Kernel Methods -6
Applications of Kernel Methods

Nello Cristianini
nelly@support-vector.net
UCB – CS-294

News

• Course web-page updated, online material, homework, etc. etc.
  www.cs.berkeley.edu/~nelly
• Also: handouts to help with homework
Overview

• General summary of ideas (and take-home message of the course)
• Examples of practical applications of KMs in real-world problems
• Three main areas:
  – Bioinformatics
  – Text Categorization
  – Handwritten Digits Recognition
• Kernel Methods and Other Pattern Recognition Systems …

We have seen:

– General structure of a kernel-based PR system
  (modularity, embedding, learning module based on learning-theory and optimization)
– Properties of kernel functions (characterization; Mercer and RKHS; kernel combination; diagonal dominance and overfitting; kernels and prior knowledge; alignment)
– Properties of kernel algorithms (duality; Representer’s theorem; optimization; working in embedding space; avoiding overfitting …)
– Examples of Kernel-Based Algorithms (kernel perceptron; Support Vector Machines; Kernel PCA; kernel Ridge Regression; Parzen Windows; clustering)
– Learning kernels (kernel matrices) from data (and using them for transduction; clustering; etc)
Basic Message

- There are advantages in looking for patterns in data by embedding them in a suitable space.
- There is a methodology to do so ‘systematically’. Algorithms are efficient and well understood.
- Computational and Generalization aspects can be modeled and controlled.
- They can be used for classification; clustering; feature selection; novelty detection; regression; etc.
- It is a very open, very exciting field to do research in … everybody is welcome ! 😊

Today:

- We will end, looking at some of the most promising directions in which this technology is being applied.
- We will survey - rather than discussing in detail – a number of applications in the fields of:
  - Bioinformatics
  - Text categorization
  - Handwriting recognition
About bioinformatics problems

• Types of data:
  sequences (DNA; or proteins)
  gene expression data
  SNP; proteomics; etc. etc
• Types of tasks:
  diagnosis; gene function prediction
  protein fold prediction; drugs design; …
• Types of problems:
  high dimensional; noisy; very small or very large datasets; heterogeneous data; …

Gene Expression Data

• Measure expression level of thousands of genes simultaneously, in a cell or tissue sample
  (genes make proteins by producing RNA; a gene is expressed when its RNA is present…)
• Very high dimensionality; noise
• Can either characterize tissues or genes (transposing matrix)
Gene Function Prediction

- Predict functional roles for yeast genes based on their expression profiles
- Given set of 2467 genes, observed their expression under 79 conditions (from Eisen et al.)
- Assigned genes to 5 functional classes (from MIPS yeast genome database).
  TCA cycle; respiration; cytoplasmic ribosomes; proteasome; histones.
- SVM: learn to predict class based on expression profile.

Gene Function Prediction

- SVMs compared with 5 other algorithms, performed best (parzen windows; fisher discriminant; decision trees; etc).
- Results on paper (and website).
- Also used to assign to their functional class ‘new’ genes.
- Often mistakes have biological interpretation ….

- www.cse.ucsc.edu/research/compbio
Gene Function Prediction

- Notice: not all functional classes can be expected to be predicted on the basis of expression profiles
- The 5 classes were chosen using biological knowledge: expected to show correlation.
- Also: chosen for control a class not expected to have correlation: helix-turn-helix.

Heterogeneous Information

- Diverse sources can be combined. An example.
- Phylogenetic Data obtained from comparison of a given gene with other genomes
- Simplest Phylogenetic Profile: a bit string in which each bit indicates whether the gene of interest has a close homolog in the corresponding genome
- More detailed: negative log of the lowest E value reported by BLAST in a search against a complete genome
- Merged with Expression data to improve performance in Function Identification
Heterogeneous Data

- Similar pattern of occurrence across species could indicate 1) functional link (they might need each other to function, so they occur together). Could also simply indicate 2) sequence similarity
- Used 24 genomes from the Sanger Centre website
- Again: only some functional classes can benefit from this type of data.
- Generalization improves, but mostly for effect 2): a way to summarize sequence similarity information

