Kernel Methods for Text Analysis

Nello Cristianini
nello@support-vector.net

Text Analysis

- Text categorization:
  eg email filtering or
  assigning document to a taxonomy like Yahoo
- Text retrieval
  possibly multi-language,
  possibly using also link structure (hypertext)
- Clustering
  e.g., creating taxonomy
- Extracting semantics
  (e.g., partially automated extraction of a semantic net,
  or a bilingual dictionary, for other applications)

Possible Types of Data

- Corpus of documents:
  a set of documents, possibly labeled
  with one <u>or more</u> categories
- Hyperlinked corpus:
  a set of documents with a link structure (directed edges)
- Paired bi-lingual corpus:
  set of pairs of documents, each the translation of the other (or: two 'aligned' translations of the same corpus)
- Usually processed by removing punctuation, stop-words, inflection, capitalization, …

Typical Tasks

- Classify elements of a corpus <u>by topic</u>
- Cluster them <u>by topic</u>
- Retrieve documents from database <u>relevant</u> to a given query
- Retrieve <u>relevant</u> documents with a query in another language

www.support-vector.net/nello.html

Remark

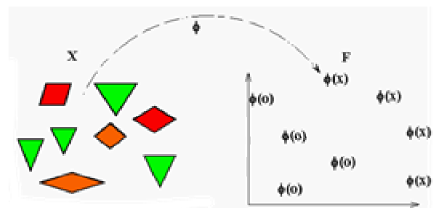
- These tasks require to operate at the level of 'topic', the document's semantic content
- Much less than full understanding, or translating
- But more serious than just processing based on easier features (eg categorize by language, by length, etc)
- We focus on problems involving the content of the document. Some level of semantic representation is required!

Overview of the Talk

- Short Review of Kernel Methods
- Vector Space Models
  - Bag of Words
  - Latent Semantic
  - Semantic Diffusion
  - Using Hypertext
  - Bi-Lingual Corpora
- String Matching

Kernel Methods for Pattern Analysis

- Work by embedding data into a vector space
- Need to know the inner product between the images of the data items (the kernel)
- Defining a suitable kernel means finding a good representation
- In our case: semantically similar documents should be mapped to nearby positions in feature space



$$x \rightarrow \phi(x)$$

$$K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$$

www.support-vector.net/nello.html

Primal and Dual

- An important property of kernel methods: instead of using directly the coordinates of the data in the embedding space,
  they represent data points by means of their inner product with the others
- If more features than documents: this is more efficient
- Dual representation: $f(x) = \langle w, x \rangle + b = \sum \alpha_i y_i \langle x_i, x \rangle + b$
- This will be relevant in the next few slides…

www.support-vector.net/nello.html

Kernel Methods

- Problem:
  how to find a semantically meaningful kernel ?
- We can:
  define it, construct it, or learn it from data…

- Successive embeddings, each closer to the semantics, are possible

Representations of Documents

- We will review two representations:

  – Bags of words

  – Symbol sequences

Vector Space Representations of Text Documents

$$d \mapsto x \mapsto \phi(x)$$

- *x* is a vector having one entry for each word in the lexicon (set of all possible words, dictionary)
- The entry x_i is the number of occurrences of word *i* in the document *d*
- We call this vector **a bag of words (BOW)**
- *Here* $\phi(x)$ is the image *of x in a* feature space (eg after normalizing, scaling or other operations, to be discussed later)

Bag of Words

- Notice that we map a document into a bag of words
- Bag = set with repetitions allowed
- We loose all information about relative positions of words
- We need to define a kernel between bags
- Possibilities:
  basic inner product between vectors
  further mappings, to improve quality of embedding $d \mapsto x \mapsto \phi(x)$

Bag of Words

- Bag of Words pioneered in Information Retrieval by Salton and his group since the ’70s
- Many alternative schemes developed to improve this first embedding, by weighting words based on their ‘relevance’, and by introducing some degree of ‘word similarity’
- All this forms the family of ‘vector space models’

Linear Feature Mapping

- An important case is when the map ϕ is linear

$$d \mapsto x \mapsto \phi(x) \qquad \phi(x) = Px$$

  - *P=diag(idf(t1),…,idf(tn))*
  - *P=diag(h(t1),…,h(tn))*
- This adjusts the weight of the different terms according to their information content
  (*idf* and *h* are some popular choices, not important here)
- More on this soon…

Nonlinear Mapping

- One could use polynomial kernels of degree *d* in order to map in the space of all possible *d*-ples of terms
- Just replace $K(x,z)$ by $K(x,z)^d$

- In the same way, one can further map by means of gaussian kernels…
- Can make a chain of many simple mappings, to construct a complex kernel …

