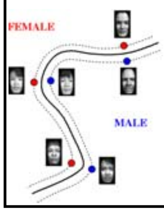
Appearance-based recognition & detection II

Kristen Grauman
UT-Austin

Tuesday, Nov 11

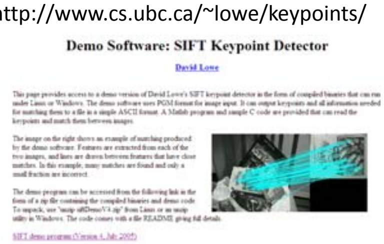


Last time

- **Appearance-based recognition**: using global appearance descriptions within a window to characterize a class.
  - Classification: basic idea of supervised learning
    - Skin color detection example
  - Sliding windows: detection via classification
    - Make a yes/no decision at every window
    - Face detection example using boosting and rectangular features [Viola-Jones 2001]

Misc notes

- Extra disk space
- SIFT extraction
  - http://www.cs.ubc.ca/~lowe/keypoints/

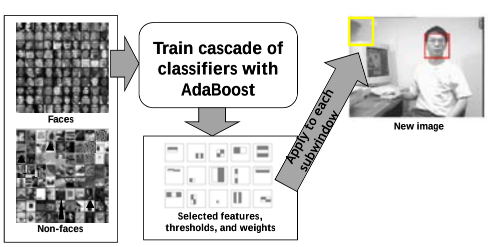


Today

- Additional classes well-suited by global appearance representations
- Discriminative classifiers
  - Boosting (last time)
  - Nearest neighbors
  - Support vector machines
    - Application to pedestrian detection
    - Application to gender classification

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Summary



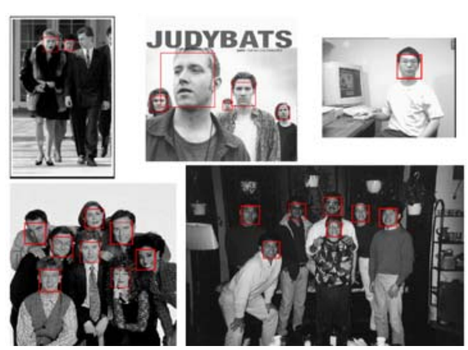
- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in final layer
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

K. Grauman, B. Leibe

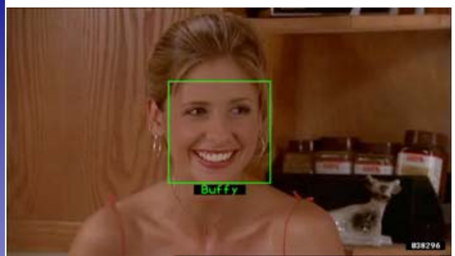
5

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results



Example application



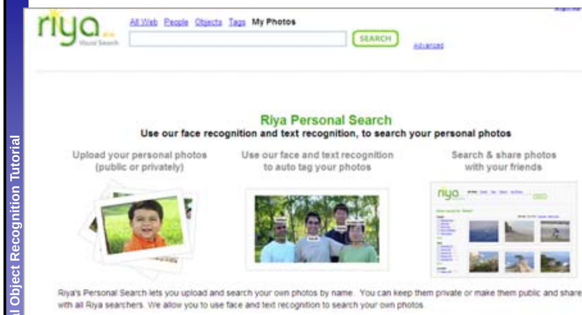
Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006.
http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

K. Grauman, B. Leibe 7

Visual Object Recognition Tutorial

Example application: faces in photos



Visual Object Recognition Tutorial

- Other classes that might work with global appearance in a window?

