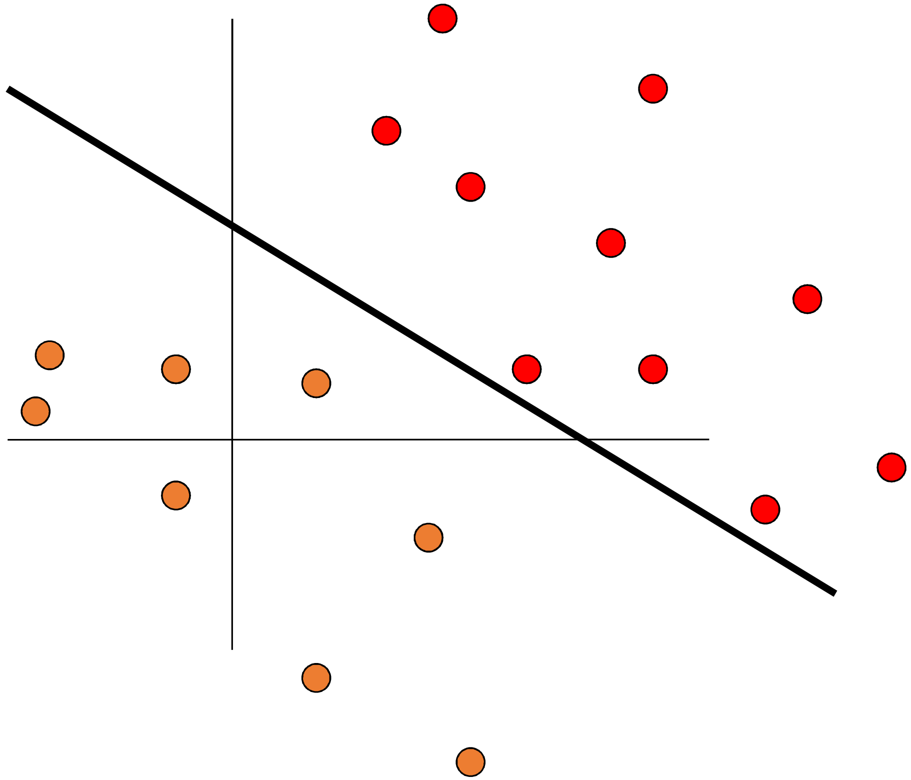
Introduction to Machine Learning and Data Mining Lecture-10: Support Vector Machine and Kernel Methods

Prof. Eugene Chang

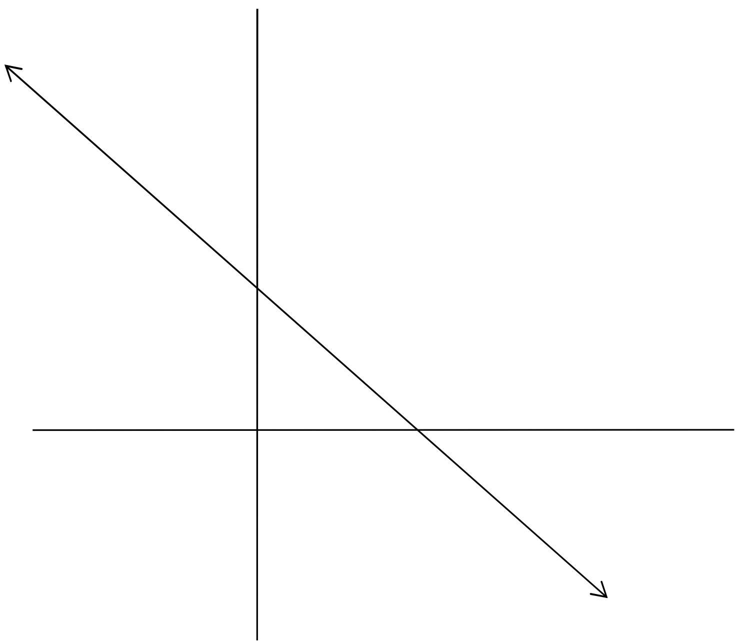
Today

- Support Vector Machines
  - Soft margin
  - Kernels method
- Slides based on materials from Prof. Kristen Grauman, University of Texas at Austin, and Prof Jiawei Han, University of Illinois at Urbana-Champaign

