Variable Bit Quantisation for Large Scale Search

Sean Moran

Final year PhD student
Institute of Language, Cognition and Computation

12th September 2014

Variable Bit Quantisation for Large Scale Search

Variable Bit Quantisation for Large Scale Search

Nearest Neighbour Search

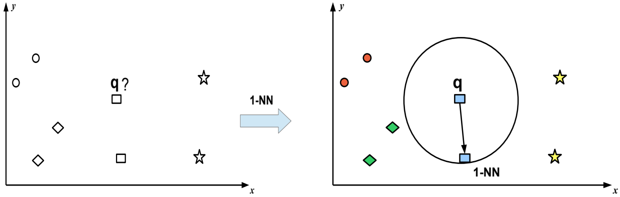
Variable Bit Quantisation for LSH

Evaluation

Summary

Nearest Neighbour Search

- Given a query $\mathbf{q}$ find the *nearest neighbour* NN($\mathbf{q}$) from $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_N\}$ where $NN(\mathbf{q}) = argmin_{\mathbf{x} \in \mathbf{X}} dist(\mathbf{q}, \mathbf{x})$
- dist($\mathbf{q}$,$\mathbf{x}$) is a *distance measure* - e.g. Euclidean, Cosine etc



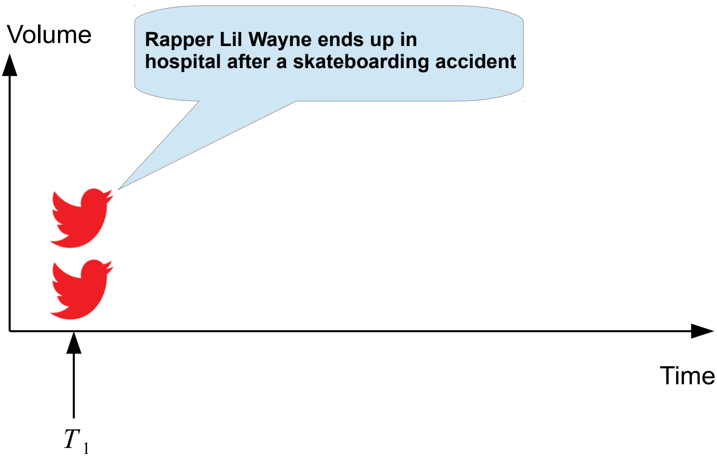
- Generalised variant: K-nearest neighbour search (KNN($\mathbf{q}$))
- Compare query to all N database items - $\mathcal{O}(N)$ query time

Example: First Story Detection in Twitter

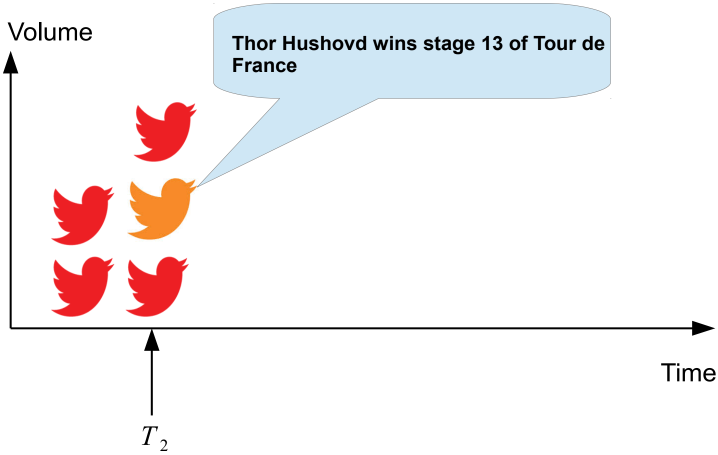
- Real-time detection of *first stories* in the Twitter stream
- Haiti earthquake struck **21:53**, first story at **22:17 UTC**
  - **22:17:43** *justinholtweb NOT expecting Tsunami on east coast after haiti earthquake. good news.*
- State-of-the-art FSD uses NN search under the bonnet
- Problems: dimensionality (1 million+) and data volume (250Gb/day)
- Hashing-based *approximate* NN operates in O(1) time [1]

[1] **Real-Time Detection, Tracking and Monitoring of Discovered Events in Social Media**. S. Moran et al. In *ACL'14*.

