

CS294-43: Multiple Kernel and Segmentation methods

Prof. Trevor Darrell
Spring 2009

April 7th, 2009

Last Lecture – Category Discovery

- R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," ICCV vol. 2, 2005
- L.-J. Li, G. Wang, and L. Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," in Computer Vision and Pattern Recognition, 2007. CVPR '07
- F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," in Computer Vision, 2007. ICCV 2007
- T. Berg and D. Forsyth, "Animals on the Web". In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
- K. Saenko and T. Darrell, "Unsupervised Learning of Visual Sense Models for Polysemous Words". Proc. NIPS, December 2008

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008,
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Multiple Kernel Learning

Manik Varma

Microsoft Research India

Alex Berg, Anna Bosch, Varun Gulshan, Jitendra Malik & Andrew Zisserman

Shanmuganathan Raman & Lihi Zelnik-Manor

Rakesh Babu & C. V. Jawahar

Debajyoti Ray

SVMs and MKL

- SVMs are basic tools in machine learning that can be used for classification, regression, *etc.*
- MKL can be utilized whenever a single kernel SVM is applicable.
- SVMs/MKL find applications in
 - Video, audio and speech processing.
 - NLP, information retrieval and search.
 - Software engineering.
- We focus on examples from vision in this talk.

Object Categorization



Chair



Schooner



?
=

Ketch



Taj



Panda



Novel image to be classified

Labelled images comprise training data

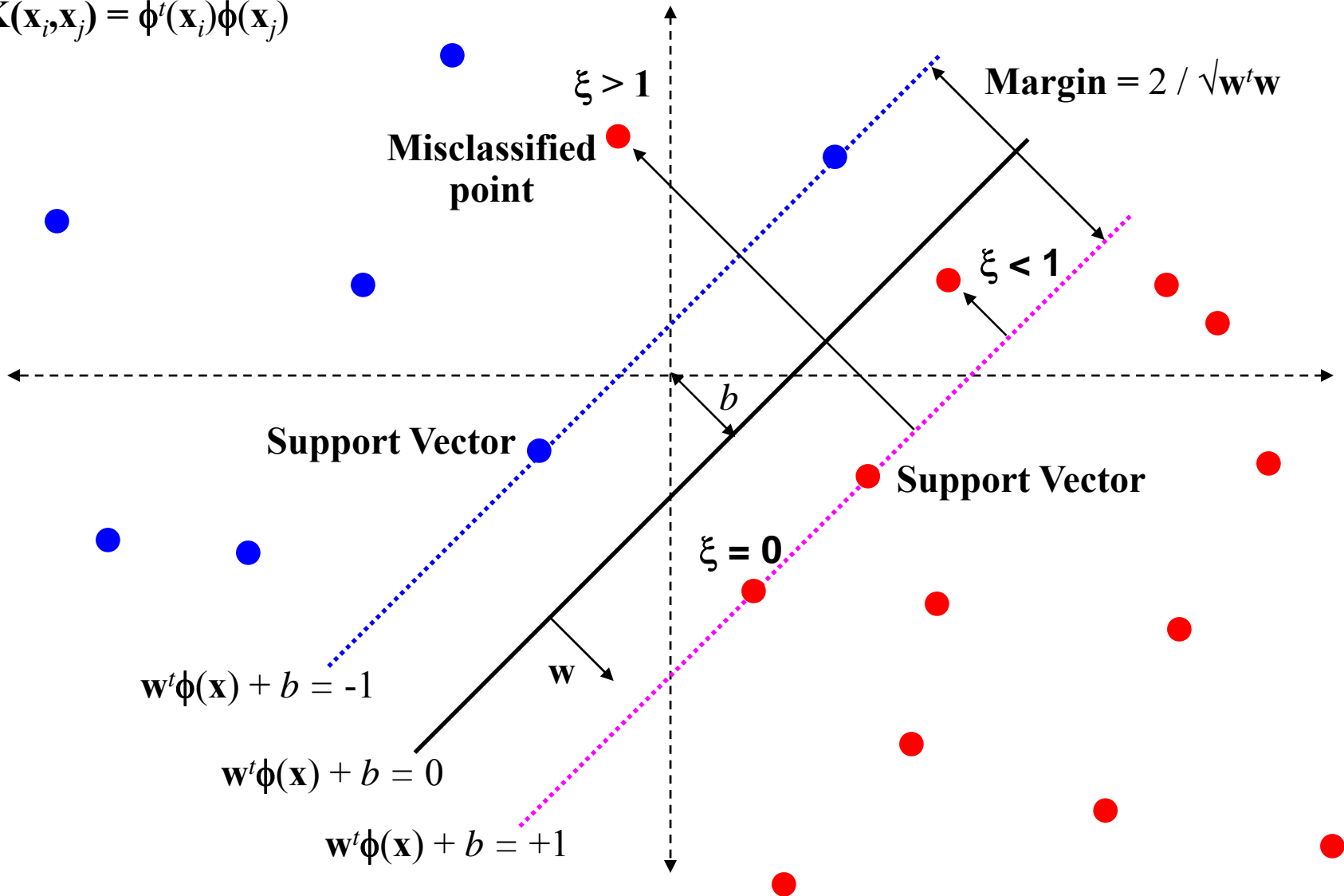
Outline of the Talk

- Introduction to SVMs and kernel learning.
- Our Multiple Kernel Learning (MKL) formulation.
- Application to object recognition.
- Extending our MKL formulation.
- Applications to feature selection and predicting facial attractiveness.

Introduction to SVMs and Kernel Learning

Binary Classification With SVMs

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi^t(\mathbf{x}_i)\phi(\mathbf{x}_j)$$



The C-SVM Primal Formulation

- Minimise \mathbf{w}, b, ξ $\frac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_i \xi_i$
 - Subject to
 - $y_i [\mathbf{w}^t \phi(\mathbf{x}_i) + b] \geq 1 - \xi_i$
 - $\xi_i \geq 0$
 - where
 - (\mathbf{x}_i, y_i) is the i^{th} training point.
 - C is the misclassification penalty.

The C-SVM Dual Formulation

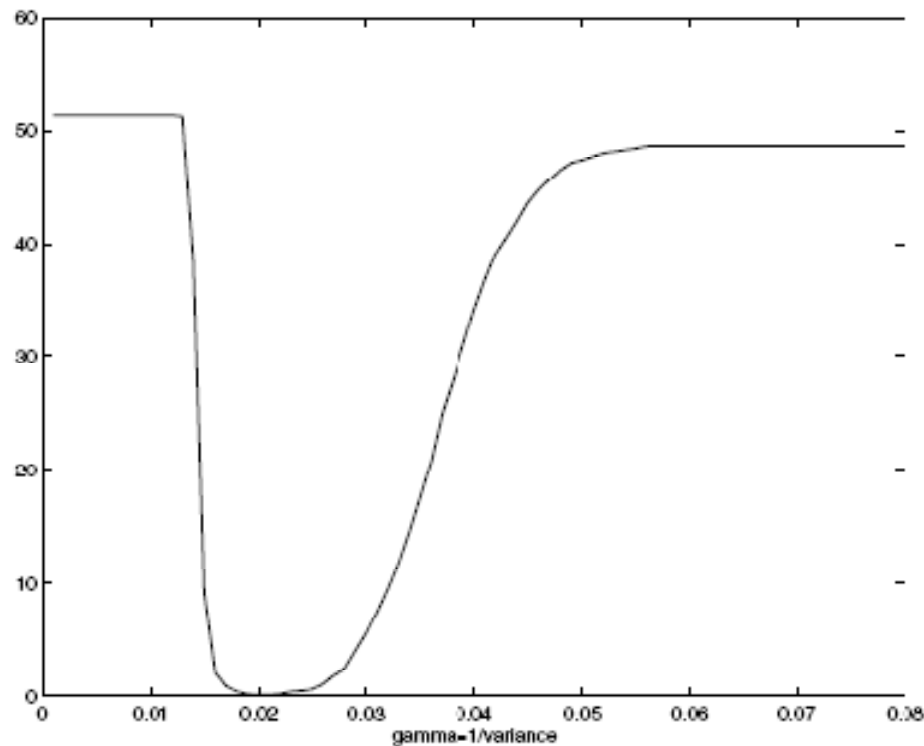
- Maximise $\alpha \mathbf{1}^t \alpha - \frac{1}{2} \alpha^t \mathbf{Y} \mathbf{K} \mathbf{Y} \alpha$
 - Subject to
 - $\mathbf{1}^t \mathbf{Y} \alpha = 0$
 - $\mathbf{0} \leq \alpha \leq \mathbf{C}$
 - where
 - α are the Lagrange multipliers corresponding to the support vector coeffs
 - \mathbf{Y} is a diagonal matrix such that $Y_{ii} = y_i$
 - \mathbf{K} is the kernel matrix with $\mathbf{K}_{ij} = \phi^t(\mathbf{x}_i) \phi(\mathbf{x}_j)$

Some Popular Kernels

- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^t \Sigma^{-1} \mathbf{x}_j$
- Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^t \Sigma^{-1} \mathbf{x}_j + c)^d$
- RBF: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sum_k \gamma_k (\mathbf{x}_{ik} - \mathbf{x}_{jk})^2)$
- Chi-Square: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \chi^2(\mathbf{x}_i, \mathbf{x}_j))$

Advantages of Learning the Kernel

- Learn the kernel parameters
 - Improve accuracy and generalisation



Advantages of Learning the Kernel

- Learn the kernel parameters
 - Improve accuracy and generalisation
 - Perform feature component selection

$$\text{Learn } K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sum_k \gamma_k (\mathbf{x}_{ik} - \mathbf{x}_{jk})^2)$$

Advantages of Learning the Kernel

- Learn the kernel parameters
 - Improve accuracy and generalisation
 - Perform feature component selection
 - Perform dimensionality reduction

Learn $K(\mathbf{P}_\theta \mathbf{x}_i, \mathbf{P}_\theta \mathbf{x}_j)$ where \mathbf{P} is a low dimensional projection matrix parameterised by θ .