Linear Classifiers



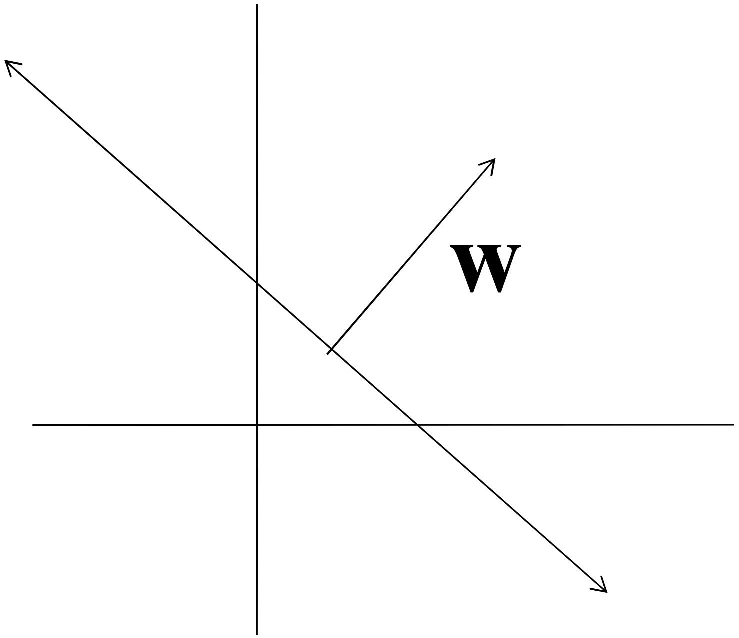
Lines in R^2



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

$$ax + cy + b = 0$$

Lines in R^2



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

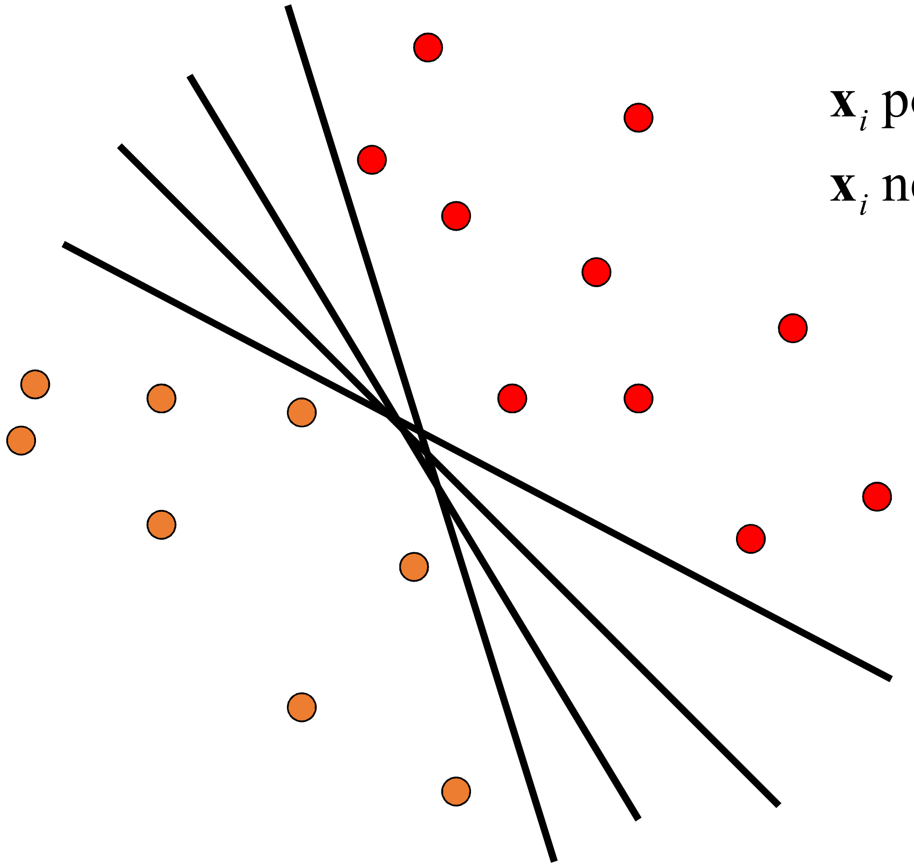
$$ax + cy + b = 0$$

$$\Updownarrow$$

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

Linear Classifiers

- Find linear function to separate positive and negative examples

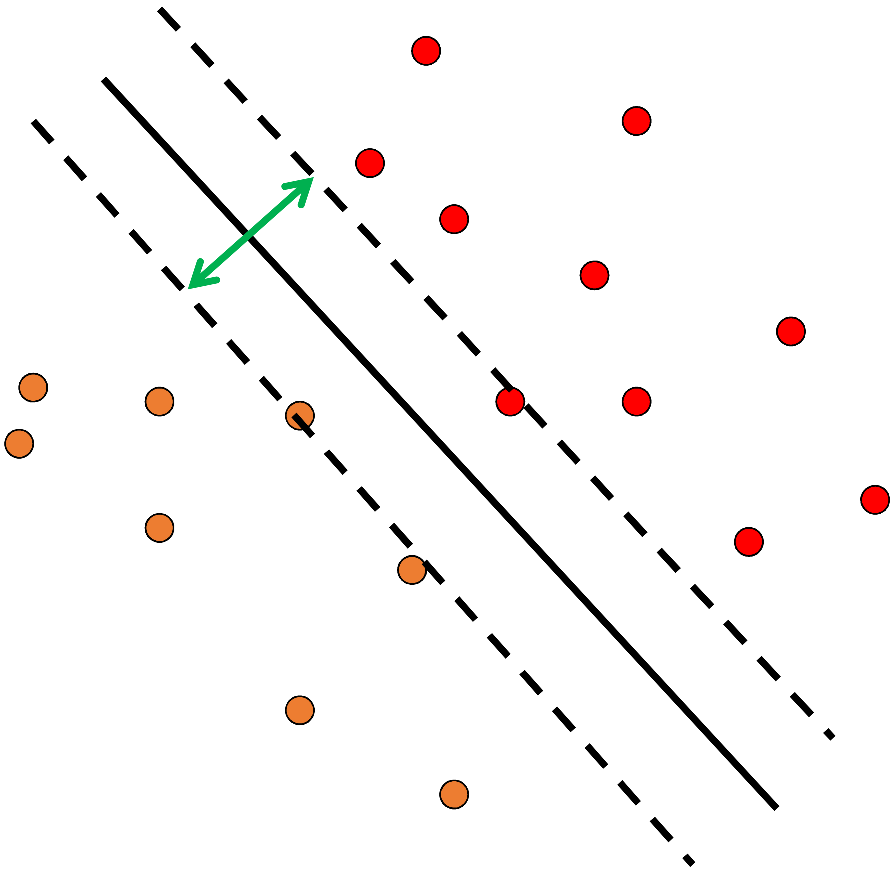


$\mathbf{x}_i$ positive: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

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

Which line is the best?

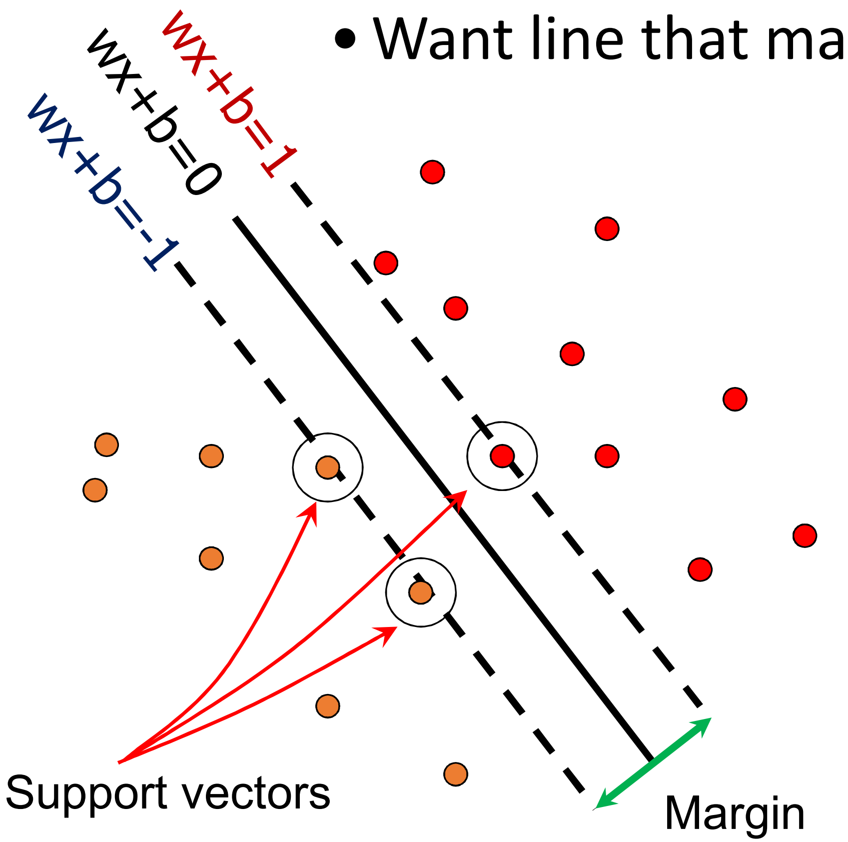
Support Vector Machines (SVMs)



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

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

Support Vector Machines



- Want line that maximizes the margin.

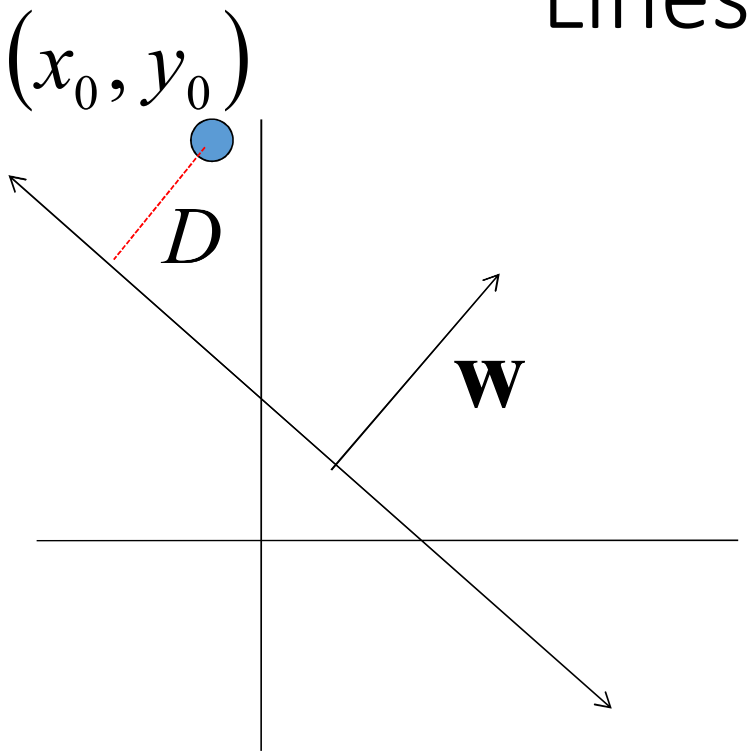
$$\mathbf{x}_i \text{ positive}(y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

