Kernel Dependency Estimation

J. Weston, O. Chapelle, A. Elisseeff, B. Schoelkopf and V. Vapnik, NIPS, 2002.

Presented by Alex Ainslie | Advanced Machine Learning | CS 6784 | February 18, 2010

Motivation

- Learning problem
- Find a dependency between a general class of objects and another
- Relies on kernel functions because it uses similarity measures in both input and output spaces
- Encodes complex costs and outputs

Learning

- Inputs $\mathbf{x} \in \mathcal{X}$
- Outputs $\mathbf{y} \in \mathcal{Y}$
- Learn the function $f(\mathbf{x}, \alpha^*)$
- Minimum value of risk function

$$R(\alpha) = \int_{\mathcal{X} \times \mathcal{Y}} L(\mathbf{y}, f(\mathbf{x}, \alpha)) dP(\mathbf{x}, \mathbf{y})$$

- Requires a priori knowledge of similarity measure (the loss function for outputs)

Complex Cost

- This loss function can be simple:
  - pattern recognition (zero-one loss)
  - regression (squared loss)
- or more complicated:
  - mapping to images
  - mixture of drugs

Kernel Functions

- A kernel k is:
  - a symmetric function
  - an inner product in some Hilbert space $\mathcal{F}$ (same class:high, different class:low)

    $\Phi_k : \mathcal{X} \to \mathcal{F}$ such that $k(\mathbf{x}, \mathbf{x}') = (\Phi_k(\mathbf{x}) \cdot \Phi_k(\mathbf{x}'))$
- EX: $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^p$

Kernel Examples

- M-class pattern recognition

  $\ell_{pat}(\mathbf{y}, \mathbf{y}') = \frac{1}{2}[\mathbf{y} = \mathbf{y}']$

  $\Phi_\ell(\mathbf{y}) = (0, 0, \ldots, \frac{\sqrt{2}}{2}, \ldots, 0)$ where the $\mathbf{y}^{th}$ coordinate is nonzero
- Regression estimation

  $\ell_{reg}(\mathbf{y}, \mathbf{y}') = (\mathbf{y} \cdot \mathbf{y}')$
- Strings

  exponential decay

  $$\ell(\mathbf{s}, \mathbf{t}) = \sum_{\mathbf{u} \in \Sigma^r} \psi_{\mathbf{u}}(\mathbf{s}) \cdot \psi_{\mathbf{u}}(\mathbf{t}) = \sum_{\mathbf{u} \in \Sigma^r} \sum_{\mathbf{i}:\mathbf{u}=\mathbf{s}[\mathbf{i}]} \lambda^{l(\mathbf{i})} \sum_{\mathbf{j}:\mathbf{u}=\mathbf{t}[\mathbf{j}]} \lambda^{l(\mathbf{j})}$$

  ordered subsequences of length r

Algorithm (KDE)

- Minimize the risk function using the feature space F induced by the kernel k and the loss function measured in the space L induced by the kernel l
- Decomposition of outputs
- Learning the map
- Solving the pre-image

Decompose

- Construct kernel matrix L on training data
- Perform kernel PCA

$$\mathbf{L}' = (\mathbf{I} - \tfrac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top)\mathbf{L}(\mathbf{I} - \tfrac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top)$$

n^{th} principal component $\mathbf{v}^n = \sum_{i=1}^m \alpha_i^n \Phi_\ell(\mathbf{y}_i)$

$$(\mathbf{v}^n \cdot \Phi_\ell(\mathbf{y})) = \sum_{i=1}^m \alpha_i^n \ell(\mathbf{y}_i, \mathbf{y}).$$

Map

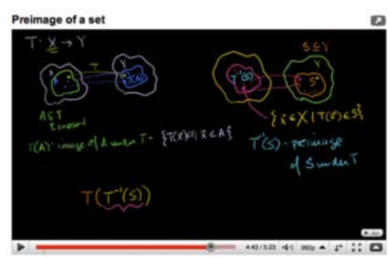
- Using the p principal components
- Perform kernel ridge regression
- Estimator:

$$f_n(\mathbf{x}) = \sum_{i=1}^m \beta_i^n k(\mathbf{x}_i, \mathbf{x}), \qquad \beta^n = (\mathbf{K} + \gamma \mathbf{I})^{-1} \hat{\mathbf{y}}^n$$