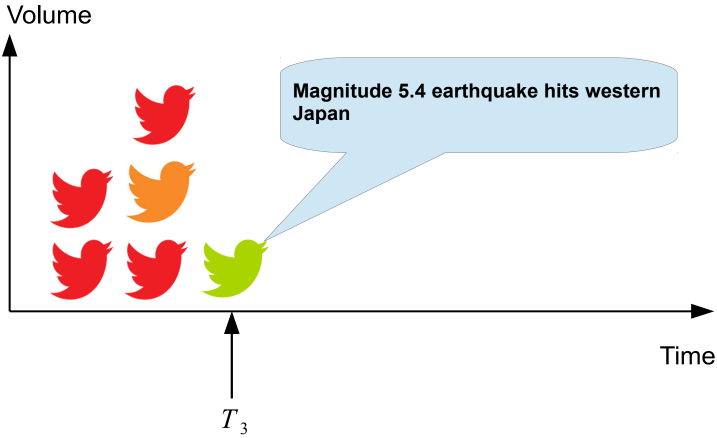
Example: First Story Detection in Twitter



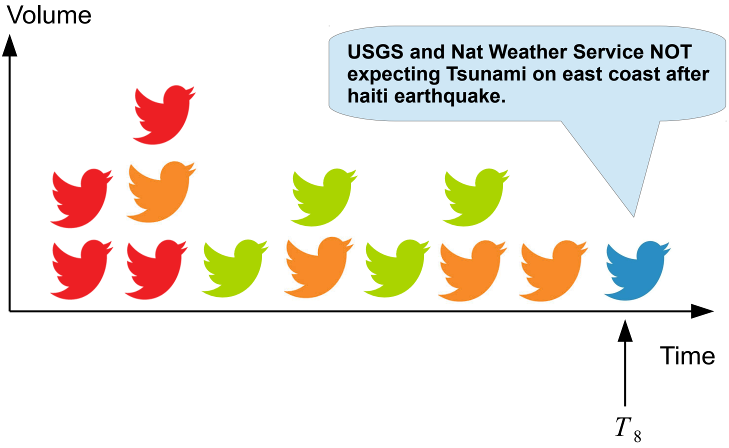
Example: First Story Detection in Twitter



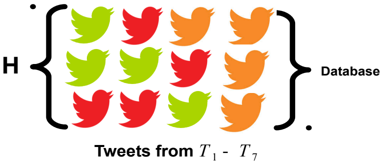
Example: First Story Detection in Twitter



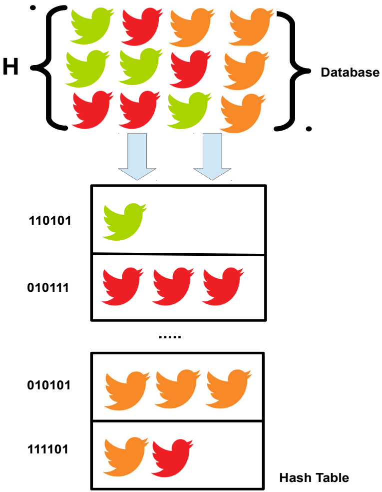
Example: First Story Detection in Twitter