B.O.W. Kernels

- Thorsten Joachims 1998:
  use BOW representation to design kernels

$$K(d_1, d_2) = \langle \phi(x_1), \phi(x_2) \rangle$$

- Significant improvement in classification performance over std approaches
- Discussion of SVM + BOW by Joachims: how and why it worked …
- IR invented this and other representations …(salton responsible for the vector space)

Problems with BOW

- Although BOW works well, many well known problems:
  it only compares documents using the terms they have in common.
  how to deal with semantically related terms ?
- Ideally, two documents could be similar even with no terms in common …

More Linear Mappings

- Problem: standard bag-of-words fails to capture semantic relations between words (and hence to recognize similarity between documents that contain synonyms)
- One solution: design a map *P* that encodes such relation, I.e. if z and x share no terms, but some of them are synonymous, *K(x,z)>0*
- Try to achieve $Pz \approx Px$
  if *x* and *z* are semantically similar

Vector Space Representations

- General form: $K(d_1, d_2) = d_1 PP' d_2'$

- Different *P* will give different methods.
- Vector *d*: one entry for each possible term, weighted according to its importance.
- Standard in I.R., they can all be used to build kernels (and hence for categorization, and other tasks).

Basic Vector Space Model

- (Salton et al) In this framework, the Basic Model used with kernel algorithms by Joachims98, has a diagonal *P* (either *I* or containing term weights).

- *P=I* $K(d_1, d_2) = d_1 PP' d_2' = d_1 d_2'$

Semantic Mapping

- Problem:
  how to design (or learn) a matrix P
  that contains meaningful terms-similarity

- How to use it efficiently

- Notice: using P is like 'expanding' the two documents, augmenting them with synonyms of their terms, increasing chances of a match

Inserting Semantic Knowledge

- One would like to 'expand' the representation of a document to include all synonyms of terms in the document
- The term by document matrix would be much less sparse

www.support-vector.net/nello.html

Example

	Astro naut						Cosmo naut		
D1	1	1	1	1	0	0	0	0	0
D2	0	0	0	0	0	1	1	1	1

	Astro naut						Cosmo naut		
D1	1	1	1	1	0	0	***1***	0	0
D2	***1***	0	0	0	0	1	1	1	1

A first attempt:
Generalised Vector Space Model

- (Wang et al): *P=D*, where *D* is the data matrix

$$K(d_1, d_2) = d_1 D D' d_2'$$

- This represents a term by the set of documents that contain it. Two terms with similar occurrence pattern are considered as related
- Not very strong results…
  – but interesting perspective.
  We will see more on this soon
- (Computationally: just square the K matrix up …)

Terms and Documents

- Term by term matrix
  TxT
  entries: level of similarity between terms
- Term by document matrix
  TxD
  (each document represented by row of features, each term by column of documents)
- Document by document matrix
  DxD (analogous to kernel matrix)
  entries: level of similarity between documents

Primal / Dual

- Primal and Dual in kernel methods correspond to
  term-based and document-based representation in the vector space model

The Kernel Matrix

- The central structure in kernel machines

K=

K(1,1)	K(1,2)	K(1,3)	…	K(1,m)
K(2,1)	K(2,2)	K(2,3)	…	K(2,m)
…	…	…	…	…
K(m,1)	K(m,2)	K(m,3)	…	K(m,m)

Semantic Smoothing of VSM

- Solias & d'Alche-Buc, 2000:

P is hand-built with a semantic network (WordNet).

- (if 2 terms t_i and t_j have graph distance *d,* the matrix will have entry *P(ij)=1/d*

Then a gaussian kernel is also applied:

$$\|Pd_1 - Pd_2\| = (d_1 - d_2)PP'\,(d_1 - d_2)'$$

- Improvements reported

Latent Semantic Indexing

- Compare two documents in a semantic space
- Capture semantic correlations by detecting co-occurrences
- Assumption: two documents are semantically related if they co-occur frequently in same documents
- Used in Retrieval, better performance than VS, introduced by Deerwester et al.

Joint work with Shawe-Taylor and Lodhi

Latent Semantic Indexing

- *Do this automatically: consider SVD of term-doc matrix:*

$$D = U\Sigma V'$$

- Remove small singular values
- This realizes another <u>bottleneck mapping</u>.
- Property: *co-occurring terms will be merged into a unique direction (semantically related terms).*
- *Known from IR to capture synonymy information (LSI)*

Latent Semantic Indexing

- Semantic information given by co-occurrence analysis
- Co-occurrence information given by SVD of term-by-doc matrix

- LSI introduced by (Deervester et al, 90) for IR
- Projects data into lower dimensional space. New coordinates are groups of related terms (concepts)

Latent Semantic Kernels

$$D = U\Sigma V' \qquad\qquad P = U_k$$

Can be computed directly on the kernel matrix, no need for term-vectors to be processed (and can be done AFTER a first kernel mapping).