Fifth International Penguin Conference, Ushuaia, Tierra del Fuego, Argentina, September 2004

Fifth International Penguin Conference
Ushuaia, Tierra del Fuego, Argentina

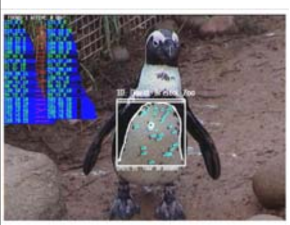
Automated Visual Recognition of Individual African Penguins

Tilo Burghardt,
Barry Thomas, Peter J Barham, Janko Čalić

University of Bristol, Department of Computer Science,
MVB Woodland Road, Bristol BS8 1UB, United Kingdom,
September 2004

burghard@cs.bris.ac.uk

Penguin detection & identification



African penguins *(Spheniscus demersus)* carry a pattern of black spots on their chests that does not change from season to season during their adult life. Further, as far as we can tell, no two penguins have exactly the same pattern. We have developed a real-time system that can confidently locate African penguins whose chests are visible within video sequences or still images. An extraction of the chest spot pattern allows the generation of a unique biometrical identifier for each penguin. Using these identifiers an authentication of filmed or photographed African penguins against a population database can be performed. This paper provides a detailed technical description of the developed system and outlines the scope and the conditions of application ■

This project uses the Viola-Jones Adaboost face detection algorithm to detect penguin chests, and then matches the pattern of spots to identify a particular penguin.

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

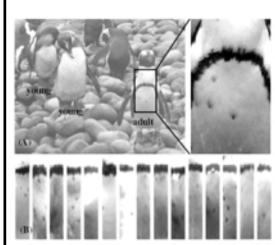
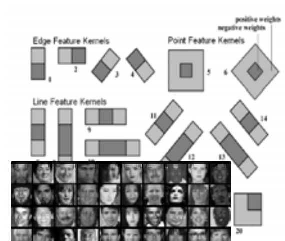


Figure 6. *Distinctive Chest Stripe of Adult African Penguins:* (A) Adult African penguins carry a distinctive and stable black stripe on their chest whilst young members of the species still change the colour of their chest feathers. (B) Various chests of adult African penguins under different lighting conditions. (figure source [19])



Use rectangular features, select good features to distinguish the chest from non-chests with Adaboost

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

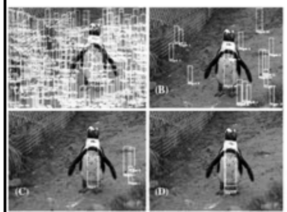


Figure 12. *Application of Attentional Cascades on Chests:* (A) Image areas that are accepted as likely to represent a chest after one stage are marked as white rectangles. (B) After three stages… (C) After five stages… (D) …and after seven stages with final result. (figure source [18], [19])

Attentional cascade

Figure 10. *AoI Detector Spotting Frontal Penguin Chests:* The detector was tested on a series of black and white still images and footage. Some result images are shown above. The detector might fire several times on one and the same chest instance. (figure

Penguin chest detections

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

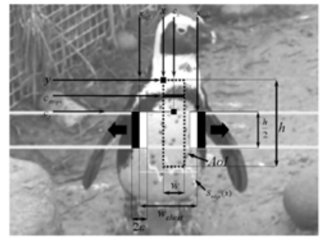


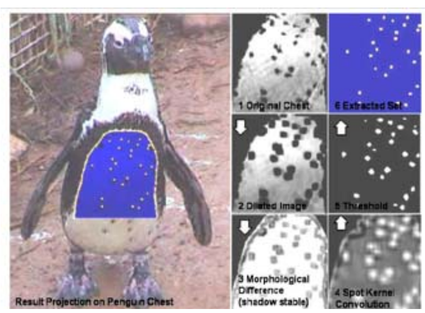
Figure 14. *Visual Description of the Chest Width Measurement:* Starting from an upper central point of the chest AoI two locally operating edge detectors moving apart search for the left and right boundary of the assumed chest. (figure source [17])

Given a detected chest, try to extract the whole chest for this particular penguin.

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

Example detections

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.



Perform **identification** by matching the pattern of spots to a database of known penguins.