- Pavlidis, Weston, Cai, Grundy, “Gene Functional Classification from Heterogeneous Data”, International Conference on Computational Molecular Biology, 2001

Cancer Detection

- Task: automatic classification of tissue samples
- Case study: ovarian cancer
- Dataset of 97808 cDNAs for each tissue! (each of which may or may not correspond to a gene)
- Just 31 tissues of 3 types: ovarian cancer; normal ovarian tissue; other normal tissues. (15 positive and 16 negatives)

- Furey, Cristianini, Duffy, Bednarski, Schummer, Haussler, “Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data” Bioinformatics
Ovarian Cancer

- Main goal: decide whether a given sample is cancerous or not
- Secondary goal: locate genes potentially responsible for classification
- Problem: overfitting due to curse of dimensionality

Results

- Cross validation experiments (l.o.o.).
- Located a consistently misclassified point. The sample was considered cancerous by the SVM (and dubious by humans that originally labelled it as OK). Re-labelled.
- The only non-ovarian tissue is also misclassified consistently. Removed.
- After its removal: perfect generalization
- Attempt to locate most correlated genes provides less interesting results (used Fisher score for ranking, independence assumption).
- Only 5 of the top 10 are actually genes, only 3 cancer related. More on this later.
Colon Tumour Dataset

- 40 tumour and 22 normal colon tissues measured for 2000 human genes (publicly available)
- Data used in clustering experiments by Alon et al.
- Using all the features, and also selecting the 1000 best features (fisher score), SVMs make 6 mistakes
- Good results also from p-norm ensemble perceptron

Cancer Classification

- Task: distinguish between Acute Myeloid Leukemia and Acute Lymphoblastic Leukemia (public dataset from )
- Training 38 (27 ALL / 11 AML) and testing 34 (20 ALL / 14 AML)
- Correctly predicted between 30 and 32 test samples (using features and examples that performed best on training set, loo).
- Also: predict success of chemotherapy (data only available for 15 AML patients)
Protein Homology

• Special kernels can be designed for comparing protein sequences, based on HMMs
• The generative model used as ‘feature extractor’ for designing a kernel (‘Fisher kernel’).
• Successfully used to detect remote protein homology
  • Jaakkola, Diekhans, Haussler “Using the Fisher Kernel Method to Detect Remote Protein Homologies”, AAAI press

Promoters

• Similar technology used to classify genes based on the patterns in their regulatory region
• Task: identify co-regulated genes based on promoter sequences
  • Pavlidis, Furey, Liberto, Haussler, Grundy, “Promoter Region-Based Classification of Genes”, Pacific Symposium on Biocomputing, 2001
Promoters

- Simple way to represent promoters: presence of motifs that function as binding sites of TFs
- Small size and other problems make approach very noisy
- More abstract features: exploit presence of multiple copies and combinations of motifs; spacing between two motifs; sequence flanking the motifs; presence of more general – less conserved – patterns.
- Features of the promoter region not only the TFBSs motifs themselves.

Fisher Kernels

- Capture information about presence and relative position of motifs
- 1) build a motif-based HMM from a collection of TFBS motifs
- 2) extract Fisher kernel and use in in SVM
- 3) discriminate between a given set of promoters from co-regulated genes and a second set of negative example promoters
- Result: predicted coregulation of unannotated genes. Predictions validated with expression profiles or other annotation sources.
String Matching Kernels

- Different approach, very promising: dynamic programming method to detect similarity between strings
- So far: used in text categorization. Being tested on protein data.
- LATER MORE ON THIS
- Other work, with different kernels: detection of translation initiation sites.

More on Bioinformatics

- Different types of data, very noisy and from different sources
- Problem: How to combine them?
- One possible answer: kernel combination …
Transcription Initiation Site

• Parts of DNA are junk, others encode for proteins. They are transcribed into RNA and then translated into proteins
• The transcription starts at ATG; but not all ATGs are transcription initiation sites …
• Problem: predict if a given ATG is a TIS based on its neighbors …. 