Algorithmically same as *kernel-PCA* (Schoelkopf et al)
find directions corresponding to correlated terms,
map documents in that subspace …

www.support-vector.net/nello.html

Another Interpretation

- Building semantic networks
- Consider the graph having one node per term, connection between nodes given by co-occurrence in same document of corpus

  (simple semantic network)
- The spectrum of a graph: eigenvectors of higher order used for graph partitioning.
- LSK finds regions of the semantic network.

- Define precision, recall, F1 measure … !!

Speed Up Techniques

- Gram-Schmidt approximation of SVD

(iteratively choose vector with largest projection on subspace orthogonal to current set of vectors)

- Other low rank approximations are possible
- (see Smola et al for kernel-GS)

Three Directions to Explore…

- The GVSM:
  just a first approximation of term similarity.
  How can we extend it to longer range correlations?

- The terms-graph idea: can we push it further ?

- LSK was unsupervised … can we find BETTER directions by using supervision ???

Semantics from Equilibrium Conditions

- Two documents are similar if they contain similar terms
- Two terms are similar if they appear in similar documents

Joint work with Shawe-Taylor and Kandola

www.support-vector.net/nello.html

Semantics from Equilibrium Conditions

- We can write the resulting system as follows:
- K is doc/doc matrix
  G is term/term matrix

$$\hat{K} = \lambda X'\hat{G}X + K$$

$$\hat{G} = \lambda X\hat{K}X' + G$$

- Its solution yields:

$$\hat{K} = K(I - \lambda K)^{-1}$$

$$\hat{G} = G(I - \lambda G)^{-1}$$

www.support-vector.net/nello.html

Semantic Proximity

- And we can regard the new kernel matrix as defined by a semantic proximity matrix as follows:

$$\hat{G} = XSX' = XP'PX'$$

$$S = P'P = \lambda\hat{K} + I$$

- parameter λ controls decay rate of influence between correlated documents…

Some experiments…

Table 2: Medline dataset - Mean and associated standard deviation alignment, F1 and SVC error values for a SVC trained using the Bag of Words kernel (A) and the von Neumann ($\hat{K}$). The index represents the percentage of training points.

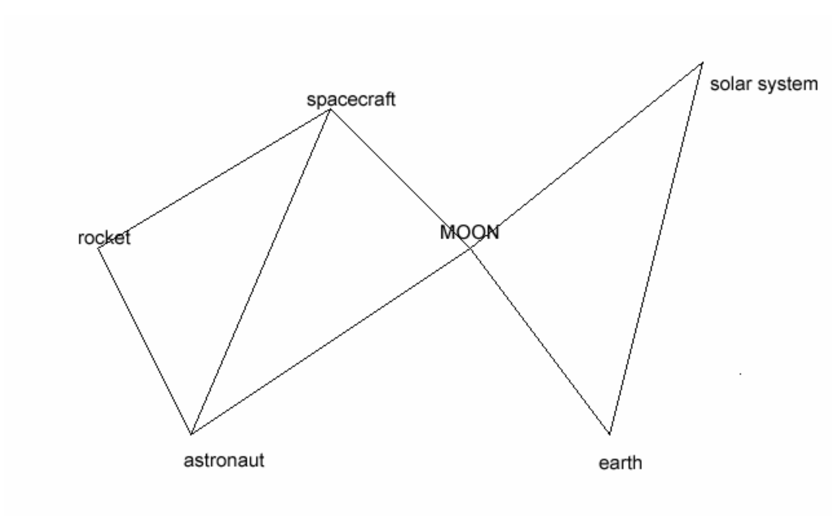
	Train Align	SVC Error	F1	λ
$\hat{K}_{80}$	0.758 (0.015)	0.017 (0.005)	0.881 (0.020)	0.032 (0.001)
A_{80}	0.423 (0.007)	0.022 (0.007)	0.256 (0.351)	-
$\hat{K}_{50}$	0.766 (0.025)	0.018 (0.006)	0.701 (0.066)	0.039 (0.008)
A_{50}	0.390 (0.009)	0.024 (0.004)	0.456 (0.265)	-
$\hat{K}_{20}$	0.728 (0.012)	0.028 (0.004)	0.376 (0.089)	0.029 (0.07)
A_{20}	0.325 (0.009)	0.030 (0.005)	0.349 (0.209)	-

Parameter λ tuned automatically using only training set information

Semantic Diffusion Kernels

- Consider the graph whose nodes are terms, and edges exist between terms that co-occur in the corpus
- We can consider the diffusion process from one node to the other, in order to refine the similarity notion given by co-occurrence analysis
- Kondor et al.-2002 studied diffusion kernels