Advantages of Learning the Kernel

- Learn the kernel parameters
 - Improve accuracy and generalisation
 - Perform feature component selection
 - Perform dimensionality reduction
- Learn a linear combination of base kernels
 - $K(\mathbf{x}_i, \mathbf{x}_j) = \sum_k d_k K_k(\mathbf{x}_i, \mathbf{x}_j)$
 - Combine heterogeneous sources of data
 - Perform feature selection

Advantages of Learning the Kernel

- Learn the kernel parameters
 - Improve accuracy and generalisation
 - Perform feature component selection
 - Perform dimensionality reduction
- Learn a linear combination of base kernels
- Learn a product of base kernels
 - $K(\mathbf{x}_i, \mathbf{x}_j) = \prod_k K_k(\mathbf{x}_i, \mathbf{x}_j)$

Advantages of Learning the Kernel

- Learn the kernel parameters
 - Improve accuracy and generalisation
 - Perform feature component selection
 - Perform dimensionality reduction
- Learn a linear combination of base kernels
- Learn a product of base kernels
- Combine some of the above

Linear Combinations of Base Kernels

- Learn a linear combination of base kernels

- $K(\mathbf{x}_i, \mathbf{x}_j) = \sum_k d_k K_k(\mathbf{x}_i, \mathbf{x}_j)$

$$\phi = \begin{array}{c} \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{blue} \\ \text{blue} \\ \text{green} \\ \text{green} \\ \text{green} \\ \text{green} \end{array} \begin{array}{l} \sqrt{d_1} \phi_1 \\ \sqrt{d_2} \phi_2 \\ \sqrt{d_3} \phi_3 \end{array}$$

Linear Combinations of Base Kernels

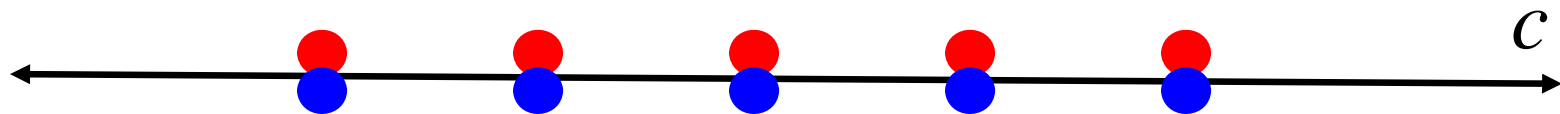
Schooner



Ketch



Simplistic 1D colour feature



- Linear colour kernel : $K_c(c_i, c_j) = \phi^t(c_i)\phi(c_j) = c_i c_j$
- Classification accuracy = 50%

Linear Combinations of Base Kernels

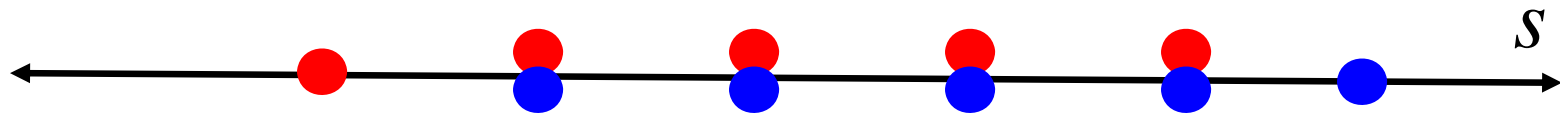
Schooner



Ketch



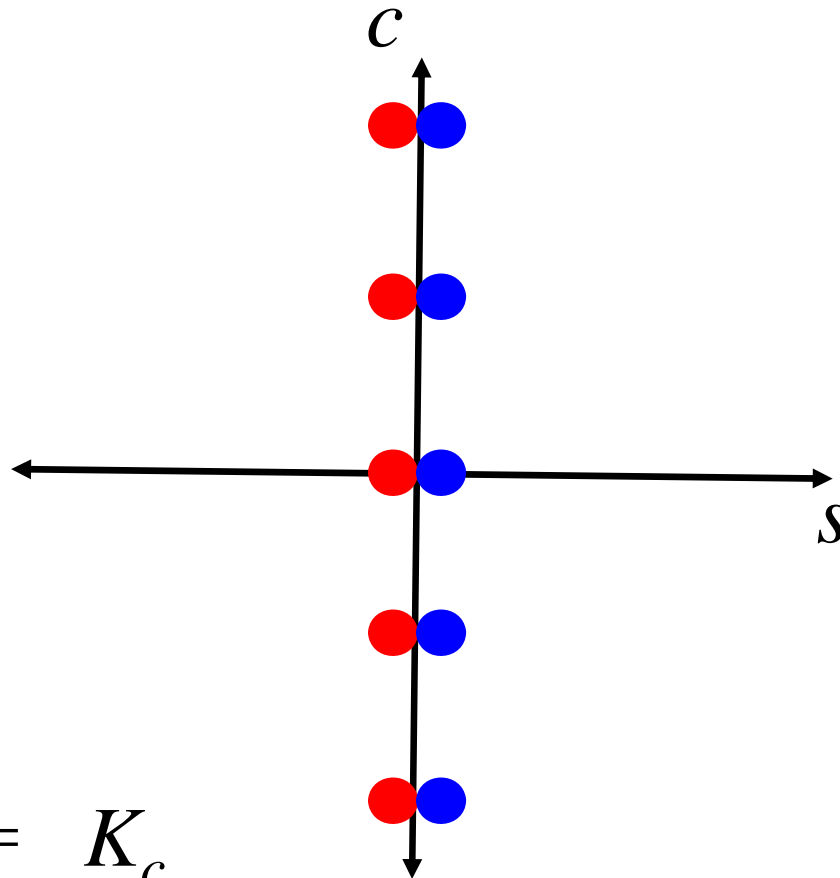
Simplistic 1D shape feature



- Linear shape kernel : $K_s(s_i, s_j) = \phi^t(s_i)\phi(s_j) = s_i s_j$
- Classification accuracy = 50% + ϵ

Linear Combinations of Base Kernels

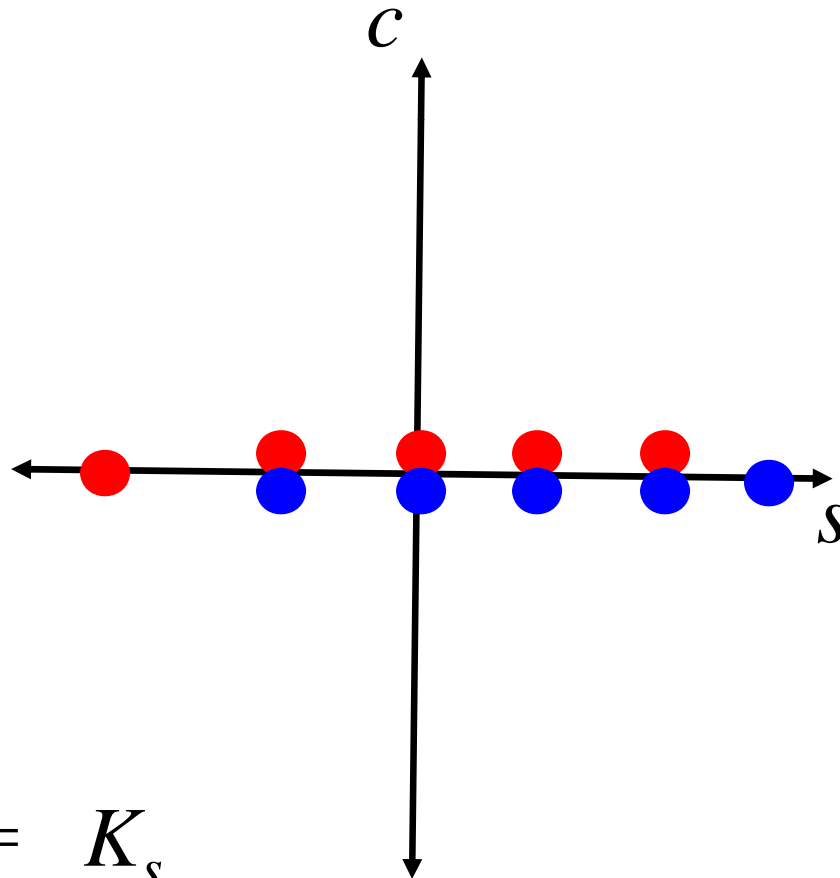
- We learn a combined colour-shape feature space.
- This is achieved by learning $K_{\text{opt}} = d K_s + (1-d) K_c$



$$d = 0 \Rightarrow K_{\text{opt}} = K_c$$

Linear Combinations of Base Kernels

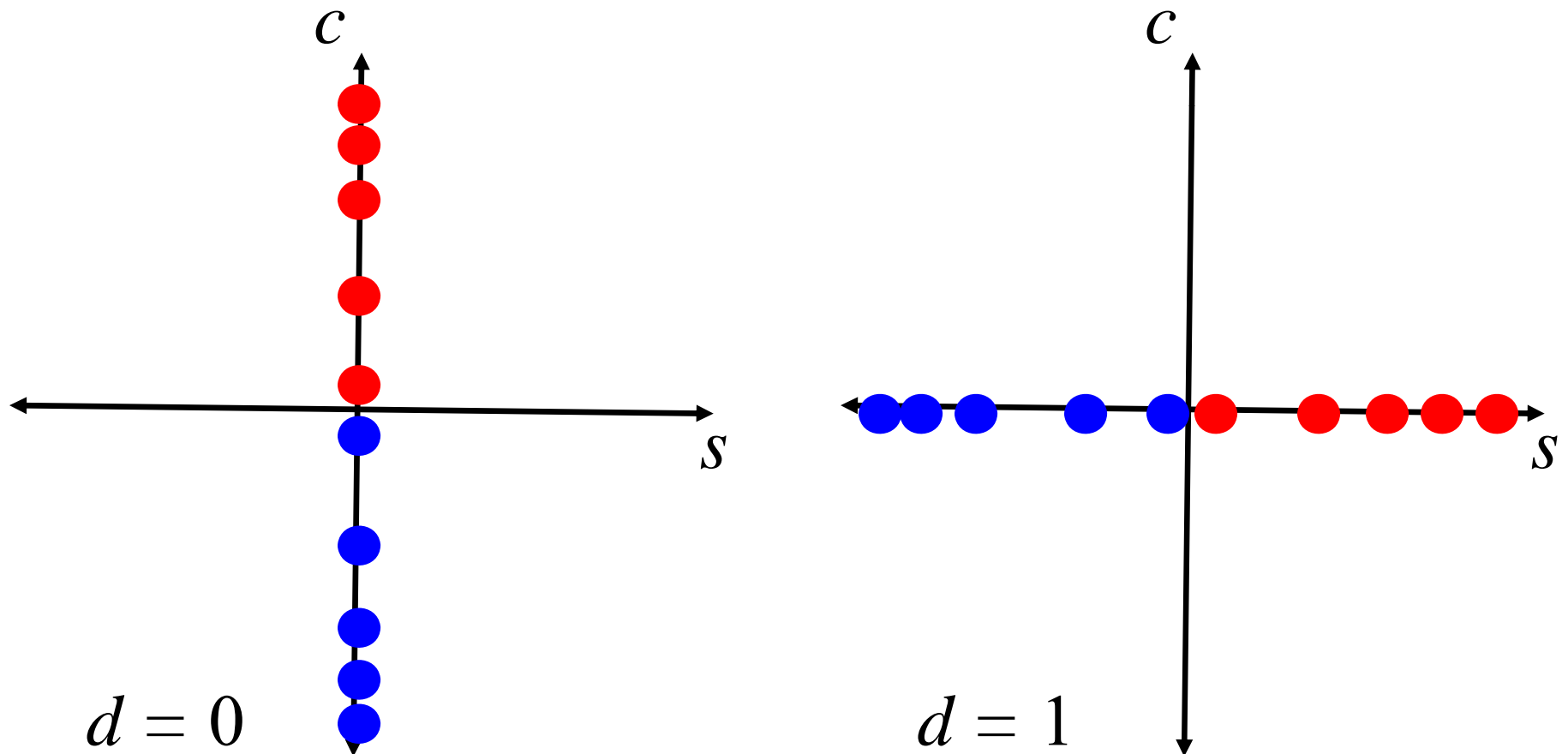
- We learn a combined colour-shape feature space.
- This is achieved by learning $K_{\text{opt}} = d K_s + (1-d) K_c$