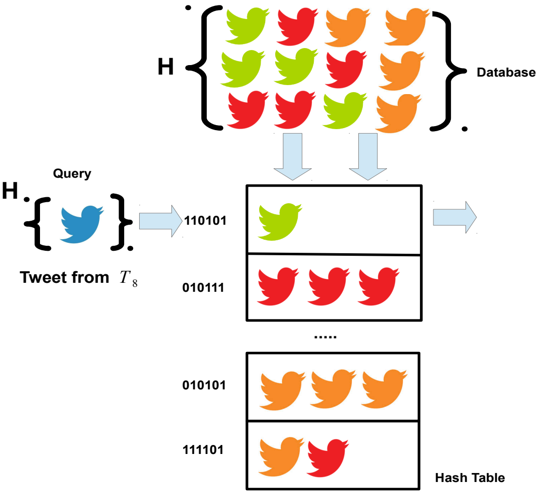
Hashing-based approximate NN search



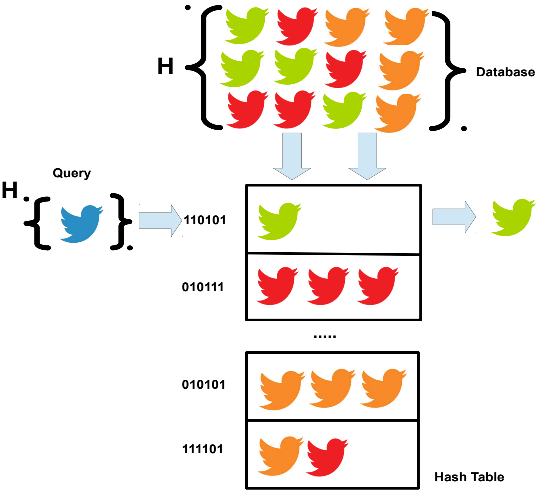
Hashing-based approximate NN search



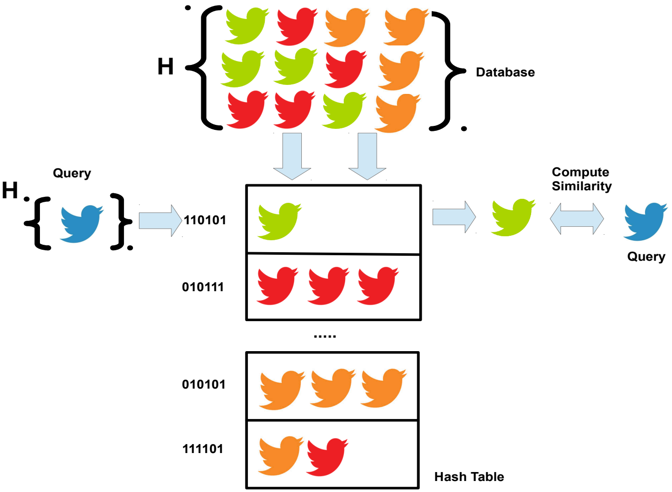
Hashing-based approximate NN search



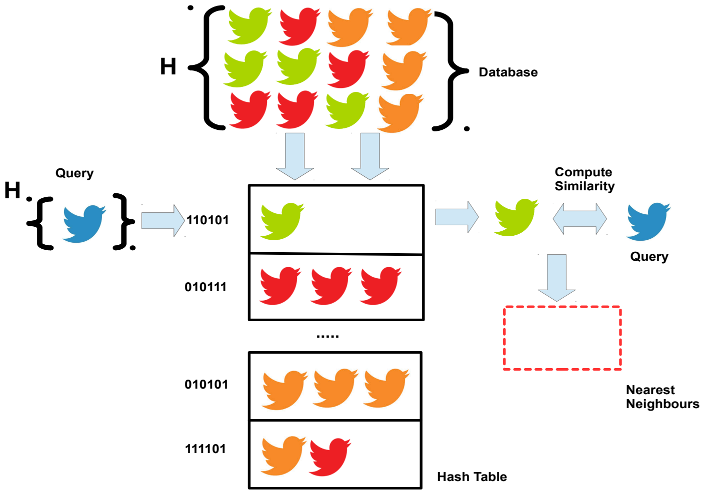
Hashing-based approximate NN search



