# Adapting Codes and Embeddings for Polychotomies

**Gunnar Rätsch**, Alex Smola & Sebastian Mika

**Australian National University, Canberra**
**Fraunhofer FIRST, Berlin**

THE AUSTRALIAN NATIONAL UNIVERSITY

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

## Motivation: Chemistry

THE AUSTRALIAN NATIONAL UNIVERSITY

| Name | Amino Acid Sequence |
|---|---|
| DEHUE7 | MARTVVLITGCSSGIGLHLAVRLASDPSQSFKVYATLRDLKTQGF |
| S41120 | MDLSAASHRIPLSDGNSIPIIGLGTYSEPKSTPKGACATSVKVAID |
| A57407 | MDPKYQRVELNDGHFMPVLGFGTYAPPEVPRNRAVEVTKLAIE |
| DEHULH | MATLKEKLIAPVAEEEATVPNNKITVVGVGQVGMACAISILGKSI |
| DEHULM | MATLKDQLIYNLLKEEQTPQNKITVVGVGAVGMACAISILMKDL |
| DEMSLM | MATLKDQLIVILLKEEQAPQNKITVVGVGAVGMACAISILMKDLA |
| DEMSMM | MLSALARPAGAALRRSFSTSAQNNAKVAVLGASGGIGQPLSLLLK |
| XOMSDH | MTRTTVDELVFFVNGKKVVEKNADPETTLLVYLRRKLGLCGTK |
| JQ1004 | MSSPAQPRVPAPLADLKIQHTKIFINNEWHNSVSGKKFPVLNPAT |
| ... | |



## Motivation: Chemistry

THE AUSTRALIAN NATIONAL UNIVERSITY

| Name \ Struct. |  |  |  |  |  |  |  | ... |
|---|---|---|---|---|---|---|---|---|
| DEHUE7 | | | | | | | | |
| S41120 | | | | | | | | |
| A57407 | | | | | | | | |
| DEHULH | | | | | | | | |
| DEHULM | | | | | | | | |
| DEMSLM | | | | | | | | |
| DEMSMM | | | | | | | | |
| XOMSDH | | | | | | | | |
| JQ1004 | | | | | | | | |
| ... | | | | | | | | |



## Motivation: Chemistry

THE AUSTRALIAN NATIONAL UNIVERSITY

| Name \ Struct. |  |  |  |  |  |  |  | . |
|---|---|---|---|---|---|---|---|---|
| DEHUE7 | | X | | | | | | |
| S41120 | X | | | | | | | |
| A57407 | | | X | | | | | |
| DEHULH | | | | | | X | | |
| DEHULM | | | | X | | | | |
| DEMSLM | | X | | | | | | |
| DEMSMM | | | | | | | X | |
| XOMSDH | | | | | | X | | |
| JQ1004 | X | | | | | | | |
| ... | | | | | | | | |

# Motivation: Chemistry

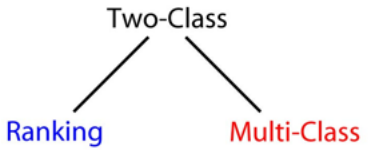
THE AUSTRALIAN NATIONAL UNIVERSITY

| Name \ Struct. |  |  |  |  |  |  |  | ... |
|---|---|---|---|---|---|---|---|---|
| DEHUE7 | | 1.0 | | | | 0.7 | 0.2 | |
| S41120 | 1.0 | | | | 0.6 | | | |
| A57407 | | | 1.0 | 0.3 | | | | |
| DEHULH | | 0.4 | | | | 1.0 | | |
| DEHULM | | | 0.7 | 1.0 | | | | |
| DEMSLM | | 1.0 | | | | 0.6 | | |
| DEMSMM | | 0.5 | | | | | 1.0 | |
| XOMSDH | | | | | | 1.0 | | |
| JQ1004 | 1.0 | | | | 0.4 | | | |
| ... | | | | | | | | |