$$d = 1 \Rightarrow K_{\text{opt}} = K_s$$

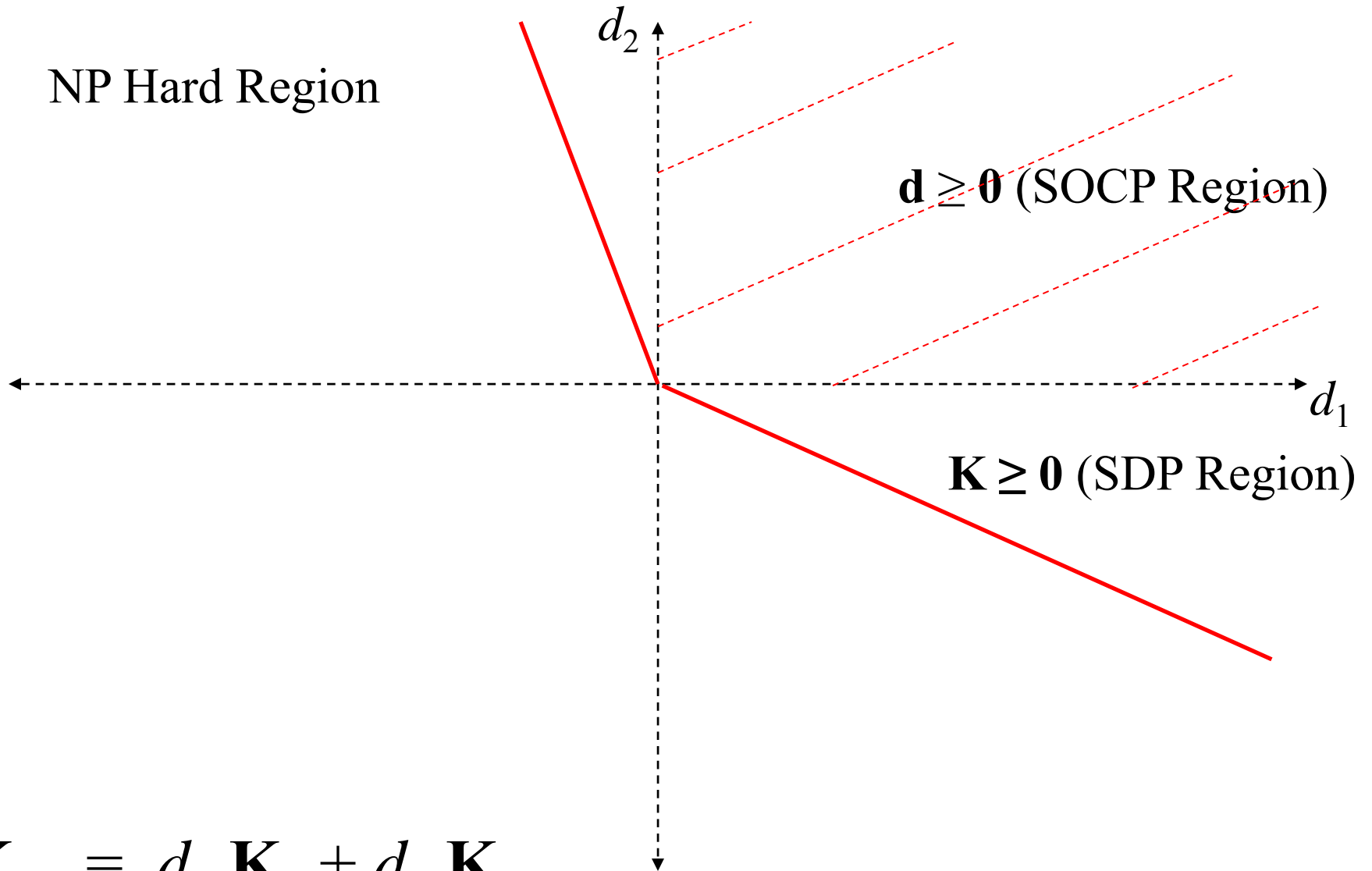
Linear Combinations of Base Kernels

- We learn a combined colour-shape feature space.
- This is achieved by learning $K_{\text{opt}} = d K_s + (1-d) K_c$



Our MKL Formulation

Multiple Kernel Learning [Varma & Ray 07]



$$\mathbf{K}_{\text{opt}} = d_1 \mathbf{K}_1 + d_2 \mathbf{K}_2$$

Multiple Kernel Learning Primal Formulation

- Minimise $\mathbf{w}, b, \xi, \mathbf{d}$ $\frac{1}{2}\mathbf{w}^t\mathbf{w} + C \sum_i \xi_i + \boldsymbol{\sigma}^t\mathbf{d}$
 - Subject to
 - $y_i [\mathbf{w}^t \boldsymbol{\phi}_{\mathbf{d}}(\mathbf{x}_i) + b] \geq 1 - \xi_i$
 - $\xi_i \geq 0$
 - $d_k \geq 0$
 - where
 - (\mathbf{x}_i, y_i) is the i^{th} training point.
 - C is the misclassification penalty.
 - $\boldsymbol{\sigma}$ encodes prior knowledge about kernels.
 - $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \sum_k d_k \mathbf{K}_k(\mathbf{x}_i, \mathbf{x}_j)$

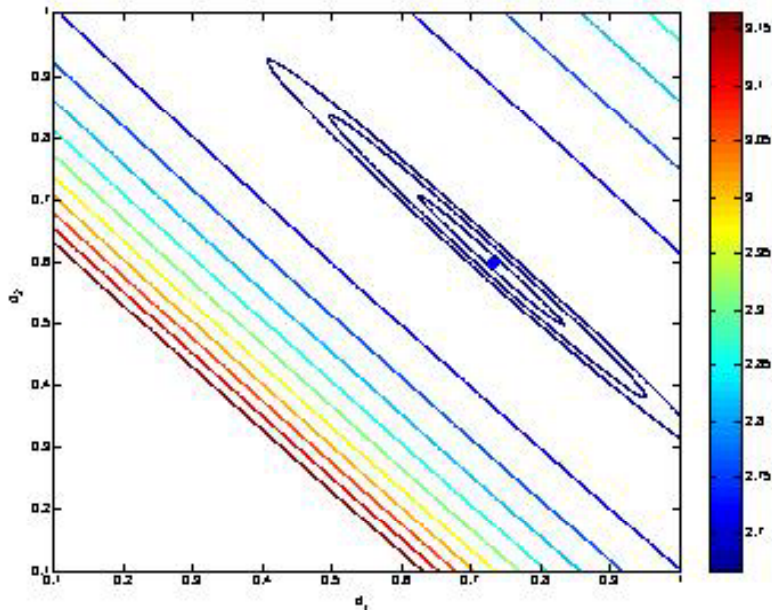
Multiple Kernel Learning Dual Formulation

- Maximise $\alpha^t \mathbf{1}$
 - Subject to
 - $0 \leq \alpha \leq \mathbf{C}$
 - $\mathbf{1}^t \mathbf{Y} \alpha = 0$
 - $\frac{1}{2} \alpha^t \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha \leq \sigma_k$
- The dual is a QCQP and can be solved by off-the-shelf solvers.
- QCQPs do not scale well to large problems.

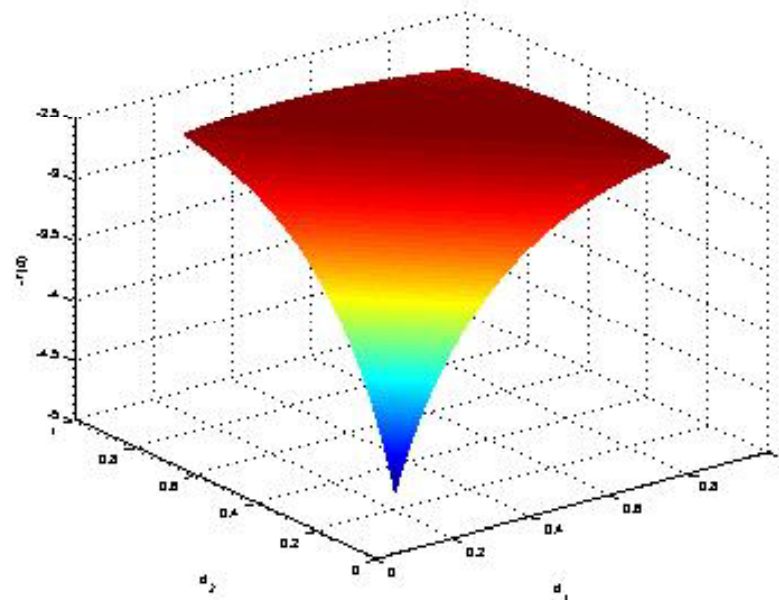
Large Scale Reformulation

- Minimise \mathbf{d} $T(\mathbf{d})$ subject to $\mathbf{d} \geq \mathbf{0}$
- where $T(\mathbf{d}) = \text{Min}_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_i \xi_i + \boldsymbol{\sigma}^t \mathbf{d}$
 - Subject to
 - $y_i [\mathbf{w}^t \boldsymbol{\phi}(\mathbf{x}_i) + b] \geq 1 - \xi_i$
 - $\xi_i \geq 0$
- In order to minimise T using gradient descent we need to
 - Prove that $\nabla_{\mathbf{d}} T$ exists.
 - Calculate $\nabla_{\mathbf{d}} T$ efficiently.

Large Scale Reformulation



Contour plot of $T(\mathbf{d})$



Surface plot of $-T(\mathbf{d})$

- We turn to the dual of T to prove differentiability and calculate the gradient.

Dual - Differentiability

- $W(\mathbf{d}) = \text{Max}_{\alpha} \mathbf{1}^t \alpha + \boldsymbol{\sigma}^t \mathbf{d} - \frac{1}{2} \sum_k d_k \alpha^t \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha$
 - Subject to
 - $\mathbf{1}^t \mathbf{Y} \alpha = 0$
 - $\mathbf{0} \leq \alpha \leq \mathbf{C}$
- $T(\mathbf{d}) = W(\mathbf{d})$ by the principle of strong duality.
- Differentiability with respect to \mathbf{d} comes from Danskin's Theorem [Danskin 1947].
- It can be guaranteed by ensuring that each \mathbf{K}_k is strictly positive definite.