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

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

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

Lines in R^2



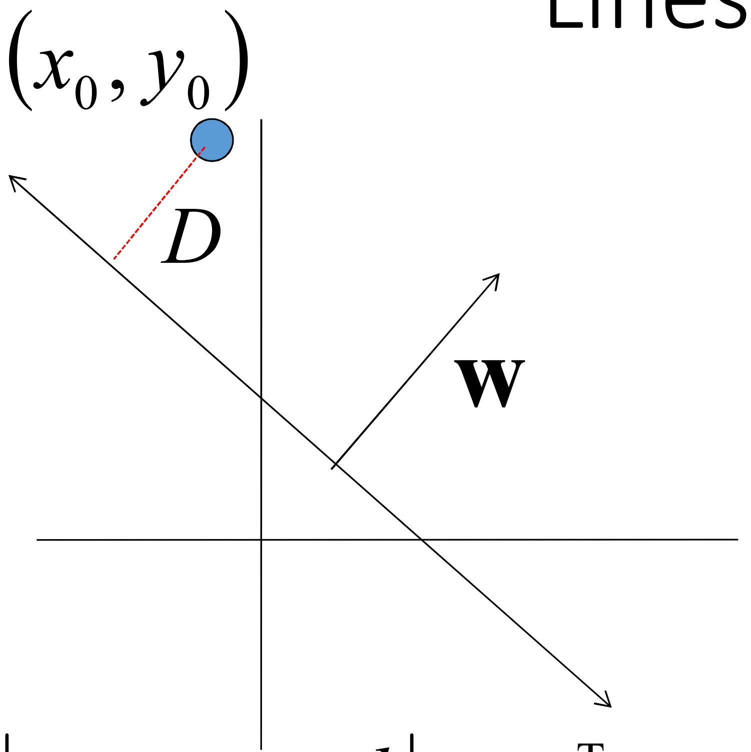
Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\Updownarrow$$

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

Lines in R^2



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

$$ax + cy + b = 0$$

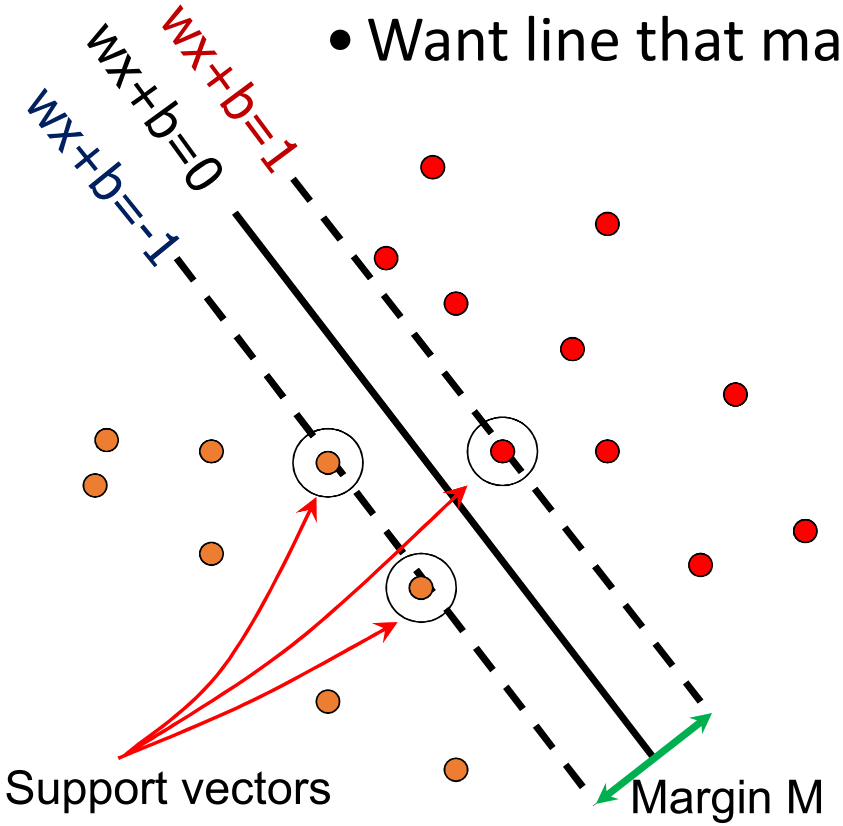
$$\Updownarrow$$

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

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

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

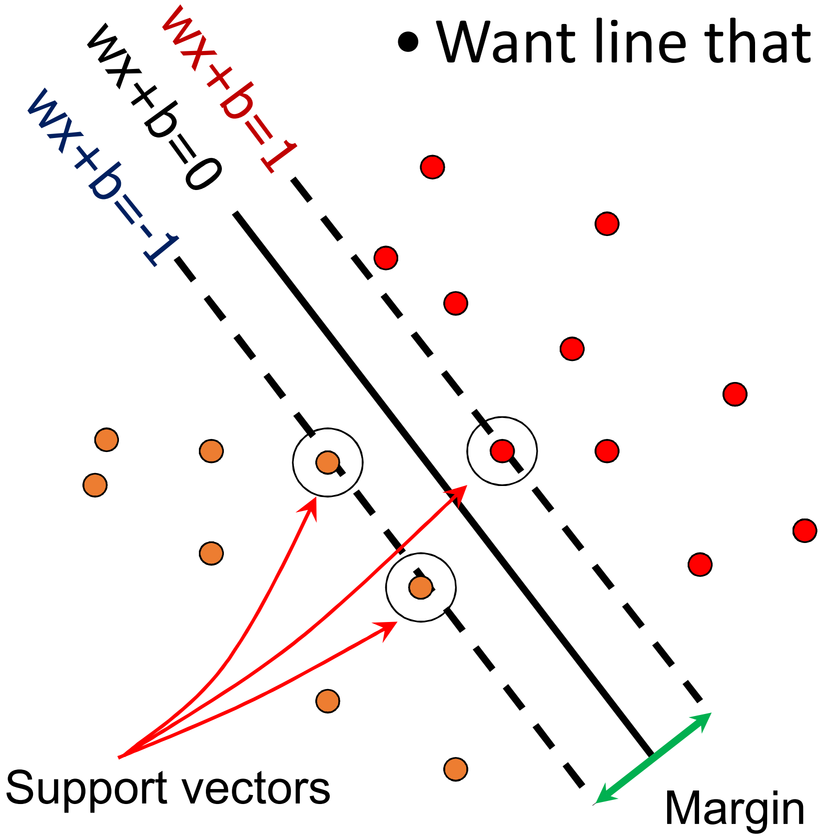
For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

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

For support vectors:

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

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 $2 / ||\mathbf{w}||$

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

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:

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

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

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} \; + b$$

- Classification function:

$$f(x) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + \text{b})$$

$$= \text{sign}\left(\sum_i \alpha_i \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

If f(x) < 0, classify as negative, if f(x) > 0, classify as positive

- Notice that it relies on an *inner product* between the test point $\boldsymbol{x}$ 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

Why is SVM Popular

- They work
  - Good generalization
- Easily interpreted.
  - Decision boundary is based on the data in the form of the **support vectors**.
- Principled bounds on testing error from Learning Theory (VC dimension)

Questions

- **What if the training data is noisy?**
- What if the features are not 2d?
- What if the data is not linearly separable?
- What if we have more than just two categories?

