SALSAS: Sub-linear Active Learning Strategy with Approximate k-NN Search

David Gorisse[1] Matthieu Cord[2] Frédéric Precioso[1,2]

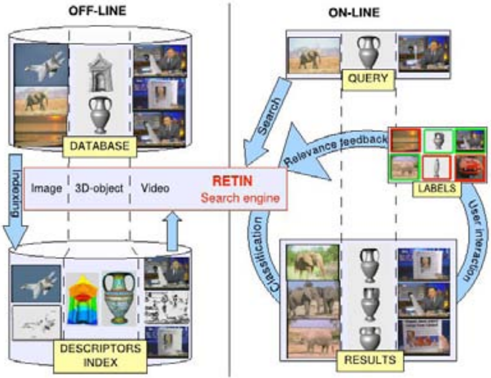
(1)ETIS lab, ENSEA-UCP-CNRS, France

(2)LIP6-UPMC Paris 6, Sorbonnes Universités

3 Novembre 2010

Introduction : CBIR classification framework

- Content-based
- Category Retrieval
- Interactivity and Learning



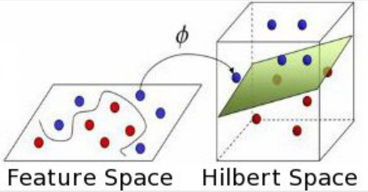
Category search $=>$ a two-class problem :

- Relevant class : image set fitting to the user query concept
- Irrelevant class : all other images from the database

Introduction : similarity function – kernel

Kernel definition :

- Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $\quad \mathbf{x}, \mathbf{y} \longmapsto k(\mathbf{x}, \mathbf{y})$
- k is a Kernel *iif* : $\exists \phi | \forall (\mathbf{x}, \mathbf{y}), k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$
- with ϕ, an embedding function into a Hilbert space.



Advantage : Machine learning friendly (Artificial neural network, SVM, . . .)

- Relevance function (without b) : $f_{\mathcal{A}}(\mathbf{x}) =< \mathbf{w}, \Phi(\mathbf{x}) >= \sum_{p=1}^{|\mathcal{A}|} \alpha_p k(\mathbf{a}_p, \mathbf{x})$

Kenels :

- RBF : $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{d(\mathbf{x}, \mathbf{y})^2}{2\sigma^2}}$
- l_2-RBF : with $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2$, $\quad \chi^2$-RBF : with $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i^p \frac{(\mathbf{x}_i - \mathbf{y}_i)^2}{(\mathbf{x}_i + \mathbf{y}_i)}}$

Introduction : interactive / active learning

Goal

To choose the best images for annotation to increase the training set thus optimally improving the search

Which ones ?

- The most relevant ones : TOPN
- The most uncertain [Tong02] and the most diversified : Angle Diversity [Brinker03] ($\mathcal{U}$ unlabeled dataset) :

$$i^{\star} = \arg\min_{\mathbf{x}_i \in \mathcal{U}} (\lambda |f_{\mathcal{A}}(\mathbf{x}_i)| + (1-\lambda) \max_{\mathbf{x}_j \in \mathcal{A}} \frac{k(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{k(\mathbf{x}_i, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_j)}})$$

Scalable interactive learning

CBIR systems

- Time retrieval ok for several thousands image databases
- But is it scalable ? what is the search time complexity ?

Scalable interactive learning

CBIR systems

- Time retrieval ok for several thousands image databases
- But is it scalable ? what is the search time complexity ?

Fast online retrieval

- Nice fast similarity search schemes recently introduced [Datar04, Valle08, Chum, Perdoch09] based on indexing structures for database representation and knn search (trees KDTree, hashing LSH, clustering Inverted files...)
- Can we speed-up the interactive learning using similar strategies ?

Scalable interactive learning

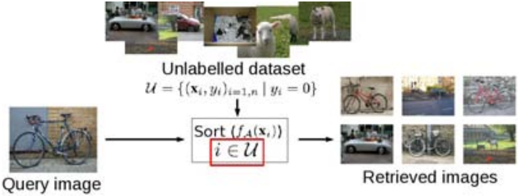
CBIR systems

- Time retrieval ok for several thousands image databases
- But is it scalable ? what is the search time complexity ?

