Background and Related Work

Nearest neighbor search for large databases
- Exhaustive nearest neighbor search not practical for large scale databases with high dimensional points
- Approximate nearest neighbor (ANN) search achieves sub-linear query time
- Popular ANN search techniques include tree-based approaches and hashing approaches

Related work
- Different choices of projections:
  - random projections
  - locality sensitive hashing (LSH, Indyk et al. 98)
  - shift invariant kernel based hashing (SIKH, Raginsky et al. 09)
- Principal projections
- Different forms of hash functions
  - identity function: LSH & Boosted SSC (RSSC, Shakhnarovich, 05)
  - sinusoidal function: SH & SIKH
- Other related work
  - Restricted boltzmann machines (RBMs, Hinton et al. 06)
  - Jun et al. 08 (metric learning)
  - Kulik et al. 09 & Mu et al. 10 (kernelized)

Main issues
- Existing hashing methods mostly rely on random or principal projections, which are not very compact or accurate
- Simple metrics are usually not enough to express semantic similarity

Semi-Supervised Hashing - Formulation

Setting and notations
\[ h_k(x) = \text{sgn}(w_k^T x + b_k), \quad b_k = \text{mean}(w_k^T x) \]

In the setting of SSH, one is given the data \( X \), and a subset \( X_\in \in \mathbb{R}^{d \times 1} \) with pair-wise labels:

\[
\begin{aligned}
(x_i, x_j) \in M & \quad \text{neighbor pair} \\
(x_i, x_j) \in C & \quad \text{nonneighbor-pair}
\end{aligned}
\]

Define the pairwise label matrix \( S \in \mathbb{R}^{d \times 1} \) as:

\[
S_{ij} = \begin{cases} 
1 & : (x_i, x_j) \in M \\
-1 & : (x_i, x_j) \in C \\
0 & : \text{otherwise}
\end{cases}
\]

Empirical Fitness

Variance as Regularizer

\[ p(y_k(x)) = \frac{1}{2} = p(h_k(x) = -1) \iff \text{max} \text{var}[h_k(x)] \]

Information theoretic term: maximum variance partition equals to balanced partition

\[
\text{max} \text{var}[h_k(x)] \geq \frac{1}{2} \text{var}[w_k^T x]
\]

Lower bound of maximum bit variance:

\[ w_k^T X \]

Unsupervised Sequential Projection Learning (USPLH) - Extension

Generate pseudo pair-wise label data from existing hash bits

\[ h_0(x) = -1, \quad h_1(x) = 1 \]

Use pseudo label matrix to learn projections

\[
S_{SC} : \begin{cases} 
(x_i, x_j) \in M & \Rightarrow S_{SC}(i, j) = 1 \\
(x_i, x_j) \in C & \Rightarrow S_{SC}(i, j) = -1
\end{cases}
\]

\[ M_z = \frac{1}{k} \sum_{i=0}^{k} X_{k+1} S_{SC} S_{SC}^T + \alpha XX^T \]

Pseudo label matrix from kth hash bit

\[ "\text{adjusted}\" \text{covariance matrix with pseudo labels} \]

Experiment and Results

Datasets
- MNIST Digits: 70K, 1K training labels, 1K query tests (semi-supervised)
- SIFT Feature: 1 Million, 2K pseudo labels, 10K query tests (unsupervised)

Experimental protocol
- Hash lookup (precision within hamming radius 2)
- Empty return treated as failed query with zero precision
- Hamming ranking (Mean Average Precision)
- Compared methods: LSH, SH, SIKH, RBMs, RSSC, PCAH

Performance evaluation and comparison

Computational evaluation

Conclusion:
- A semi-supervised paradigm for hashing learning (empirical fitness with information theoretic regularization)
- Sequential learning idea for error correction
- Extension of unsupervised case
- Easy implementation and highly scalable

Future work:
- Theoretical analysis of performance guarantee
- Weighted hamming embedding