The idea…



The idea

- Similarity between two nodes determined by all possible paths connecting them (weighted to reduce long range effects).
- From a local measure (co-occurrence) to a global measure (hopefully closer to semantic similarity).
- Trying to capture the idea that the meaning depends on the way a word is used, and hence on global usage patterns…

The Kernel

Somewhat similar to case before, but if we add a much faster decay rate, we obtain … (an extreme version of the generalized vector space model)

$$\hat{K}(\lambda) = K \sum_{i=1}^{\infty} \frac{\lambda^i K^i}{i!} = K \exp(\lambda K)$$

Some results…

Table 1: Medline dataset - Mean and associated standard deviation alignment, F1 and SVC error values for a SVC trained using the Bag of Words kernel (A) and the exponential kernel (K). The index represents the percentage of training points.

	TRAIN ALIGN	SVC ERROR	F1	λ
K_{80}	0.851 (0.012)	0.017 (0.005)	0.795 (0.060)	0.197 (0.004)
A_{80}	0.423 (0.007)	0.022 (0.007)	0.256 (0.351)	-
K_{50}	0.863 (0.025)	0.018 (0.006)	0.783 (0.074)	0.185 (0.008)
A_{50}	0.390 (0.009)	0.024 (0.004)	0.456 (0.265)	-
K_{20}	0.867 (0.029)	0.019 (0.004)	0.731 (0.089)	0.147 (0.04)
A_{20}	0.325 (0.009)	0.030 (0.005)	0.349 (0.209)	-

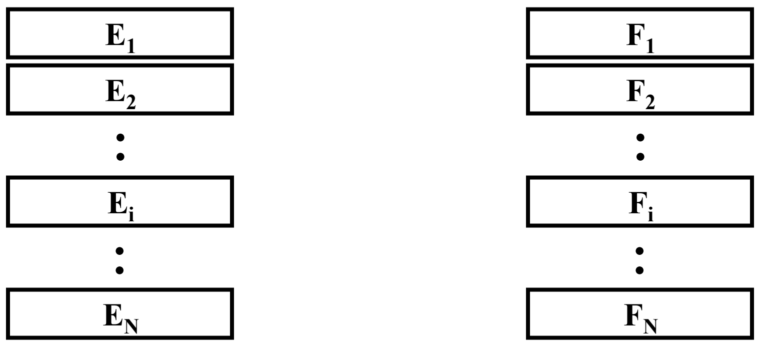
Exploiting Bilingual Corpora

- Both for cross-language analysis, and as a way to learn a semantic mapping for 1 language …

- Given a bilingual aligned corpus (e.g., english and french, from canadian parliament)

Joint work with Shawe-Taylor and Vinokurov

aligned text



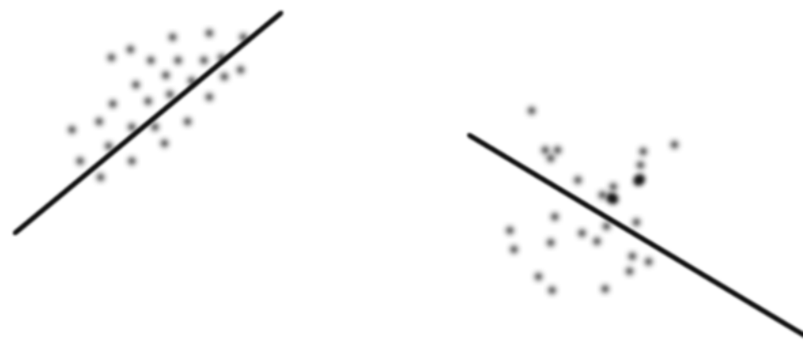
(CCA) Canonical Correlation Analysis

- 1- correlation between 2 random variables:
(assume observations 1…m performed)

$$a = (a_1, a_2, \ldots, a_m)$$
$$b = (b_1, b_2, \ldots, b_m)$$
$$\text{zero mean}, \sigma = 1$$

$$C(a,b) = \frac{\langle a,b \rangle}{\sqrt{\langle a,a \rangle}\sqrt{\langle b,b \rangle}}$$

- 2- here: random variable is projection of data point *x* onto a given direction *w*



CCA

- Given 2 sets of points in bijection
  (or a set of pairs of points, generally in different spaces X1 and X2)
- Find a direction w1 in X1 and w2 in X2, such that the projection of the datasets onto the respective directions is maximally correlated

Formally…

$$\max_{w_1, w_2} C(\langle w_1, x^i{}_1 \rangle, \langle w_2, x^i{}_2 \rangle)$$

- Maximize correlation of random variables <w1,x1> and <w2,x2> over choice of w1 and w2

- This leads to a generalized eigenvalue problem

Generalized eigen-problem

- This leads to a generalized eigenvalue problem, both in the primal and in the dual…
  - Av=λBv

