

String Kernels



Huma Lodhi
University of Sheffield

Plan

- ❖ Motivation
- ❖ Kernels for Sequences
- ❖ Word Kernel (WK)
- ❖ n-gram Kernel (NGK)
- ❖ String Subsequence Kernels (SSK)
- ❖ Sequence Kernels in Practice
- ❖ Applications of SSK in Human Motion Analysis and Identification

Kernel Methods: Main Idea

Map the data into feature space via mapping ϕ .
The mapping may be assessed via the kernel function.

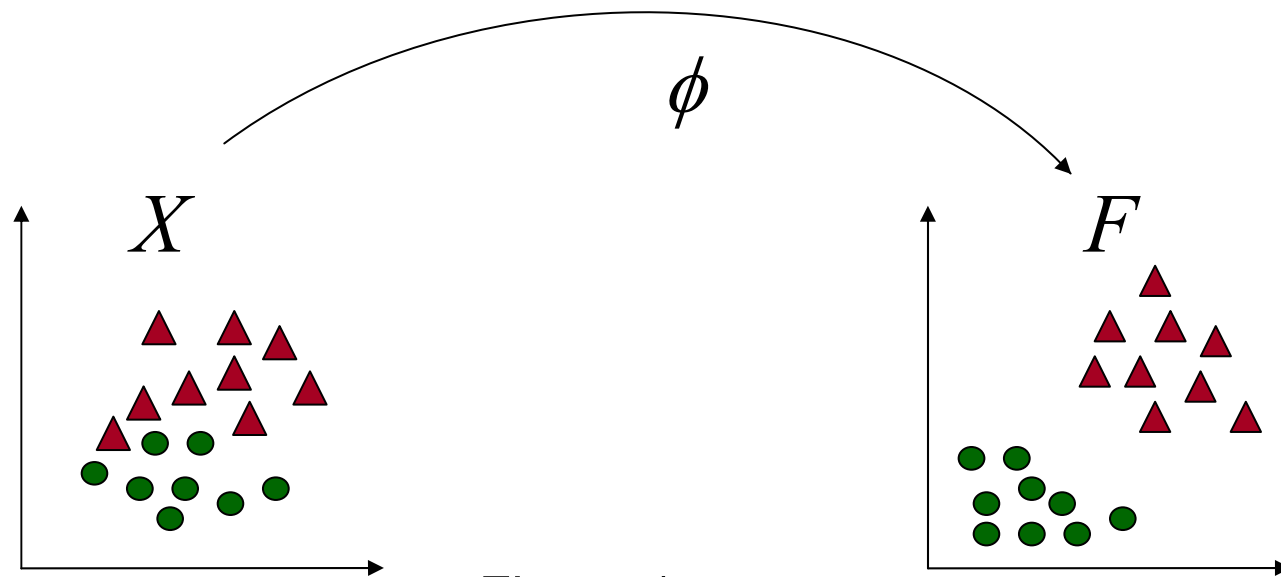


Figure 1

Construct a linear function in feature space

Kernel Methods: Main Idea

Kernel function, a similarity measure, returns inner product between mapped instances

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

Example: Kernel Function Between a Pair of Instances

$$x_i = (x_{i_1}, x_{i_2}) \qquad x_j = (x_{j_1}, x_{j_2})$$

The mapping is

$$\phi(x_i) = (x_{i_1}^2, x_{i_2}^2, \sqrt{2}x_{i_1}x_{i_2}) \qquad \phi(x_j) = (x_{j_1}^2, x_{j_2}^2, \sqrt{2}x_{j_1}x_{j_2})$$

The kernel is

$$\langle \phi(x_i), \phi(x_j) \rangle = (x_{i_1}^2 x_{j_1}^2, x_{i_2}^2 x_{j_2}^2, 2x_{i_1}x_{i_2}x_{j_1}x_{j_2}) = \langle x_i, x_j \rangle^2$$