Relax the Constraints

- There can be outliers on the other side of the decision boundary, or leading to a small margin.
- To allow errors in data, we relax the margin constraints by introducing **slack** variables, ξ_i (≥ 0) as follows:

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1 - \xi_i \qquad \text{for } y_i = 1$$

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1 + \xi_i \qquad \text{for } y_i = \text{-}1$$

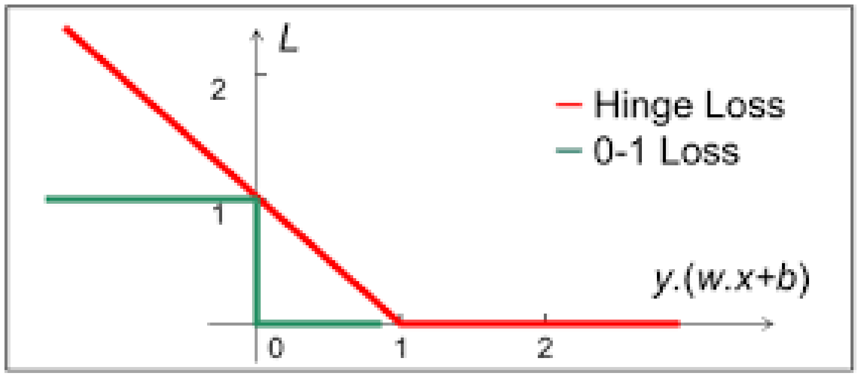
- The new constraints:

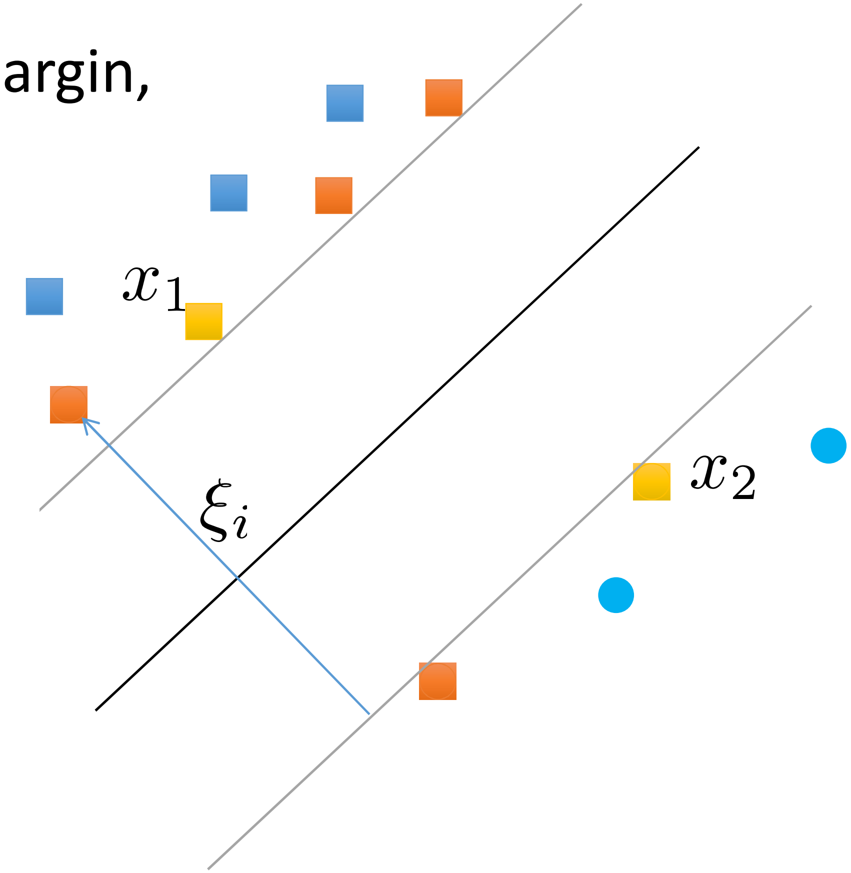
  Subject to: $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$, i =1, …, r, $\xi_i \geq 0$, i =1, 2, …, r.

Soft Margin Example

- Points are allowed within the margin, but cost is introduced.

Hinge Loss





Soft Margin Classification

- Solution: Introduce a penalty term to the constraint function

$$\min \|\vec{w}\| + C \sum_{i=0}^{N-1} \xi_i$$

$$\text{where } t_i(\vec{w}^T x_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0$$

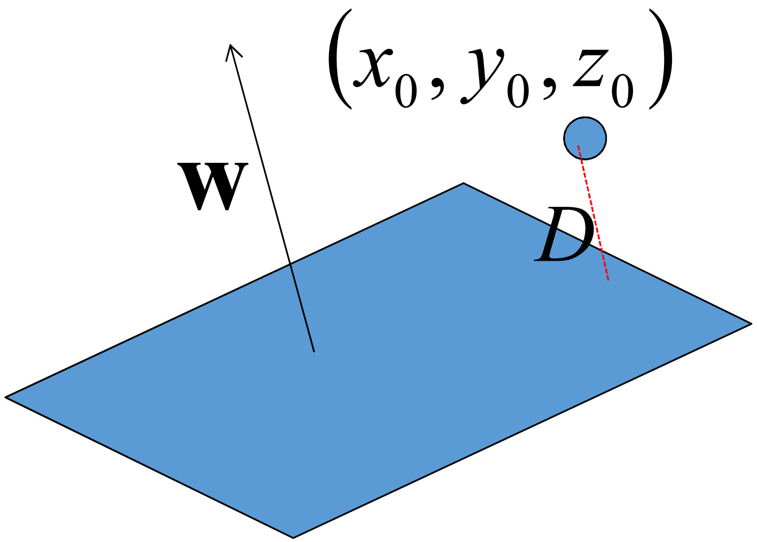
$$L(\vec{w}, b) = \frac{1}{2}\vec{w} \cdot \vec{w} + C \sum_{i=0}^{N-1} \xi_i - \sum_{i=0}^{N-1} \alpha_i [t_i((\vec{w} \cdot \vec{x_i}) + b) + \xi_i - 1]$$

- C is a regularization parameter:
  - small C allows constraints to be easily ignored → large margin
  - large C makes constraints hard to ignore → narrow margin
  - C = ∞ enforces all constraints: hard margin

Questions

- **What if the features are not 2d?**
  - **Generalizes to d-dimensions – replace line with “hyperplane”**
- What if the data is not linearly separable?
- What if we have more than just two categories?

Planes in R^3



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

$$ax + by + cz + d = 0$$

$$\Updownarrow$$

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

$$D = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} = \frac{\mathbf{w}^T \mathbf{x} + d}{\|\mathbf{w}\|}$$

distance from point to plane

Hyperplanes in R^n

Hyperplane *H* is set of all vectors $\mathbf{x} \in R^n$ which satisfy:

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b = 0$$

$$\Updownarrow$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$$

$$D(H, \mathbf{x}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x} + b}{\|\mathbf{w}\|}$$

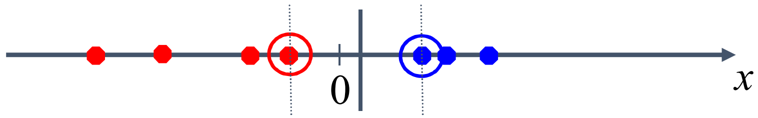
distance from point to hyperplane

Questions

- What if the training data is noisy?
- What if the features are not 2d?
- **What if the data is not linearly separable?**
- What if we have more than just two categories?