Dual - Derivative

- $W(\mathbf{d}) = \text{Max}_{\alpha} \mathbf{1}^t \alpha + \boldsymbol{\sigma}^t \mathbf{d} - \frac{1}{2} \sum_k d_k \alpha^t \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha$

- Subject to

- $\mathbf{1}^t \mathbf{Y} \alpha = 0$

- $\mathbf{0} \leq \alpha \leq \mathbf{C}$

- Let $\alpha^*(\mathbf{d})$ be the optimal value of α so that

$$W(\mathbf{d}) = \mathbf{1}^t \alpha^* + \boldsymbol{\sigma}^t \mathbf{d} - \frac{1}{2} \sum_k d_k \alpha^{*t} \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha^*$$

$$\implies \partial W / \partial d_k = \sigma_k - \frac{1}{2} \alpha^{*t} \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha^* + (\dots) \partial \alpha^* / \partial d_k$$

$$\implies \partial W / \partial d_k = \sigma_k - \frac{1}{2} \alpha^{*t} \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha^*$$

Dual - Derivative

- $W(\mathbf{d}) = \text{Max}_{\alpha} \mathbf{1}^t \alpha + \sigma^t \mathbf{d} - \frac{1}{2} \sum_k d_k \alpha^t \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha$
 - Subject to
 - $\mathbf{1}^t \mathbf{Y} \alpha = 0$
 - $\mathbf{0} \leq \alpha \leq \mathbf{C}$
- Let $\alpha^*(\mathbf{d})$ be the optimal value of α so that
$$\partial T / \partial d_k = \partial W / \partial d_k = \sigma_k - \frac{1}{2} \alpha^{*t} \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha^*$$
- Since \mathbf{d} is fixed, $W(\mathbf{d})$ is the standard SVM dual and α^* can be obtained using any SVM solver.

Final Algorithm

1. Initialise \mathbf{d}^0 randomly
2. Repeat until convergence
 - a) Form $\mathbf{K}(x,y) = \sum_k d_k^n \mathbf{K}_k(x,y)$
 - b) Use any SVM solver to solve the standard SVM problem with kernel \mathbf{K} and obtain α^* .
 - c) Update $d_k^{n+1} = d_k^n - \varepsilon^n (\sigma_k - \frac{1}{2} \alpha^{*t} \mathbf{Y} \mathbf{K}_k \mathbf{Y} \alpha^*)$
 - d) Project \mathbf{d}^{n+1} back onto the feasible set if it does not satisfy the constraints $\mathbf{d}^{n+1} \geq 0$

Application to Object Recognition

Object Categorization



Chair



Schooner



Ketch



Taj



Panda



Novel image to be classified

?
=

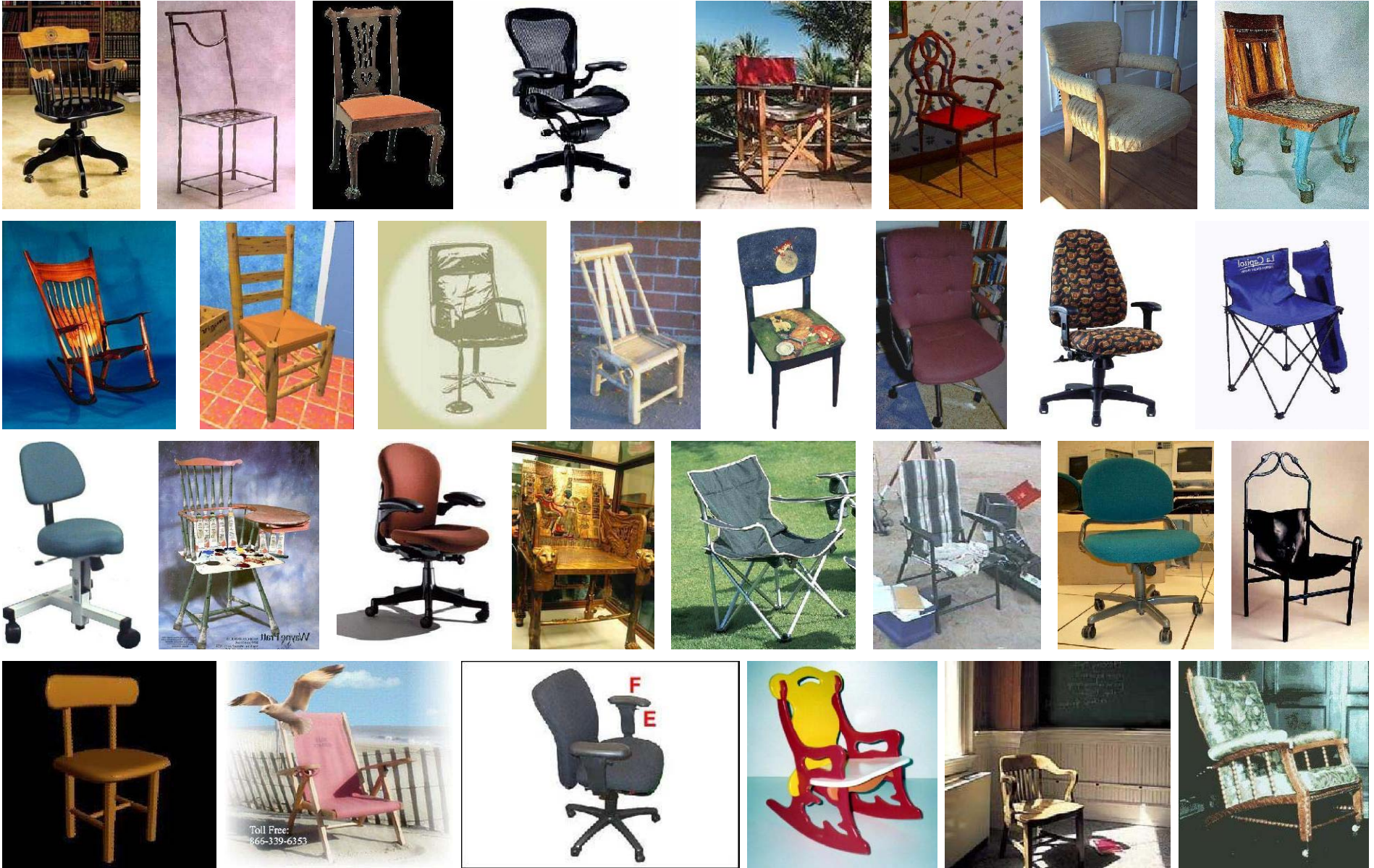
Labelled images comprise training data

The Caltech 101 Database

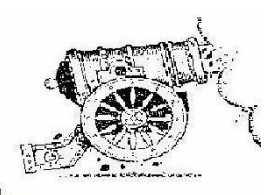
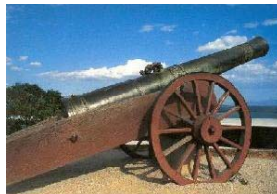
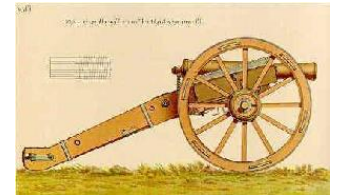
- Object category database collected by Fei-Fei *et al.* [PAMI 2006].



The Caltech 101 Database – Chairs



The Caltech 101 Database – Cannons



Caltech 101 – Experimental Setup

- Experimental setup kept identical to that of Zhang *et al.* [CVPR 2006]
 - 102 classes, 30 images / class = 3060 images.
 - 15 images / class used for training and the other 15 for testing.
 - Results reported over 20 random splits

Descriptor	1NN	SVM (1-vs-1)
GB	$39.67 \pm 1.02\%$	$57.33 \pm 0.94\%$
GBDist	$45.23 \pm 0.96\%$	$59.30 \pm 1.00\%$
AppGray	$42.08 \pm 0.81\%$	$52.83 \pm 1.00\%$
AppColour	$32.79 \pm 0.92\%$	$40.84 \pm 0.78\%$
Shape180	$32.01 \pm 0.89\%$	$48.83 \pm 0.78\%$
Shape360	$31.17 \pm 0.98\%$	$50.63 \pm 0.88\%$

Table 3. Classification results on the Caltech 101 dataset. The MKL-Block l_1 method of [4] achieves $76.55 \pm 0.84\%$ for 1-vs-1 classification when combining all the descriptors. Our results are **$78.43 \pm 1.05\%$** (1-vs-1) and **$87.82 \pm 1.00\%$** (1-vs-All).

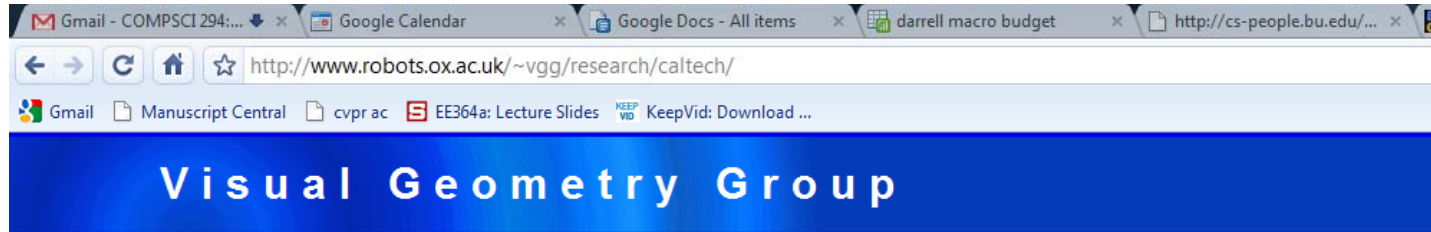


Image classification - Caltech datasets

[Anna Bosch](#) and [Andrew Zisserman](#)

Warning: the Kernel matrices that were previously available from this site were found to contain errors that positively benefited classification performance. We are currently investigating the problem and will report the findings here.

We are grateful to Nicolas Pinto and Peter Gehler for alerting us to these errors.

Overview

The objective of this work is classifying images by the object categories they contain. To this end we combine shape and appearance representations over a region of interest to learn the object model.

Challenges

We need to be able to recognize objects in images despite within class variations (see examples below) and imaging variations such as: scale, viewpoint, lighting and background. Additionally, we need to learn which are the best descriptors to classify each specific object category.



Experimental Results – Caltech 101

	1-NN	SVM (1-vs-1)	SVM (1-vs-All)
Shape GB1	39.67 ± 1.02	57.33 ± 0.94	62.98 ± 0.70
Shape GB2	45.23 ± 0.96	59.30 ± 1.00	61.53 ± 0.57
Self Similarity	40.09 ± 0.98	55.10 ± 1.05	60.83 ± 0.84
Gist	30.41 ± 0.85	45.56 ± 0.82	51.46 ± 0.79
MKL Block l_1		62.09 ± 0.40	72.05 ± 0.56
Our		65.73 ± 0.77	77.47 ± 0.36

Comparisons to the state-of-the-art

- Zhang *et al.* [CVPR 06]: $59.08 \pm 0.38\%$ by combining shape and texture cues. Frome *et al.* add colour and get $60.3 \pm 0.70\%$ [NIPS 2006] and 63.2% [ICCV 2007].
- Lin *et al.* [CVPR 07]: 59.80% by combining 8 features using Kernel Target Alignment
- Boiman *et al.* [CVPR 08]: $72.8 \pm 0.39\%$ by combining 5 descriptors.

