Locality-Sensitive Codes from Shift-Invariant Kernels

Maxim Raginsky (Duke) and Svetlana Lazebnik (UNC)
Goal

- We want to design a binary encoding of data such that similar data points (similarity measures by a kernel) map to binary strings with low Hamming distance.

Related work

- Locality-sensitive hashing (Indyk et al., 1999 – present)
- Semantic hashing (Salakhutdinov and Hinton, 2007)
- Small codes and large databases (Torralba et al., 2008)
- Spectral hashing (Weiss et al., 2008)
- LSH for kernels (Kulis and Grauman, 2009)
Overview

- **Assume**
  - Data are initially embedded in $\mathbb{R}^D$, and a shift-invariant kernel $K(\cdot, \cdot)$ is defined on that space.

- **Approach**
  - Start with a randomized continuous mapping that is guaranteed to preserve kernel values with high probability (Rahimi and Recht's *random Fourier features*).
  - Binarize that mapping in a way that also preserves the kernel values.

- **Advantages**
  - Simple, data-independent mapping
  - Theoretical convergence guarantees
Random Fourier features

Let $K(x, y)$ be a Mercer kernel which is translation-invariant and normalized, i.e., $K(x, y) = K(x - y)$ and $K(0) = 1$. 
Random Fourier features

- Let $K(x, y)$ be a Mercer kernel which is translation-invariant and normalized, i.e., $K(x, y) = K(x - y)$ and $K(0) = 1$.

Then Bochner’s theorem says that $K$ is a Fourier transform of a uniquely defined probability measure $P_K$:

$$K(x - y) = \int_{\mathbb{R}^D} e^{i\omega \cdot (x-y)} dP_K(\omega).$$

- If $K$ is the Gaussian kernel $K(x - y) = \exp(-\gamma \|x - y\|^2/2)$, then $P_K = \mathcal{N}(0, \gamma \mathbf{I}_{D \times D})$. 
Let $K(x, y)$ be a Mercer kernel which is translation-invariant and normalized, i.e., $K(x, y) = K(x - y)$ and $K(0) = 1$.

Then Bochner’s theorem says that $K$ is a Fourier transform of a uniquely defined probability measure $P_K$:

$$K(x - y) = \int_{\mathbb{R}^D} e^{i\omega \cdot (x - y)} dP_K(\omega).$$

If $K$ is the Gaussian kernel $K(x - y) = \exp(-\gamma \|x - y\|^2 / 2)$, then $P_K = N(0, \gamma I_{D \times D})$.

Thus, we can write

$$K(x - y) = \mathbb{E}_{\omega \sim P_K} [e^{i\omega \cdot (x - y)}]$$

$$= \frac{1}{2} \mathbb{E}_{\omega \sim P_K} [\cos(\omega \cdot (x - y))]$$
From Bochner’s theorem, it follows that the kernel value $K(x - y)$ can be approximated in expectation using random Fourier features (Rahimi and Recht, 2007):

$$K(x - y) = \mathbb{E}_{\omega, b} [\Phi(x) \Phi(y)],$$

where $\Phi(x) = \sqrt{2} \cos (\omega \cdot x + b)$, $\omega \sim P_K$, and $b \sim \text{Unif}[0, 2\pi]$. 

Random Fourier features
Random Fourier features

- From Bochner’s theorem, it follows that the kernel value $K(x - y)$ can be approximated in expectation using random Fourier features (Rahimi and Recht, 2007):

$$K(x - y) = \mathbb{E}_{\omega, b}[\Phi(x)\Phi(y)], \quad \text{where } \Phi(x) = \sqrt{2} \cos(\omega \cdot x + b),$$

$\omega \sim P_K$, and $b \sim \text{Unif}[0, 2\pi]$.

- If we concatenate $n$ RFF’s into

$$\Phi^n(x) = (\Phi_1(x), \ldots, \Phi_n(x)) \in \mathbb{R}^n$$

with independently drawn $\omega$ and $b$ for each feature, then the *normalized* dot product $\Phi^n(x) \cdot \Phi^n(y)/n$ sharply concentrates around $K(x - y)$. 
The proposed scheme

Given: kernel $K$ and desired code length $n$.