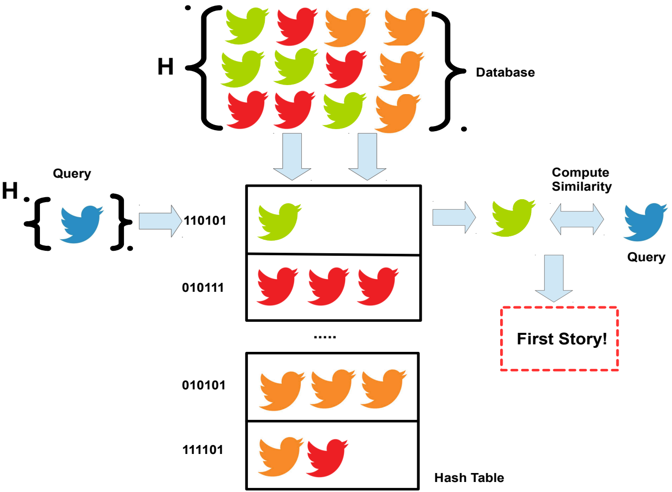
Hashing-based approximate NN search



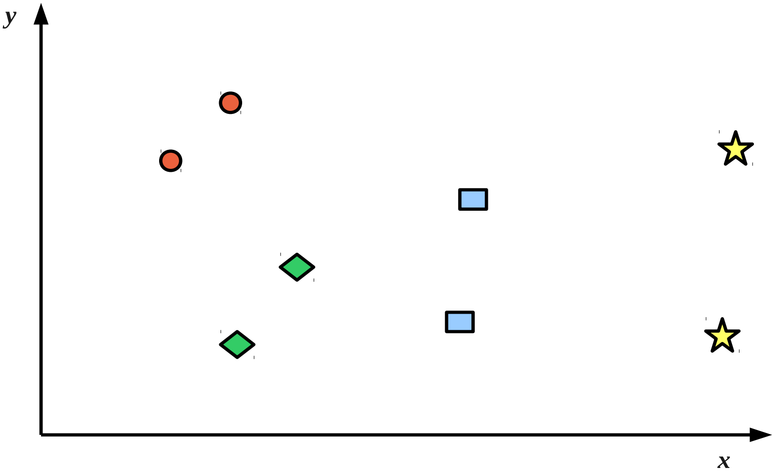
Hashing-based approximate NN search



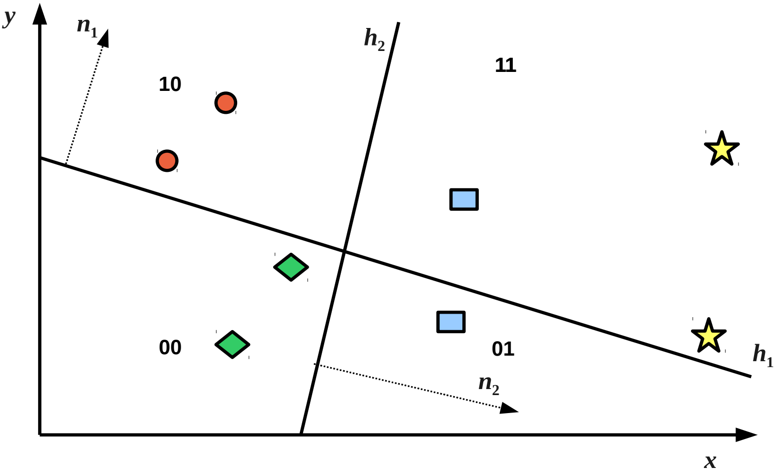
Hashing-based approximate NN search



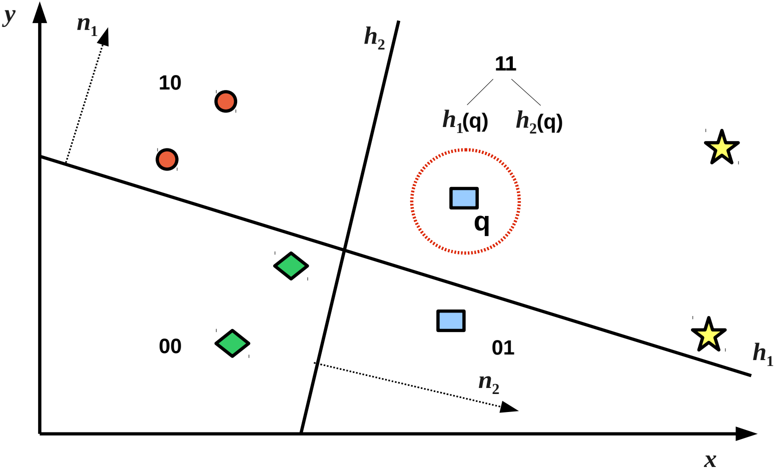
Locality Sensitive Hashing (LSH)