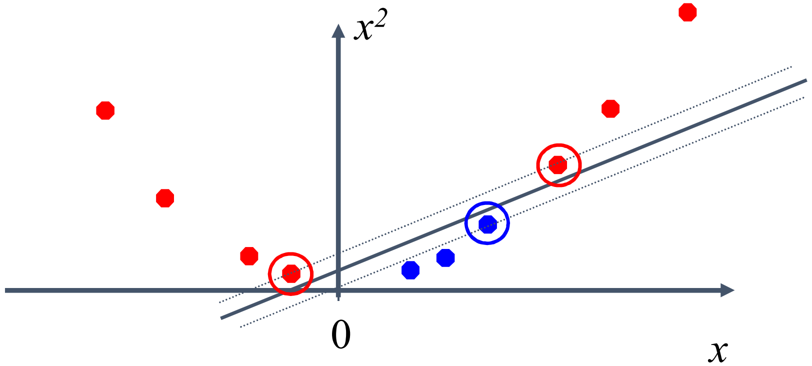
Non-linear SVMs

- Datasets that are linearly separable with some noise work out great:



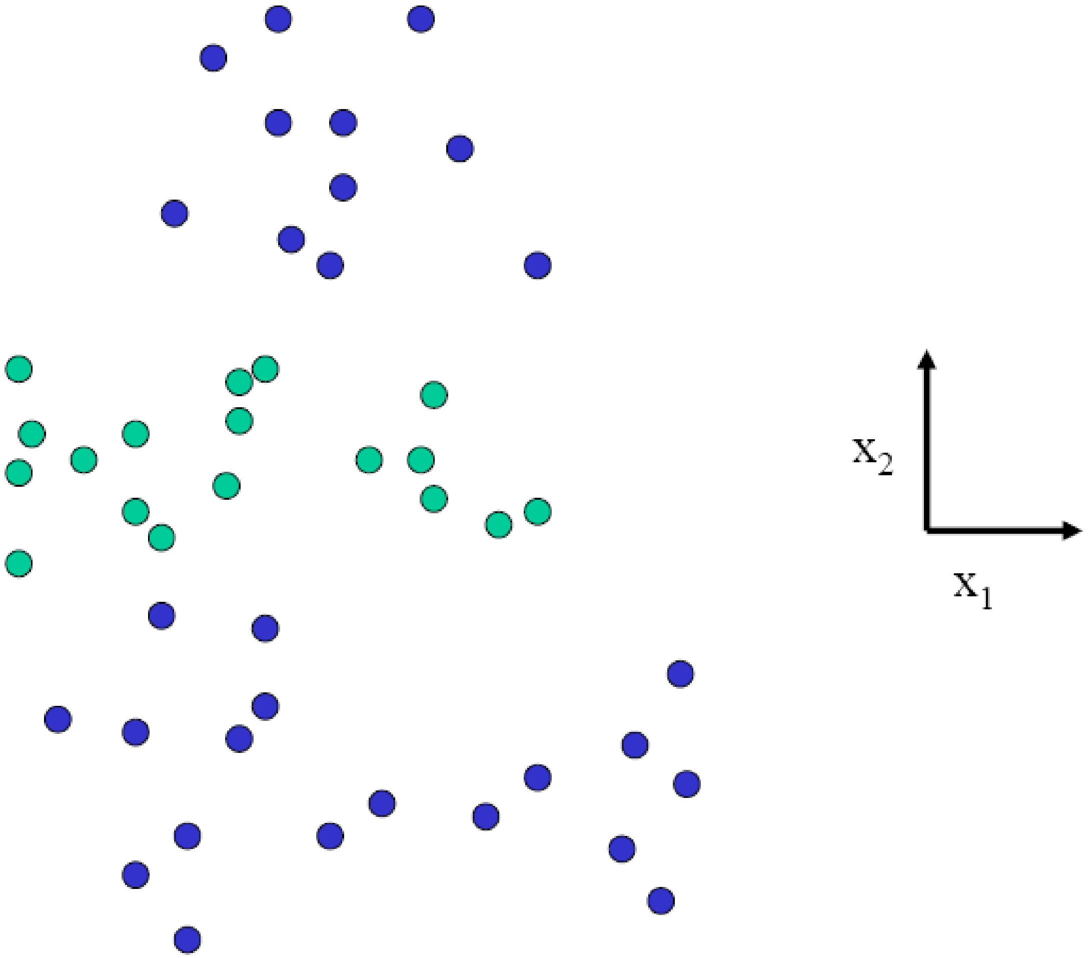
- But what are we going to do if the dataset is just too hard?

- How about… mapping data to a higher-dimensional space:

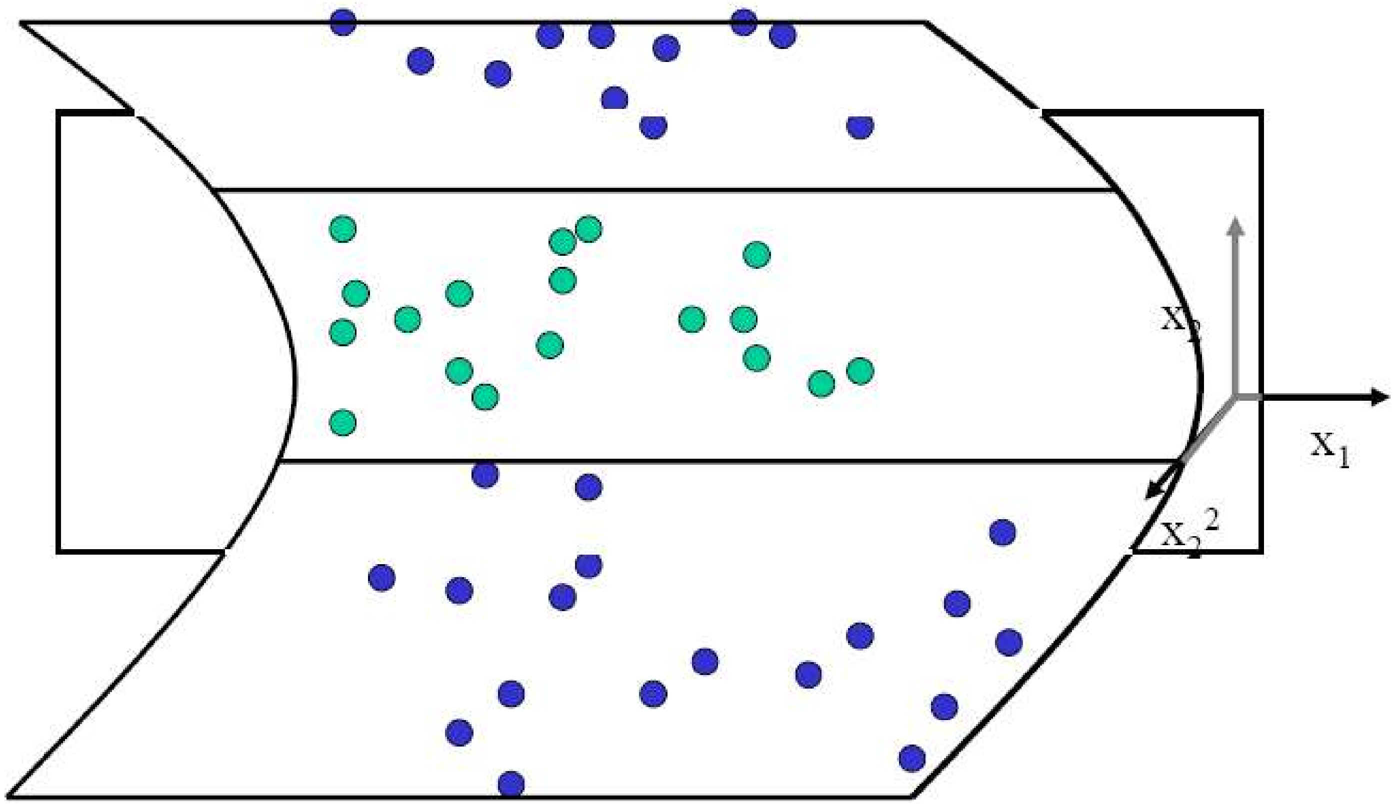


Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Non-separable by a hyperplane in 2-d