Fast online retrieval

- Nice fast similarity search schemes recently introduced [Datar04, Valle08, Chum, Perdoch09] based on indexing structures for database representation and knn search (trees KDTree, hashing LSH, clustering Inverted files...)
- Can we speed-up the interactive learning using similar strategies ?

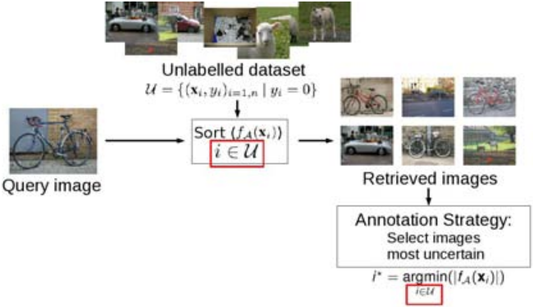
Scalability problems in interactive search



- TOPN relevant images for intermediate results.
  Computation : $f_{\mathcal{A}}(\mathbf{x})\ \forall \mathbf{x} \in \mathcal{U} +$ Ranking

$\Rightarrow$ Complexity : $O(\mathcal{U} ln(\mathcal{U}))$

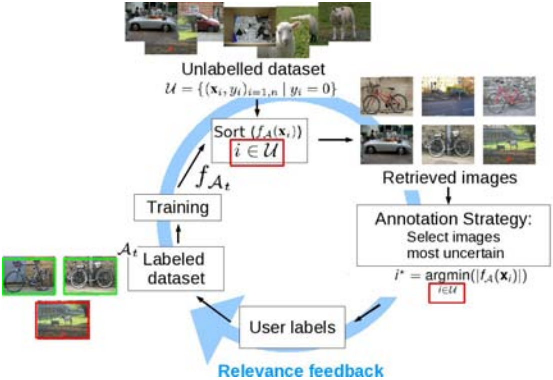
Scalability problems in interactive search



- Most uncertain images for annotation strategy
  Computation : Angle Diversity score $\forall \mathbf{x} \in \mathcal{U}$ + Ranking

$\Rightarrow$ Complexity : $O(\mathcal{U} ln(\mathcal{U}))$

Scalability problems in interactive search

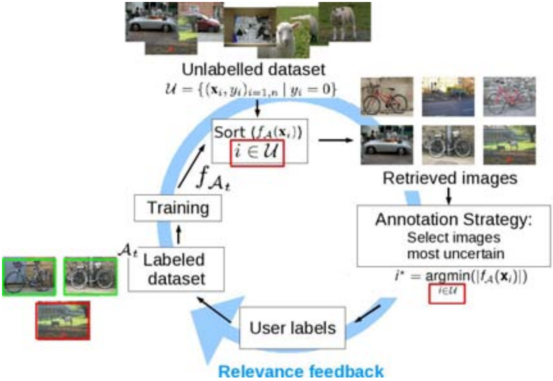


- Training

  Complexity $O(\mathcal{A}^2)$, with $\mathcal{A} \ll \mathcal{U}$

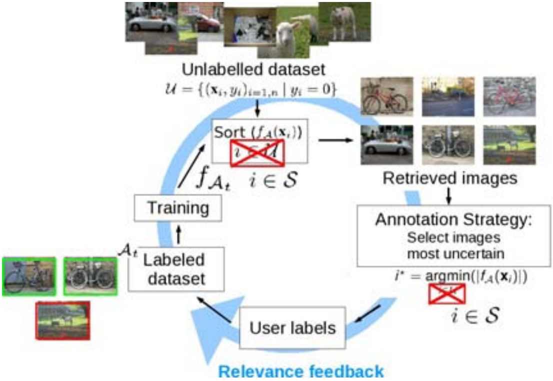
$\Rightarrow$ Complexity negligible

Scalability problems in interactive search



- Complexity at least linear regarding the size of the database
  $\Rightarrow$ impracticable for large databases

Scalability problems in interactive search



- Complexity at least linear regarding the size of the database
  $\Rightarrow$ impracticable for large databases
- Sublinear solution : only consider a relevant subset $\mathcal{S}$ instead of $\mathcal{U}$

How to decrease search complexity ?