- We skip all the details, we give directly the dual problem (leading to the α coefficients for the directions w1 and w2)

www.support-vector.net/nello.html

CCA

Let *x* and *y* be random variables with zero mean
And x=w_x'x and y=w_y'y be their projections in the directions w_x and w_y

$$C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix} = E\left\{ \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}^T \right\}$$

$$\rho = \frac{E\{\mathrm{xy}\}}{\sqrt{E\{\mathrm{xx}\}E\{\mathrm{yy}\}}} = \frac{E\{\hat{w}_x{}^T x y^T \hat{w}_y\}}{\sqrt{E\{\hat{w}_x{}^T x x^T \hat{w}_x\}E\{\hat{w}_y{}^T y y^T \hat{w}_y\}}} = \frac{w_x{}^T C_{xy} w_y}{\sqrt{w_x{}^T C_{xx} w_x w_y{}^T C_{yy} w_y}}$$

Skipping some steps…

$$\begin{cases} C_{xy}\hat{w}_y = \rho\lambda_x C_{xx}\hat{w}_x \\ C_{yx}\hat{w}_x = \rho\lambda_y C_{yy}\hat{w}_y \end{cases}$$

$$\lambda_x = \lambda_y{}^{-1} = \sqrt{\frac{w_y{}^T C_{yy} w_y}{w_x{}^T C_{xx} w_x}}$$

$$\begin{pmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{pmatrix} \begin{pmatrix} w_x \\ w_y \end{pmatrix} = \mu \begin{pmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{pmatrix} \begin{pmatrix} w_x \\ w_y \end{pmatrix}$$

kCCA

- This can be kernelized,
  (by replacement w=Kα)
  and the dual is:

$$\begin{pmatrix} 0 & K_1K_2 \\ K_2K_1 & 0 \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \rho \begin{pmatrix} K_1{}^2 & 0 \\ 0 & K_2{}^2 \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$$

- At least 4 authors have done this
  independendly in the last year or so!
  (I used Bach & Jordan)

kCCA

- Very promising method, when used in conjunction with kernels

- Tomorrow we will see an application of this to cross-language analysis
- Work in progress in bioinformatics

kCCA

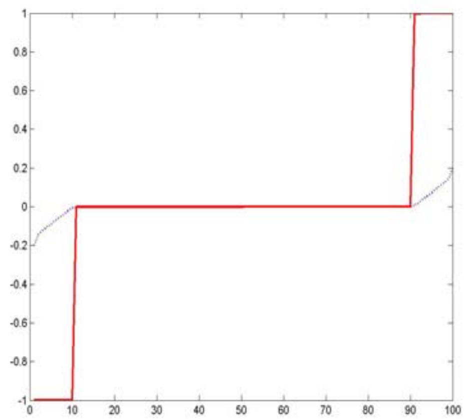
- Important to understand its ‘overfitting’ behaviour (to avoid it).
- Usually $B \leftarrow B + \lambda I$
- This constrains the norm of vectors w, making the system less flexible …

Some artificial examples…

- 50 random points in 10 dimensions
- Correlated with itself
- And with randomized version of self

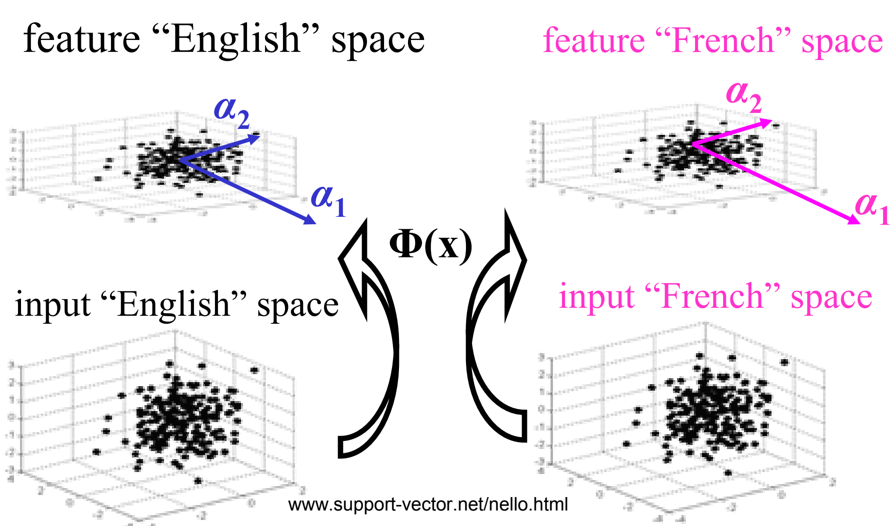
- We expect only 10 positive eigenvalues
- If full freedom is given, they will be =1
- We can reduce their freedom …