Extending MKL

Extending Our MKL Formulation

- Minimise $\mathbf{w}, b, \mathbf{d}$ $\frac{1}{2}\mathbf{w}^t\mathbf{w} + \sum_i L(f(\mathbf{x}_i), y_i) + l(\mathbf{d})$
 - Subject to constraints on \mathbf{d}
 - where
 - (\mathbf{x}_i, y_i) is the i^{th} training point.
 - $f(\mathbf{x}) = \mathbf{w}^t\phi_{\mathbf{d}}(\mathbf{x}) + b$
 - L is a general loss function.
 - $\mathbf{K}_{\mathbf{d}}(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function parameterised by \mathbf{d} .
 - l is a regularizer on the kernel parameters.

Kernel Generalizations

- The learnt kernel can now have any functional form as long as
 - $\nabla_{\mathbf{d}}K(\mathbf{d})$ exists and is continuous.
 - $K(\mathbf{d})$ is strictly positive definite for feasible \mathbf{d} .
 - For example, $K(\mathbf{d}) = \sum_k d_{k0} \prod_l \exp(-d_{kl} \chi^2)$
- However, not all kernel parameterizations lead to convex formulations.
- This does not appear to be a problem for the applications we have tested on.

Regularization Generalizations

- Any regulariser can be used as long as it has continuous first derivative with respect to \mathbf{d}
 - We can now put Gaussian rather than Laplacian priors on the kernel weights.
 - We can have other sparsity promoting priors.
 - We can, once again, have negative weights.

Loss Function Generalizations

- The loss function can be generalized to handle
 - Regression.
 - Novelty detection (1 class SVM).
 - Multi-class classification.
 - Ordinal Regression.
 - Ranking.

Multiple Kernel Regression Formulation

- Minimise $\mathbf{w}, b, \mathbf{d}$ $\frac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_i \xi_i + l(\mathbf{d})$
 - Subject to
 - $|\mathbf{w}^t \phi_{\mathbf{d}}(\mathbf{x}_i) + b - y_i| \leq \varepsilon + \xi_i$
 - $\xi_i \geq 0$
 - $d_k \geq 0$
 - where
 - (\mathbf{x}_i, y_i) is the i^{th} training point.
 - C and ε are user specified parameters.

Probabilistic Interpretation for Regression

- MAP estimation with the following priors and likelihood

$$p(b) = \text{const} \quad (\text{improper prior})$$

$$p(\mathbf{d}) = \begin{cases} \text{const} e^{-\lambda l(\mathbf{d})} & \text{if } \mathbf{d} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$p(\boldsymbol{\alpha}|\mathbf{d}) = |(\lambda/2\pi)\mathbf{K}_d|^{1/2} e^{-1/2\lambda\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha}}$$

$$p(y|\mathbf{x},\boldsymbol{\alpha},\mathbf{d},b) = 1/2(1+\varepsilon)^{-1} e^{-\text{Max}(0, |b - \boldsymbol{\alpha}'\mathbf{K}(:,\mathbf{x}) - y| - \varepsilon)}$$

is equivalent to the optimization involved in our Multiple Kernel Regression formulation

$$\text{Min}_{\boldsymbol{\alpha},\mathbf{d},b} 1/2\boldsymbol{\alpha}'\mathbf{K}_d\boldsymbol{\alpha} + C \sum_i \text{Max}(0, |b - \boldsymbol{\alpha}'\mathbf{K}_d(:,\mathbf{x}) - y| - \varepsilon) + l(\mathbf{d}) - 1/2C \log(|\mathbf{K}_d|)$$

- This is very similar to the Marginal Likelihood formulation in Gaussian Processes.

Predicting Facial Attractiveness



Anandan



Chris

What is their rating on the milliHelen scale?

Hot or Not – www.hotornot.com

Over **12 Billion** votes counted &
25,987,000 photos submitted.

Official Rating

9.9

based on 65535 votes



You rated her: 9

She checked her score:
45 months ago

Over **12 Billion** votes counted &
25,987,000 photos submitted.

Official Rating

8.7

based on 116 votes

[See Profile](#) [Meet Me](#)



[Email](#) [Send Flower](#)

You rated her: 9

She checked her score:
11 hours ago

Over **12 Billion** votes counted &
25,987,000 photos submitted.

Official Rating

9

based on 335 votes

[See Profile](#) [Meet Me](#)



[Email](#) [Send Flower](#)

You rated her: 8

She checked her score:
11 hours ago

Regression on Hot or Not Training Data



7.3



6.5



7.5



9.4



7.7



8.7



6.9



6.5

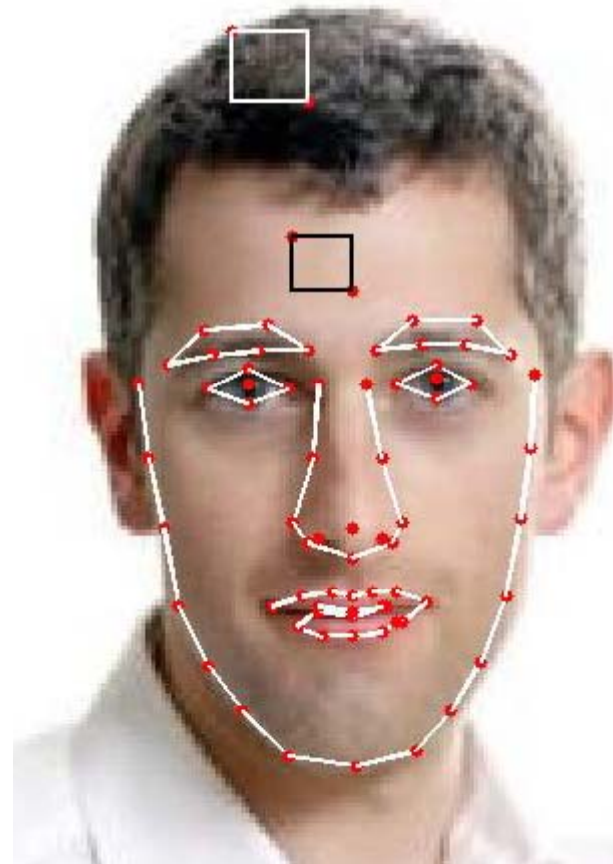


7.4



7.7

Predicting Facial Attractiveness – Features



- Geometric features: Face, eyes, nose and lip contours.
- Texture patch features: Skin and hair.
- HSV colour features: Skin and hair.

Predicting Facial Attractiveness

Francis Bach



8.85

Luc Van Gool



8.58

Phil Torr



8.38

Richard Hartley



8.35

Jean Ponce



8.30

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.

- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008,
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
 - but first, Lampert et. al., CVPR2008....
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008,
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Beyond Sliding Windows: Object Localization by *Efficient Subwindow Search*

Christoph H. Lampert[†], Matthew B. Blaschko[†], & Thomas Hofmann[‡]



Max Planck Institute for Biological Cybernetics[†]
Tübingen, Germany

Google, Inc.[‡]
Zürich, Switzerland



MAX-PLANCK-GESELLSCHAFT



BIOLOGISCHE KYBERNETIK

Learning to Localize Objects with Structured Output Regression

Matthew B. Blaschko and Christoph H. Lampert



Max Planck Institute for Biological Cybernetics
Tübingen, Germany

October 13th, 2008



MAX-PLANCK-GESELLSCHAFT



BIOLOGISCHE KYBERNETIK

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Object Recognition by Integrating Multiple Image Segmentations

Caroline Pantofaru^{1*}, Cordelia Schmid², and Martial Hebert¹

¹ The Robotics Institute, Carnegie Mellon University, USA

² INRIA Grenoble, LEAR, LJK, France

`crp@ri.cmu.edu, cordelia.schmid@inrialpes.fr, hebert@ri.cmu.edu`

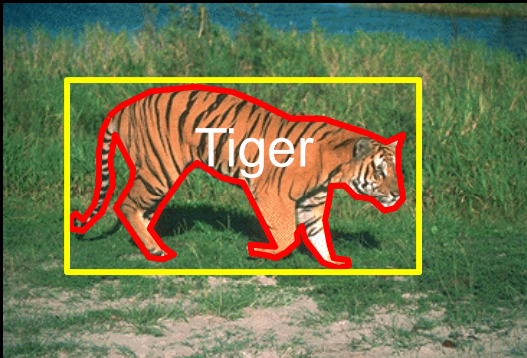
Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Recognition using Regions