# Polychotomies

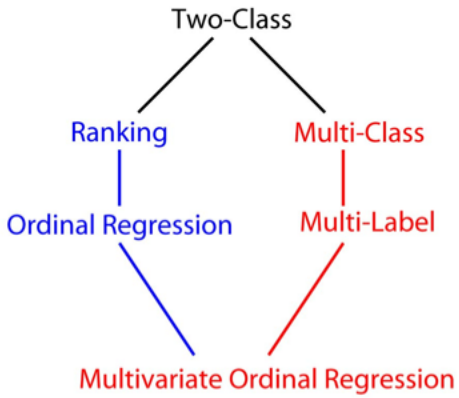
Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik






# Polychotomies

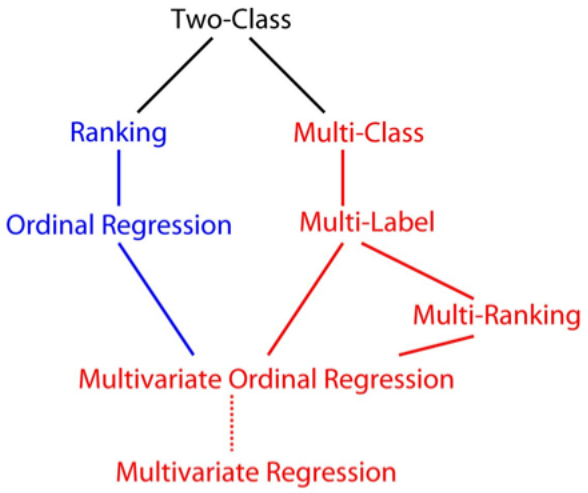
Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik





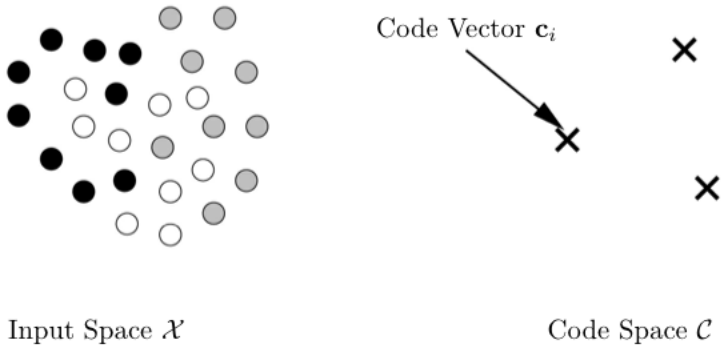

# Polychotomies

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik



## Embeddings and Codes

THE AUSTRALIAN NATIONAL UNIVERSITY

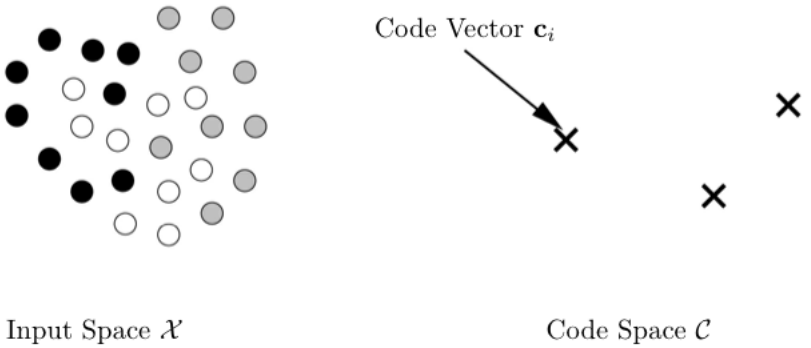



Input Space $\mathcal{X}$ Code Space $\mathcal{C}$



## Embeddings and Codes

THE AUSTRALIAN NATIONAL UNIVERSITY

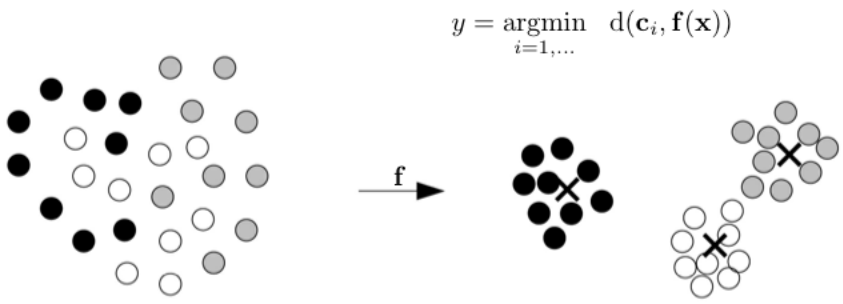



Input Space $\mathcal{X}$ Code Space $\mathcal{C}$

**Gunnar Rätsch**, Alex Smola & Sebastian Mika: Adapting Codes and Embeddings for Polychotomies

## Embeddings and Codes

THE AUSTRALIAN NATIONAL UNIVERSITY

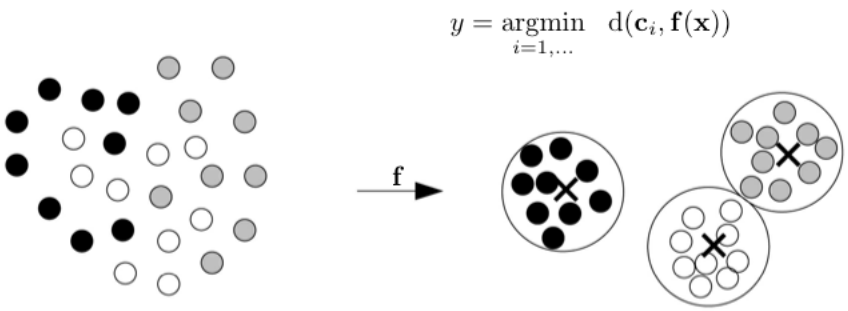



Input Space $\mathcal{X}$ Code Space $\mathcal{C}$

**Gunnar Rätsch**, Alex Smola & Sebastian Mika: Adapting Codes and Embeddings for Polychotomies 11

## Embeddings and Codes

THE AUSTRALIAN NATIONAL UNIVERSITY



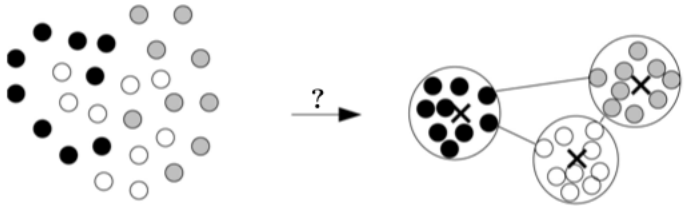

Input Space $\mathcal{X}$ Code Space $\mathcal{C}$

Questions
- Which metric $\mathrm{d}(\cdot,\cdot)$?
- Which embedding $\mathbf{f}$?
- Which code vectors $\mathbf{c}_1, \ldots$?

**Gunnar Rätsch**, Alex Smola & Sebastian Mika: Adapting Codes and Embeddings for Polychotomies

# Margin = Difference of Distances

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik




**Margin:** $\mathrm{d}(\mathbf{c}_i, \mathbf{f}(\mathbf{x})) - \mathrm{d}(\mathbf{c}_y, \mathbf{f}(\mathbf{x}))$ for all $i \neq y$

**Find embedding that maximizes the (soft-) margin**

**Lemma:** *Margin is convex in* $\mathbf{c}$ *and* $\mathbf{f}(\mathbf{x})$*, only if distance can be written as:*

$$\mathrm{d}(\mathbf{c}, \mathbf{f}(\mathbf{x})) = \mathrm{d}_1(\mathbf{c}) + \mathrm{d}_2(\mathbf{f}(\mathbf{x})) + \mathbf{c}^\top D \mathbf{f}(\mathbf{x}),$$

*where* $D$ *is a symmetric matrix.*

**Can only use linear or quadratic ($\sim$ Euclidean) distances**



# Finding the Embedding

THE AUSTRALIAN NATIONAL UNIVERSITY

- Why codes? Model class relationships!
- Which metric? No big choice! (cf. Lemma)
- Which embedding? Standard Convex optimization for fixed code.



