Towards On-Line Classification

Vartika Singh

vsingh2@cubs.buffalo.edu

Outline

- Problem Statement
- Proposed Approach
- Results
- Future Work

Problem Statement

- Problem Definition:
  - Data keeps arriving.
  - Frequent retraining with new data expensive.
  - Many classes.
  - In case of 'no' classification of new data, it should be added and labelled as a new class and thus labelled.
- Motivation:
  - OCR not sufficient.
  - Often a bottleneck.
- Application Space:
  - Document Images
  - For now: journal and conference papers
  - Also, any problem which fill the description of problem definition.
- Goals:
  - Features.
  - Also, data representation.
  - Since, class label not known beforehand, how to classify.
  - Discriminative and/or Generative model?

Approach

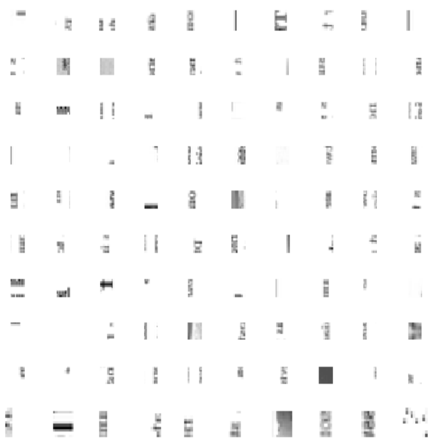
- Generative models:
  - Train a model over each class or document
  - Good if we are certain about the generative model beforehand – techniques like EM can be used to make optimization faster.

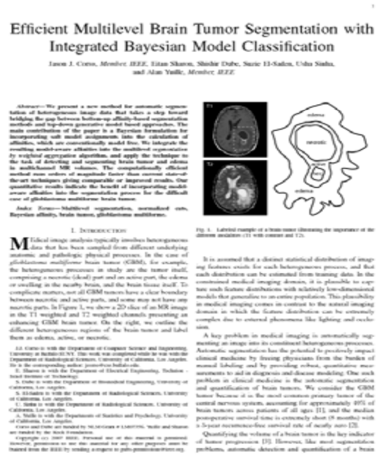
- Content Based Image Retrieval:
  - Segments images into known/meaningful objects.
  - Use them for semantic representation of image.
  - Query – string of objects
  - Documents retrieved in a ranked order

- Vector-Space model:
  - Traditional representation of documents in vector-space.
  - Query – also in the same vector space.
  - Relevant documents retrieved in a ranked order.

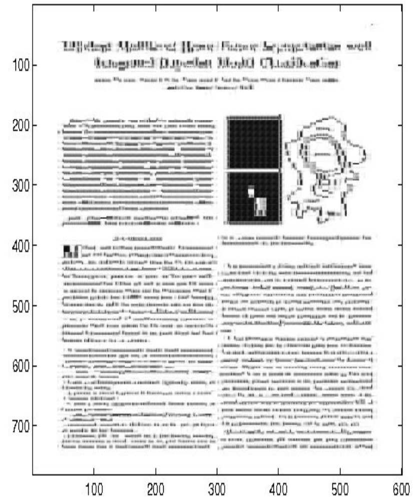
- Combine all three to form a hybrid generative and discriminative model.
  - Cutting up the images in patches gives the 'meaningful' objects.
  - Vector space model gives a way of representing the documents using the labelled objects..

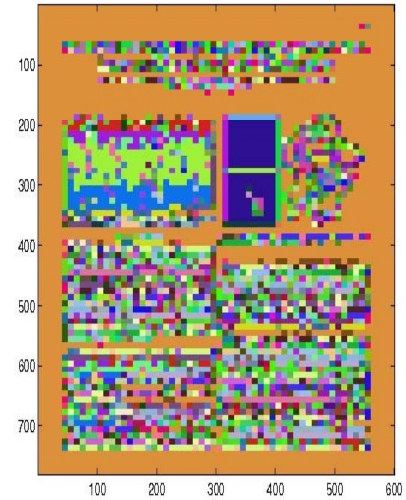
cse@buffalo





Efficient Multilevel Brain Tumor Segmentation with Integrated Bayesian Model Classification





i) **Means of the patches**

ii) **Original document image**

iii) **Data coded document image**

iv) **Coded document image**

●Features :