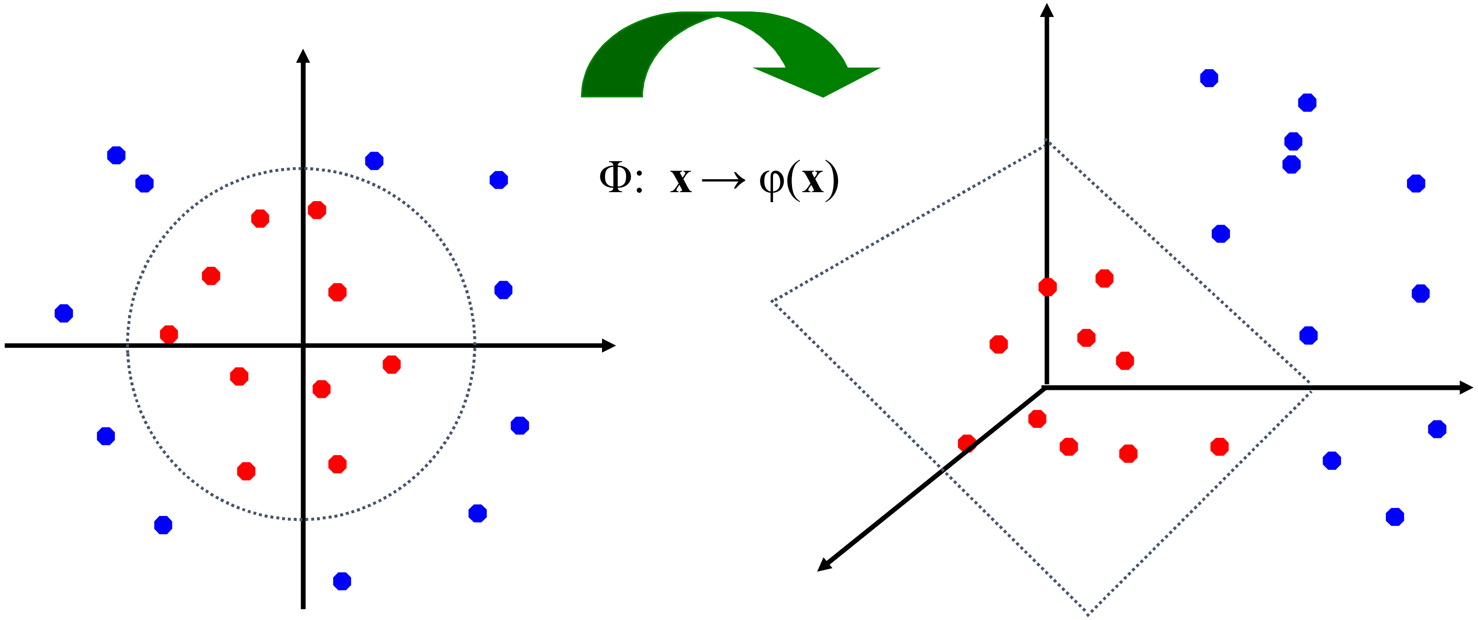
Separable by a hyperplane in 3-d



Source: Bill Freeman

Non-linear SVMs: Feature Spaces

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



Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Nonlinear SVMs

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

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

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

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

Examples of General Purpose Kernel Functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j})= \mathbf{x_i}^T\mathbf{x_j}$

- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j})= (1+ \mathbf{x_i}^T\mathbf{x_j})^p$

- Gaussian (radial-basis function network):

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

Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Questions

- What if the training data is noisy?
- What if the features are not 2d?
- What if the data is not linearly separable?
- **What if we have more than just two categories?**

Multi-class SVMs

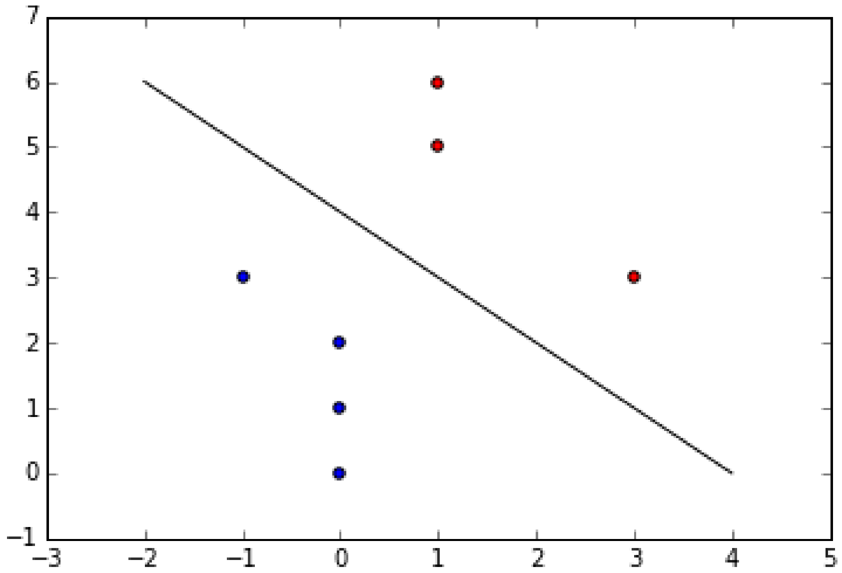
- Achieve multi-class classifier by combining a number of binary classifiers

longer time -> less classifier -

- **One vs. all**
  - Training: learn an SVM for each class vs. the rest
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- **One vs. one** ← shorter time → more classifier.
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

SVM Examples

- Consider building an SVM for the following two-class training data:
  - Positive class : [−1, 3], [0, 2], [0, 1], [0, 0]
  - Negative class : [1, 5], [1, 6], [3, 3]
- Plot the training points and, by inspection, draw a linear classifier that separates the data with maximum margin.
- SVM is parameterized by h(x) = $w^t x + b$. What are w and b?
- What are the support vectors?

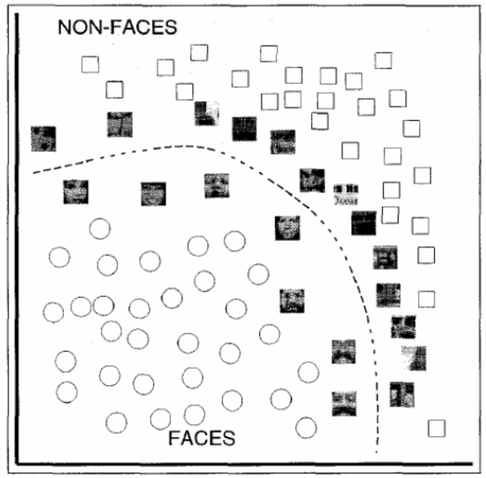


Sk-learn SVM Examples

- svm_demo.py
  - Demonstrate concepts
  - Create toy world svm classifier
- svm_demo2.py
  - Digit recognition
  - Iris data classification
- svm_gui.py
  - Interactive tool to create samples and adjust SVM parameters
  - Linear, RBF, and polynomial
- svm-kernels.pdf
  - Explains in details the effects of various sklearn svm parameters and kernels

SVMs for Recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Given this “kernel matrix” to SVM optimization software to identify support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.