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

Penguin detection & identification

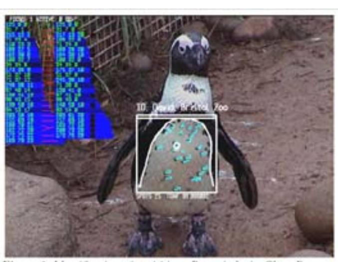


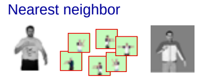
Figure 1. *Identification of an African Penguin by its Chest Pattern:* Screenshot of Software Prototype: African penguins carry a unique pattern of black spots on their chest. The detection of the chest location and the decomposition of the spot pattern allow checking a photographed individual (here penguin 'David' from Bristol Zoo) against a population database. (figure source [18], [19])

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

Visual Object Recognition Tutorial

Discriminative classifiers

Nearest neighbor



Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

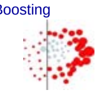
Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines

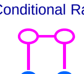
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



Viola, Jones 2001, Torralba et al. 2004, Opelt et al. 2006,…

Conditional Random Fields



McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba

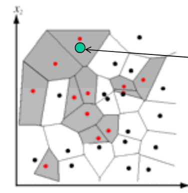
Today

- Additional classes well-suited by global appearance representations
- Discriminative classifiers
  - Boosting (last time)
  - Nearest neighbors
  - Support vector machines
    - Application to pedestrian detection
    - Application to gender classification

Nearest Neighbor classification

- Assign label of nearest training data point to each test data point

Black = negative
Red = positive



from Duda *et al.*

Novel test example

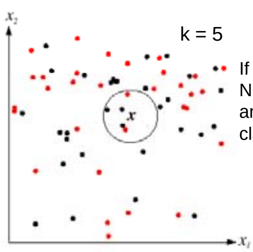
Closest to a positive example from the training set, so classify it as positive.

Voronoi partitioning of feature space for 2-category 2D data

K-Nearest Neighbors classification

- For a new point, find the k closest points from training data
- Labels of the k points “vote” to classify

Black = negative
Red = positive

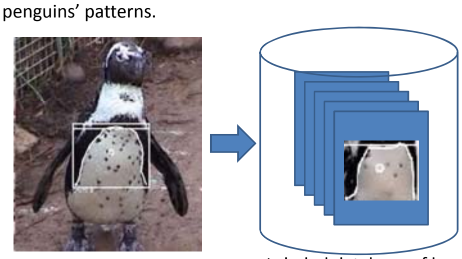


If query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

Source: D. Lowe

Example: nearest neighbor classification

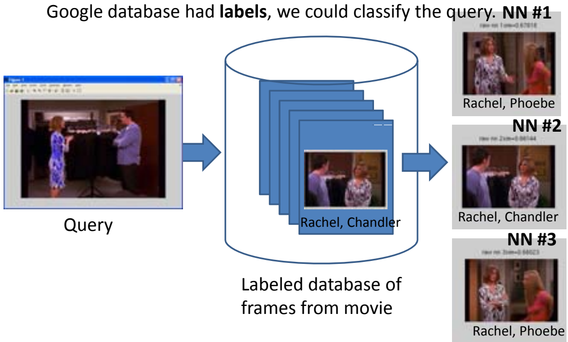
- We could identify the penguin in the new view based on the distance between its chest spot pattern and all the stored penguins’ patterns.



Labeled database of known penguin examples

Example: nearest neighbor classification

- Similarly, if the video frames we were indexing in the Video Google database had **labels**, we could classify the query.