# Finding the Embedding

THE AUSTRALIAN NATIONAL UNIVERSITY

- Why codes? Model class relationships!
- Which metric? No big choice! (cf. Lemma)
- Which embedding? Standard Convex optimization for fixed code.
- Which code? Standard Convex optimization for fixed embedding.
- **Algorithm:** Alternate between optimizing embedding and code
- Does it converge? Yes, to a **local** minimum.



# Finding the Embedding

THE AUSTRALIAN NATIONAL UNIVERSITY

- Why codes? Model class relationships!
- Which metric? No big choice! (cf. Lemma)
- Which embedding? Standard Convex optimization for fixed code.
- Which code? Standard Convex optimization for fixed embedding.
- **Algorithm:** Alternate between optimizing embedding and code
- Does it converge? Yes, to a **local** minimum.
- Which code to start with?
  - random code :-(
  - unsupervised learning
  - trivial code (one-against-the-rest) or use prior knowledge for init.

# Finding the Embedding

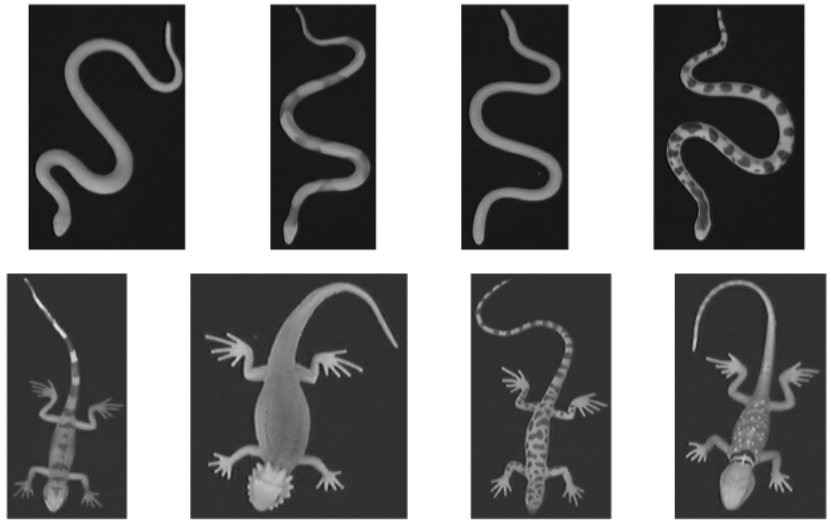
THE AUSTRALIAN NATIONAL UNIVERSITY

- Why codes? Model class relationships!
- Which metric? No big choice! (cf. Lemma)
- Which embedding? Standard Convex optimization for fixed code.
- Which code? Standard Convex optimization for fixed embedding.
- **Algorithm:** Alternate between optimizing embedding and code
- Does it converge? Yes, to a local minimum.
- Which code to start with?
  - random code :-(
  - unsupervised learning
  - trivial code (one-against-the-rest) or use prior knowledge for init.
- **Does it work?**



# Computer Vision Illustration (1)

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

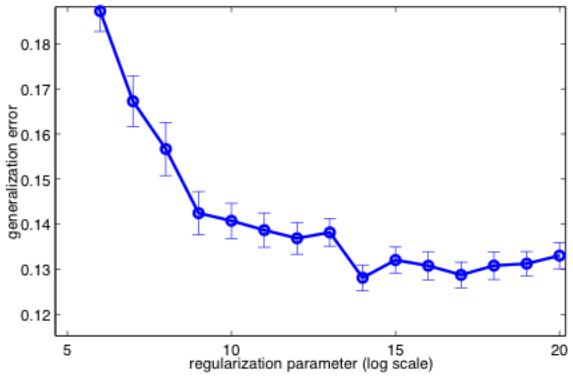


8 objects from the Nelson database (Nelson & Selinger, 1998)