SVMs

• Encoding: window of 200 nucleotides each side around the candidate ATG
• Each nucleotide encoded with a 5 bits word (00001, 00010, 00100, 01000, 10000) (for A,C,G,T, and N – unknown).
• Comparisons of these 1000-dim bitstrings should reveal which ones contain actual TIS
Naïve Approach

• Linear kernels:
  \( <x,z> \)

• Polynomial kernels:
  \( <x,z>^d \)

Special Kernels

• Poly kernels consider all possible k-ples, even very distant ones
• We assume that only short range correlations matter
• We need a kernel that discards long range correlations
‘locality improved’ kernels

• First consider a window of length \(2l+1\) around each position. We will compare two sequences ‘locally’, by moving this window along them …

\[
\text{win}_p(x, z) = \left( \sum_{j=-l}^{l} w(j) \text{match}_{p+j}(x, z) \right)^{d_z}
\]

\[
K(x, z) = \left( \sum_{p=1}^{I} \text{win}_p(x, z) \right)^{d_z}
\]

Notice: these are all kernel-preserving operations on basic kernels. Hence the result is still a valid kernel. Weights chosen to penalize long range correlations.

TIS detection with Locality Improved kernels

• Performed better than polynomial kernels
• Better than best neural network (state of the art on that benchmark)

Work by Zien et al., appeared on Bioinformatics
Protein Fold

• Problem is: given sequence of aminoacids forming a protein, predict which overall shape the molecule will assume
• Problem: defining the right set of features, the right kernel
• Work in progress

Predicting Activity of Drugs Based on Structure

• Features of a molecule: high dim vectors
  (electron-density-derived molecular surface area)
  (quantitative structure activity relations)
• Number of observations /number of variables:
  197/640; 62/694; 46/769; 66/626; 64/620; 76/1181; 76/1181
• Standard chemometrics tool:
  partial least squares, designed for high spaces
• SVM regression was compared to it: significantly better in 6 out of 7 sets
• (by Demiriz et al)
APPLICATIONS OF kPCA

• Used to extract features, later fed to a regression machine
• Performed better than linear PCA
• Applied to regression problem, by Rosipal et al,

• Break
Text Categorization

• Problem: map a text document to a category, based on its content

• Kernels used for this see KerMIT project in Europe, an analogous NSF project here, a workshop at NIPS 2001 ….

Kernels for Text

• The rest of the talk focuses on the design of kernel functions for text.

• 3 approaches discussed:
  – Latent semantic kernels
  – String matching kernels
  – Hypertext kernels and kernel combination
Vector Space Representations of Text Documents

\[ d \mapsto x \mapsto \phi(x) \]

- \( x \) is a vector, \( \text{dim}(x) = \# \text{dictionary} \)
- \( x_i \propto \# \{t_i \in \text{doc. } d\} \) \text{[BAG OF WORDS]}
  (...with various normalizations ...)

\( \phi(x) \) feature vector

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

Linear Feature Mapping

- An important case is when the map \( \phi \) is linear
  \[ d \mapsto x \mapsto \phi(x) \]
  \[ \phi(x) = Px \]

- \( P=\text{diag}(\text{idf}(t_1),\ldots,\text{idf}(t_n)) \)
- \( P=\text{diag}(h(t_1),\ldots,h(t_n)) \)
- This adjusts the weight of the different terms according to their information content
- 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 …

More Linear Mappings

- Problem: standard bag-of-words fails to capture semantic relations between words
- 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, $K(x,z)>0$

$$P_z \approx P_x$$
A Semantic Mapping

- Solias & D’Alche-Buc handcrafted such a matrix $P$, by using WordNet to assess the semantic proximity between terms $P(i,j)$.
- Used Support Vector Machines with this kernel (further combined with a gaussian K).
- This is very expensive, we will present ideas to perform similar operation implicitly and more generally, using 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

