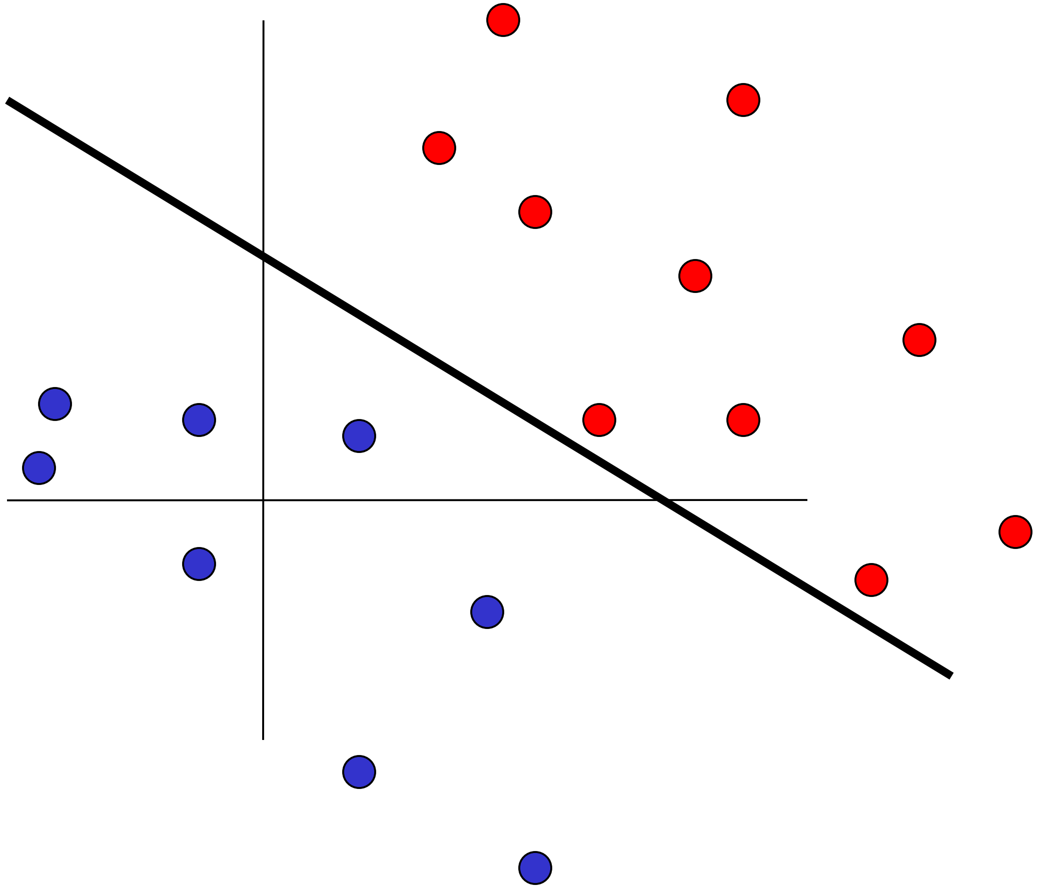
James Hays

5-nearest neighbor



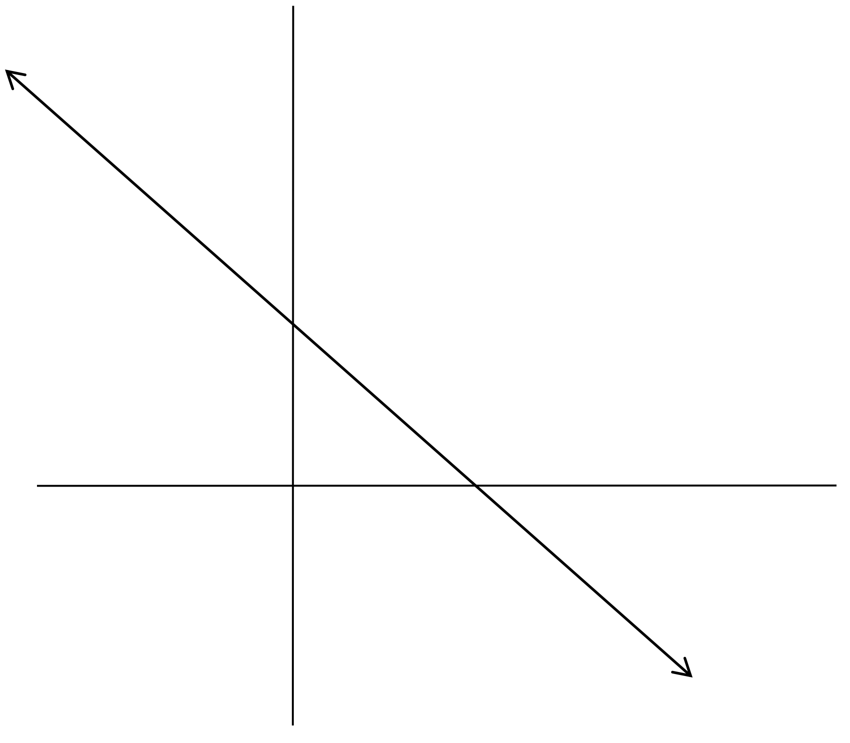
James Hays

Linear classifiers



C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Lines in R^2

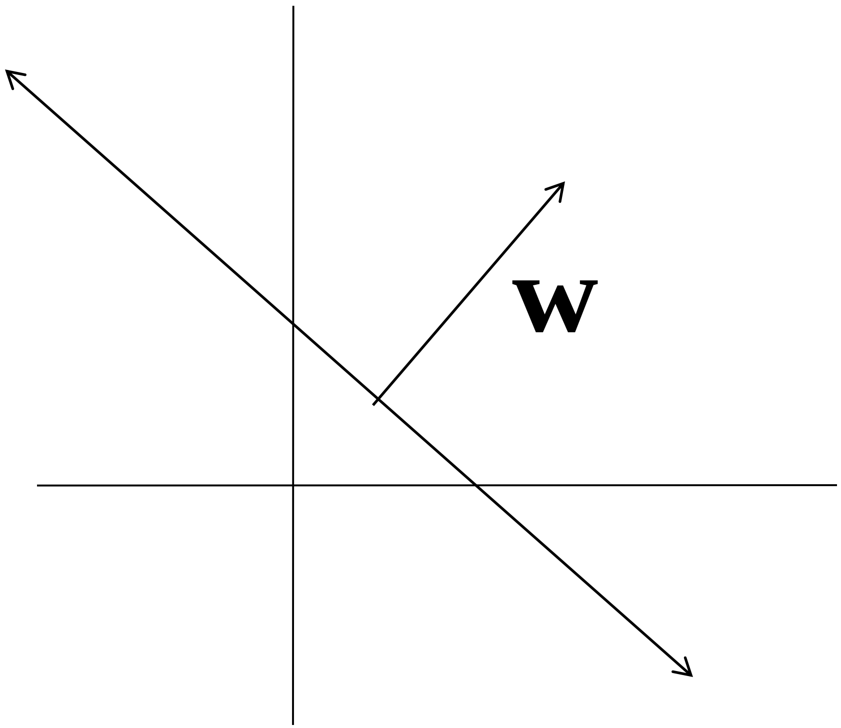


Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

Kristen Grauman

Lines in R^2



Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

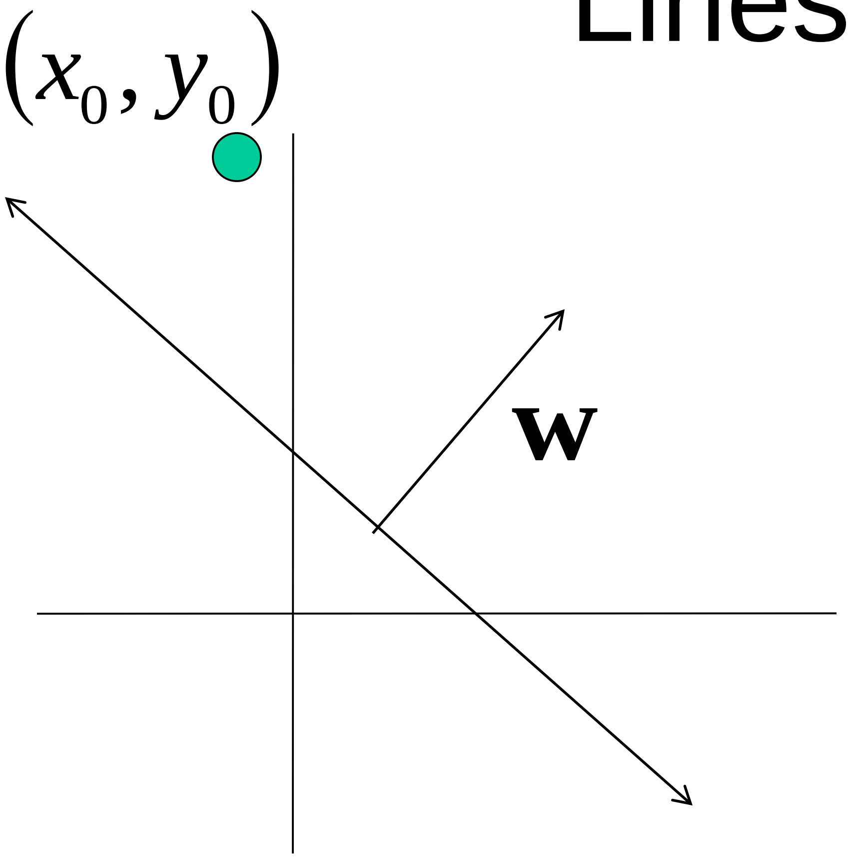