Regularization of kCCA…



www.support-vector.net/nello.html

cross-lingual kernel canonical correlation analysis



www.support-vector.net/nello.html

- Two ways to look at it:
  - either french documents as sophisticated labels for the english ones, in supervised feature extraction task
  - Or: 'unsupervised' task from a paired corpus…

Comparing with random pairings …

The correlation between **E** and **F** is higher than between **E** and **rand(E)**
And lower than between **E** and **E**, or **F** and **F**
As expected

kCCA components

PENSIONS PLAN?	
pension	regime
plan	pensions
cpp	rpc
canadians	prestatior
benefits	canadiens
retiremen	retraite
fund	cotisation
tax	fonds
investmer	discours
income	impôt
finance	revenu
young	jeunes
years	ans
rate	pension
superann	argent
disability	regimes
taxes	investisse
mounted	milliards
future	prestatior
premiums	plan
seniors	finances
country	pays
rates	avenir
jobs	invalidit
pay	resolutior

AGRICULTURE?	
wheat	bl
board	commissi
farmers	agriculteu
newfound	producteu
grain	canadienr
party	grain
amendme	parti
producers	conseil
canadian	commerci
speaker	neuve
referendu	ministre
minister	administr
directors	modificati
quebec	qubec
speech	terre
school	formistes
system	partis
marketing	grains
provinces	op
constituti	nationale
throne	lus
money	bloc
section	nations
rendum	chambre
majorit	administr

CANADIAN LANDS?	
park	parc
land	autochtor
aboriginal	terres
yukon	ches
marine	vall
governme	ressource
valley	yukon
water	nord
boards	gouverne
territories	offices
board	marin
north	eaux
parks	territoires
resource	parcs
agreemen	nations
northwes	territorial
resources	revendica
developm	ministre
treaty	cheurs
nations	ouest
territoire	entente
work	rights
territory	office
atlantic	atlantique
programs	ententes

FISHING INDUSTRY	
fisheries	pêches
atlantic	atlantique
operative	pêcheurs
fishermen	pêche
newfound	probl
fishery	coop
problem	ans
operative	industrie
fishing	poisson
industry	neuve
fish	terre
years	ouest
problems	stocks
wheat	ratives
coast	ministre
oceans	sant
west	saumon
salmon	affaiblies
tags	facult
minister	secteur
communit	programn
program	gion
commissi	scientifiqu
motion	travailler
stocks	conduite

Details…

- we compute the product <alphas>*<training examples>
- we get d vectors (d- number of alpha vectors)
- which we treat as documents and extract from them 30 most "frequent" words
- ("frequency" is a component in the vectors)

Hypertext Kernels

- Document = bag of links
- The adjacency matrix A is analogous to the term-document matrix
- Inspired on Kleinberg's HITS algorithm (and PageRank)
- Represent documents by their connectivity pattern (similar documents have similar connections).
- Same operations as before are possible:

Can also be merged with text informati

Joint work with Joachims and Shawe-Taylor

Hypertext

- Typical example: hypertext.
- Two different representations of web pages (by words and by links). Both known to be informative, expected to be independent
- Combination of them should improve performance

Co-citation

- THE COCITATION MATRIX:
  introduced in bibliometrics. Two documents have positive score if cited by same document
- The co-link matrix: obvious extension. Positive score if pointed to by same webpage.
- The cocitation kernel: this matrix is also a Gram matrix.
  Feature space dimensionality = corpus size.

Kernel Combination

- If K_1 and K_2 are kernels, and $a>0$, $b>0$, then $K_{comb}=aK_1+bK_2$ is also a kernel
- When is K_{comb} better than K_1 and K_2 ?
- Answer: if they are both ‘good’ and ‘different'
- Boosting type of idea: combine independent ‘experts’ …

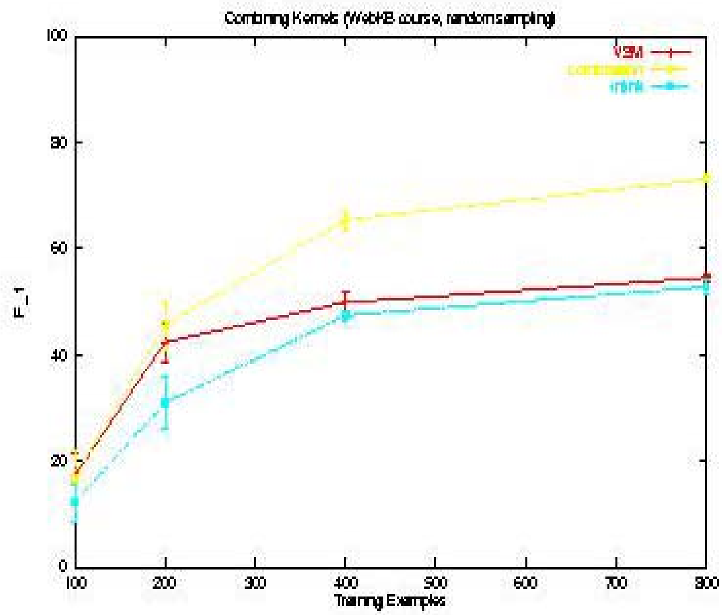
- Analysis of this kind of kernel combination is possible (eg based on the concept of alignment, or others), we will not do it here