- Can equivalently be done on doc-by-doc matrix
- Can be done AFTER or BEFORE other non-linear mappings (eg poly kernels)
  - $K = U \Sigma V'$
  - $K^* = U_k \Sigma V_k'$
  - Here $P = U_k = U_l_k$ (another linear mapping)
- Joint work with Huma Lodhi, John Shawe-Taylor

Polynomial kernel $d=1$, soft-margin, Medline Query 20 (100 random splits, 930+103)
Polynomial kernel $d=4$, soft-margin, Medline

Does not always work well …

Considerations

- Very expensive to do full SVD decomposition
- Can use Lanczos; Gram-Smidt; diagonal

- If performed BEFORE poly kernel mapping, we can capture ‘d-ples of concepts’. 
## String Matching Kernels

- Feature space: non-contiguous symbol sequences of size $n$.
- Example, $n=3$: `seminar`, `seminar`, `seminar`, `seminar`, `seminar`, `seminar`, ...
- ‘sparse’ sequences are penalised
- (Exponential dim.)

- Joint work with Huma Lodhi, John Shawe-Taylor, Chris Watkins,

### For any $0 < \lambda < 1$, for $n=2$

<table>
<thead>
<tr>
<th></th>
<th>C-A</th>
<th>C-T</th>
<th>A-T</th>
<th>B-A</th>
<th>B-T</th>
<th>C-R</th>
<th>A-R</th>
<th>B-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>$\lambda^2$</td>
<td>$\lambda^3$</td>
<td>$\lambda^2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car</td>
<td>$\lambda^2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\lambda^3$</td>
<td>$\lambda^2$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Bat</td>
<td>0</td>
<td>0</td>
<td>$\lambda^2$</td>
<td>$\lambda^2$</td>
<td>$\lambda^3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bar</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\lambda^2$</td>
<td>0</td>
<td>$\lambda^2$</td>
<td>$\lambda^3$</td>
<td></td>
</tr>
</tbody>
</table>

So $K(\text{car,cat})=\lambda^4$, $K(\text{cat,cat})=2\lambda^4 + \lambda^6$
Recursive Computation

- Exponential number of features,
- Kernel can be computed efficiently with recursive definition

- If $s$, $t$ sequences, and $x$ symbol, can express $K_n(sx,t)$ as function of $K_n(s,t)$ and $K_{n-1}$ (more or less)
- Exact definition a bit complicated, see paper

Results on Reuters

- Despite successes in speeding up, still very expensive computation
- Tested on subset of Reuters (380 training, 90 testing); $n=5,6$; $\lambda=0.5$; 4 categories (earn; acq; crude; corn).
- Compared with bag-of-words kernels
### How to merge information?

**Hypertext**

- Many possible kernels for text
- Text: bag of words $\rightarrow$ **Hypertext: bag of links**.
  (Same ideas for link-weighting-schemes, etc)
- How to combine them?
- Exploit modularity of kernels, to combine many simple ones in a complex (better?) one.
Some Preliminary Experiments

- We designed kernels for hypertext exploiting link structure.
- We tested simplest ones, in principle also LSI and other stuff could be applied to them (connection with HITS, PageRank).
- Used dataset WebKB (Sean Slattery, ICML2000)
  - Joint with Thorsten Joachims, John Shawe-Taylor (All experiments by Thorsten)

Data

- 4 Universities WebKB dataset as compiled by Sean Slattery for ICML00
- 4168 examples
- 623 words selected by frequency (done by Sean Slattery)
- three binary tasks (student homepages, faculty homepages, and course homepages)
Experiments

- We designed kernels for hypertext exploiting link structure.
- We tested simplest ones, in principle also LSI and other stuff could be applied to them (connection with HITS, PageRank).
- Used dataset WebKB (Sean Slattery, ICML2000)

Data

- 4 Universities WebKB dataset as compiled by Sean Slattery for ICML00
- 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*: $VSM + \text{inlink}$ kernel added with equal weight.