# Computer Vision Illustration (2)

THE AUSTRALIAN NATIONAL UNIVERSITY




**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i - y_i)^2}{\sum_i |x_i + y_i|}\right)$

Some experimental details:

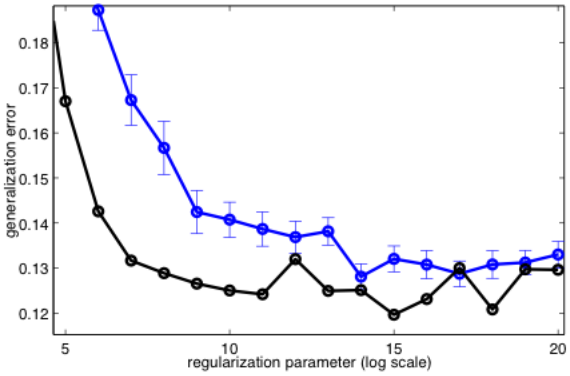
- 8 objects (4 snakes/lizards)
- 100 training examples, about 12 views per object
- 424 test examples
- averaged over 100 runs
- Multidim. receptive Field Hist. with Gaussian derivatives filters along $x/y$-axis
- shown is *test object identification error vs. regul. parameter* (C=1-1000 in log)

**Joint work with Barbara Caputo.**



# Computer Vision Illustration (2)

THE AUSTRALIAN NATIONAL UNIVERSI




**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i - y_i)^2}{\sum_i |x_i + y_i|}\right)$

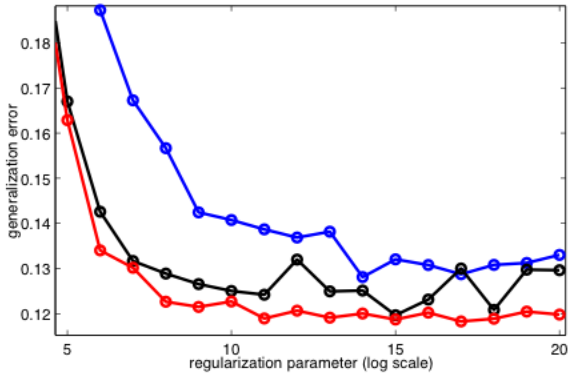
$\ell_2$**-Embedding &** $\ell_1$**-Code** in 8 dimensions

Some experimental details:

- 8 objects (4 snakes/lizards
- 100 training example about 12 views per object
- 424 test examples
- averaged over 100 runs
- Multidim. receptive Fiel Hist. with Gaussian deriva tives filters along $x/y$-axis
- shown is *test object ident fication error vs. regul. p rameter* (C=1-1000 in log

# Computer Vision Illustration (2)

THE AUSTRALIAN NATIONAL UNIVERSITY




Some experimental details:

- 8 objects (4 snakes/lizards)
- 100 training examples, about 12 views per object
- 424 test examples
- averaged over 100 runs
- Multidim. receptive Field Hist. with Gaussian derivatives filters along $x/y$-axis
- shown is *test object identification error vs. regul. parameter* (C=1-1000 in log)

**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i-y_i)^2}{\sum_i |x_i+y_i|}\right)$

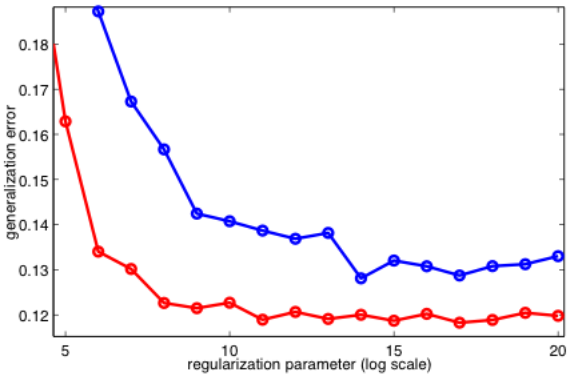
**$\ell_2$-Embedding & $\ell_1$-Code** in 8 dimensions