$$ax + cy + b = 0$$

$$\Updownarrow$$

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

Kristen Grauman

Lines in R^2



Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

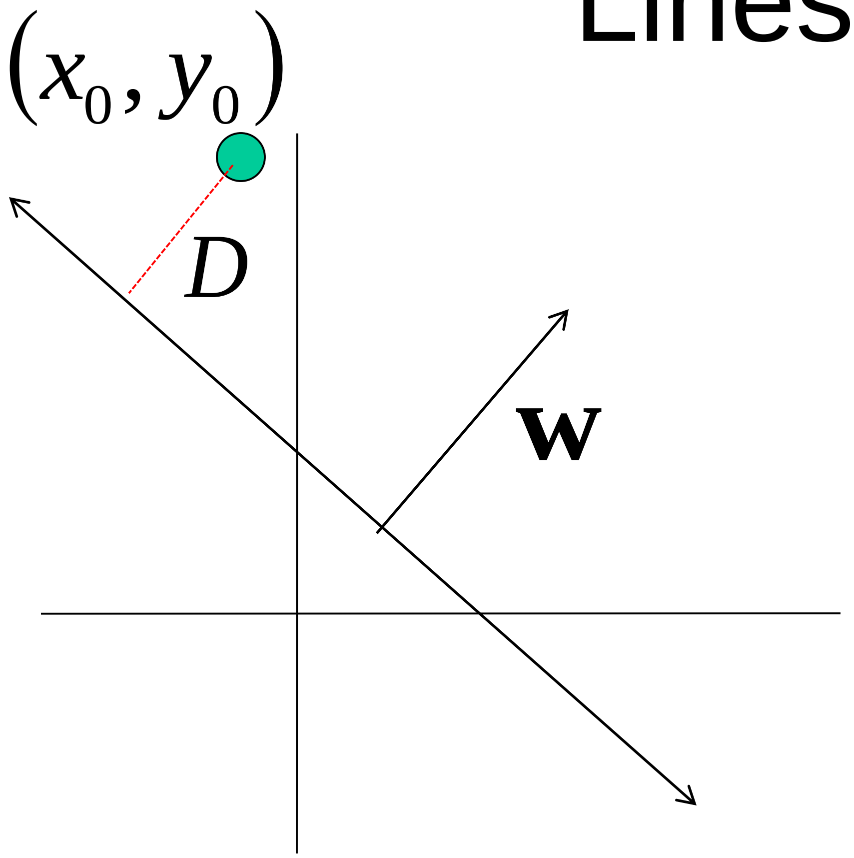
$$ax + cy + b = 0$$

$$\Updownarrow$$

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

Kristen Grauman

Lines in R^2



Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\Updownarrow$$

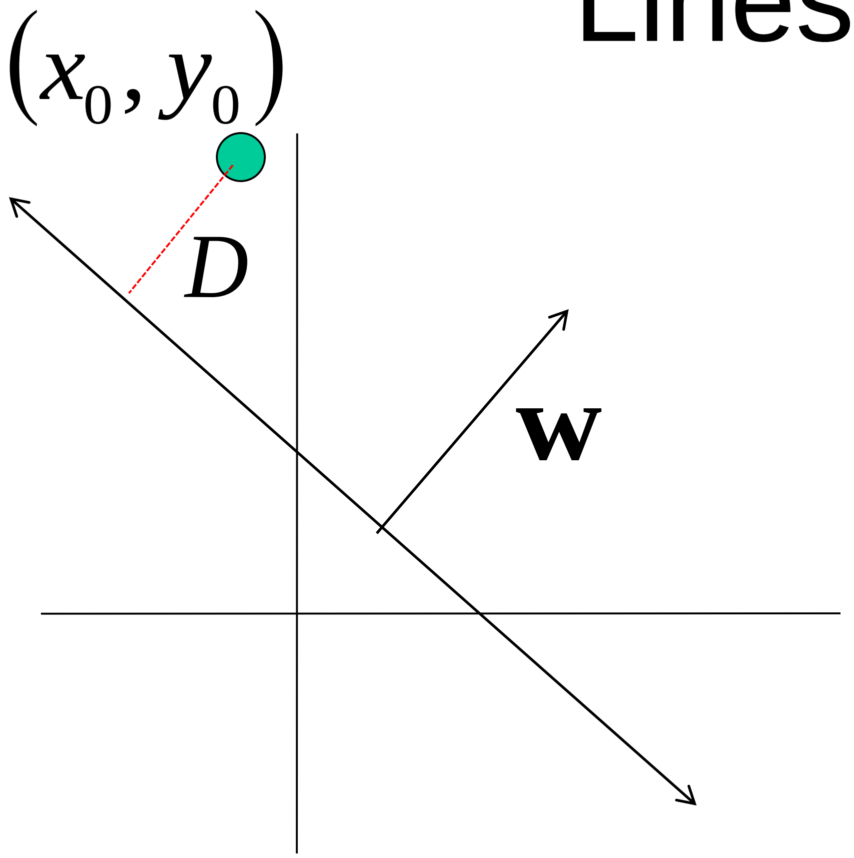