SVM-HOG (git). and svmlight

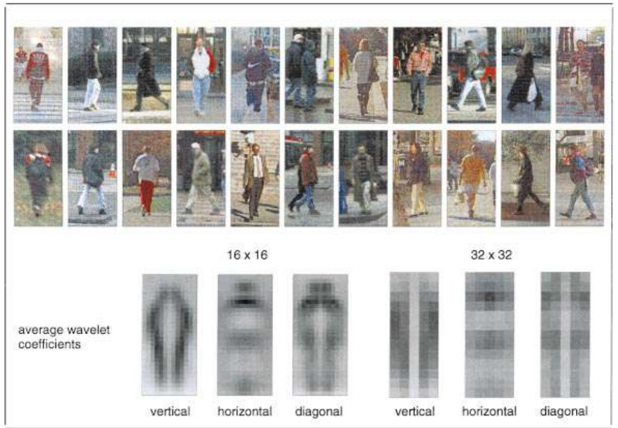
HOG + SVM for Zebra Detection in Photos

- Training sets
  - Positive samples
  - Negative samples

- HOG feature extraction
- SVM training
- Zebra detection using SVM classification

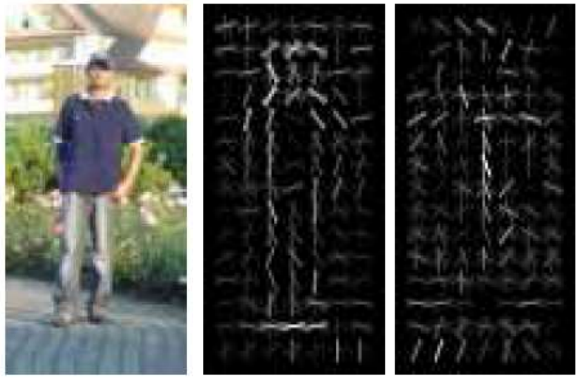
Pedestrian Detection

- Detecting upright, walking humans also possible using sliding window's appearance/texture; e.g.,



SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]

Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]

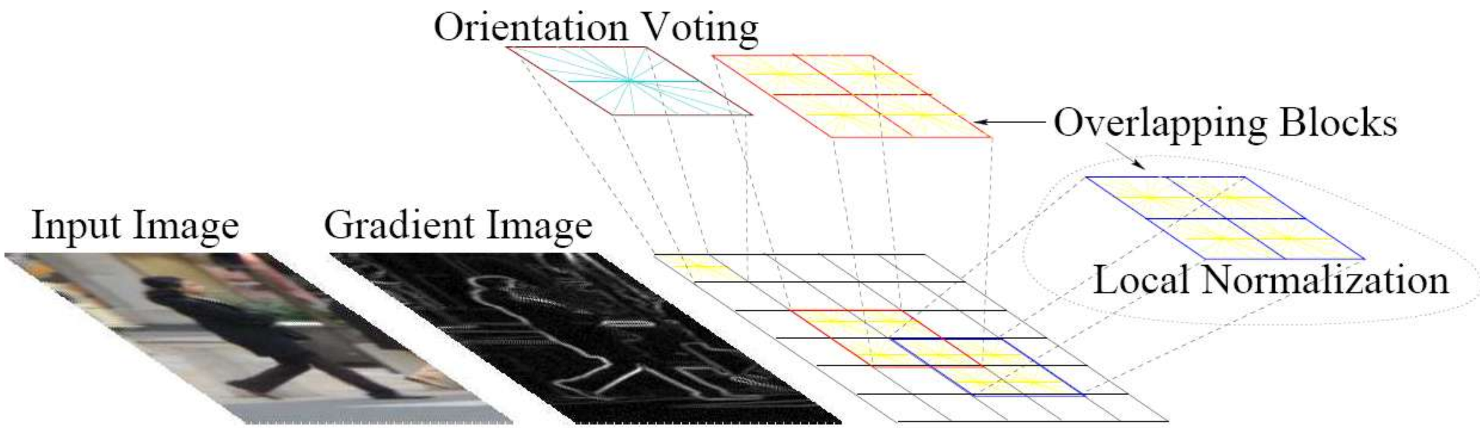


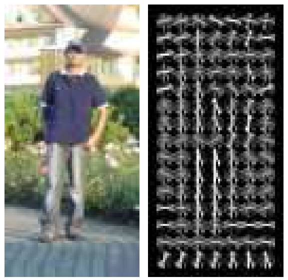
SVM with HoGs [Dalal & Triggs, CVPR 2005]

feature extracted

K. Grauman, B. Leibe

Example: pedestrian detection with HoG's and SVM's





- **Map each grid cell in the input window to a histogram counting the gradients per orientation.**
- **Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.**

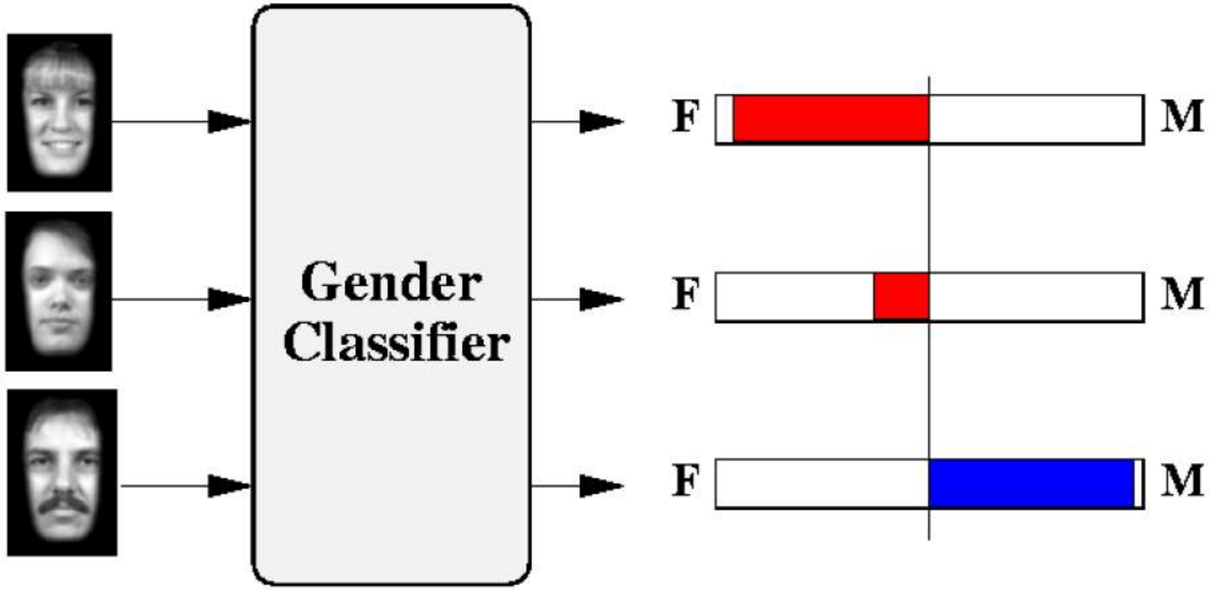
Dalal & Triggs, CVPR 2005

Code available: http://pascal.inrialpes.fr/soft/olt/

Pedestrian Detection with HoG's & SVM's

- Histograms of Oriented Gradients for Human Detection, [Navneet Dalal](), [Bill Triggs](), International Conference on Computer Vision & Pattern Recognition - June 2005
- http://lear.inrialpes.fr/pubs/2005/DT05/