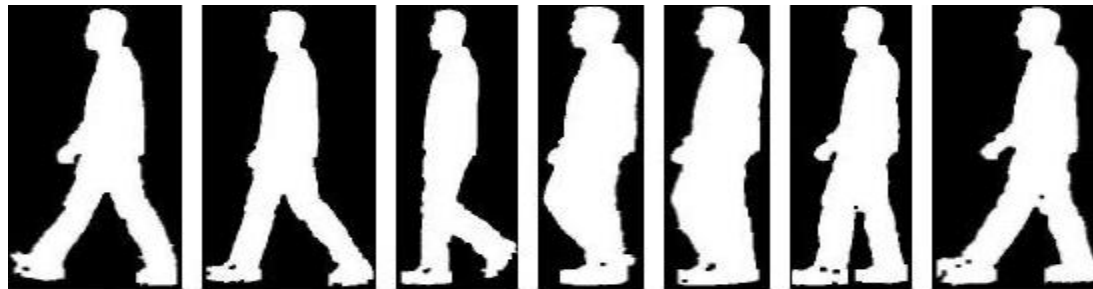
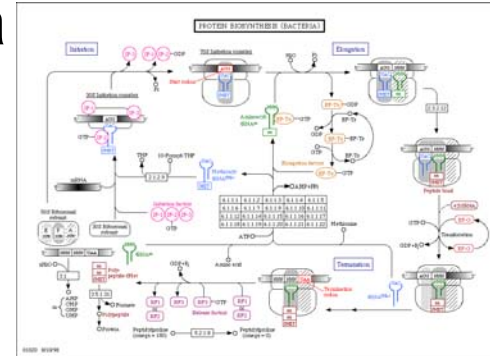
Motivation

This is a text document. It contains only plain text (which can be Unicode). You can edit raw text in Ultra Recall but the formatting capabilities are disabled for this type of data.

Text document

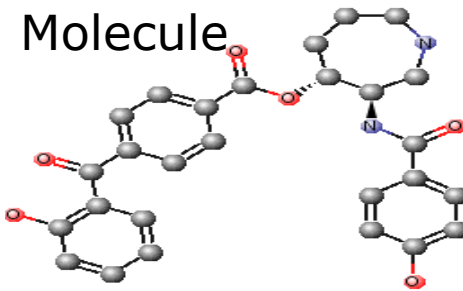
Real World Data

Metabolic pathway



Human motion sequence

Molecule



Bio sequence

MSETENQALTFAKRLKADTTAVHDSVDNLMVSVQPFVSKENYIKFLKLQSVFHKAVDHIY
MSETENQALTFAKRLKADTTAVHDSVDNLMVSVQPFVSKENYIKFLKLQSVFHKAVDHIY
MSETENQALTFAKRLKADTTAVHDSVDNLMVSVQPFVSKENYIKFLKLQSVFHKAVDHIY

KDAELNKAIPELEYMARYDAVTQDLADLGDKPYEYKPLPHETGNKAIGWLYCAEGSNLG
KDAELNKAIPELEYMARYDAVTQDL DLG++PY++ K LP+E GNKAIGWLYCAEGSNLG
KDAELNKAIPELEYMARYDAVTQDLKDLGEEPYKFDKELPYEAGNKAIGWLYCAEGSNLG

Motivation

Problem:

Real world data is non vectorial,
How to apply machine learning
algorithms to discrete instances?

Solution:

Develop and apply kernels for
strings, graphs and trees

Kernels for Sequences

- Word Kernels (WK)
- n-grams Kernels (NGK)
- String Subsequence Kernels (SSK)