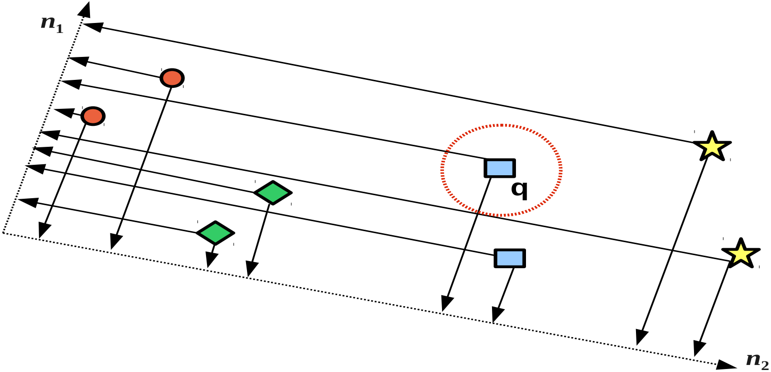
Locality Sensitive Hashing (LSH)



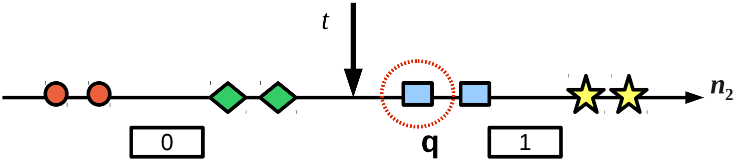
Locality Sensitive Hashing (LSH)



Step 1: Projection



Step 2: Single Bit Quantisation (SBQ)



- Threshold typically zero (sign function): $sgn(\mathbf{n}_2.\mathbf{q})$
- Generate full 2 bit hash key (bitcode) by concatenation:

$$\begin{aligned} g(\mathbf{q}) = h_1(\mathbf{q}) \oplus h_2(\mathbf{q}) &= sgn(\mathbf{n}_1.\mathbf{q}) \oplus sgn(\mathbf{n}_2.\mathbf{q}) \\ &= 1 \oplus 1 = 11 \end{aligned}$$

Many more methods exist...

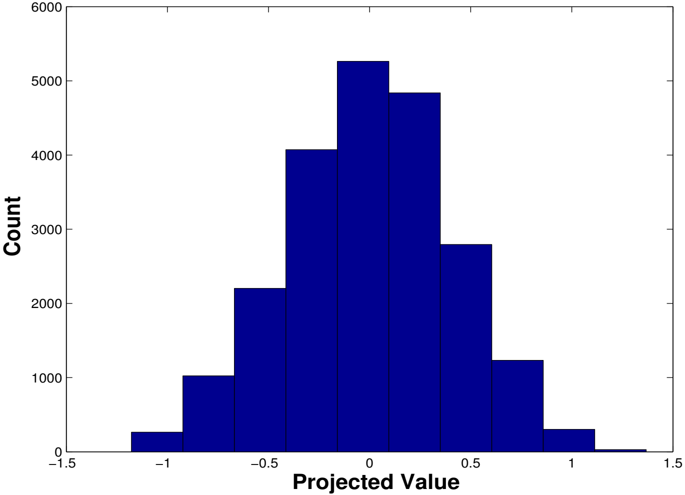
- Very active area of research:
  - Kernel methods [3]
  - Spectral methods [4] [5]
  - Neural networks [6]
  - Loss based methods [7]
- Commonality: all use single bit quantisation (SBQ)

[3] M. Raginsky and S. Lazebnik. *Locality-sensitive binary codes from shift-invariant kernels*. In NIPS '09.
[4] Y. Weiss and A. Torralba and R. Fergus. *Spectral Hashing*. NIPS '08.
[5] J. Wang and S. Kumar and SF. Chang. *Semi-supervised hashing for large-scale search*. PAMI '12.
[6] R. Salakhutdinov and G. Hinton. *Semantic Hashing*. NIPS '08.
[7] B. Kulis and T. Darrell. *Learning to Hash with Binary Reconstructive Embeddings*. NIPS '09.