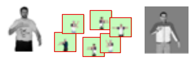
Nearest neighbors: pros and cons

- Pros:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- Cons:
  - Large search problem to find nearest neighbors
  - Storage of data
  - Must know we have a meaningful distance function

Discriminative classifiers

Visual Object Recognition Tutorial

Nearest neighbor

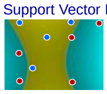


Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines

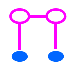


Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting

Viola, Jones 2001, Torralba et al. 2004, Opelt et al. 2006,…

Conditional Random Fields



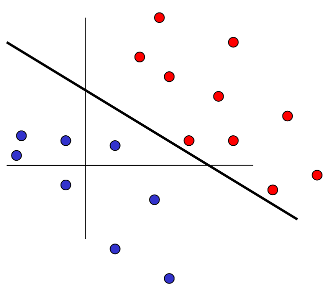
McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

Slide adapted from Antonio Torralba

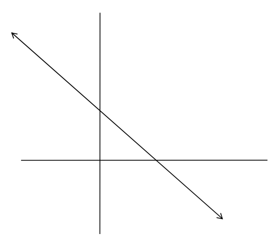
Today

- Additional classes well-suited by global appearance representations
- Discriminative classifiers
  - Boosting (last time)
  - Nearest neighbors
  - Support vector machines
    - Application to pedestrian detection
    - Application to gender classification

Linear classifiers



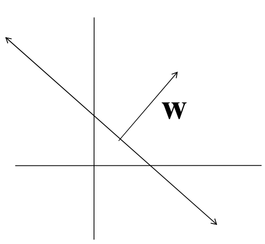
Lines in R^2



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

$$ax + cy + b = 0$$

Lines in R^2

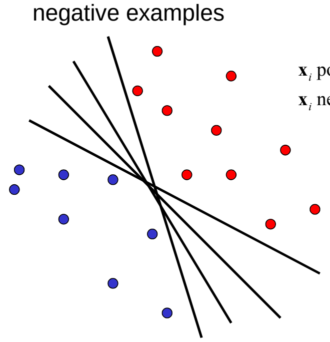


Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

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

Linear classifiers

- Find linear function to separate positive and negative examples

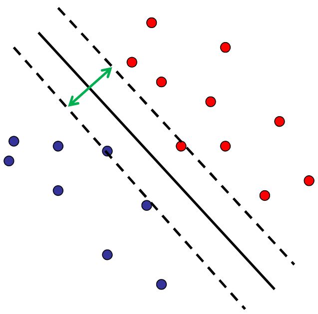


$\mathbf{x}_i$ positive: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

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

Which line is best?

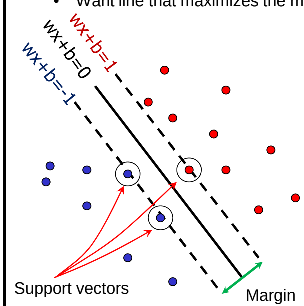
Support Vector Machines (SVMs)



- 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.



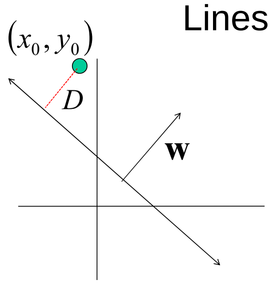
$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ 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

Lines in R^2



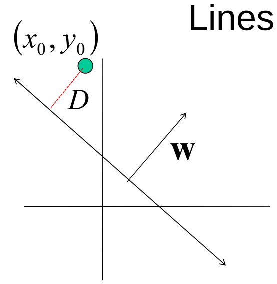
Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\updownarrow$$

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

Lines in R^2



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

$$ax + cy + b = 0$$

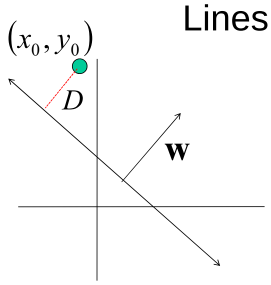
$$\updownarrow$$

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

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

distance from point to line

Lines in R^2



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

$$ax + cy + b = 0$$

$$\updownarrow$$

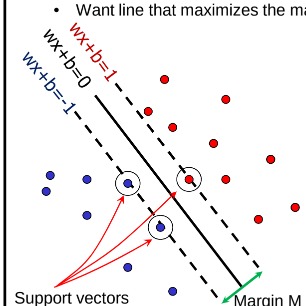
$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