$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}}$$

} distance from point to line

Kristen Grauman

Lines in R^2



Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\Updownarrow$$

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

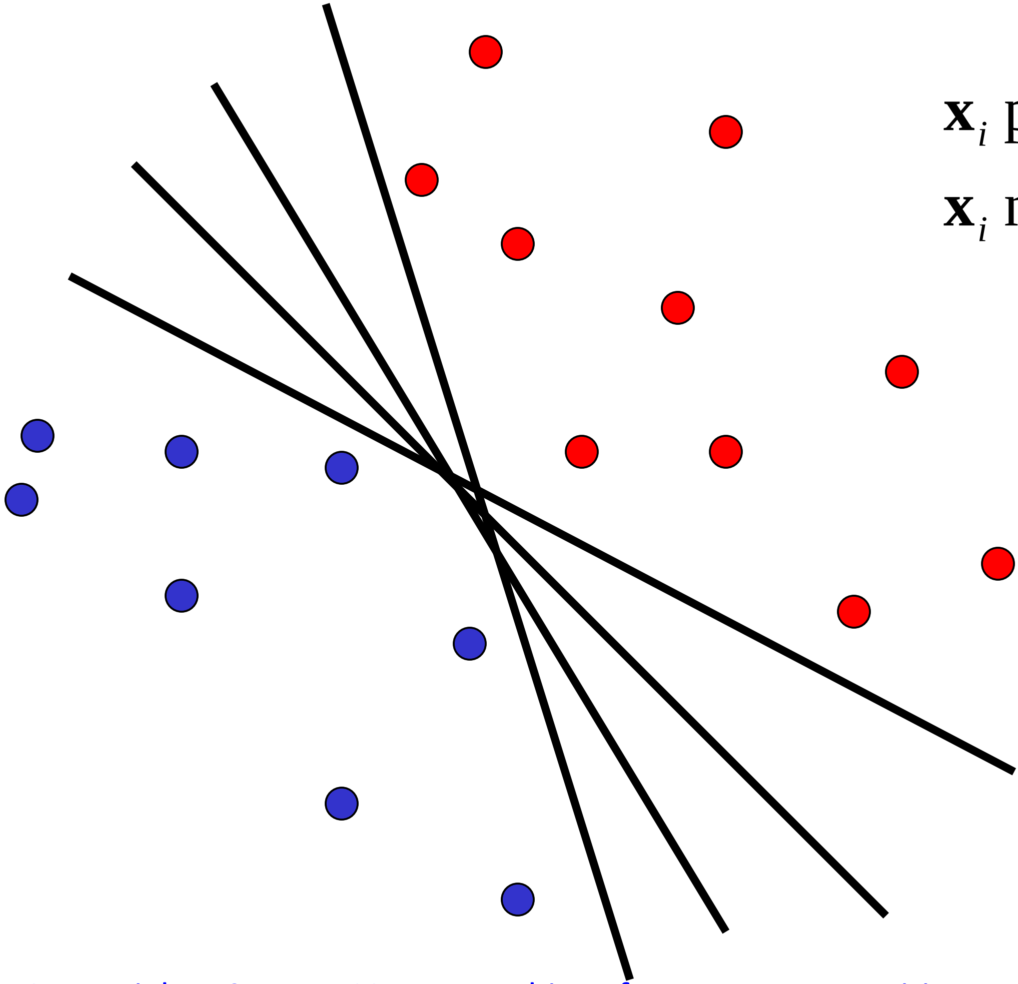
$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}} = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x} + b}{\|\mathbf{w}\|}$$