Kernels for Sequences

Words

Sequence of characters followed by punctuation or space

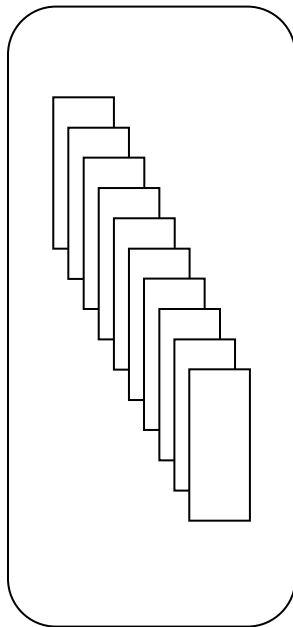
n-grams

Sequence of n consecutive characters

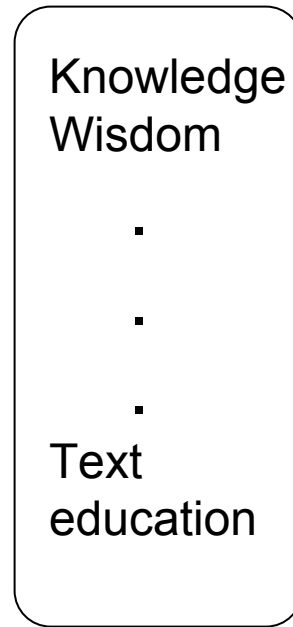
Example: support vector

3-grams = sup upp ppo por ort rt_ t_v _ve
ect cto tor

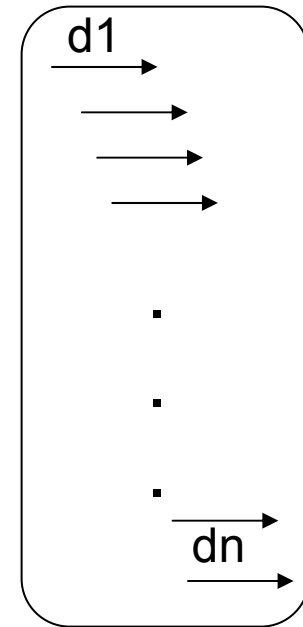
Word Kernels



Documents



Word (term)
extraction



Document
vectors

Text representation: Vector Space Model

Word Kernels

- ❖ Strings (documents) are mapped into very high dimensional feature vectors, where dimensionality of the feature space is equal to the number of words in a corpus
- ❖ Each entry of the vector represents the occurrence or non-occurrence of a **word** by a number

Kernel

- Inner product between mapped sequences give a sum over all **common (weighted) words**

n-grams Kernels

- ❖ n-grams maps strings (documents) into high dimensional feature vectors
- ❖ Each entry of the vector represents occurrence or non-occurrence of a **contiguous subsequence** by a number

Kernel

- Inner product between mapped sequences give a sum over all **common (weighted) contiguous subsequences**

String Subsequence Kernels

Basic Idea

- ❖ Computes the similarity between two strings without explicitly extracting the features
- ❖ The more subsequences two strings have in common, the more similar they are considered

Characteristics

- ❖ Feature space is generated by all the substrings of bounded length
- ❖ Substrings can be non-contiguous, gaps are taken into account
- ❖ Substrings are weighted according to the degree of contiguity in a string by a decay factor λ

String Kernels: Example

Consider strings

- ❖ sectionalization
- ❖ segmentation

and a substring

- ❖ “s-e-t”

Length l of

- ❖ “s-e-t” in **sectionalization** = 4
- ❖ “s-e-t” in **segmentation** = 7

String Kernels: Example

| | f-o | f-g | o-g | f-b | o-b |
|--------------------|-------------|-------------|-------------|-------------|-------------|
| $\phi(\text{fog})$ | λ^2 | λ^3 | λ^2 | 0 | 0 |
| $\phi(\text{fob})$ | λ^2 | 0 | 0 | λ^3 | λ^2 |

$$k(\text{fog}, \text{fog}) = 2\lambda^4 + \lambda^6$$

$$k(\text{fob}, \text{fob}) = 2\lambda^4 + \lambda^6$$

$$k(\text{fog}, \text{fob}) = \frac{k(\text{fog}, \text{fob})}{\sqrt{k(\text{fog}, \text{fog})k(\text{fob}, \text{fob})}} = \frac{\lambda^4}{2\lambda^4 + \lambda^6} = \frac{1}{2 + \lambda^2}$$

String Kernels

Alphabet

Let Σ be finite alphabet

String

A string s is a finite sequence of characters from alphabet with length $|s|$

Subsequence

Let $i = (i_1, \dots, i_n)$ be a set of indices (sorted in ascending order) in s and a subsequence is given by $u = s[i]$, length of subsequence is $l(i) = i_n - i_1 + 1$