**Using hierarchy info.** by separate losses



# Computer Vision Illustration (2)

THE AUSTRALIAN NATIONAL UNIVERSI




Some experimental details:

- 8 objects (4 snakes/lizards)
- 100 training example about 12 views per object
- 424 test examples
- averaged over 100 runs
- Multidim. receptive Fiel Hist. with Gaussian deriva tives filters along $x/y$-axis
- shown is *test object ident fication error vs. regul. p rameter* (C=1-1000 in log

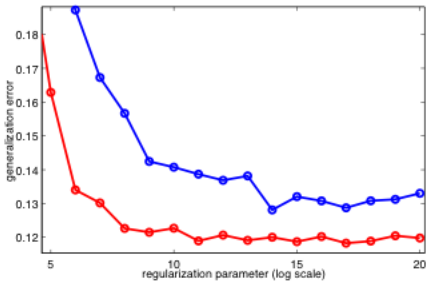
**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i-y_i)^2}{\sum_i |x_i+y_i|}\right)$

**Using hierarchy info.** by separate losses



# Computer Vision Illustration (2)

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik




**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i-y_i)^2}{\sum_i |x_i+y_i|}\right)$

**Using hierarchy info.** by separate losses

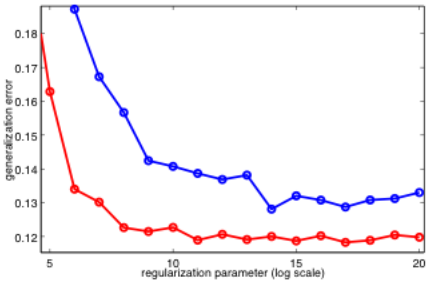
Some experimental details:

- 8 objects (4 snakes, 4 lizards)
- 100 training examples, ∼12 views per object
- 424 test examples, averaged over 100 runs



# Computer Vision Illustration (2)

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik



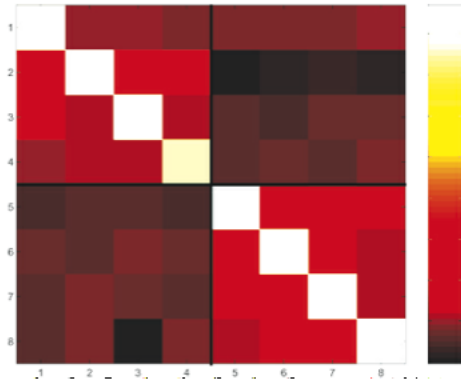

**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i-y_i)^2}{\sum_i |x_i+y_i|}\right)$

**Using hierarchy info.** by separate losses

Some experimental details:

- 8 objects (4 snakes, 4 lizards)
- 100 training examples, ∼12 views per object
- 424 test examples, averaged over 100 runs

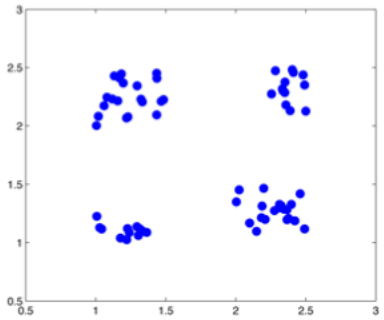
Resulting Code Vectors:



- positive interactions within category
- negative interactions between cat.