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

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

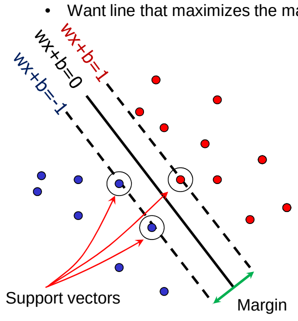
Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

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: $\quad \frac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

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:

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

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

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,

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,

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)

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

- Classification function:

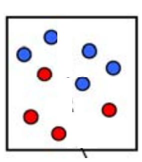
$$f(x) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + \text{b})$$
$$= \text{sign}\left(\sum_i \alpha_i \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

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 ***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,

How is the SVM objective different from the boosting objective?



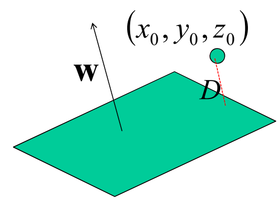
Questions

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

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

$$ax + by + cz + d = 0$$
$$\Updownarrow$$
$$\mathbf{w} \cdot \mathbf{x} + d = 0$$

$$D = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x} + d}{\|\mathbf{w}\|}$$ distance from point to plane

Hyperplanes in R^n

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

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b = 0$$
$$\Updownarrow$$
$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$$

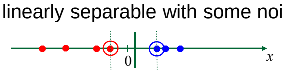
$$D(H, \mathbf{x}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x} + b}{\|\mathbf{w}\|}$$ distance from point to hyperplane

Questions

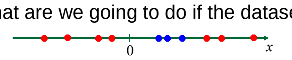
- What if the features are not 2d?
- **What if the data is not linearly separable?**
- What if we have more than just two categories?

Non-linear SVMs

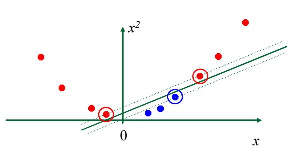
- Datasets that are linearly separable with some noise work out great:



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



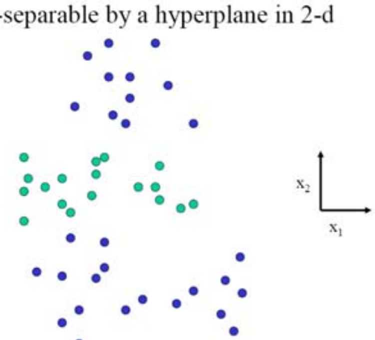
- How about… mapping data to a higher-dimensional space:



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

Another example:

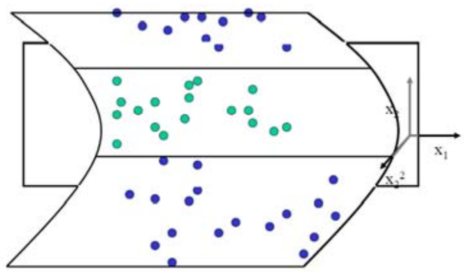
Non-separable by a hyperplane in 2-d



Source: Bill Freeman

Another example:

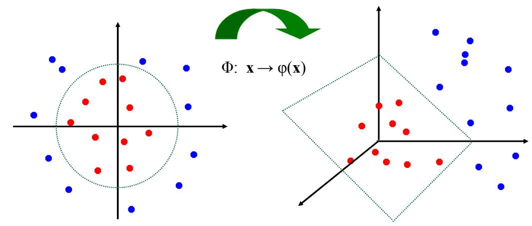
Separable by a hyperplane in 3-d



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 $\varphi(\mathbf{x})$, define a kernel function K such that

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

- 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):

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

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

Questions

- 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
- **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**
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

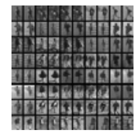
SVMs for recognition

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.