Idea : subsampling

- Only work (ranking with the relevant function $f_{\mathcal{A}}$) on the pool $\mathcal{S}$ wrt $N < |\mathcal{S}| \ll n = \mathcal{U}$
- How to find $\mathcal{S}$? with images that have a *high probability* to be in the TOPN

Methods

- Subsampling of $\mathcal{U}$ – ramdom selection ?
- Hierarchical sampling based on clustering [Panda06] and focusing on "interesting" clusters
- Our strategy : sampling using knn search with an optimized LSH indexing structure over a $f_{\mathcal{A}}$ approximation

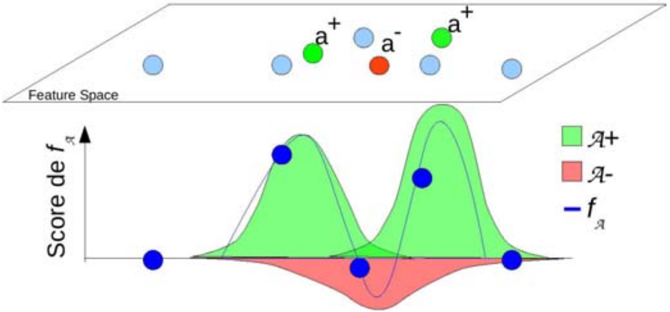
Our Subsampling Strategy

TOPN

- Looking for TOPN images = maximize the relevance function :

$$f_{\mathcal{A}}(\mathbf{x}) = \sum_{p=1}^{|\mathcal{A}^+|} \alpha_p K(\mathbf{a}_p^+, \mathbf{x}) - \sum_{n=1}^{|\mathcal{A}^-|} \alpha_n K(\mathbf{a}_n^-, \mathbf{x}) = f_{\mathcal{A}^+}(\mathbf{x}) - f_{\mathcal{A}^-}(\mathbf{x})$$

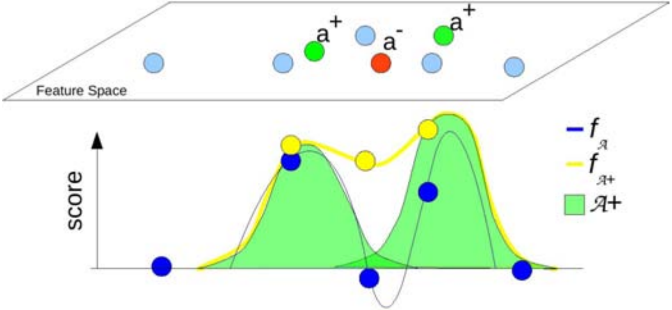
with $\mathcal{A}^+$ training set of positively annotated images
and $\mathcal{A}^-$ training set of negatively annotated images



Subsampling Strategy

Selection : to build $\mathcal{S}$

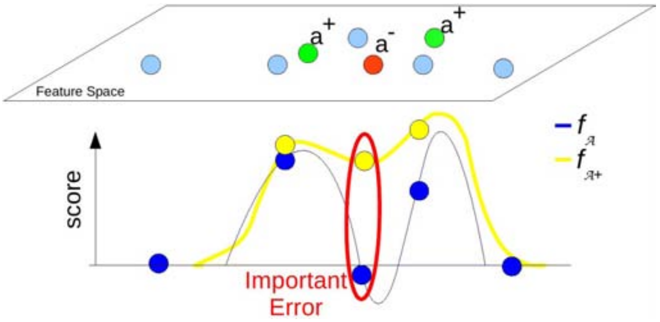
- approximation of $f_{\mathcal{A}}$ by focusing on $f_{\mathcal{A}^+}$.
- fast maximization of $f_{\mathcal{A}^+}$



Subsampling Strategy

Pruning

- some images of $\mathcal{S}$ can have a low $f_{\mathcal{A}}$ score because of $f_{\mathcal{A}-}$
- To compute exact $f_{\mathcal{A}}$ score for all images in $\mathcal{S}$ in order to filter lower score images.