distance from point to line

Kristen Grauman

Linear classifiers

- Find linear function to separate positive and negative examples



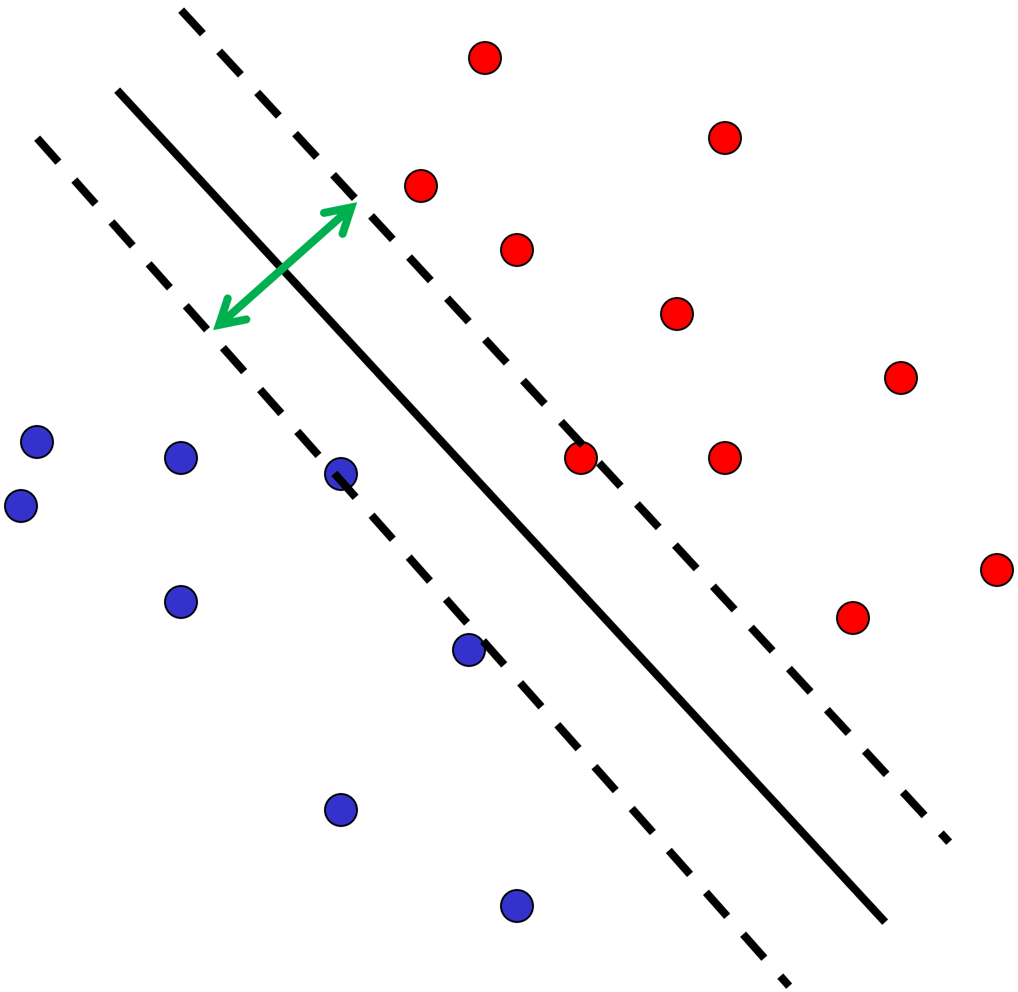
$\mathbf{x}_i$ positive : $\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

$\mathbf{x}_i$ negative : $\mathbf{x}_i \cdot \mathbf{w} + b < 0$

Which line is best?

C. Burges, [A Tutorial on Support Vector Machines for Pattern Recognition](), Data Mining and Knowledge Discovery, 1998

Support vector machines



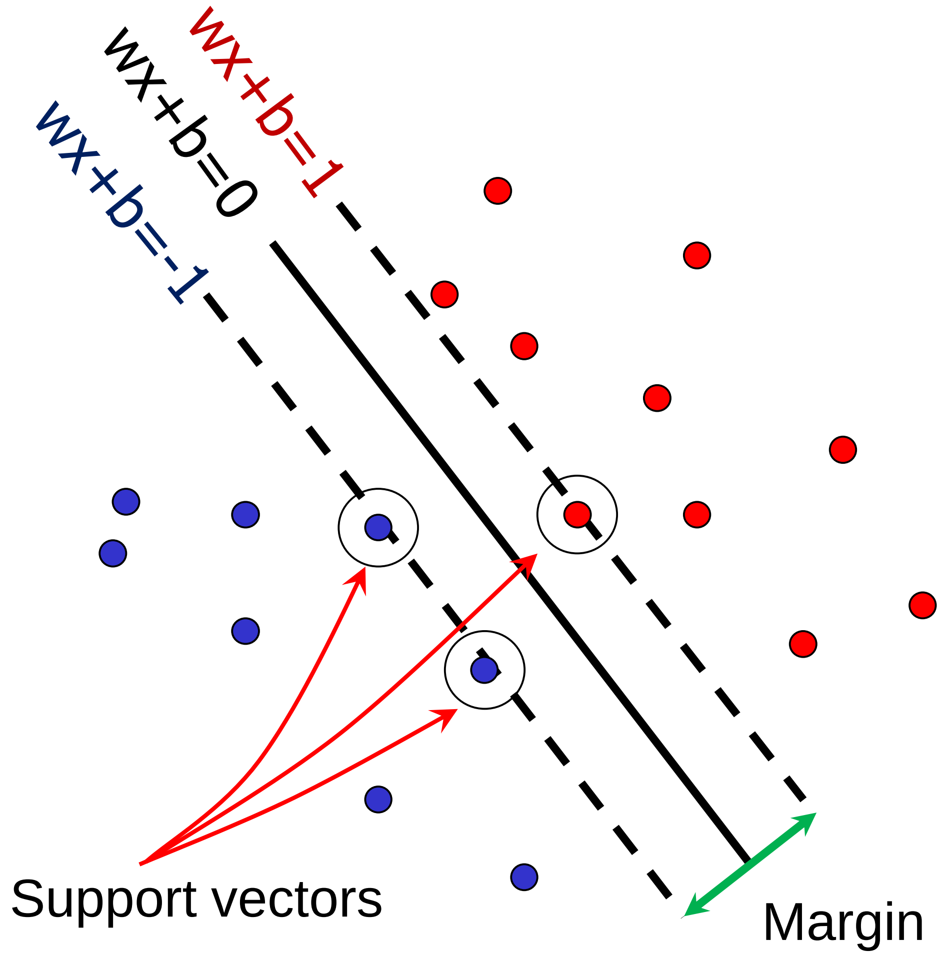
- Discriminative classifier based on *optimal separating line (for 2d case)*