Pedestrian detection

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

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



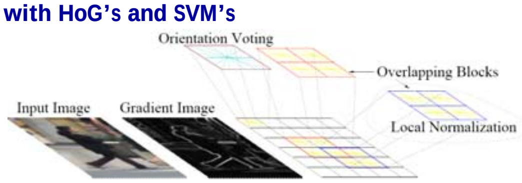
Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]

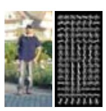
SVM with HoGs [Dalal & Triggs, CVPR 2005]

Visual Object Recognition Tutorial

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/

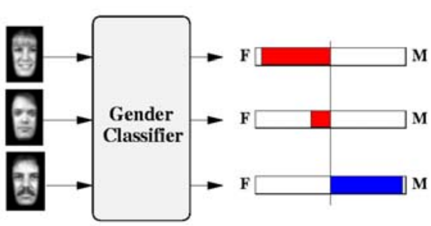
Visual Object Recognition Tutorial

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/

Visual Object Recognition Tutorial

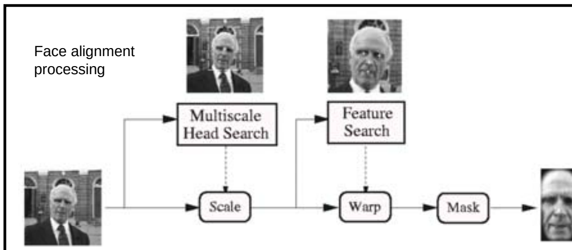
Example: learning gender with SVMs



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

Moghaddam and Yang, Face & Gesture 2000.

Face alignment processing



Processed faces

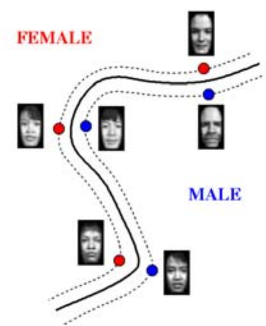
Moghaddam and Yang, Learning Gender with Support Faces, TPAMI 2002.

Learning gender with SVMs

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

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

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.

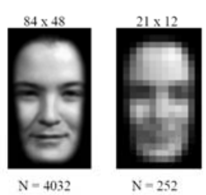
Gender perception experiment: How well can humans do?

- Subjects:
  - 30 people (22 male, 8 female)
  - Ages mid-20's to mid-40's
- Test data:
  - 254 face images (6 males, 4 females)
  - Low res and high res versions
- Task:
  - Classify as male or female, forced choice
  - No time limit

Moghaddam and Yang, Face & Gesture 2000.

Gender perception experiment: How well can humans do?

Stimuli →



Results →

High-Res	Low-Res
6.54%	30.7%
Error	Error

σ = 3.7%

Moghaddam and Yang, Face & Gesture 2000.

Human vs. Machine

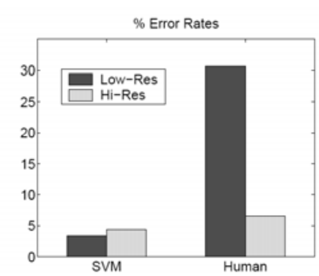


Figure 6. SVM vs. Human performance

- SVMs performed better than any single human test subject, at either resolution

Hardest examples for humans

Top five human misclassifications

Moghaddam and Yang, Face & Gesture 2000.

SVMs: Pros and cons

- Pros
  - Many publicly available SVM packages: http://www.kernel-machines.org/software
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - Kernel-based framework is very powerful, flexible
  - Often a sparse set of support vectors – compact at test time
  - Work very well in practice, even with very small training sample sizes
- Cons
  - No "direct" multi-class SVM, must combine two-class SVMs
  - Can be tricky to select best kernel function for a problem
  - Computation, memory
    - During training time, must compute matrix of kernel values for every pair of examples
    - Learning can take a very long time for large-scale problems

Adapted from Lana Lazebnik

Summary: today

- Additional classes well-suited by global appearance representations
- Discriminative classifiers
  - Boosting (last time)
  - Nearest neighbors
  - Support vector machines
    - Application to pedestrian detection
    - Application to gender classification