Chunhui Gu, Joseph Lim,
Pablo Arbelaez, Jitendra Malik
UC Berkeley

Grand Recognition Problem

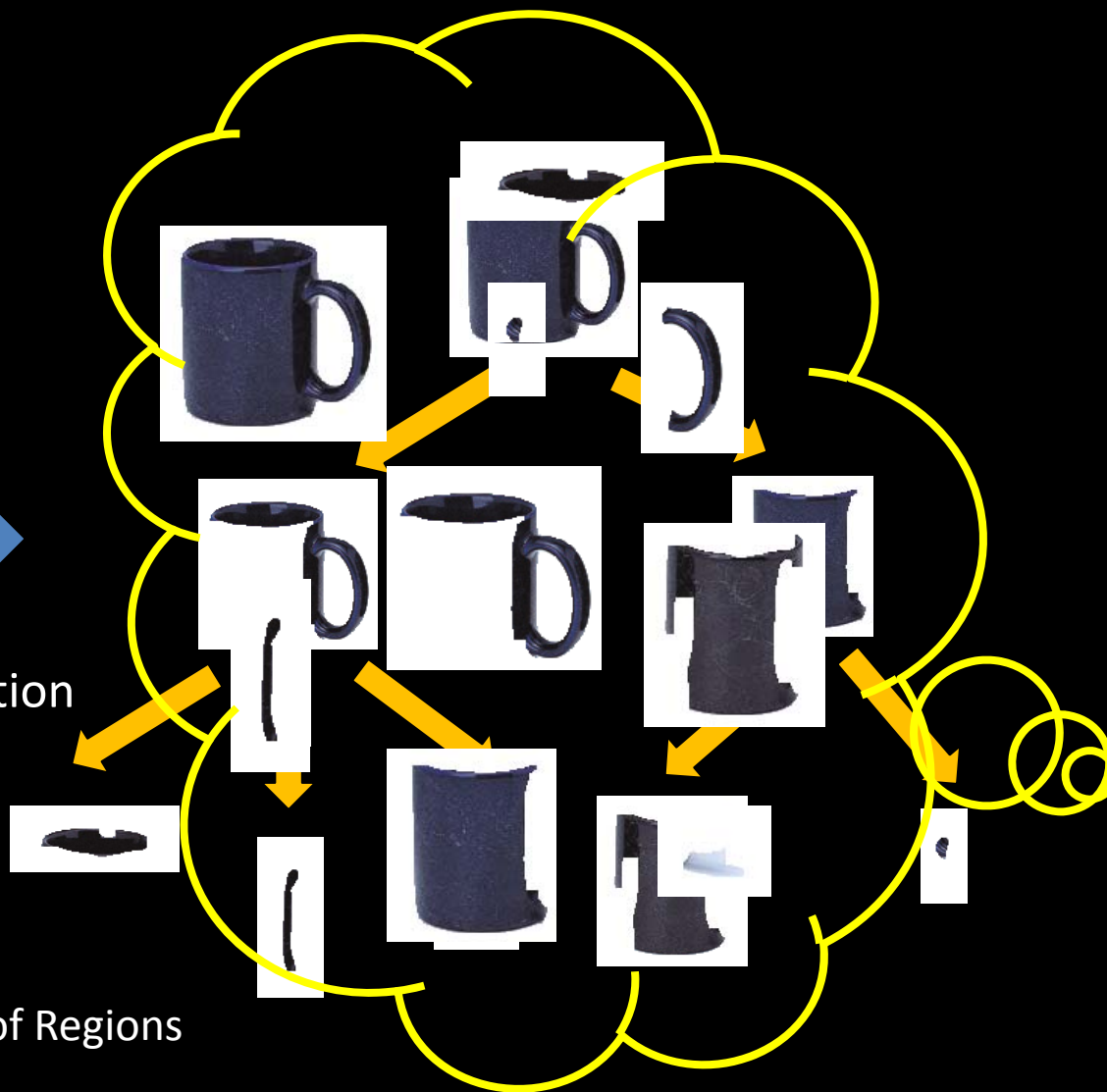


- Detection
- Segmentation
- Classification

Region Extraction

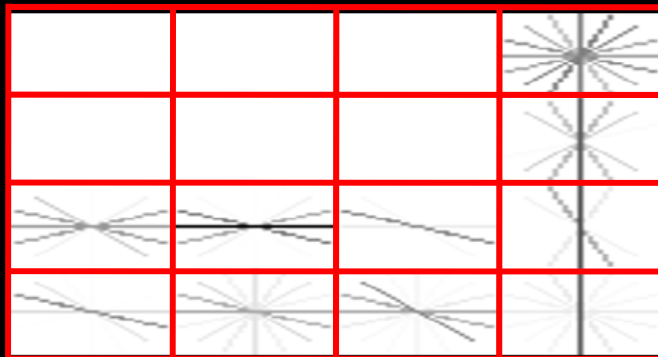
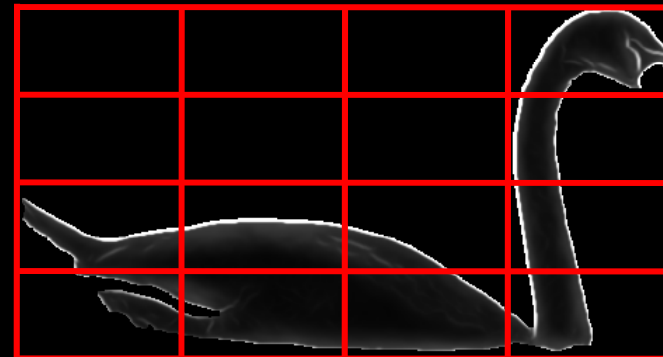


Region
Segmentation



Bag of Regions

Region Description



Weight Learning

- Not all regions are equally important



image J

D_{IJ}



exemplar I

D_{IK}



image K

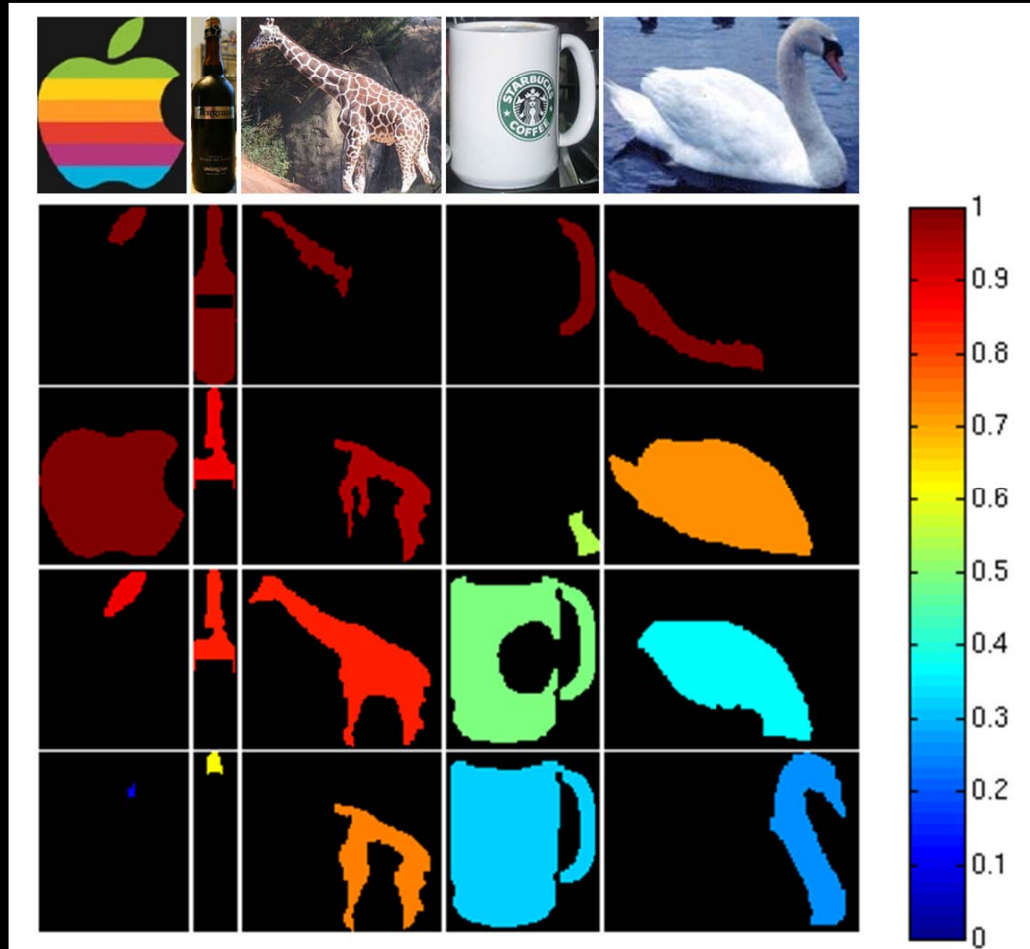


$$D_{IJ} = \sum_i w_i \cdot d_i^J \quad \text{and} \quad d_i^J = \min_j \chi^2(f_i^I, f_j^J)$$

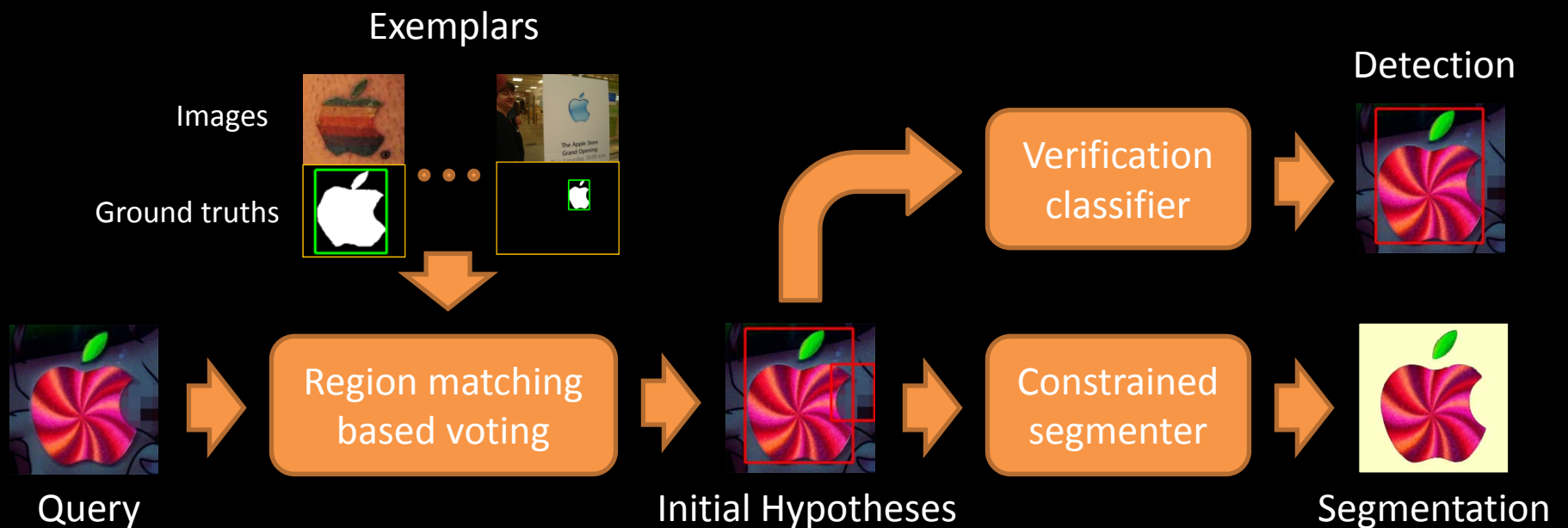
want: $D_{IK} > D_{IJ}$

Max-margin formulation results in a sparse solution of weights.

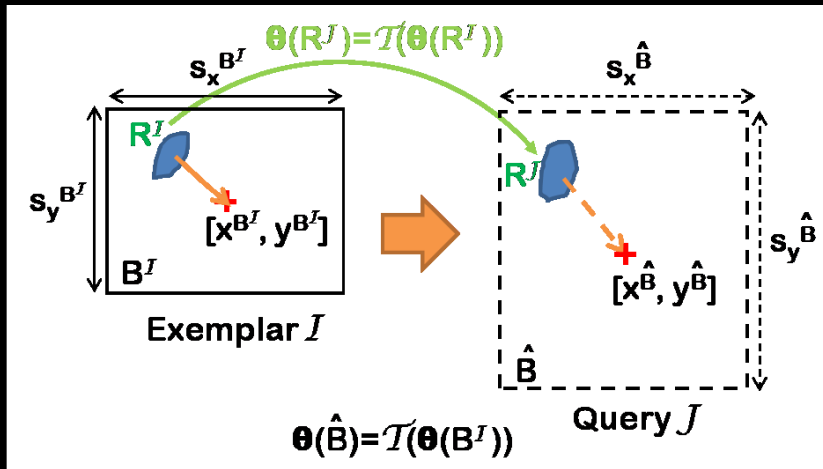
Weight Learning



Detection/Segmentation Algorithm



Voting



$$x^{\hat{B}} = x^{R^J} + (x^{B^I} - x^{R^I}) \cdot s_x^{R^J} / s_x^{R^I}$$

$$s_x^{\hat{B}} = s_x^{B^I} \cdot s_x^{R^J} / s_x^{R^I}$$



Exemplar



Query

Verification

- For exemplar $1, 2, \dots, N$ of class C , define likelihood function of query image J :