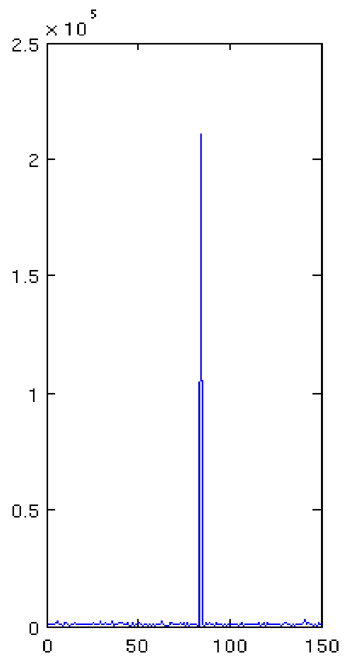
- Wavelets, SIFT: Where and how many.
- Classical document features: X-Y cuts, point pattern matching and so on. Not generic.
- We use image patches.
- We, cut up the image in patches. 10 by 10 pixels size. Use them all as image 'words'.

●Moving away from on-line aspect:

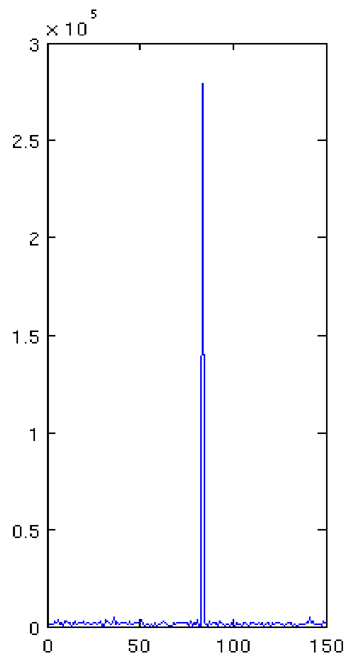
- We do clustering to get representative 150 patches.
- With each new incoming image the cluster centres will have to be recomputed.
- Could we approximate the a set from the same domain as another given set?

●Features : making them independent

- Documents which we get or are expected to get are usually text documents with some images thrown in.
  - •Let, Cluster centres = Words in a vocabulary.
  - •Small set of words used frequently.
  - •From a given set, approximate another set from the same domain.
- Figures.
- Thus, we do not have to recompute the cluster centres every time a new document comes in.