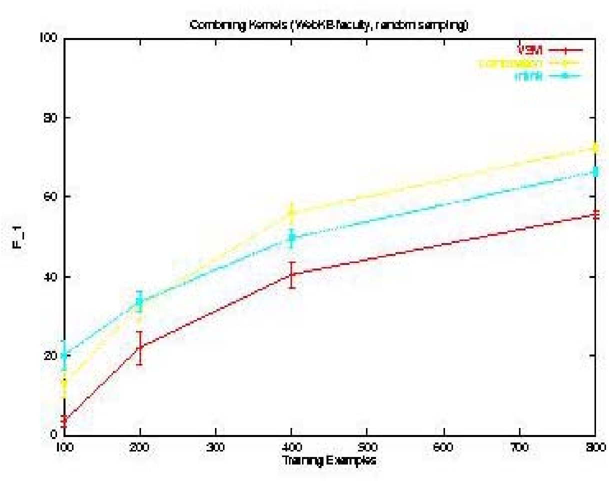
Data

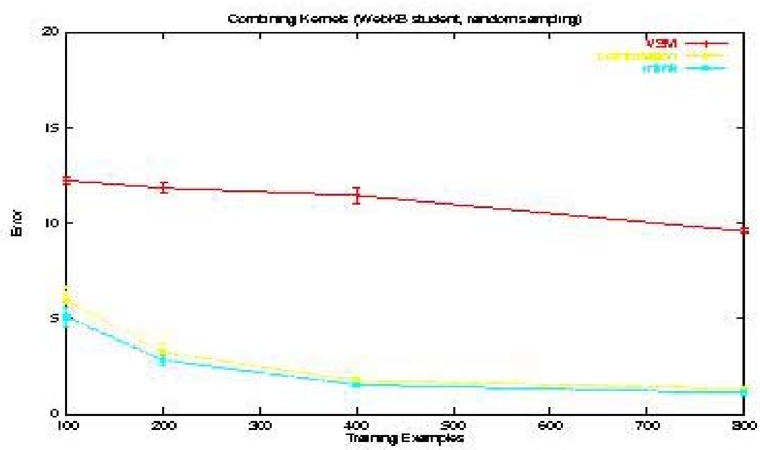
- 4 Universities WebKB dataset as compiled by Sean Slattery for ICML00
- http://www.cs.cmu.edu/~WebKB/ICML2000-data.html
- 4168 examples
- 623 words selected by frequency (done by Sean Slattery)
- three binary tasks (student homepages, faculty homepages, and course homepages)

Hypertext Results

- Tried several kernels, and combination of inlink + VSM.
- *inlink*: Binary representation of all links pointing to the page. Examples normalized to unit length
- *combination*: *VSM+inlink* kernel added with equal weight.


Combining Kernels (WebKB course, random sampling)
VSM
combination
rlink
F_1
Training Examples


Combining Kernels (WebKB faculty, random sampling)
VSM
combination
rlink
F_1
Training Examples



Fisher Kernels

- Fisher Kernel:
  given a generative model, it sees what parameters of the model need to be adapted to accommodate a given new data point (Introduced by Jaakkola and Haussler)
- Two points are similar if they 'stretch the model' in the same way
- Hoffmann's probabilistic model of text: latent variables (~topics) generate the documents…
- Trained the model to fit a corpus, a kernel can now be defined…
- Probabilistic LSI …

- END OF VSM
- NOW WE DO STRING MATCHING …

String Representations

- Documents as symbol sequences (symbols can be: letters, syllables, words, etc)
- "Soft" matching functions can reveal the degree of similarity of two sequences (developed for bioinformatics by …)
- Map sequence into feature space formed by all sub-strings of …

- START WITH A SIMPLE EXAMPLE, THEN WE COMPLICATE IT …

A <u>simple</u> kernel for sequences

Consider a space with dimensions indexed by all possible finite substrings from alphabet A.

