Pattern Recognition (PR)

Winter Term 2015/16

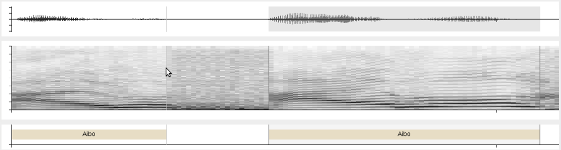
Elmar Nöth
Computer Science Dept. 5
(Pattern Recognition)

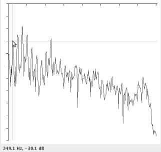
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Kernels for Feature Sequences

Example: string kernels

- In speech recognition we do not have feature vectors but sequences of feature vectors.
- In order to use kernel methods we need a kernel for time series.





Kernels for Feature Sequences (cont.)

Example: string kernels (cont.)

- Feature vectors are considered in $\mathbb{R}^d = \mathcal{X}$.
- Sequences of feature vectors are elements of $\mathcal{X}^*$.
- Problem: How to define a kernel over the sequence space $\mathcal{X}^*$?

Implications:

- PCA on feature sequences – COOL!
- SVM for feature sequences – EVEN COOLER!

Kernels for Feature Sequences (cont.)

Example: string kernels (cont.)

Comparison of sequences via *dynamic time warping* (DTW):

Given the feature sequences ($p, q \in \{1, 2, \ldots\}$):

$$
\begin{aligned}
\langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p \rangle &\in \mathcal{X}^* \\
\langle \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_q \rangle &\in \mathcal{X}^*
\end{aligned}
$$

Kernels for Feature Sequences (cont.)

Example: string kernels (cont.)

- Distance is computed by DTW:

$$D(\langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p \rangle, \langle \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_q \rangle) = \frac{1}{p} \sum_{k=1}^{p} \| \boldsymbol{x}_{v(k)} - \boldsymbol{y}_{w(k)} \|_2$$

where v, w define the mapping of indices to indices.

Kernels for Feature Sequences (cont.)

Example: string kernels (cont.)

- Distance is computed by DTW:

$$D(\langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p \rangle, \langle \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_q \rangle) = \frac{1}{p} \sum_{k=1}^{p} \| \boldsymbol{x}_{v(k)} - \boldsymbol{y}_{w(k)} \|_2$$

  where v, w define the mapping of indices to indices.

- The DTW kernel can be defined as:

$$k(\boldsymbol{x}, \boldsymbol{y}) = e^{-D(\langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p \rangle, \langle \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_q \rangle)}$$

Fisher Kernels

Now we design kernels building on probability density functions $p(\boldsymbol{x}; \boldsymbol{\theta})$.

- Fisher score:

$$\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{x}; \boldsymbol{\theta})$$

Fisher Kernels

Now we design kernels building on probability density functions $p(\boldsymbol{x};\boldsymbol{\theta})$.

- Fisher score:

$$\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta})$$

- Fisher information matrix:

$$\boldsymbol{I}(\boldsymbol{x}) = E_{\boldsymbol{x}}[\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x})\boldsymbol{J}_{\boldsymbol{\theta}}^T(\boldsymbol{x})]$$

Fisher Kernels

Now we design kernels building on probability density functions $p(\boldsymbol{x};\boldsymbol{\theta})$.

- Fisher score:

$$\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\frac{\partial}{\partial\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta})$$

- Fisher information matrix:

$$\boldsymbol{I}(\boldsymbol{x}) = E_{\boldsymbol{x}}[\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x})\boldsymbol{J}_{\boldsymbol{\theta}}^T(\boldsymbol{x})]$$

Note:

The Fisher information matrix is the curvature of the Kullback-Leibler divergence.

Fisher Kernels (cont.)

The Fisher kernel can be defined in two different ways:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{J}_{\boldsymbol{\theta}}^T(\boldsymbol{x})\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x}')$$

or

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{J}_{\boldsymbol{\theta}}^T(\boldsymbol{x})\boldsymbol{I}^{-1}(\boldsymbol{x})\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{x}')$$

Fisher Kernels (cont.)

Application: learning from partially labeled data

Fisher Kernels (cont.)

Application: learning from partially labeled data

- Some classification approaches require huge collections of data (e. g. for text or speech recognition).

Fisher Kernels (cont.)

Application: learning from partially labeled data

- Some classification approaches require huge collections of data (e. g. for text or speech recognition).
- Labeling of the data can be time-consuming and costly.

Fisher Kernels (cont.)

Application: learning from partially labeled data

- Some classification approaches require huge collections of data (e. g. for text or speech recognition).
- Labeling of the data can be time-consuming and costly.
- If the data can be modeled with a small number of well separated components (with each component corresponding to a distinct category), little labeled data would suffice to assign a proper label to each of them.

Fisher Kernels (cont.)

Application: learning from partially labeled data

- Some classification approaches require huge collections of data (e. g. for text or speech recognition).
- Labeling of the data can be time-consuming and costly.
- If the data can be modeled with a small number of well separated components (with each component corresponding to a distinct category), little labeled data would suffice to assign a proper label to each of them.
- A machine learning approach that makes use of only partially labeled data usually achieves much better classification performance than using only the labeled data alone.

Fisher Kernels (cont.)

Application: learning from partially labeled data

- Some classification approaches require huge collections of data (e. g. for text or speech recognition).
- Labeling of the data can be time-consuming and costly.
- If the data can be modeled with a small number of well separated components (with each component corresponding to a distinct category), little labeled data would suffice to assign a proper label to each of them.
- A machine learning approach that makes use of only partially labeled data usually achieves much better classification performance than using only the labeled data alone.
- Fisher kernels describe a generative model that can be used in a discriminative approach (e. g. SVM).