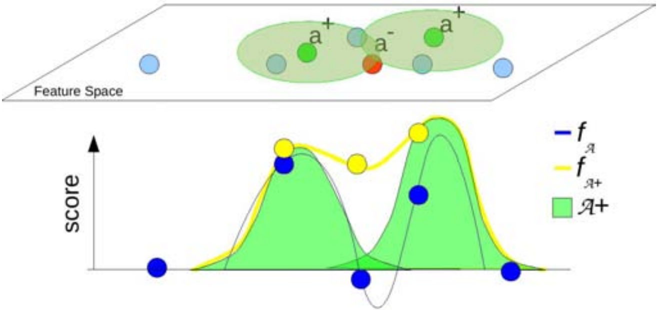


Subsampling Strategy

Salsas

- Approximation of maximization of $f_{\mathcal{A}^+}$

$\Rightarrow$ by selecting images of $\mathcal{U}$ that are Nearest Neighbor of $\mathcal{A}^+$ images



Subsampling Strategy

Annotation Strategy

- Pool $\mathcal{S}$ quite larger than the TOPN
- As long as user not satisfied uncertain images in $\mathcal{S}$

⇒ Looking for the most uncertain and diversified images in $\mathcal{S}$:

$$i^\star = \arg\min_{\mathbf{x}_i \in \mathcal{S}} (\lambda |f_{\mathcal{A}}(\mathbf{x}_i)| + (1-\lambda) \max_{\mathbf{x}_j \in \mathcal{A}} \frac{k(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{k(\mathbf{x}_i, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_j)}})$$

Pros

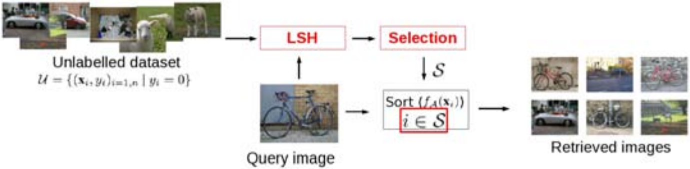
- Very fast : benefit of the previous stage
- Rebalance the problem : much more irrelevant images than relevant ones (amplified with database size growing)

Cons

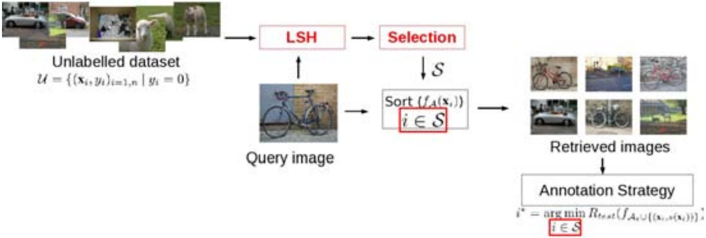
- No more theoretical validity [Tong02]

⇒ But “valid in experiments”

