Spectral Hashing: Learning to Leverage 80 Million Images

Yair Weiss, Antonio Torralba, Rob Fergus

Hebrew University, MIT, NYU
Outline

- Motivation: Brute Force Computer Vision.
- Semantic Hashing.
- Spectral Hashing.
Motivation

“Brute Force” Computer Vision using millions of labeled images (Torralba et al, Hays and Efros, S naveley et al)
Tiny Images dataset

- Query search engines with \( \approx 80K \) nouns in English.
- One thousand images each \( \Rightarrow 80 \text{ Million Images} \)
“Twin Jet”
“Mohammed”
“Killer Whale”
Brute Force Recognition?
Brute Force Recognition?
Why this won’t work

- Grandmother cell reborn.
- Similarity between images.
- Noisy Labels.
- Efficient search.
Some Inspiration

principle component analysis - Wikipedia, the free encyclopedia

Principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of...

A tutorial on Principal Components Analysis

Before getting to a description of PCA, this tutorial first introduces mathematical .... Finally we come to Principal Components Analysis (PCA) ....
Why this won’t work

- Grandmother cell reborn.
- Similarity between images.
- Noisy Labels.
- Efficient search.
Semantic Hashing

Short \((32\text{bit})\) codes. Hamming distance \(\approx\) semantic distance. (Salakhutdinov and Hinton, 2007)
Constructing Codes so that Hamming distance $\approx$ semantic distance.

- Deep Neural Network (Salakhutdinov and Hinton 07)
- Random Projections “LSH” (Andoni and Indyk 06)
- Boosting (Shakhnarovich et al. 03)
Deep Neural Network

Figure 2: Left panel: The deep generative model. Middle panel: Pretraining consists of learning a stack of RBM’s in which the feature activations of one RBM are treated as data by the next RBM. Right panel: After pretraining, the RBM’s are “unrolled” to create a multi-layer autoencoder that is fine-tuned by backpropagation.

(Salakhutdinov and Hinton 07)
Claim: If \( y_k \) are random linear thresholds, then Hamming distance monotonic with Euclidean distance asymptotically. (Andoni and Indyk 06)
Our Approach

- Optimization problem for best hashing code.
- NP Hard $\Rightarrow$ spectral relaxation
- Eigenvectors $\Rightarrow$ Eigenfunctions $\Rightarrow$ simple alg.
- State-of-the-art results.
Optimization

- Input: \( \{ x_i \} \in \text{semantic feature space} \)
  \( W_{ij} = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right) \)
- Output: \( y_i \in \{-1, 1\}^k \)
- Good Code: (1) Small Hamming Distance between neighbors (2) Bits fire 50% and independent.
Graph Partitioning

\[
\begin{align*}
\text{minimize} & : \sum_{ij} W_{ij} \| y_i - y_j \|^2 \\
\text{subject to} & : y_i \in \{-1, 1\}^k \\
& \sum_i y_i = 0 \\
& \frac{1}{N} \sum_i y_i y_i^T = I
\end{align*}
\]
**Graph Partitioning**

\[
\text{minimize : } \sum_{ij} W_{ij} \| y_i - y_j \|^2 \\
\text{subject to : } y_i \in \{-1, 1\}^k \\
\sum_i y_i = 0 \\
\frac{1}{N} \sum_i y_i y_i^T = I
\]

Observation: NP Hard even for one bit.
Graph Partitioning

\[
\begin{align*}
\text{minimize} & : \sum_{ij} W_{ij} \| y_i - y_j \|^2 \\
\text{subject to} & : y_i \in \{-1, 1\}^k \\
& \sum_i y_i = 0 \\
& \frac{1}{N} \sum_i y_i y_i^T = I
\end{align*}
\]

Relaxation \(\Rightarrow\) Smallest eigenvectors of graph Laplacian.
Out of Sample Extension

Nystrom ? Too expensive
Calculating Nystrom as expensive as exhaustive nearest neighbor.
Assume $x$ IID samples from $p(x)$.
Calculate limit of eigenvectors as number of points $\rightarrow \infty$.
(Coifman et al. 05, Belkin Niyogi 07, Bengio et al. 04, Nadler et al. 08).
Graph Partitioning

\[
\text{minimize :} \sum_{ij} W_{ij} \| y_i - y_j \|^2
\]

subject to:

\[
\sum_i y_i = 0
\]

\[
\frac{1}{N} \sum_i y_i y_i^T = I
\]

Relaxation \(\Rightarrow\) Smallest eigenvectors of graph Laplacian.
Out of Sample Extensions with Eigenfunctions

\[
\begin{align*}
\text{minimize} & : \\
\int \| y(x_1) - y(x_2) \|^2 W(x_1 - x_2) \quad p(x_1)p(x_2)dx_1x_2 \\
\text{subject to} & : y(x) \in \{-1, 1\}^k \\
\int y(x)p(x)dx & = 0 \\
\int y(x)y(x)^T p(x)dx & = I
\end{align*}
\]

Relaxation \(\Rightarrow\) Smallest eigenfunctions of Laplace-Beltrami.
Analytical Eigenfunctions for ND uniform

If each dimension is uniform $[a_i, b_i]$ then eigenfunctions are product of 1D sinusoids.

$$
\Phi_k(x) = \sin \left( \frac{\pi}{2} + \frac{k \pi}{b-a} x \right)
$$

$$
\lambda_k = 1 - e^{-\frac{\epsilon^2}{2}} \left| \frac{k \pi}{b-a} \right|^2
$$
Pairwise Independence too weak

- 3 Thresholded eigenfunctions can be deterministic functions
- Current solution: use only single-dimension eigenfunctions.
Experiments - Synthetic

Training samples

LSH

RBM (two hidden layers)

stumps boosting SSC

RBM

Proportion good neighbors for hamming distance < 2

number of bits

Spectral hashing

Boosting + spectral hashing

RBM+

spectral hashing

RBM

LSH

Spectral hashing

Boosting SSC

RBM (two hidden layers)

LSH

Spectral hashing

a) 2D uniform distribution

b) 10D uniform distribution

a) 3 bits

b) 7 bits

c) 15 bits
Approximate $p(x)$ with multidimensional rectangle. Works well despite bad assumption.
“Semantic Distance” $\approx$ Euclidean Distance in GIST descriptor.
LabelMe dataset

- Spectral hashing
- RBM
- Boosting SSC
- LSH
80 Million Image dataset

Gist neighbors

Spectral hashing: 32 bits

64 bits

Retrieval time: microseconds.
Limitations

- Three professors, no students.
- $p(x)$ uniform assumption.
- Higher order dependencies between bits.
- “Rounding” problem.
Why this won’t work

- Grandmother cell reborn.
- Similarity between images.
- Noisy Labels.
- Efficient search.
Conclusions

- Brute force computer vision using hundreds of millions of images.
- Hashing allows retrieval in microseconds.
- Spectral hashing - simple learning that outperforms the state-of-the-art.

Code Available: Google “spectral hashing”