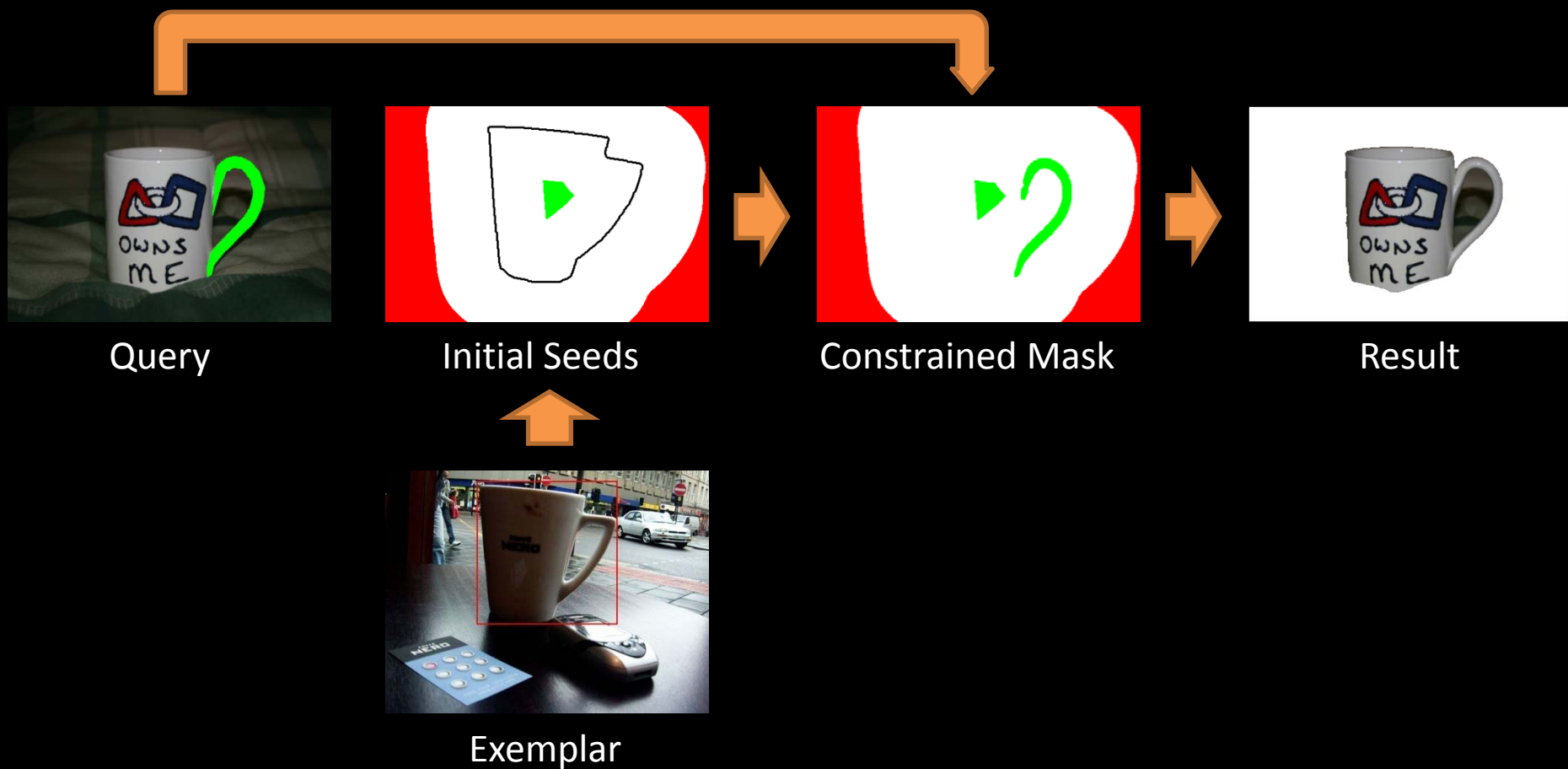
$$L_J(C) = \frac{1}{N} \sum_{I=1}^N f_I(D_{IJ})$$

where f converts a distance to a similarity measure (e.g. logistic regression, negation)

- The predicted category label for image J :

$$\tilde{C}_J = \arg \max_C (L_J(C))$$

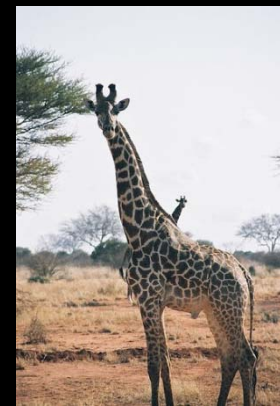
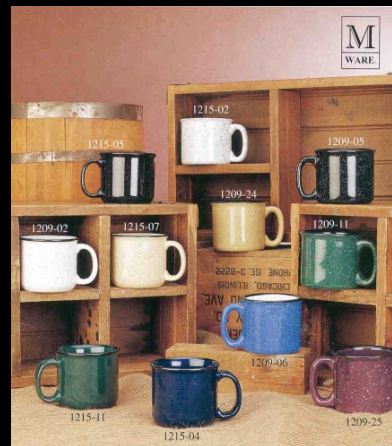
Segmentation



Results

ETHZ Shape (Ferrari et al. 06)

- Contains 255 images of 5 diverse shape-based classes.



Detection/Segmentation Results

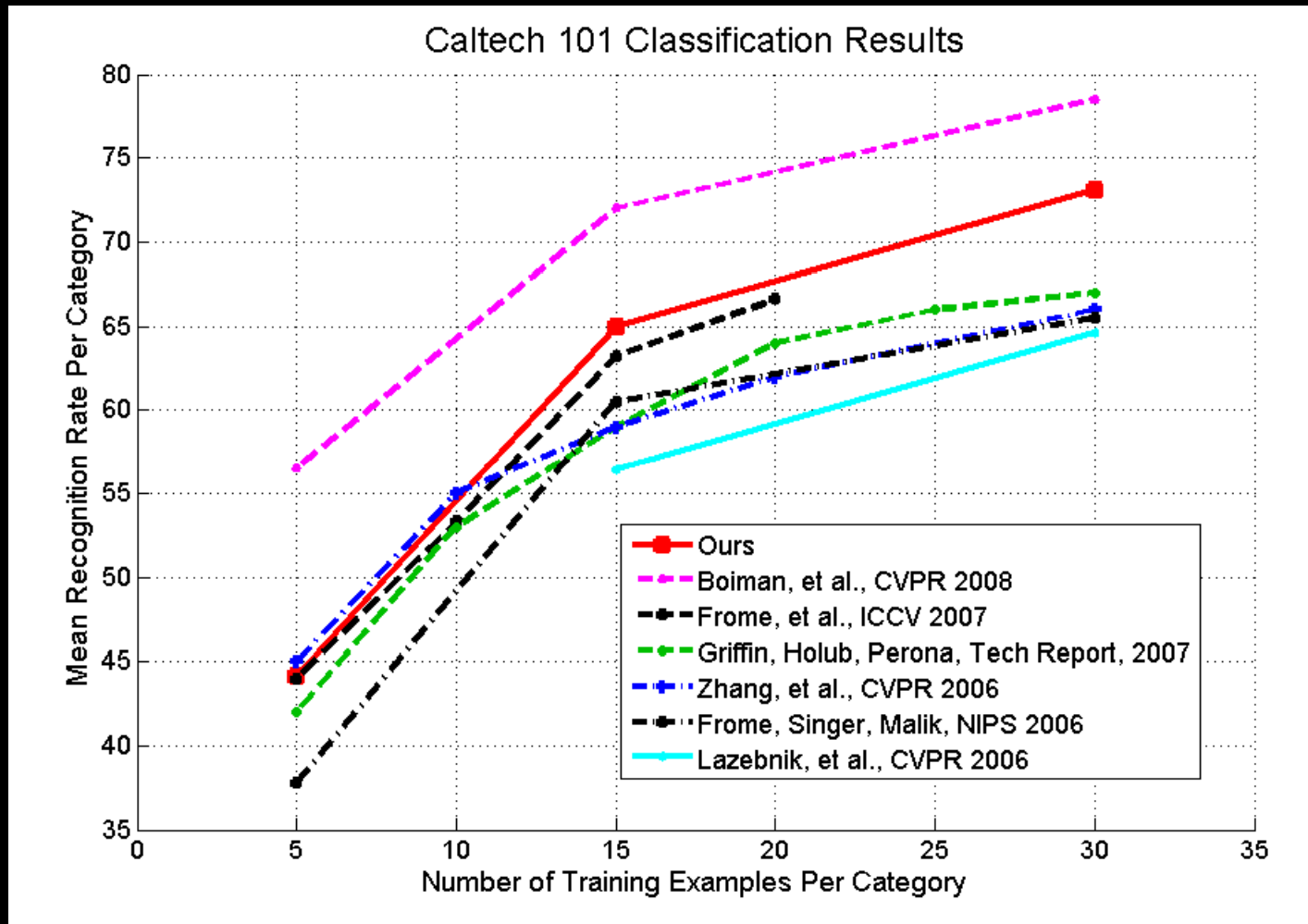
- Significantly outperforms the state of the art (Ferrari et al. CVPR07) – 87.1% vs. 67.2% det. rate at 0.3 FFPI
- 75.7% Average Precision rate in segmentation, >20% boost w.r.t. bounding boxes



A Comparison to Sliding Window

Categories	Sld. Windows	Regions	Bnd. Boxes
Applelogos	~ 30,000	115	3.1
Bottles	~ 1,500	168	1.1
Giraffes	~ 14,000	156	6.9
Mugs	~ 16,000	189	5.3
Swans	~ 10,000	132	2.3

Classification Rate in Caltech 101



Results with Cue Combination

Image cues	5 train	15 train	30 train
(R) Contour shape	41.5	55.1	60.4
(R) Edge shape	30.0	42.9	48.0
(R) Color	19.3	27.1	27.2
(R) Texture	23.9	31.4	32.7
(R) All	40.9	59.0	65.2
(P) GB	42.6	58.4	63.2
(R) Contour shape+(P) GB	44.1	65.0	73.1
(R) All + (P) GB	45.7	64.4	72.5

Table 4. Mean classification rate (%) in Caltech 101 using individual and combinations of image cues. (R) stands for region-based, and (P) stands for point-based. (R)All means combining all region cues (Contour shape+Edge shape+Color+Texture). We notice that cue combination boosts the overall performance significantly.