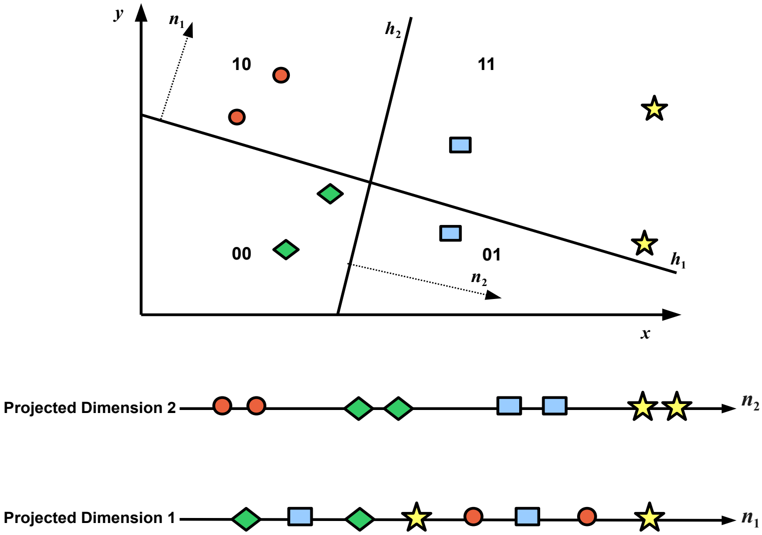
Variable Bit Quantisation for Large Scale Search

Problem 1: SBQ leads to high quantisation errors

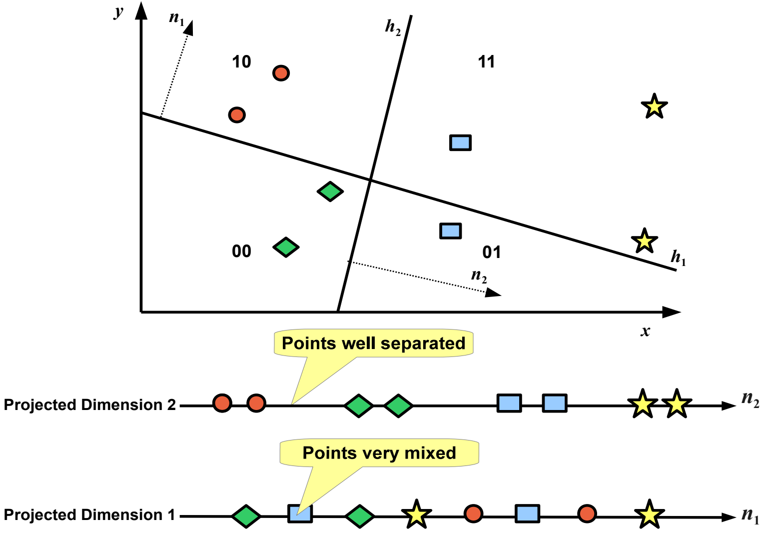
- Threshold at zero can separate many related Tweets:



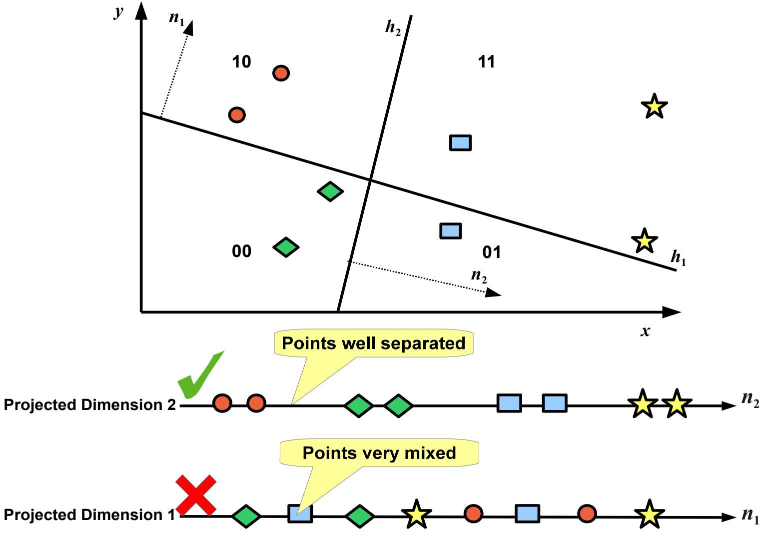
Problem 2: some hyperplanes are better than others