- Maximize the *margin* between the positive and negative training examples

C. Burges, [A Tutorial on Support Vector Machines for Pattern Recognition](), Data Mining and Knowledge Discovery, 1998

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

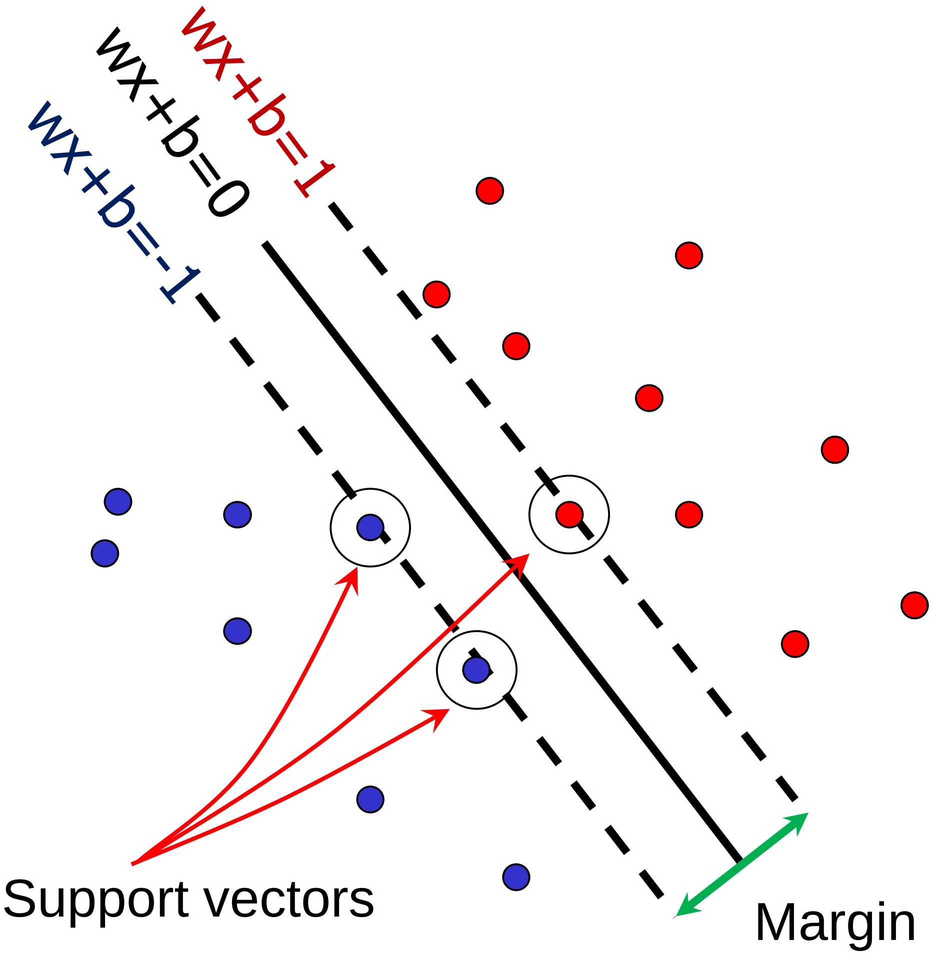
$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