i) **<u>Distribution for 75 images.</u>**



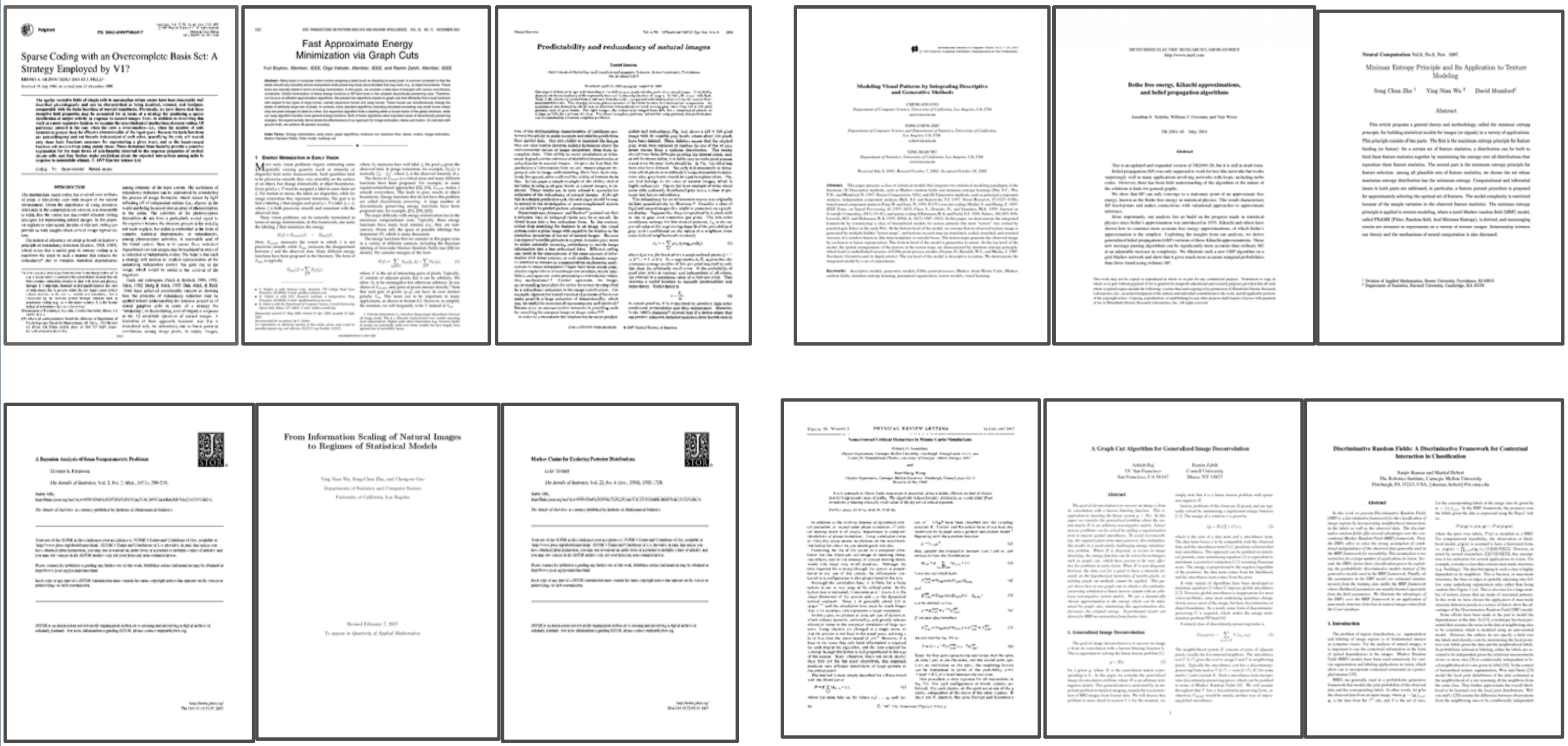
ii) **<u>Distribution for 100 images.</u>**

●Using the Vector-Space Model:

- The traditional Tf-Idf scheme.
- $O(N^2)$ time to build the vector space representation of a document.
- If the index files are not re-computed every time a new document comes in, the new time, O(N).
- $td_{ij} = tf_{ij} * idf_{ij}$, where
  - **tf_{ij} = length of d_j / Number of times the word occurs in d_j.**
  - **Idf_{ij} = K / number of documents in which w_j occurs at least once.**

●Some results

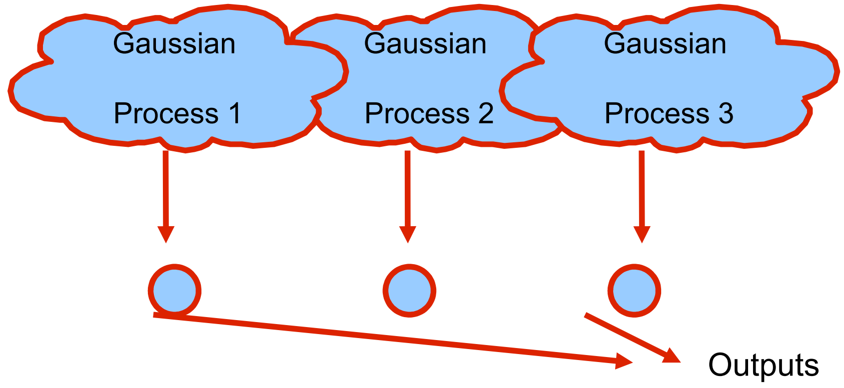
- The document image data set was represented in the vector space using the tf-idf scheme. The length of the vector is equal to the cluster number size.
- Just for an initial test, these vectors were clustered using a binary-decision tree using K-Means. The results at the second level of the tree look as follows:



Modelling relation between patches.

