**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.

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

**Neighbourhood Preserving Quantisation (NPQ)**

- Assigns multiple bits per hyperplane using multiple thresholds.
- $F_\ell$ optimisation using pairwise constraints matrix $S$: if $S_{ij} = 1$ then points $x_i, x_j$ with projections $y_i, y_j$ 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 $F_\ell$:

$$Z_{npq} = \alpha F_1 + (1 - \alpha)(1 - \Omega(T_{1,u}))$$

where:

$$\Omega(T_{1,u}) = \frac{1}{\sigma} \sum_{r \in \mathcal{T}} \sum_{i=1}^{N} (y_i - \mu_{r_i})^2$$

where: $\sigma = \sum_{i=1}^{n} (y_i - \mu_{r_i})^2$, $\mu_{r_i}$ is dimension mean. $\mu_{r_i}$ is mean of region $r_i$.

- Random restarts used to optimise $Z_{npq}$. Time complexity $\sim O(N^2)$, where $N$ is # data points in training dataset.

**Evaluation Protocol**

- **Task:** Image retrieval on three image datasets: 22 LabelMe, CIFAR-10 and 100k TinyImages. Images encoded with GIST descriptors.
- **Projections:** LSH, Shift-invariant kernel hashing (SIKH), Iterative Quantisation (ITQ), 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 across different projection methods at 32 bits:**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LabelMe</th>
<th>CIFAR</th>
<th>TinyImages</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBQ</td>
<td>0.406</td>
<td>0.476</td>
<td>0.460</td>
</tr>
<tr>
<td>MQ</td>
<td>0.476</td>
<td>0.440</td>
<td>0.410</td>
</tr>
<tr>
<td>DBQ</td>
<td>0.406</td>
<td>0.476</td>
<td>0.460</td>
</tr>
<tr>
<td>NPQ</td>
<td>0.476</td>
<td>0.440</td>
<td>0.410</td>
</tr>
<tr>
<td>ITQ</td>
<td>0.227</td>
<td>0.354</td>
<td>0.308</td>
</tr>
<tr>
<td>SIKH</td>
<td>0.049</td>
<td>0.072</td>
<td>0.077</td>
</tr>
<tr>
<td>SH</td>
<td>0.156</td>
<td>0.138</td>
<td>0.123</td>
</tr>
<tr>
<td>PCAH</td>
<td>0.080</td>
<td>0.221</td>
<td>0.182</td>
</tr>
</tbody>
</table>

- NPQ can quantise a wide range of projection functions.
- NPQ + cheap projection (e.g. LSH) can outperform SBQ + expensive projection (e.g. PCA). NPQ is faster for $N < $ data dimensionality.
- AUPRC vs. Number of bits for LabelMe, CIFAR and TinyImages:

**Future Work**

- Variable bits per hyperplane: refer to our recent ACL’13 paper.
- Evaluation of NPQ in a hash lookup based retrieval scenario.