Conclusion

- Introduce a unified framework for object detection, segmentation and classification
- Regions encode shape and scale information of objects naturally
- Cue combination improves recognition performance
- Region-based Hough voting significantly reduces number of candidate windows for detection

Thank You

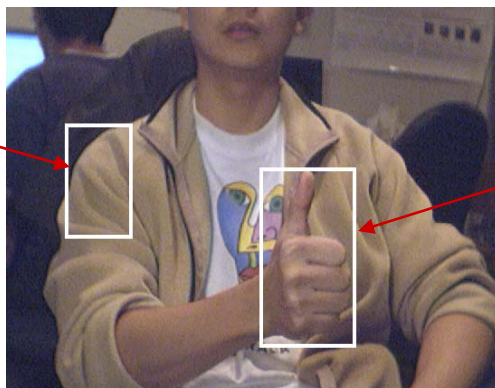
Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008,
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Multiplicative kernels for parameterized detectors

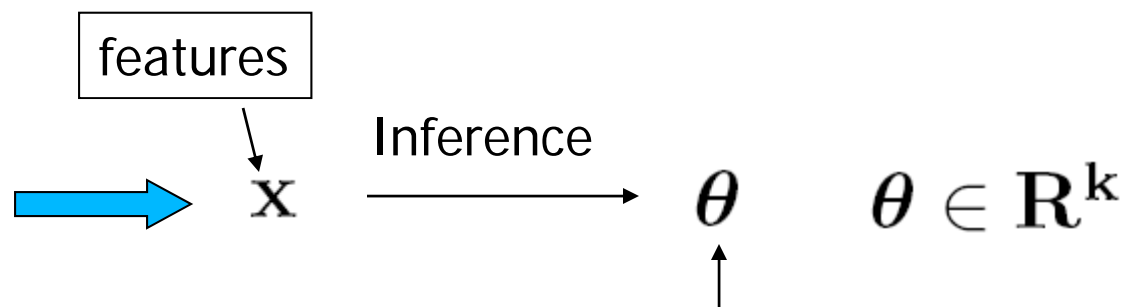
- Object Detection (binary classification)

Umm, this doesn't look like a hand



A hand is here!

- Foreground State Estimation



Angles between fingers, view angle...

Proposed Approach

We try to learn a function $C(\mathbf{x}, \theta)$ which tells whether \mathbf{x} is an instance of the object with foreground state θ .

$$C(\mathbf{x}, \theta) \begin{cases} > 0, \mathbf{x} \text{ is an instance of the object with } \theta \\ \leq 0, \textit{ otherwise.} \end{cases}$$

Why $C(\mathbf{x}, \theta)$ - intuitions

- If we fix θ , $C(\cdot, \theta)$ is a detector of a specific θ
- Parameter estimation can be achieved by searching of best θ via $C(\mathbf{x}, \theta)$.

$$C(\mathbf{x}, \theta)$$

Learning of function C

- Assume C can be factorized into a feature space mapping $\phi_x(\mathbf{x})$ and a weight vector $\mathbf{w}(\boldsymbol{\theta})$:

$$C(\mathbf{x}, \boldsymbol{\theta}) = \phi_x(\mathbf{x})^T \mathbf{w}(\boldsymbol{\theta})$$

where $\phi_x(\mathbf{x}) = [\phi_x^0(\mathbf{x}), \phi_x^1(\mathbf{x}), \dots, \phi_x^N(\mathbf{x})]^T$

Approximation of W

- $\mathbf{w}(\boldsymbol{\theta})$ is approximated by basis function expansion:

$$\mathbf{w}(\boldsymbol{\theta}) = \sum_{i=0}^M \mathbf{v}_i \phi_{\boldsymbol{\theta}}^i(\boldsymbol{\theta}) = \mathbf{V} \boldsymbol{\phi}_{\boldsymbol{\theta}}(\boldsymbol{\theta})$$

where vectors \mathbf{v}_i are unknowns and

$$\mathbf{V} = [\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_M]$$

$$\boldsymbol{\phi}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = [\phi_{\boldsymbol{\theta}}^0(\boldsymbol{\theta}), \phi_{\boldsymbol{\theta}}^1(\boldsymbol{\theta}), \dots, \phi_{\boldsymbol{\theta}}^M(\boldsymbol{\theta})]^T.$$

Kernel Representation

- If we plug $\mathbf{w}(\boldsymbol{\theta})$ and $\phi_x(\mathbf{x})$ into $C(\mathbf{x}, \boldsymbol{\theta})$

$$\begin{aligned} C(\mathbf{x}, \boldsymbol{\theta}) &= \phi_x(\mathbf{x})^T \mathbf{V} \phi_\theta(\boldsymbol{\theta}) \\ &= \begin{bmatrix} \phi_\theta^0(\boldsymbol{\theta}) \phi_x(\mathbf{x}) \\ \phi_\theta^1(\boldsymbol{\theta}) \phi_x(\mathbf{x}) \\ \vdots \\ \phi_\theta^M(\boldsymbol{\theta}) \phi_x(\mathbf{x}) \end{bmatrix}^T \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_M \end{bmatrix} \\ &= \phi_{\mathbf{x}, \boldsymbol{\theta}}^T \mathbf{v}, \end{aligned}$$

The unknowns are in \mathbf{v} . $\phi_{\mathbf{x}, \boldsymbol{\theta}}$ is the data term.

Kernel Representation


- If we solve the binary classification problem by SVM, the actual kernel is,

$$\begin{aligned}k_c(\mathbf{y}, \mathbf{y}') &= \phi_{\mathbf{x}, \boldsymbol{\theta}}^T \phi_{\mathbf{x}', \boldsymbol{\theta}'} \\ &= [\phi_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T \phi_{\boldsymbol{\theta}}(\boldsymbol{\theta}')] [\phi_x(\mathbf{x})^T \phi_x(\mathbf{x}')] \\ &= k_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\theta}') k_x(\mathbf{x}, \mathbf{x}'),\end{aligned}$$

where $\mathbf{y} = [\mathbf{x}^T, \boldsymbol{\theta}^T]^T$

Kernel Representation

- After SVM learning, the classification function becomes

$$\begin{aligned} C(\mathbf{x}, \boldsymbol{\theta}) &= \sum_{i \in SV} \alpha_i k_{\theta}(\boldsymbol{\theta}_i, \boldsymbol{\theta}) k_x(\mathbf{x}_i, \mathbf{x}) \\ &= \sum_{i \in SV} \alpha'_i(\boldsymbol{\theta}) k_x(\mathbf{x}_i, \mathbf{x}), \end{aligned}$$


where α_i is the weight of i -th support vector and

$$\alpha'_i(\boldsymbol{\theta}) = \alpha_i k_{\theta}(\boldsymbol{\theta}_i, \boldsymbol{\theta}).$$

Short Summary

- We have a way to learn $C(\mathbf{x}, \boldsymbol{\theta})$ which evaluates tuples $(\mathbf{x}, \boldsymbol{\theta})$.
- $C(\mathbf{x}, \boldsymbol{\theta})$ corresponds to a family of detectors tuned to different $\boldsymbol{\theta}$.
- Feature sharing is implicit via sharing of support vectors.

Training Process

Each training sample is in the form of a tuple (\mathbf{x}, θ) .

$$\mathbf{x}_1 = \text{img}_1, \theta_1 = 0$$

$$\mathbf{x}_2 = \text{img}_2, \theta_2 = 30$$

$$\mathbf{x}_3 = \text{img}_3, \theta_3 = 70$$

Foreground tuples

$$\mathbf{x} = \text{img}_4, \theta = 0, 10, 20, \dots$$

For a background tuple, the parameter can be random.

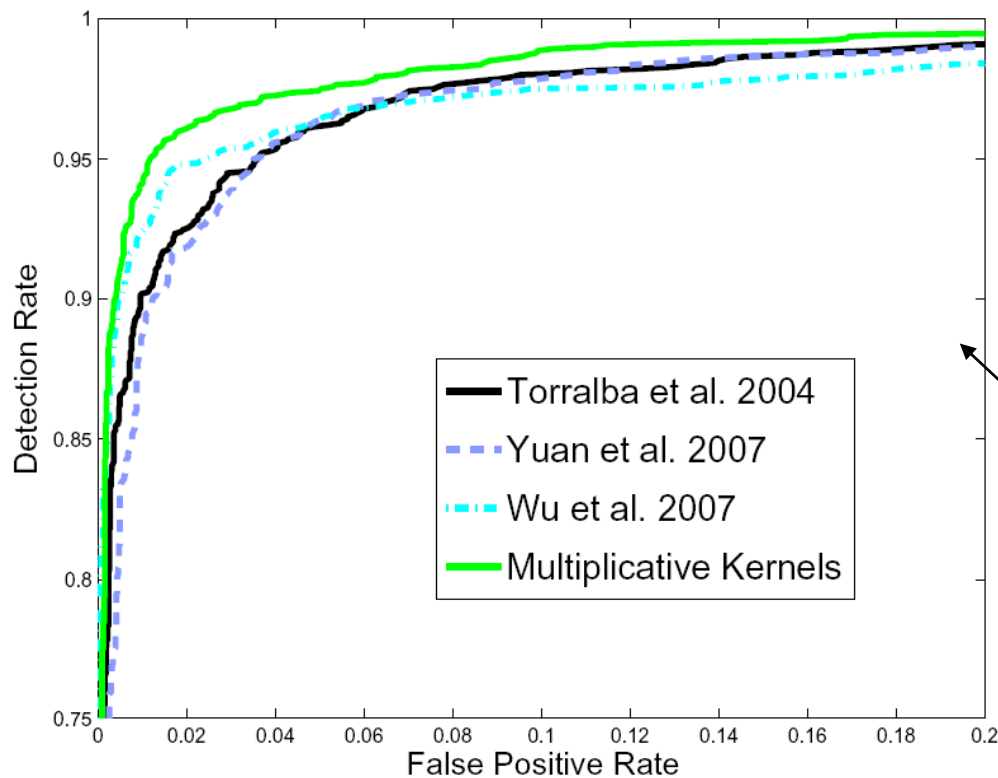
We propose an iterative bootstrap training process to avoid potentially huge number of background training tuples.

Feasibility Experiment III

- Vehicle data set, 12 view categories.



Feasibility Experiment III



k_θ is an RBF kernel defined with Euclidean distance in HOG space kernel. k_x is a linear kernel.

Detection accuracy compared with previous detection methods.

Feasibility Experiment III

Vehicle view angle estimation (multi-class classification)

Baseline method (12 subclass detectors): 62.4%

Torralba et al. 2004: 65.0%

In both methods, view labels of all FG training samples are given (1405 in total).

Multiplicative Kernels:

view labels needed in our approach

#modes	400	600	800
accuracy	65.6%	68.0%	71.7%

Today – Kernel Combination, Segmentation, and Structured Output

- M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007,
- Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Multiplicative kernels: Object detection, segmentation and pose estimation," in Computer Vision and Pattern Recognition, 2008. CVPR 2008
- M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in ECCV 2008.
- C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," CVPR 2008,
- Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, Jitendra Malik, [Recognition using Regions](#), CVPR 2009, to appear

Next Lecture – Image Context

- A. Torralba, K. P. Murphy, and W. T. Freeman, "Contextual models for object detection using boosted random fields," in Advances in Neural Information Processing Systems 17 (NIPS), 2005.
- D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," in Computer Vision and Pattern Recognition, 2006
- L.-J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in Computer Vision, 2007.
- G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in ECCV 2008, pp. 30-43.
- S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Messner, G. R. Bradski, P. Baumstarck, S. Chung, A. Y. Ng: Peripheral-Foveal Vision for Real-time Object Recognition and Tracking in Video. IJCAI 2007
- Y. Li and R. Nevatia, "Key object driven multi-category object recognition, localization and tracking using spatio-temporal context," in ECCV 2008