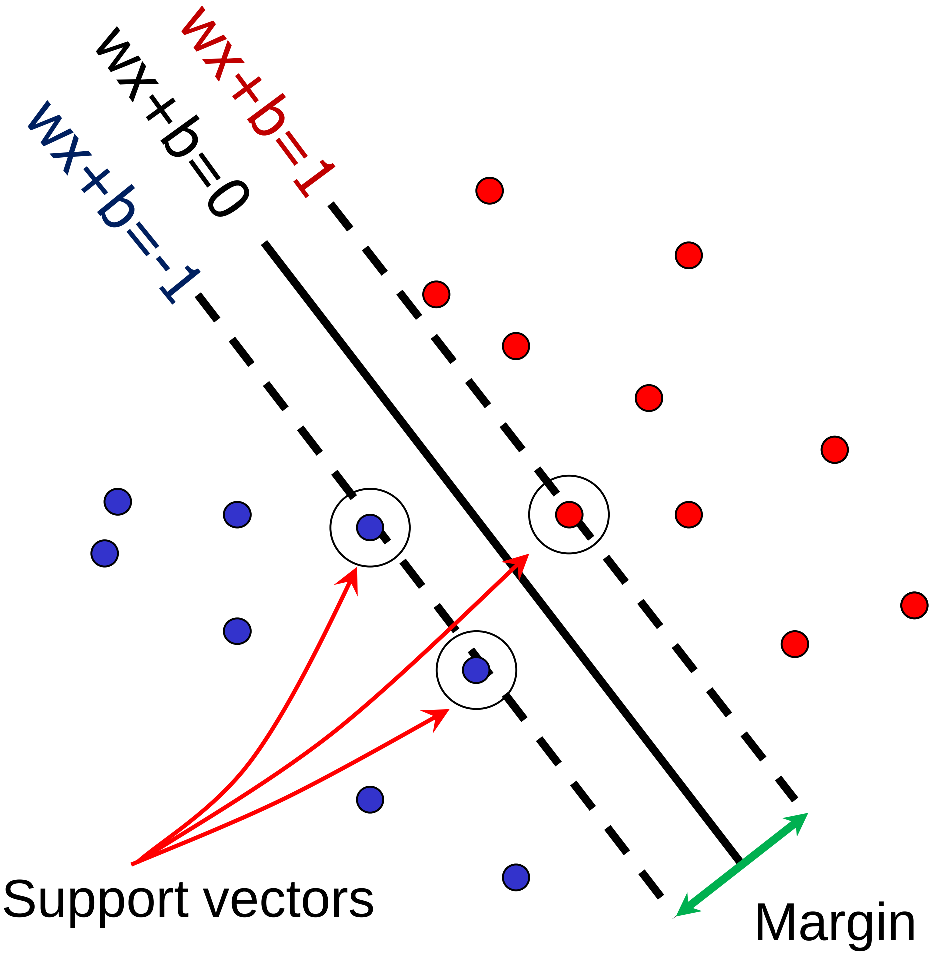
For support vectors:

$$\frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|} \qquad M = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{||\mathbf{w}||}$

Therefore, the margin is $2 / ||\mathbf{w}||$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Finding the maximum margin line

1. Maximize margin $2/||\mathbf{w}||$
2. Correctly classify all training data points:

$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

Quadratic optimization problem:

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$

One constraint for each training point.

Note sign trick.

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

Learned weight

Support vector

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ (for any support vector)

- Classification function:

$$f(x) = \text{sign}\ (\mathbf{w} \cdot \mathbf{x} + \mathrm{b})$$
$$= \text{sign}\left(\sum_{\mathrm{i}} \alpha_{\mathrm{i}} y_i \mathbf{x}_i \cdot \mathbf{x} + b\right)$$

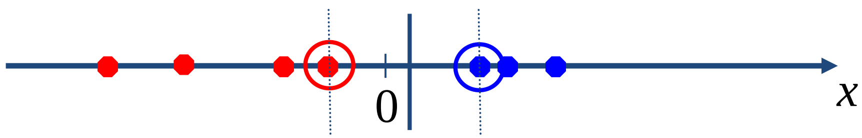
If f(x) < 0, classify as negative, otherwise classify as positive.

- Notice that it relies on an *inner product* between the test point ***x*** and the support vectors $\boldsymbol{x}_i$
- (Solving the optimization problem also involves computing the inner products $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ between all pairs of training points)

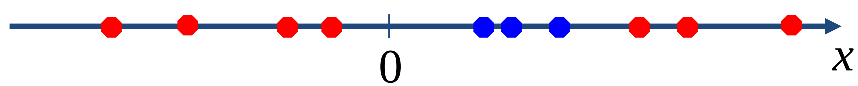
C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Nonlinear SVMs

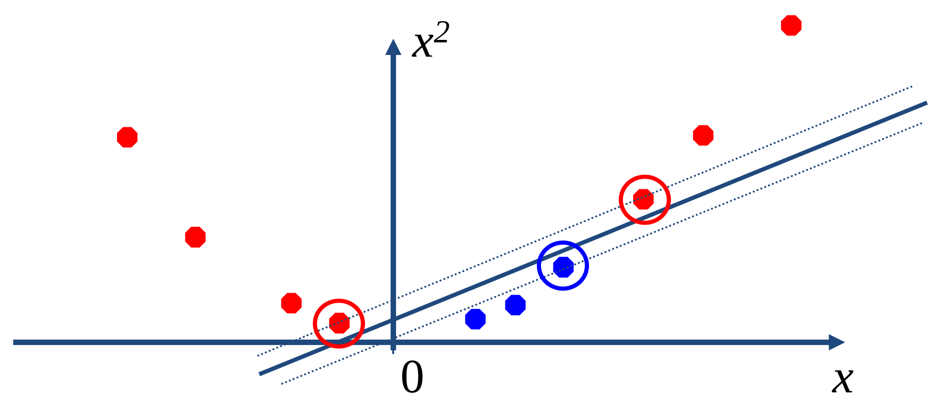
- Datasets that are linearly separable work out great:



- But what if the dataset is just too hard?