Example

$s = \text{segmentation}$, $i = [1, 2, 7]$

String Kernels

The inner product between two mapped strings is a sum over all the common weighted subsequence

$$\begin{aligned}k_n(s, t) &= \sum_{u \in \Sigma^n} \langle \phi_u(s), \phi_u(t) \rangle = \sum_{u \in \Sigma^n} \sum_{i:u=s[i]} \lambda^{l(i)} \sum_{j:u=t[j]} \lambda^{l(j)} \\ &= \sum_{u \in \Sigma^n} \sum_{i:u=s[i]} \sum_{j:u=t[j]} \lambda^{l(i)+l(j)}\end{aligned}$$

where

$$\phi_u(s) = \sum_{i:u=s[i]} \lambda^{l(i)}$$

String Kernels

$$K'_0(s, t) = 1, \text{ for all } s, t$$

$$K''_i(s, t) = 0, \text{ if } \min(|s|, |t|) < i$$

$$K'_i(s, t) = 0, \text{ if } \min(|s|, |t|) < i$$

$$K_i(s, t) = 0, \text{ if } \min(|s|, |t|) < i$$

$$K''_i(sx, t) = \sum_{j:t_j=x} K'_{i-1}(s, t[1:j-1])\lambda^{|t|-j+2}$$

$$K''_i(sx, tx) = \lambda (K''_i(sx, t) + \lambda K'_{i-1}(s, t))$$

$$K''_i(sx, tz) = \lambda K''_i(sx, t), \text{ provided } x \neq z$$

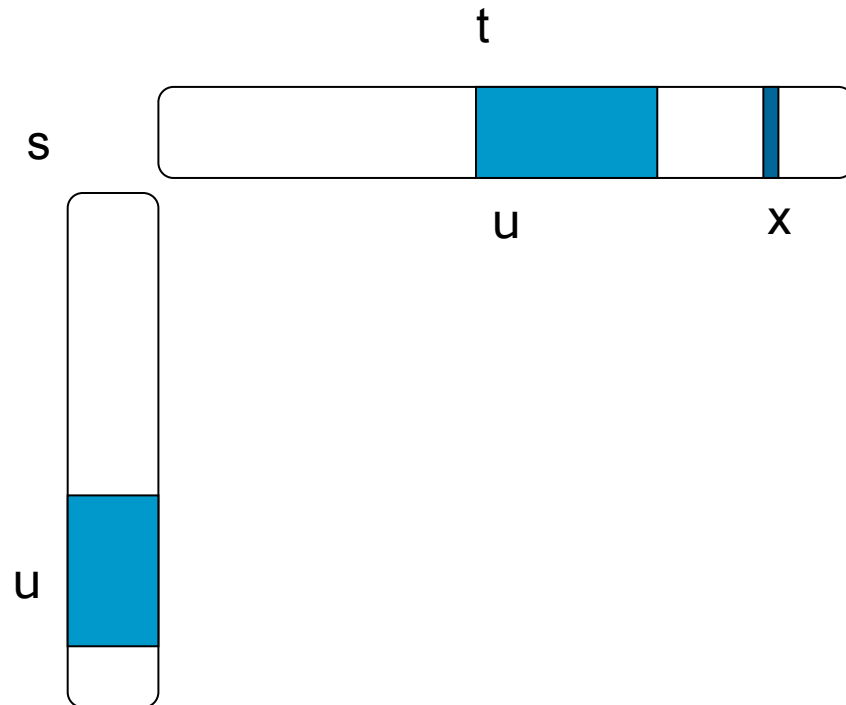
$$K'_i(sx, t) = \lambda K'_i(s, t) + K''_i(sx, t)$$

$$K_n(sx, t) = K_n(s, t) + \sum_{j:t_j=x} K'_{n-1}(s, t[1:j-1])\lambda^2$$

String Kernels

$$\begin{aligned}\hat{K}(s, t) &= \langle \hat{\phi}(s) \cdot \hat{\phi}(t) \rangle = \left\langle \frac{\phi(s)}{\|\phi(s)\|} \cdot \frac{\phi(t)}{\|\phi(t)\|} \right\rangle \\ &= \frac{1}{\|\phi(s)\| \|\phi(t)\|} \langle \phi(s) \cdot \phi(t) \rangle = \frac{K(s, t)}{\sqrt{K(s, s)K(t, t)}}\end{aligned}$$

String Kernels



Dataset

| Category | Train+ | Test+ |
|----------|--------|-------|
| earn | 152 | 40 |
| acq | 114 | 25 |
| crude | 76 | 15 |
| corn | 38 | 10 |