Problem 2: some hyperplanes are better than others

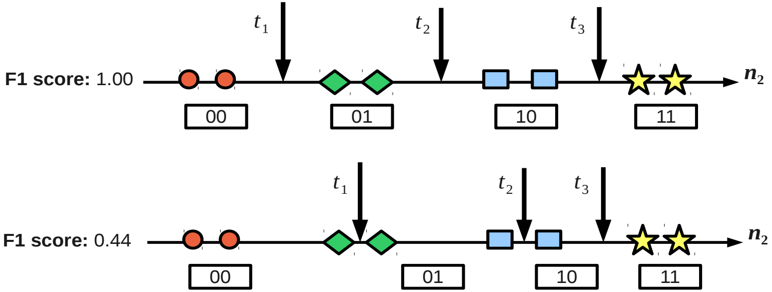


Problem 2: some hyperplanes are better than others



Threshold Positioning

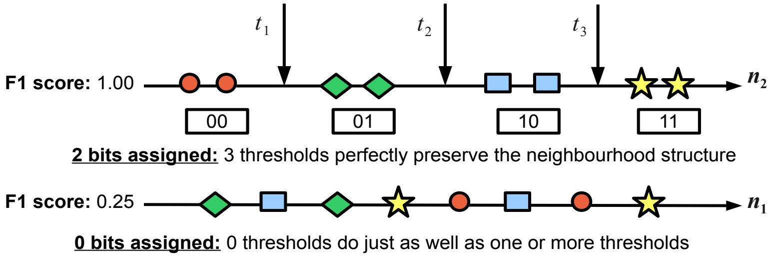
- Multiple bits per hyperplane requires multiple thresholds [8]
- *F*-score optimisation: maximise # related tweets falling inside the same thresholded regions:



[8] **Neighbourhood Preserving Quantisation for LSH**. S. Moran et al. In *SIGIR'13*

Variable Bit Allocation

- *F*-score is a measure of the *neighbourhood preservation* [9]:



- Compute bit allocation that maximises the *cumulative F*-score
- Bit allocation solved as a binary integer linear program (BILP)

[9] **Variable Bit Quantisation for LSH**. S. Moran et al. In *ACL'13*

Variable Bit Allocation

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

- $\mathbf{F}$ contains the F scores per hyperplane, per bit count
- $\mathbf{Z}$ is an indicator matrix specifying the bit allocation
- $\mathbf{D}$ is a constraint matrix
- B is the bit budget
- $\|.\|$ denotes the Frobenius L_1 norm
- $\circ$ the Hadamard product

[9] Variable Bit Quantisation for LSH. S. Moran et al. In *ACL'13*

Variable Bit Allocation

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

- $\mathbf{F}$ contains the F scores per hyperplane, per bit count
- $\mathbf{Z}$ is an indicator matrix specifying the bit allocation
- $\mathbf{D}$ is a constraint matrix
- B is the bit budget
- $\|.\|$ denotes the Frobenius L_1 norm
- $\circ$ the Hadamard product

[9] Variable Bit Quantisation for LSH. S. Moran et al. In *ACL'13*

Variable Bit Allocation

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

- $\mathbf{F}$ contains the F scores per hyperplane, per bit count
- $\mathbf{Z}$ is an indicator matrix specifying the bit allocation
- $\mathbf{D}$ is a constraint matrix
- B is the bit budget
- $\|.\|$ denotes the Frobenius L_1 norm
- $\circ$ the Hadamard product

[9] Variable Bit Quantisation for LSH. S. Moran et al. In *ACL'13*

Variable Bit Allocation

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

- $\mathbf{F}$ contains the F scores per hyperplane, per bit count
- $\mathbf{Z}$ is an indicator matrix specifying the bit allocation
- $\mathbf{D}$ is a constraint matrix
- B is the bit budget
- $\|.\|$ denotes the Frobenius L_1 norm
- $\circ$ the Hadamard product

[9] **Variable Bit Quantisation for LSH**. S. Moran et al. In *ACL'13*

Variable Bit Allocation

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

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

- **Sparse solution:** lower quality hyperplanes discarded [9].