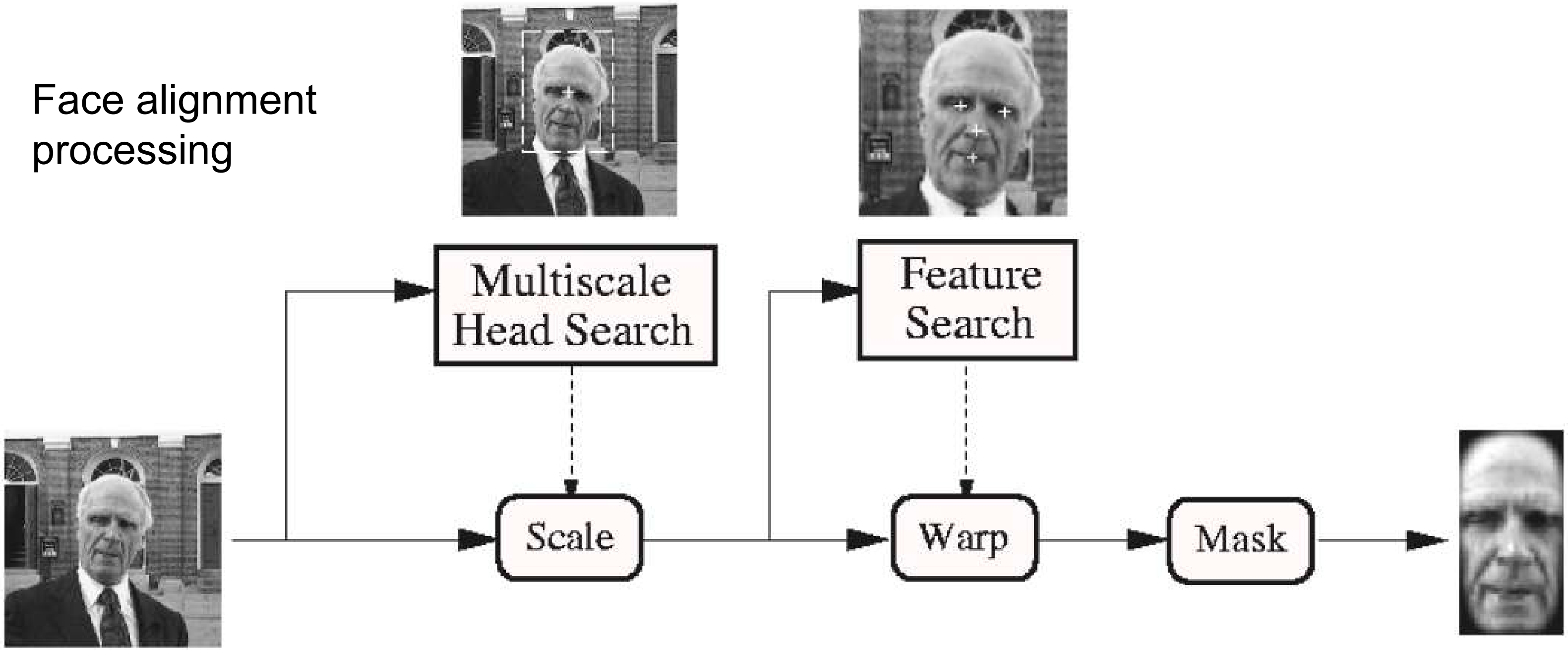
Example: learning gender with SVMs



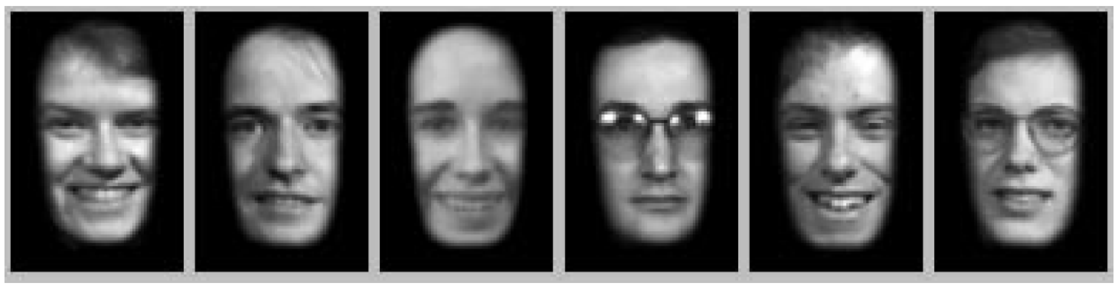
Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Moghaddam and Yang, Face & Gesture 2000.

Face alignment processing



Processed faces



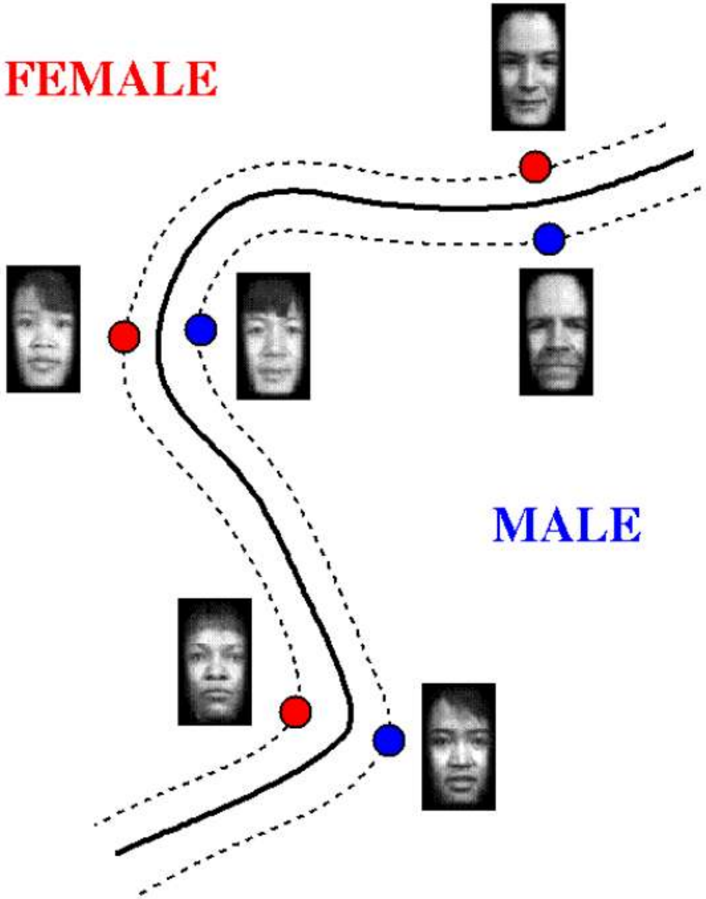
Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Learning gender with SVMs

- Training examples:
  - 1044 males
  - 713 females
- Experiment with various kernels, select Gaussian RBF

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

Support Faces



Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Classifier Performance

Classifier	Error Rate		
	Overall	Male	Female
SVM with RBF kernel	**3.38%**	**2.05%**	**4.79%**
SVM with cubic polynomial kernel	**4.88%**	**4.21%**	**5.59%**
Large Ensemble of RBF	5.54%	4.59%	6.55%
Classical RBF	7.79%	6.89%	8.75%
Quadratic classifier	10.63%	9.44%	11.88%
Fisher linear discriminant	13.03%	12.31%	13.78%
Nearest neighbor	27.16%	26.53%	28.04%
Linear classifier	58.95%	58.47%	59.45%

Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.