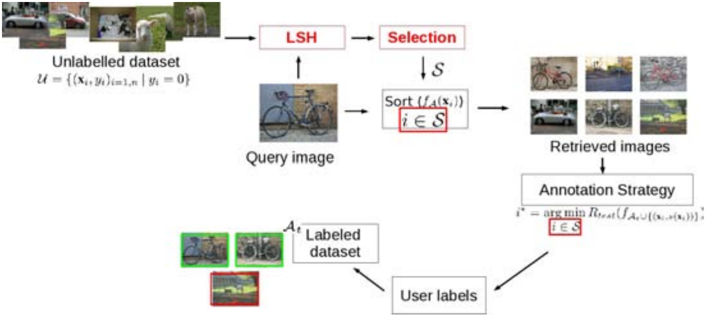
Our interactive scheme



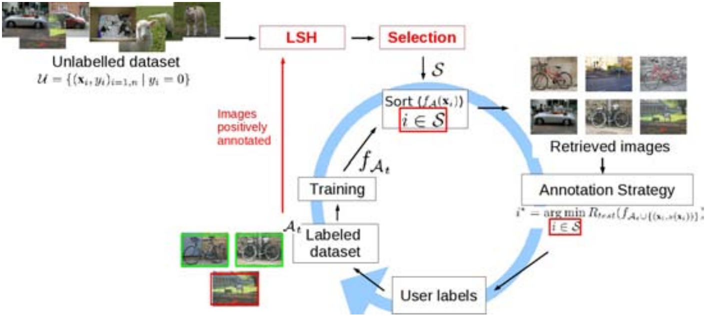
Our interactive scheme



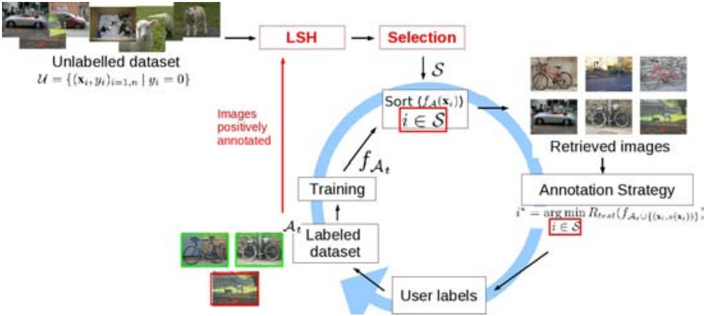
Our interactive scheme



Our interactive scheme



Our interactive scheme



$\mathcal{S}$ updating

- To be efficient, the k-nearest neighbor knn search must be very fast
  Sublinear $\Rightarrow$ Index Structure : Locality Sensitive Hashing (LSH)

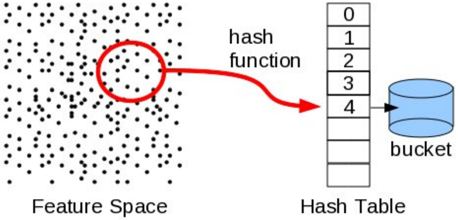
Locality Sensitive Hashing (LSH)

Definition

LSH is a space-partitioning data structure [Indyk98]

Principle

- Split database into buckets stored in a table
- Bucket accessible with a key
- Key provided by a *hash function*



Locality Sensitive goal

Hash function must be able to :

- Bring together similar images
- Sort out dissimilar images

E2-LSH : Hash function of Euclidean metric

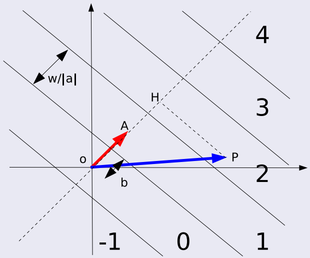
Def

A hash function to perform fast search with l_2 distance and to approximate l_2-RBF :

- Hash function $h_{\mathbf{a},b}$ based on random projection :

$$h_{\mathbf{a},b}(\mathbf{p}) = \lfloor \frac{\mathbf{a}.\mathbf{p} + b}{w} \rfloor$$

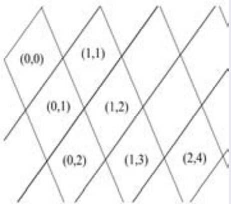
  - $\mathbf{a}$ a random vector : each component is chosen independently from a Gaussian distribution,
  - b a shift, W a bin width



E2-LSH

Def

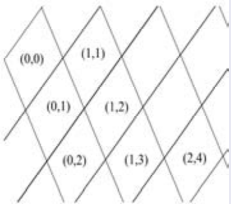
- To increase the data partitioning : concatenation of M hash functions
- Lattice to approximate Euclidean partitioning



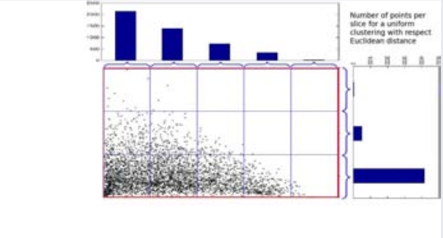
E2-LSH

Def

- To increase the data partitioning : concatenation of *M* hash functions
- Lattice to approximate Euclidean partitioning