<table>
<thead>
<tr>
<th>error (%)</th>
<th>$\text{BOW}_{\text{bin}}$</th>
<th>$\text{BOW}_{\text{tfidf}}$</th>
<th>$\text{BOL}$</th>
<th>$\text{BOW}_{\text{bin}}/\text{BOL}$</th>
<th>$\text{BOW}_{\text{tfidf}}/\text{BOL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>course</td>
<td>$4.33 \pm 0.07$</td>
<td>$5.89 \pm 0.05$</td>
<td>$4.33 \pm 0.12$</td>
<td>$2.75 \pm 0.06$</td>
<td>$4.40 \pm 0.18$</td>
</tr>
<tr>
<td>student</td>
<td>$9.57 \pm 0.16$</td>
<td>$13.55 \pm 0.11$</td>
<td>$1.10 \pm 0.03$</td>
<td>$1.28 \pm 0.04$</td>
<td>$1.55 \pm 0.05$</td>
</tr>
<tr>
<td>faculty</td>
<td>$2.67 \pm 0.10$</td>
<td>$3.79 \pm 0.06$</td>
<td>$2.12 \pm 0.06$</td>
<td>$1.60 \pm 0.06$</td>
<td>$2.57 \pm 0.08$</td>
</tr>
</tbody>
</table>

Table 1. Error rates of the word, the link, and the combined kernels on 800 training examples.

<table>
<thead>
<tr>
<th>#SV</th>
<th>$\text{BOW}_{\text{bin}}$</th>
<th>$\text{BOW}_{\text{tfidf}}$</th>
<th>$\text{BOL}$</th>
<th>$\text{BOW}_{\text{bin}}/\text{BOL}$</th>
<th>$\text{BOW}_{\text{tfidf}}/\text{BOL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>course</td>
<td>158.0</td>
<td>413.4</td>
<td>519.3</td>
<td>115.3 (102.6)</td>
<td>285.4 (268.5)</td>
</tr>
<tr>
<td>student</td>
<td>273.5</td>
<td>618.1</td>
<td>458.9</td>
<td>158.3 (156.9)</td>
<td>359.6 (354.6)</td>
</tr>
<tr>
<td>faculty</td>
<td>127.7</td>
<td>358.3</td>
<td>200.1</td>
<td>61.3 (46.3)</td>
<td>146.8 (133.6)</td>
</tr>
</tbody>
</table>

Table 2. Number of support vectors for the word and the link kernels on 800 training examples, as well as the number of common support vectors. The number in parenthesis is the expected overlap, if the SV sets were independent.
New Work

- Hoffman’s probabilistic LSI kernels, and hypertext extensions …
- Very interesting and relatively unexplored: generative models for kernel design

- Here you are ideally placed for this…
Handwritten Digits Recognition

- Standard benchmark for pattern recognition algorithms (USPS digits; MIST dataset)
- SVMs with polynomial kernels competitive with state of the art (LeNet)
- By adding knowledge to the kernel, best performance on MIST (Decoste & Schoelkopf, 2000)

Kernel Jittering

- Consider handwritten digits
- Want to incorporate knowledge that small perturbation of a handwritten character do not change its meaning
- (Small) Translations, rotations, thickenings of the line, …
- Map an image to a set of images obtained by applying these transformations
- Measure similarity between such sets… (actually their wok for some reason measures just ‘similarity’ between a point and a set)
Kernel Engineering

• Similarly to the case of TIS, also in this case simple prior knowledge insertion improved the system
• Analogous to the use of Semantic Networks for the text categorization kernel
• This is the way to go also for bioinformatics, gene expression analysis …

End of course

• Today we have seen that successful applications come with using prior knowledge together with the theories seen in the previous classes
• Creativity is needed in kernel design
• No free kernel… if a kernel knows nothing, it cannot be of much help.
The End

• Let’s keep in touch
• Contact me - ask me questions about the homework
  (not now: try and do it yourself first!)
• Talk to me about course projects if you want
• If you want more reading, look at my book!
• Or ask me about specific topics, I’ll point you to papers

The End