⇝ significant improvements

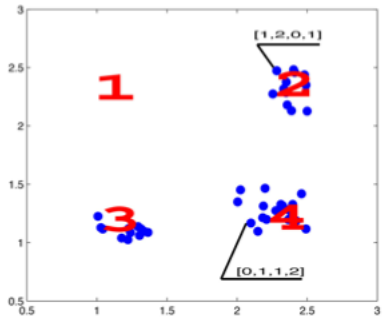
## Beyond Multi-Class: Missing Classes



- four classes in two dimensions
- training set only contains examples from three classes
- include additional information of relation between classes
- find embedding & code for all four classes using separate losses
- test on the "missing class"



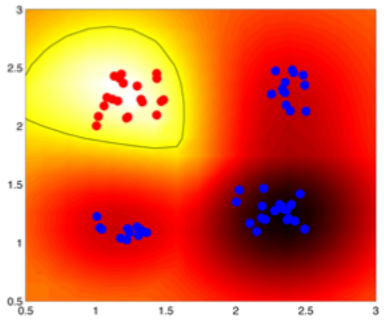
## Beyond Multi-Class: Missing Classes




- four classes in two dimensions
- training set only contains examples from three classes
- include additional information of relation between classes
- find embedding & code for all four classes using separate losses
- test on the "missing class"



## Beyond Multi-Class: Missing Classes



- four classes in two dimensions
- training set only contains examples from three classes
- include additional information of relation between classes
- find embedding & code for all four classes using separate losses
- test on the "missing class"

**Exploiting the structure of the "label space" makes learning of missing classes possible.**

This can be exploited for solving the motivating chemistry problem

(future work)



## Conclusion

- Important to extract information between classes
- Find Embedding and Code at the same time (code acts as "hidden layer")
- Leads to two convex optimization problems ...
  ... but overall problem is non-convex $\rightsquigarrow$ local minima
- Does there exist a better approach accomplishing the same?
- Leads to improvements of object identification and classification
- Can find boundaries of "missing classes"

# Conclusion

THE AUSTRALIAN NATIONAL UNIVERSITY

- Important to extract information between classes
- Find Embedding and Code at the same time (code acts as "hidden layer")
- Leads to two convex optimization problems ...

  ... but overall problem is non-convex $\rightsquigarrow$ local minima
- Does there exist a better approach accomplishing the same?
- Leads to improvements of object identification and classification
- Can find boundaries of "missing classes"

**Thanks to You (attention), Barbara Caputo (Computer Vision), Andrew Torda (Chemistry) and Bob Williamson (Machine Learning)**



# ANU Commercials

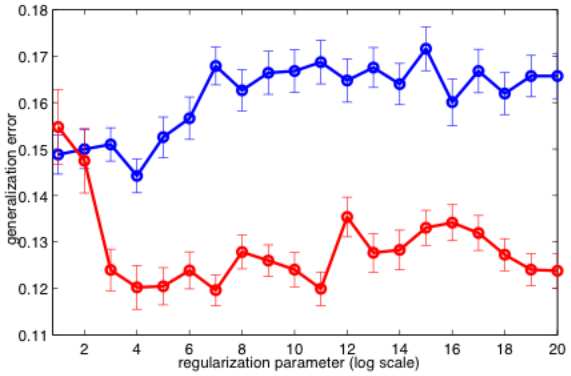
THE AUSTRALIAN NATIONAL UNIVERSI

- Machine Learning Summer School **http://mlss.cc**
  - **Canberra**, February 2-14, 2003
  - **Tübingen**, August, 2003
- vacant postdoc positions at ANU (projects with Bob Williamson, Alex Smola & Gunnar Rätsch)
  - Online Learning & Information Theory
  - Computer Vision
  - Network Intrusion Detection
- vacant positions at all levels at new **ICT** Center
- visit:
  - **http://mlss.cc**
  - **http://boosting.org**
  - **http://kernel-machines.org**



# Computer Vision Illustration (3)

THE AUSTRALIAN NATIONAL UNIVERSITY




**SVM** with $k(\mathbf{x},\mathbf{y}) = \exp\left(-10\frac{\sum_i (x_i - y_i)^2}{\sum_i |x_i + y_i|}\right)$

**Using hierarchy info.** by separate losses

Some experimental details:

- 8 objects (4 snakes/lizards)
- 100 training examples, about 12 views per object
- 424 test examples with **different snakes & lizards**
- averaged over 100 runs
- Multidim. receptive Field Hist. with Gaussian derivatives filters along $x/y$-axis
- shown is *test object category error vs. regul. parameter* (C=1-1000 in log)



# Relations to Previous Work

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

- One-Against-the-Rest method and SVM multi-class (Weston et al., 1998) uses trivial constant code: $\mathbf{c}_i = \mathbf{e}_i$
- Adapting codes approach of Cramer & Singer (2000) adapts only code, bu not embedding
- Boosting with codes (Schapire, 1997) adapts essentially only embedding, but not the code (cf. column generation technique)
- ECOC approaches (Dietterich & Bakiri, 1995; Utschik, 1998; Guruswami Sahai, 1999; Allwein et al., 2000) have to solve hard combinatorial optimization problems for finding code. We have a different motivation!
- Closely related to idea having a common Hidden-Layer in Multi-class Neural Networks
- General formulation is very similar to multivariate Ordinal Regression (Herbrich et al., 2000)

# Optimization Problems

THE AUSTRALIAN NATIONAL UNIVERSITY

Find codes that, given a fixed embedding function $\mathbf{f}$, maximize the margin:

$$\begin{aligned} &\underset{\mathbf{c}_i, \boldsymbol{\epsilon} \in \mathbf{R}^N_+}{\text{minimize}} && \sum_{i=1}^{C} \|\mathbf{c}_i\|_p^p + \lambda \sum_{n=1}^{N} \epsilon_n \\ &\text{with} && d(\mathbf{c}_i, \mathbf{f}(\mathbf{x}_n)) - d(\mathbf{c}_{y_n}, \mathbf{f}(\mathbf{x}_n)) \geq 1 - \epsilon_n \text{ for all } n \text{ and } i \neq y_n. \end{aligned}$$

Find embedding that, given a fixed code $\mathbf{c}$, maximizes the margin:

$$\begin{aligned} &\underset{\boldsymbol{\alpha} \in \mathbf{R}^J, \boldsymbol{\epsilon} \in \mathbf{R}^N_+}{\text{minimize}} && \|\boldsymbol{\alpha}\|_q^q + \lambda \sum_{n=1}^{N} \varepsilon_n \\ &\text{with} && d(\mathbf{c}_i, \mathbf{f}_{\boldsymbol{\alpha}}(\mathbf{x}_n)) - d(\mathbf{c}_{y_n}, \mathbf{f}_{\boldsymbol{\alpha}}(\mathbf{x}_n)) \geq 1 - \epsilon_n \text{ for all } n \text{ and } i \neq y_n. \end{aligned}$$

**Lemma:** One $\lambda$ is sufficient, if $d(\mathbf{x}, \mathbf{y}) = \mathbf{d}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{y}}{\|y\|}\right) \|\mathbf{x}\| \|\mathbf{y}\|$

**For instance: $\mathbf{d}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y})$**



# Problems & Ideas for solutions

THE AUSTRALIAN NATIONAL UNIVERSI

- Combined optimization problem is non-convex and has local minima
  - $\rightsquigarrow$ needs good initialization: start with the trivial code
  - $\rightsquigarrow$ use all available information about the structure of labels
- If $p = 2$ and $q = 2$: solution can be rotated
  - $\rightsquigarrow$ harder to optimize: better chose $p = 1$, $q = 2$ or $p = 2$, $q = 1$
  - $\Rightarrow$ Convergence in a few (say 5) steps
- It might be difficult to choose $C$
  - $\rightsquigarrow$ $\nu$ trick may simplify model selection
- Algorithm is slow when using general purpose optimization techniques
  - $\rightsquigarrow$ implementation of SMO-type algorithm is needed