Variable Bit Quantisation for LSH

School of informatics

Sean Moran†, Victor Lavrenko, Miles Osborne
† sean.moran@ed.ac.uk

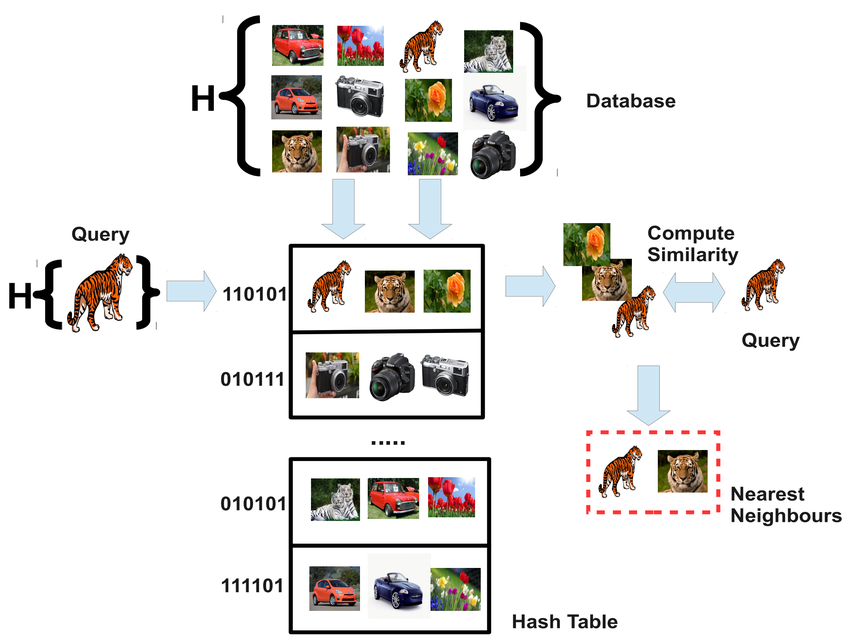
The University of Edinburgh

Research Question

- LSH uses 1 bit per hyperplane. Can we do better with multiple bits?

Introduction

- **Problem:** Fast Nearest Neighbour (NN) search in large datasets.
- **Hashing-based approach:**
  - Generate a similarity preserving binary code (fingerprint).
  - Use fingerprint as index into the buckets of a hash table.
  - If collision occurs only compare to items in the same bucket.



Content Based IR

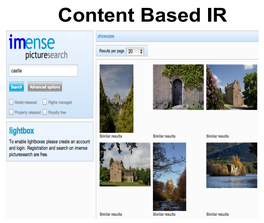


Image: Imense Ltd

Near duplicate detection

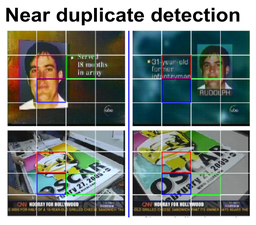


Image: Xu et al.

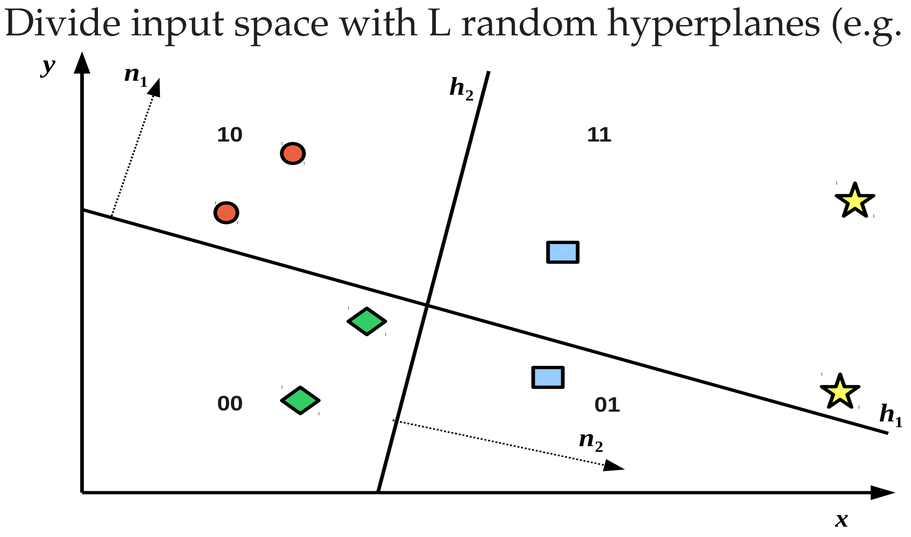
Location Recognition

Image: Doersch et al.