- Images consist of patches (cluster centres).
- Number of clusters << Number of words. Same cluster occur in many documents.
- Vector space model disregards spatial knowledge.
- Use Hierarchical Gaussian Processes.
- For now, assuming we have labelled data, then it is easy to extend the vector space representation of documents by one more dimension which will be the label of the document. Say this be an integer number from 1,2,...M.
- Then our multi-class Gaussian process can be defined on the indexed set {1,2, .. M} * X.
- Assuming audiences familiarity with mixture models.

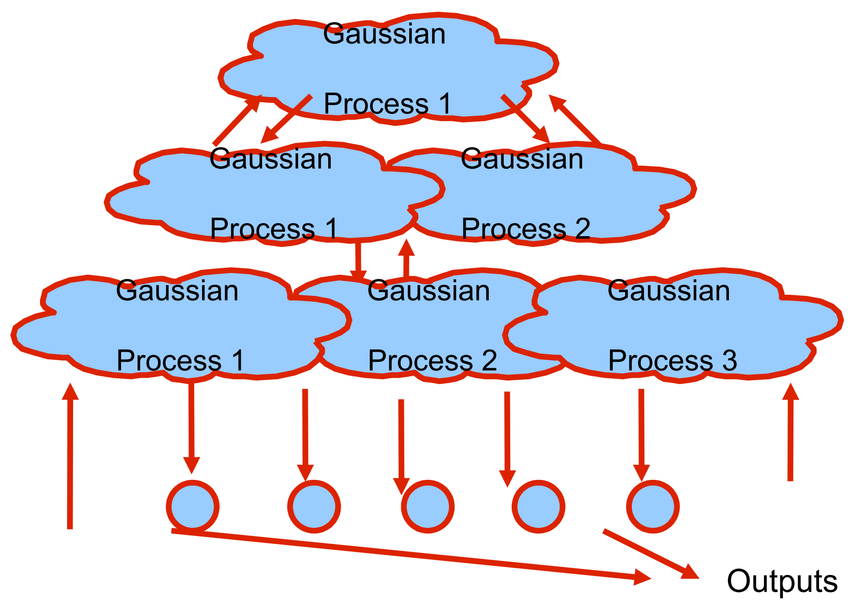
cse@buffalo



- **Multi-class Gaussian processes with**
- **hierarchical Bayesian framework.**
  - Data X, where X is the rows of vector space representation of training documents, with a single dimension for output appended in the end. If number of documents in training set is K, then X' is of dimension K * (vector length +1).
  - The relation between the documents classes i and j from the set of M can be coded into the covariance function, Cov($\kappa(i, x_a)$, $\kappa(j, x_b)$), where x_a, x_b are in X.
  - Thus, a Gaussian process over each class, with all of them being trained together. (Informative Vector machines to be used for sparse representation.)
  - Let there be hyper parameters on each dimension of columns of X'. Thus, after marginalizing the uncertainty in (vector length +1) hyper parameters, we get optimal co-dependent covariance matrices, with each of them having some hyper parameters operating over common attributes. Thus, for every new incoming document, hyper parameters to be recomputed..

Future Work

- Having a Gaussian process for each class heavy.
- To continue with the preliminary work done with simple clustering and decision trees. Replace each node with a Gaussian process.
- Each layer will have it's own objective function, which minimizes independently of other layers.
- The output is in the form of regression, with each document taking it's place in the output space.
- Cosine distances between the document vectors can be used for regression modelling.
- All in hope to make the process on-line, with the addition of new classes.



Bibliography

- Williams, C. K. I., and Rasmussen, C. E. (1996) Gaussian processes for regression. In Advances in Neural Information Processing Systems 8 , ed. by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo
- Jordan, M.I. Jacobs, R.A. (1993) Hierarchical mixtures of experts and the EM algorithm. In Neural Networks, IJCNN '93-Nagoya.
- Rasmussen, C. E. (2000) The Infinite Gaussian Mixture Model, in S. A. Solla, T. K. Leen and K.-R. Muller (editors.), Adv. Neur. Inf. Proc. Sys. 12, MIT Press, pp. 554--560
- Salton G., Buckley C.,(1987) Term Weighting Approaches in Automatic Text Retrieval, Technical Report: TR87-881 Cornell University Ithaca, NY, USA