- Limitation if data distribution not grid regular
- 4500 feature vectors randomly selected from an image database



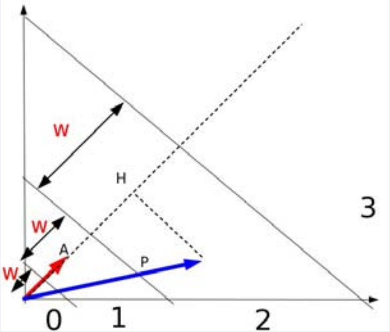
CHI2-LSH

CHI2-LSH def

A new hash function to perform fast search with χ^2 distance and to approximate χ^2-RBF

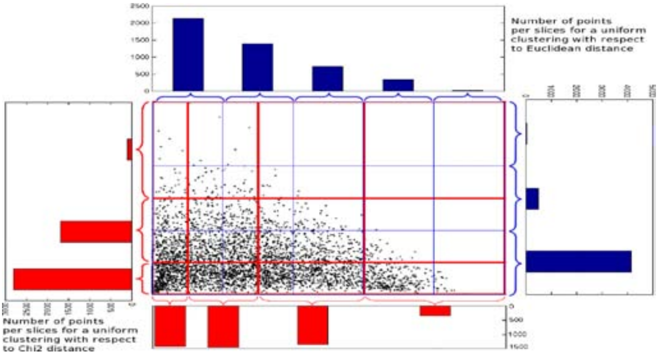
- Same principle as E2-LSH
- But distances between two consecutive hyperplans constant W with χ^2 distance

$$h_{\mathbf{a},b}(\mathbf{p}) = \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{p}}{W^2} + 1} - 1}{2} + b$$



To increase data partitioning
$->$ M hash functions, L hash tables

CHI2-LSH : splitting space differences with L2



4500 feature vectors randomly selected from an image database

Space grids for feature components with respect to χ^2 (in red) and l_2 (in blue)

Experiments

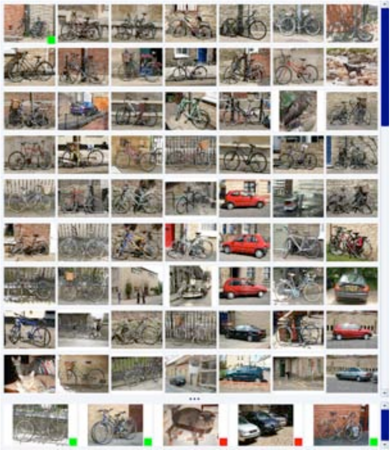
Protocol

- 5 datasets
  between 5K and 180K images from VOC 2006, 2007, 2008 + TrecVid 2007, 2008, 2009
- Feature Space
  128-dimension vector (color and texture based)
- Parameters
  1 annotation by iteration, TOP200, 100-NN search

Examples

LIN_CHI2

SALSAS



Examples

LIN_CHI2

SALSAS

Examples

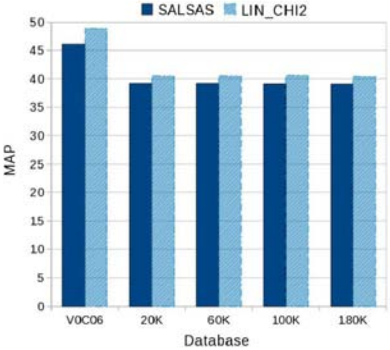
Examples

LIN_CHI2

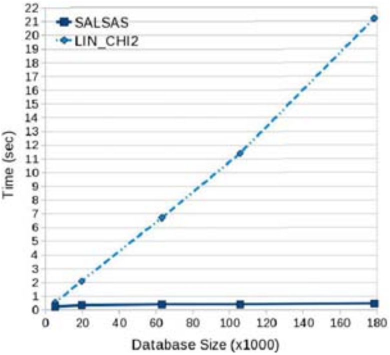
SALSAS

Evaluation (1) : exact *vs* fast search

- Accuracy and Efficiency comparison between exact and fast search
- Accuracy => MAP measurement
- Efficiency => Time speed-up measurement
- 5 datasets between 5K and 180K images