Number of relevant documents for the categories in dataset

Dataset

| Category | Maximum | Minimum | Average |
|----------|---------|---------|---------|
| earn | 3422 | 40 | 527 |
| acq | 3406 | 41 | 615 |
| crude | 3479 | 57 | 575 |
| corn | 3616 | 31 | 638 |

Evaluation Measures

Precision = relevant documents categorized
relevant/total documents categorized
relevant

Recall = relevant documents categorized
relevant/total relevant documents

F1 = $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$

Results

| Kernel | Length | F1 | Precision | Recall |
|--------|--------|-------|-----------|--------|
| SSK | 3 | 0.925 | 0.981 | 0.878 |
| | 4 | 0.932 | 0.992 | 0.888 |
| | 5 | 0.936 | 0.992 | 0.888 |
| | 6 | 0.936 | 0.992 | 0.888 |
| | 7 | 0.940 | 0.992 | 0.900 |
| | 8 | 0.934 | 0.992 | 0.885 |
| | 10 | 0.927 | 0.997 | 0.868 |
| | 12 | 0.931 | 0.981 | 0.888 |
| | 14 | 0.936 | 0.959 | 0.915 |

Table: Performance (F1, Precision and Recall) of SVM in conjunction with SSK ($\lambda = 0.5$) for Reuters category earn.

Results

| Kernel | λ | F1 | Precision | Recall |
|--------|-----------|-------|-----------|--------|
| SSK | 0.01 | 0.937 | 0.968 | 0.913 |
| | 0.03 | 0.941 | 0.968 | 0.920 |
| | 0.05 | 0.945 | 0.974 | 0.920 |
| | 0.07 | 0.945 | 0.974 | 0.920 |
| | 0.09 | 0.927 | 0.987 | 0.880 |
| | 0.10 | 0.947 | 0.980 | 0.920 |
| | 0.30 | 0.948 | 0.980 | 0.920 |
| | 0.50 | 0.936 | 0.979 | 0.900 |
| | 0.70 | 0.893 | 0.993 | 0.813 |
| | 0.90 | 0.758 | 0.810 | 0.727 |

Table: The performance (F1, Precision and Recall) of SVM in conjunction with SSK (k=5) for Reuter's category crude

Results: Effectiveness of Sequence Length

| Category | SSK | NGK | WK |
|----------|-------|-------|-------|
| earn | 0.940 | 0.944 | 0.925 |
| acq | 0.876 | 0.882 | 0.802 |
| crude | 0.936 | 0.937 | 0.904 |
| corn | 0.779 | 0.847 | 0.762 |

Table: Performance (F1) of SVM in conjunction with SSK ($\lambda = 0.5$), NGK and WK for Reuters categories.

Results: Effectiveness of Weight Decay Factor

| Category | SSK | NGK | WK |
|----------|-------|-------|-------|
| earn | 0.946 | 0.944 | 0.925 |
| acq | 0.882 | 0.882 | 0.802 |
| crude | 0.948 | 0.937 | 0.904 |
| corn | 0.845 | 0.847 | 0.762 |

Table: Performance (F1) of SVM in conjunction with SSK (k=5), NGK and WK for Reuters categories.

Results: Effectiveness of Combining Kernels

| Category | k1 | k2 | F1 |
|----------|----|----|-------|
| earn | 5 | 6 | 0.938 |
| acq | 6 | 0 | 0.876 |
| crude | 4 | 5 | 0.936 |
| corn | 4 | 0 | 0.783 |

Strings Kernels for Human Motion Analysis

Reducing Human Motion Sequences to Strings (of Characters)

Approach 1

- Apply segmentation to motion sequence

- Apply clustering to obtained segments

- Perform prototypical feature construction

- Transform motion sequence into strings of characters

Approach 2

- Transform strings into 1-dimensional time series

- Apply SAX

Strings Kernels for Human Motion Analysis

SAX

- ❖ Map obtained motion sequence to w -dimensional space, where each coordinate is computed as

$$c'_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j$$

where n = number of frames in input space

w = number of frames in reduced space

- ❖ Perform discretization

Results: Preliminary experiments show encouraging results on a small dataset