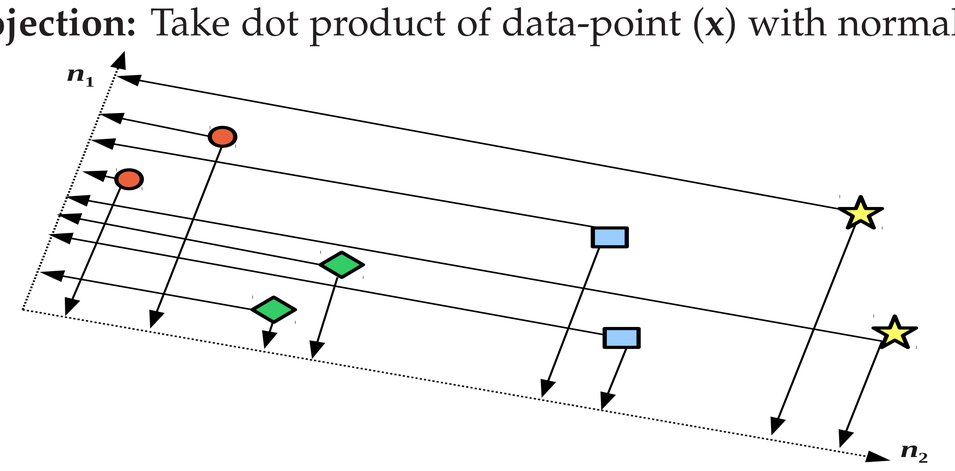
- **Advantages:**
  - Constant query time (with respect to the database size).
  - Compact binary codes are extremely storage efficient.

Locality Sensitive Hashing (LSH) (Indyk and Motwani, '98)

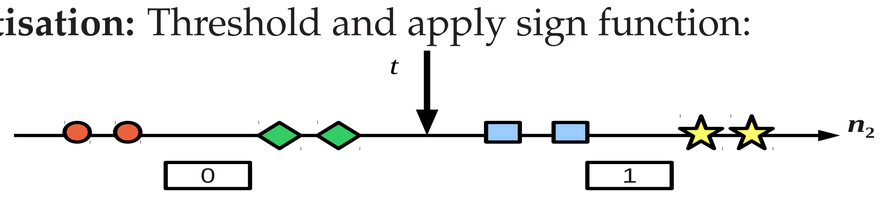
- Randomized algorithm for approximate nearest neighbour (ANN) search using binary codes.
- Probabilistic guarantee on retrieval accuracy versus search time.
- LSH for inner product similarity:
  - Divide input space with L random hyperplanes (e.g. L=2):



  - **Projection:** Take dot product of data-point (**x**) with normal (**n**.**x**):



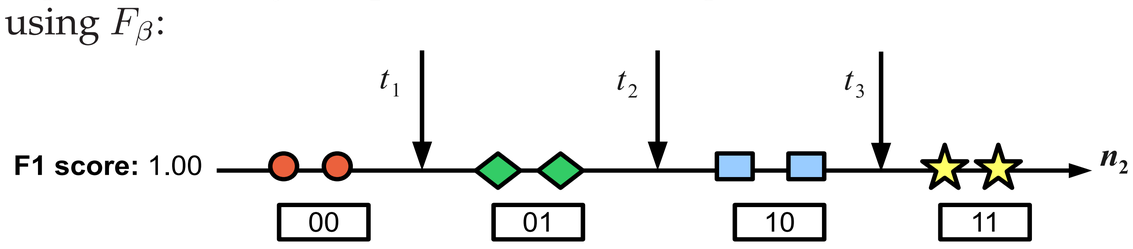
  - **Quantisation:** Threshold and apply sign function:



- **Vanilla LSH**: each hyperplane gives 1 bit of the binary code.