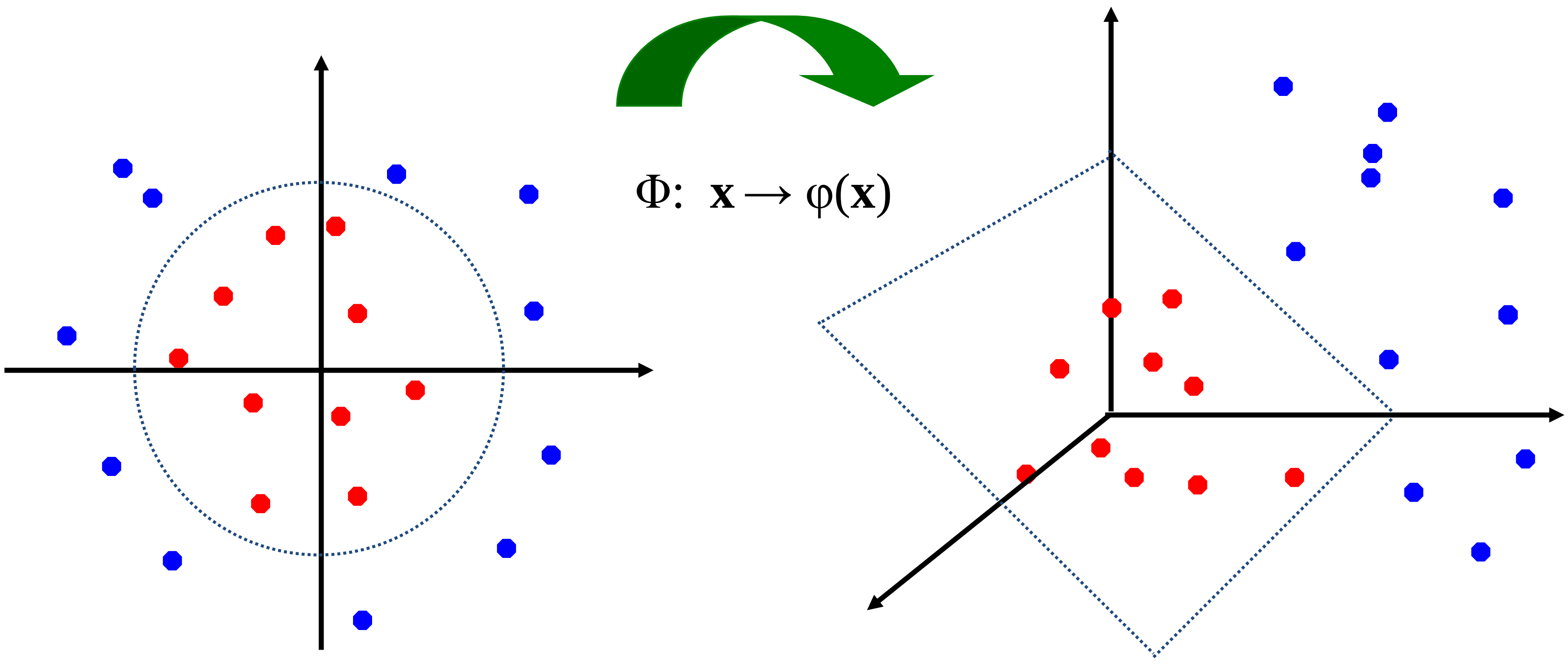
- We can map it to a higher-dimensional space:



Andrew Moore

Nonlinear SVMs

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Andrew Moore

Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\boldsymbol{\varphi}(\mathbf{x})$, define a kernel function K such that

  $$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j)$$

Examples of kernel functions

- Linear: $$K(x_i, x_j) = x_i^T x_j$$

- Gaussian RBF: $$K(x_i, x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2}\right)$$

- Histogram intersection: $$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

Summary: SVMs for image classification

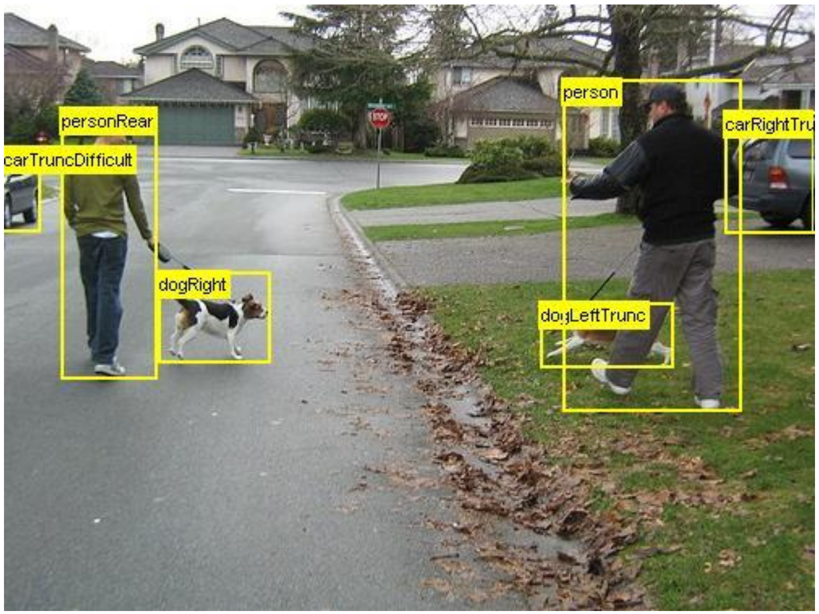
1. Pick an image representation
2. Pick a kernel function for that representation
3. Compute the matrix of kernel values between every pair of training examples
4. Feed the kernel matrix into your favorite SVM solver to obtain support vectors and weights
5. At test time: compute kernel values for your test example and each support vector, and combine them with the learned weights to get the value of the decision function

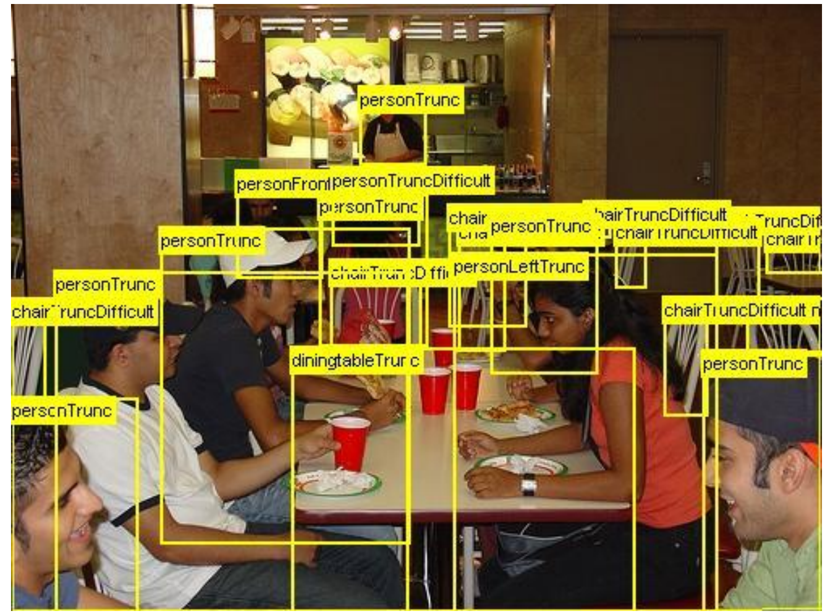
What about multi-class SVMs?