Pre-Image

- During testing to find estimate for y for a given x, we need the pre-image $\Phi_\ell(\mathbf{y})$

$$\mathbf{y}(\mathbf{x}) = \mathrm{argmin}_{\mathbf{y}\in\mathcal{Y}} \| ((\mathbf{v}^1 \cdot \Phi_\ell(\mathbf{y})), \ldots, (\mathbf{v}^p \cdot \Phi_\ell(\mathbf{y}))) - (f_1(\mathbf{x}), \ldots, f_p(\mathbf{x})) \|$$



Experiment: Images

- USPS handwritten 16 pixel digit database
- Classification

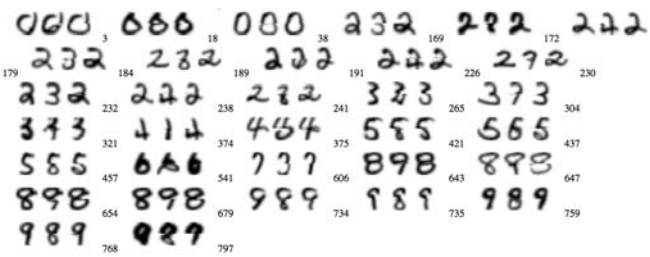
	KDE	1-vs-rest SVM	k-NN
classification loss	0.0798 ± 0.0067	0.0847 ± 0.0064	0.1250 ± 0.0075

Experiment: Images

- Image Reconstruction
- Estimate using first 8 rows

	KDE	k-NN	Hopfield net
RBF loss	0.8384 ± 0.0077	0.8960 ± 0.0052	1.2190 ± 0.0072

KDE Mistakes



Original, KDE, KNN

KNN Mistakes

Original, KDE, KNN

Toy Problem: Strings

- Predict output string from input string
- Almost classification with three classes

input string		output string
ccddddddddd	→	aabc
dccccdddcd	→	abc
adddccccccccc	→	bb
bbcdcdadbad	→	aebad
cdaaccadcbccdd	→	abad

	KDE	k-NN
string loss	0.676 ± 0.030	0.985 ± 0.029
classification loss	0.125 ± 0.012	0.205 ± 0.026

Toy Problem: Strings

- Alphabet (a,b,c,d)
- Input: random length (10 -15)
- Three classes of strings
  - transitions equally likely : abad
  - 0.7 repeat, 0.1 other : dbbd
  - 0.7 repeat, 0.1 other, only c,d : aabc
- Outputs corrupted with noise

String Subsequence Kernel

- Lodhi, H. S., C.; Shawe-Taylor, J.; Cristianini, N.; Watkins, C. (2002). Text classification using string kernels. 419-444.
- Compare text documents by substrings (not necessarily contiguous) $\lambda \in (0,1)$
  - **c-a-r** is in **car**d and **c**ust**ar**d
- Used for both inputs and outputs

Toy Problem: Strings

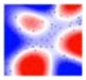
- cat, car, bat, bar $\lambda = 0.01$
- ca, ct, at, ca, cr, ar, ba, bt, at, ba, br, ar

	c-a	c-t	a-t	b-a	b-t	c-r	a-r	b-r
ϕ(cat)	λ^2	λ^3	λ^2	0	0	0	0	0
ϕ(car)	λ^2	0	0	0	0	λ^3	λ^2	0
ϕ(bat)	0	0	λ^2	λ^2	λ^3	0	0	0
ϕ(bar)	0	0	0	λ^2	0	0	λ^2	λ^3

$K(\text{car},\text{cat}) = \lambda^4$

$\lambda^2 \; \lambda^3 \; \lambda^2 \; 0 \; 0 \; 0 \; 0 \; 0 \; \bullet \; \lambda^2 \; 0 \; 0 \; 0 \; 0 \; \lambda^3 \; \lambda^2 \; 0$

Toy Problem: Strings



$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')/(\sqrt{k(\mathbf{x}, \mathbf{x})}\sqrt{k(\mathbf{x}', \mathbf{x}')})$$
$$\exp(-(k(\mathbf{x}, \mathbf{x}) + k(\mathbf{x}', \mathbf{x}') - 2k(\mathbf{x}, \mathbf{x}'))/2\sigma^2)$$

- Find this distance (similarity) measure for each pair in inputs and outputs
- Then using kernel ridge regression to finding a mapping
- Pre-image: closest training example output to the given solution

Image: Joachims, SIGIR03 Tutorial Slides

Kernel Dependency Estimation

J. Weston, O. Chapelle, A. Elisseeff, B. Schoelkopf and V. Vapnik, NIPS, 2002.

Presented by Alex Ainslie | Advanced Machine Learning | CS 6784 | February 18, 2010