[9] **Variable Bit Quantisation for LSH**. S. Moran et al. In *ACL'13*

Variable Bit Quantisation for Large Scale Search

Evaluation Protocol

- **Task:** Text and image retrieval
- **Projections:** LSH [2], Shift-invariant kernel hashing (SIKH) [3], Spectral Hashing (SH) [4] and PCA-Hashing (PCAH) [5].
- **Baselines:** Single Bit Quantisation (SBQ), Manhattan Hashing (MQ)[10], Double-Bit quantisation (DBQ) [11].
- **Evaluation:** how well do we retrieve the NN of queries?

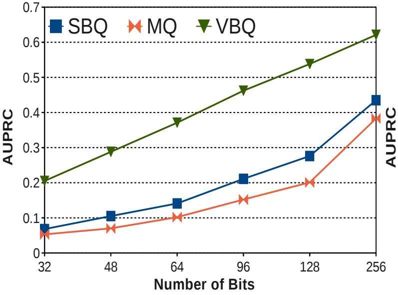
[2] P. Indyk and R. Motwani. *Approximate nearest neighbors: removing the curse of dimensionality*. In STOC '98.
[3] M. Raginsky and S. Lazebnik. *Locality-sensitive binary codes from shift-invariant kernels*. In NIPS '09.
[4] Y. Weiss and A. Torralba and R. Fergus. *Spectral Hashing*. NIPS '08.
[9] J. Wang and S. Kumar and SF. Chang. *Semi-supervised hashing for large-scale search*. PAMI '12.
[10] W. Kong and W. Li and M. Guo. *Manhattan hashing for large-scale image retrieval*. SIGIR '12.
[11] W. Kong and W. Li. *Double Bit Quantisation for Hashing*. AAAI '12.

AUPRC across different projections (variable # bits)

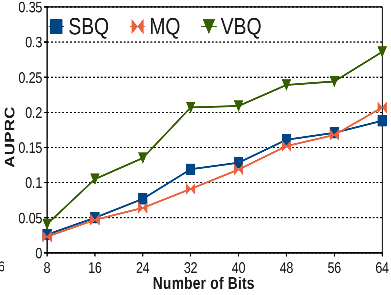
Dataset	Images (32 bits)				Text (128 bits)			
	SBQ	MQ	DBQ	**VBQ**	SBQ	MQ	DBQ	**VBQ**
SIKH	0.042	0.046	0.047	**0.161**	0.102	0.112	0.087	**0.389**
LSH	0.119	0.091	0.066	**0.207**	0.276	0.201	0.175	**0.538**
SH	0.051	0.144	0.111	**0.202**	0.033	0.028	0.030	**0.154**
PCAH	0.036	0.132	0.107	**0.219**	0.095	0.034	0.027	**0.154**

- Variable bit allocation yields substantial gains in retrieval accuracy
- VBQ is an effective *multimodal* quantisation scheme

AUPRC for LSH across a broad bit range



(a) Text



(b) Images

- VBQ is effective for both long and short bit codes (hash keys)

Variable Bit Quantisation for Large Scale Search

Nearest Neighbour Search

Variable Bit Quantisation for LSH

Evaluation

Summary

Summary

- Proposed a novel data-driven scheme (VBQ) to adaptively assign variable bits per LSH hyperplane
- Hyperplanes better preserving the neighbourhood structure are afforded more bits from budget
- VBQ substantially increased LSH retrieval performance across text and image datasets
- **Current work:** method to *couple* the quantisation and projection stages of LSH

Thank you for your attention!

- **FSD live system:** `goo.gl/Q7WQ0k` [1]
  - Running over live Twitter stream in real time
  - Sub-second detection latency via ANN search (1 CPU!)
- **Papers:** `www.seanjmoran.com` [1][8][9]
- **Contact:** `sean.moran@ed.ac.uk`

[1] **Real-Time Detection, Tracking and Monitoring of Discovered Events in Social Media**. S. Moran et al. In *ACL'14*.

[8] **Variable Bit Quantisation for LSH**. S. Moran et al. In *ACL'13*

[9] **Neighbourhood Preserving Quantisation for LSH**. S. Moran et al. In *SIGIR'13*