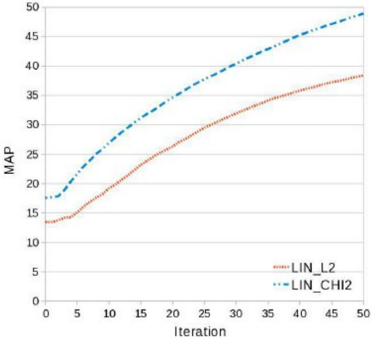
MAP of TOP200 at 50^{th} iteration
10 class VOC06 – 5 databases



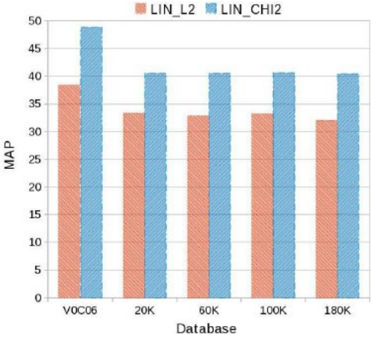
Time after 50 iterations vs database size

Evaluation (2) : χ^2 *vs* l_2-RBF

- Comparison of Acuracy and Efficiency between l_2-RBF and χ^2-RBF exact search

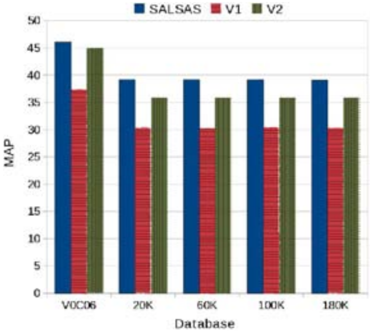


(a) MAP of TOP200 VS number of iterations on VOC06

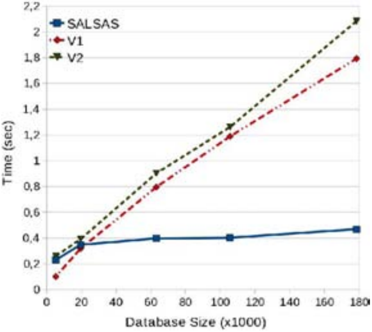


(b) MAP of TOP200 at 50th iteration VS database size

Evaluation (3) : SALSAS *vs E2LSH*



(a) MAP of TOP200 at 50th iteration vs database size



(b) Time at 50th iteration vs database size

FIG.: Evolution of the accuracy and the efficiency with the size of the database for 50 iterations with 1 label by iteration. V1 is *E2LSH* scheme combined with a l_2-RBF kernel and V2 is *E2LSH* scheme combined with a χ^2-RBF kernel.

Thanks for your attention ! Questions ?

Bibliography pieces

- SALSAS : Sub-linear Active Learning Strategy with Approximate k-NN Search, D. Gorisse, M. Cord, F. Precioso, Elsevier, Pattern Recognition, Special Issue on "Semi-Supervised Learning", to appear in 2011.
- Machine Learning Techniques for Multimedia – Case Studies on Organization and Retrieval, Springer 2008, M. Cord, P. Cunningham (Eds.).
- Active learning methods for Interactive Image Retrieval, P.H. Gosselin, M. Cord, IEEE Transactions on Image Processing, 17(7), 1200 –1211, 2008.
- Fast Approximate Kernel-based Similarity Search for Image Retrieval Task, D. Gorisse, M. Cord, F. Precioso, S. Philipp-Foliguet, ICPR 2008.
- Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval, P.H. Gosselin, M. Cord, S. Philipp, CVIU, 2008

People

- David Gorisse (ETIS, ENSEA/UCP/CNRS PhD student)
- Matthieu Cord
LIP6, Univ. UPMC-PARIS 6
matthieu.cord@lip6.fr
http ://webia.lip6.fr/∼cord/

  Frédéric Precioso
ETIS ENSEA/UCP/CNRS
frederic.precioso@lip6.fr
http ://frederic.precioso.free.fr