Embedding: if a certain substring *i* is present once in sequence *s*, then $\phi_i(s)=1$

Inner product: counts common substrings

Exponentially many coordinates, but can compute the inner product in such space in LINEAR time by using a recursive relation

Sequence-Kernel-recursion

It starts by computing kernels of small prefixes, then uses them for larger prefixes, etc

$$K(s,\Omega)=1$$

$$K(sa,t)=K(s,t)+\sum_i K(s,t[1:i-1])[t_i=a]$$

- Where *s,t* are generic sequences, *a* is a generic symbol, Ω is the empty sequence, …
- Analogous relation for *K(s,ta)* by symmetry…
- Dynamic programming techniques evaluate this in linear time !

Example

$$K(s,\Omega)=1$$
$$K(sa,t)=K(s,t)+\sum_{i}K(s,t[1:i-1])[t_i=a]$$

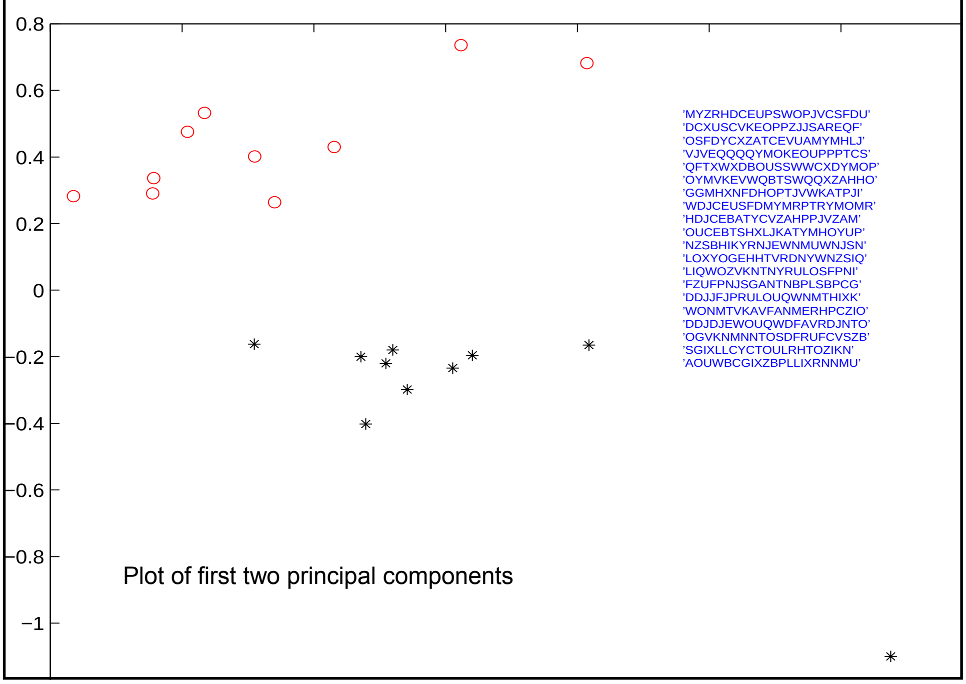
S=<u>**ABBCBBC**</u>A

T=<u>**BBABBCAB**</u>

Dynamic programming:
stored in table all the kernels for all smaller prefixes
The computation of the sum is just a matter of looking them up

More advanced sequence kernels…

- Compare substrings of length *k*, and tolerate insertions …
- Similar (but more complicated) recursions…
- Demonstrated on sets of strings (generated by 2 different markov sources)



Plot of first two principal components

Example

	C-A	C-T	A-T	B-A	B-T	C-R	A-R	B-R
Cat	λ^2	λ^3	λ^2	0	0	0	0	0
car	λ^2	0	0	0	0	λ^3	λ^2	0
Bat	0	0	λ^2	λ^2	λ^3	0	0	0
Bar	0	0	0	λ^2	0	0	λ^2	λ^3

www.support-vector.net/nello.html

Example

- Unnormalized: *K(cat,car)*= λ^4
- *K(bat, bar)*= λ^4
- *K(car, car)*= $2\lambda^4 + \lambda^6$
- Normalized: *K(cat,car)*= $1/(2+\lambda^2)$

String Alignment Kernels

- Recursive procedure to compute the kernel

$$K_n(s,t) = \sum_{u \in \Sigma^n} \left\langle \phi_u(s), \phi_u(t) \right\rangle$$

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

- A recursion can give this in time linear in the length of the sequences

Joint work with Lodhi, Watkins, Saunders, JST

www.support-vector.net/nello.html

Results

- Comparable (but not better than) bags of words…
- Interesting that no prior knowledge was inserted here
  (space just another symbol, no stemming or preprocessing …)

CONCLUSIONS

- Vector Space models natural match with kernel methods
- Many ways to iteratively improve the embedding, inserting semantic information
- Cross-linguistic correlation analysis very promising
- Hyperlinks can help
- String matching works for text (but slowly)

References

- These slides in:
  www.support-vector.net/tutorial.html

- All cited papers in:
  www.support-vector.net/text-kernels.html