Variable Bit Quantisation for LSH

- Goal: assign more bits to neighbourhood preserving hyperplanes.
- Score hyperplanes by F_β computed using pairwise constraints matrix S: if $S_{ij} = 1$ then points $\mathbf{x}_i$, $\mathbf{x}_j$ with projections $\mathbf{y}_i$, $\mathbf{y}_j$ are true nearest neighbours.
- TP: # $S_{ij} = 1$ pairs in same region. FP: # $S_{ij} = 0$ pairs in same region. FN: # $S_{ij} = 1$ pairs in different regions. Combine TP, FP, FN using F_β:



2 bits assigned: 3 thresholds perfectly preserve the neighbourhood structure

0 bits assigned: 0 thresholds do just as well as one or more thresholds

- Solve bit allocation as a binary integer linear program (BILP):

$$\begin{aligned} \max \quad & \|\mathbf{F} \circ \mathbf{Z}\| \\ \text{subject to} \quad & \|\mathbf{Z}_h\| = 1 \qquad h \in \{1 \ldots B\} \\ & \|\mathbf{Z} \circ \mathbf{D}\| \leq B \\ & \mathbf{Z} \quad \text{is binary} \end{aligned}$$

$$\mathbf{F} = \begin{array}{c} \\ b_0 \\ b_1 \\ b_2 \end{array} \begin{array}{c} \begin{array}{cc} h_1 & h_2 \end{array} \\ \begin{pmatrix} 0.25 & 0.25 \\ 0.35 & 0.50 \\ 0.40 & 1.00 \end{pmatrix} \end{array} \quad \mathbf{D} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

- **F**: matrix of F_β scores. **Z**: bit allocation. **D**: constraint matrix. B: bit budget. $\|.\|$: Frobenius L_1 norm. $\circ$: Hadamard product

Evaluation protocol

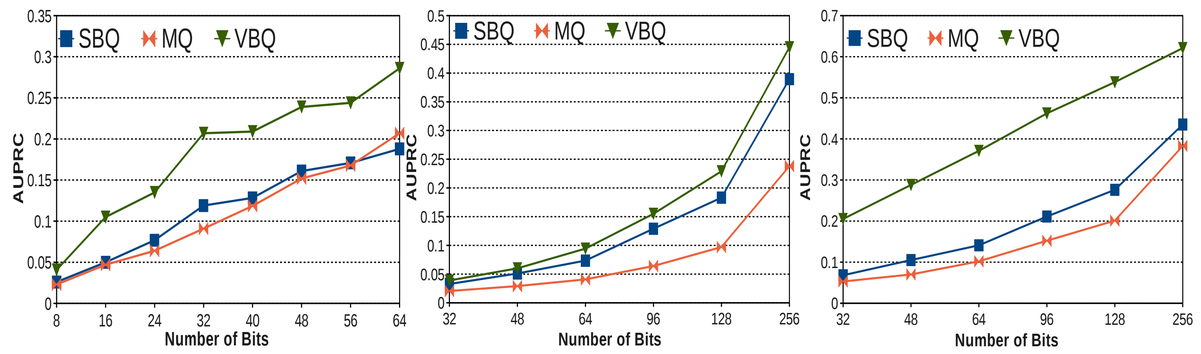
- **Task:** Retrieval on: *CIFAR-10*, *Reuters-21578* and *TDT-2* datasets.
- **Projections:** LSH, Shift-invariant kernel hashing (SIKH), Binary-LSI (BLSI), Spectral Hashing (SH) and PCA-Hashing (PCAH).
- **Baselines:** Single Bit Quantisation (SBQ), Manhattan Hashing (MQ) (Kong et al., '12), Double-Bit quantisation (DBQ) (Kong and Li, '12).
- **Hamming Ranking:** how well do we retrieve $\epsilon-$NN of queries? Quantify using area under the precision-recall curve (AUPRC).

Results

- AUPRC for projections at 32 bits (images) and 128 bits (text):

Dataset	CIFAR-10				TDT-2				Reuters			
	SBQ	MQ	DBQ	VBQ	SBQ	MQ	DBQ	VBQ	SBQ	MQ	DBQ	VBQ
SIKH	0.042	0.063	0.047	**0.161**	0.034	0.045	0.031	**0.092**	0.102	0.112	0.087	**0.389**
LSH	0.119	0.093	0.066	**0.207**	0.189	0.097	0.089	**0.229**	0.276	0.201	0.175	**0.538**
BLSI	0.038	0.135	0.111	**0.231**	0.283	0.210	0.087	**0.396**	0.100	0.030	0.030	**0.156**
SH	0.051	0.135	0.111	**0.202**	0.146	0.212	0.167	**0.370**	0.033	0.028	0.030	**0.154**
PCAH	0.036	0.137	0.107	**0.219**	0.281	0.208	0.094	**0.374**	0.095	0.034	0.027	**0.154**

- AUPRC vs. number of bits for CIFAR, TDT-2 and Reuters:



(a) CIFAR-10 **(b) TDT-2** **(c) Reuters**

Future Work

- Evaluation of VBQ in a hash lookup based retrieval scenario.