- Unfortunately, there is no “definitive” multi-class SVM formulation
- In practice, we have to obtain a multi-class SVM by combining multiple two-class SVMs
- One vs. others
  - Traning: learn an SVM for each class vs. the others
  - Testing: apply each SVM to the test example, and assign it to the class of the SVM that returns the highest decision value
- One vs. one
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM “votes” for a class to assign to the test example

Detection

PASCAL Visual Object Challenge





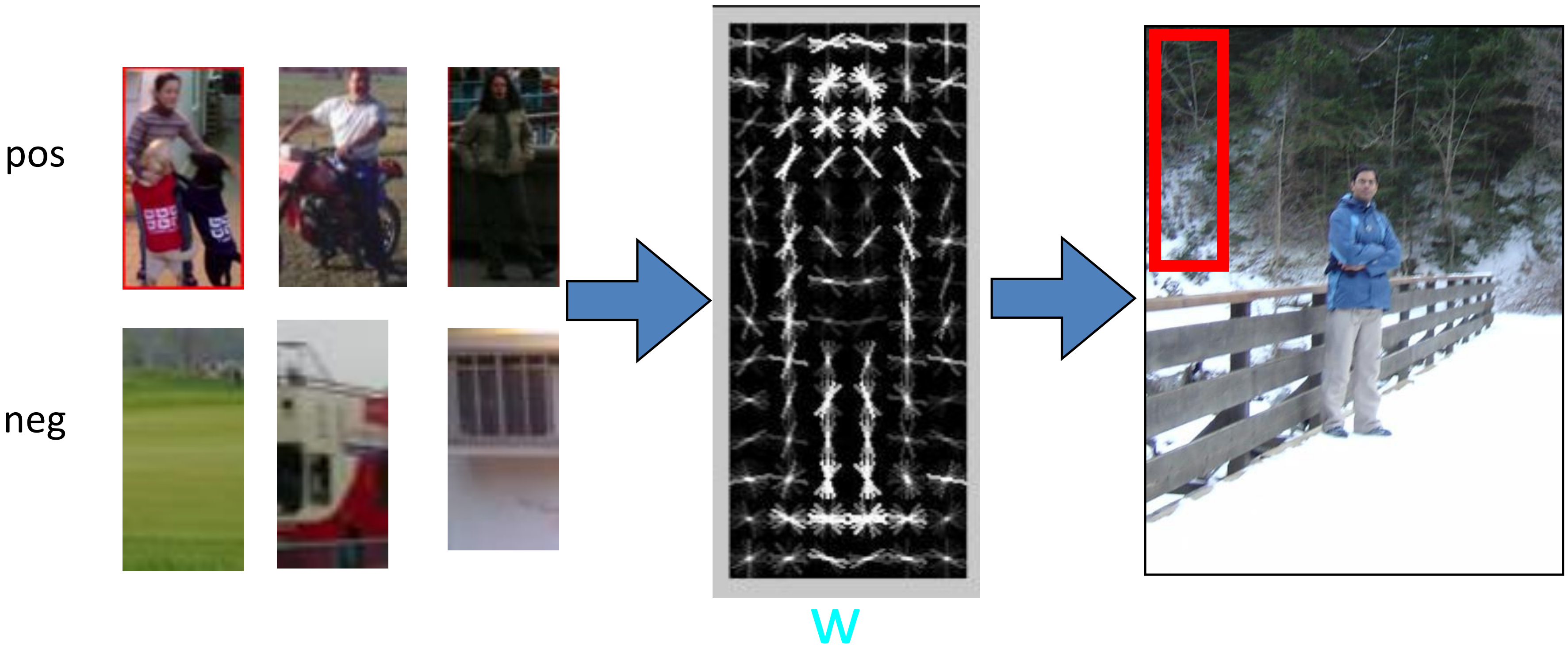
aeroplane bike bird boat bottle bus car cat chair cow table dog horse motorbike person plant sheep sofa train tv

Deva Ramanan

Detection: Scanning-window templates

Dalal and Triggs CVPR05 (HOG)

Papageorgiou and Poggio ICIP99 (wavelets)

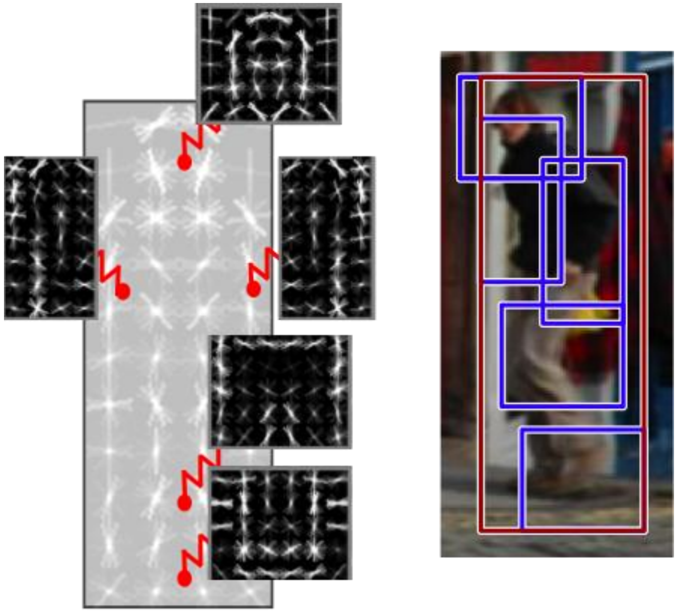


w = weights for orientation and spatial bins

$w \cdot x > 0$

Deva Ramanan

Deformable part models



Model encodes local appearance + spring deformation

Deva Ramanan

Homework for Next Time

- Paper review for Features due at 10pm on 1/14, send to **kovashka@cs.pitt.edu**