Produce: a mapping

$$F^n : \mathbb{R}^D \rightarrow \{-1, +1\}^n, \quad \text{where } F^n(x) = (F_1(x), \ldots, F_n(x))$$

Obtain each bit of the code by composing a random Fourier feature with a random sign mapping:

$$F_i(x) = \text{sgn} \left[ \sqrt{2} \cos (\omega_i \cdot x + b_i) + t_i \right], \quad i = 1, \ldots, n$$

where $\omega_i \sim P_K$, $b_i \sim \text{Unif}[0, 2\pi]$, $t_i \sim \text{Unif}[-\sqrt{2}, \sqrt{2}]$, $i = 1, \ldots, n$, are i.i.d.
The expected normalized Hamming distance between two random sign codevectors $F^n(x), F^n(y)$:

$$
\frac{1}{n} \mathbb{E} \left[ d_H(F^n(x), F^n(y)) \right] \\
= \frac{1}{n} \sum_{i=1}^{n} 1\{F_i(x) \neq F_i(y)\} \\
= \frac{8}{\pi^2} \sum_{m=1}^{\infty} \frac{1 - K(mx - my)}{4m^2 - 1} h_K(x - y)
$$
Random Fourier features vs. random signs

\[ \Phi^n(x) \cdot \Phi^n(y) \]

\[ \frac{1}{n} F^n(x) \cdot F^n(y) \]

\[ K(x - y) \]

Kernel value vs. Euclidean distance
Bounding the normalized Hamming distance

Simple upper and lower bounds:

$$\frac{4}{\pi^2} \left( 1 - K(x - y) \right) \leq h_K(x - y)$$

$$\leq \min \left\{ \frac{1}{2} \sqrt{1 - K(x - y)}, \frac{4}{\pi^2} \left( 1 - \frac{2K(x - y)}{3} \right) \right\}$$
Locality-sensitive behavior: finite sets

**Theorem 1 ("Johnson–Lindenstrauss" result).** Fix $\varepsilon, \delta \in (0, 1)$. For any finite set $\{x_1, \ldots, x_N\} \subset \mathbb{R}^D$,\

$$\left| \frac{1}{n} d_H(F^m(x_j), F^m(x_k)) - h_K(x_j - x_k) \right| \leq \delta$$

hold for all $j, k$ with probability at least $1 - \varepsilon$ whenever $n \geq (1/2\delta^2) \log(N^2/\varepsilon)$. 
**Theorem 1 ("Johnson–Lindenstrauss" result).** Fix $\varepsilon, \delta \in (0, 1)$. For any finite set $\{x_1, \ldots, x_N\} \subset \mathbb{R}^D$, \[
\frac{1}{n}d_H(F^n(x_j), F^n(x_k)) - h_K(x_j - x_k) \leq \delta\]
hold for all $j, k$ with probability at least $1 - \varepsilon$ whenever $n \geq (1/2\delta^2) \log(N^2/\varepsilon)$.

Thus, any $N$-point subset of $\mathbb{R}^D$ can be embedded, with high probability, into the binary cube of dimension $\Omega(\log N)$ in a similarity-preserving way.
Theorem 2. Suppose that the distribution $P_K$ has a finite second moment:

$$L_K \triangleq \sqrt{\mathbb{E}\|\omega\|^2} < \infty.$$ 

Let $\mathcal{X} \subset \mathbb{R}^D$ be a compact domain with intrinsic (Assouad) dimension $d_{\mathcal{X}}$. Then there exists a constant $C > 0$ independent of $D$ and $K$, such that, for any $\varepsilon, \delta \in (0, 1)$,

$$\sup_{x, y \in \mathcal{X}} \left| \frac{1}{n} d_H(F^n(x), F^n(y)) - h_K(x - y) \right| \leq \delta$$

with probability at least $1 - \varepsilon$, provided

$$n \geq \max \left\{ \frac{C L_K d_{\mathcal{X}} \text{diam} \mathcal{X}}{\delta^2}, \frac{2}{\delta^2} \log \left( \frac{2}{\varepsilon} \right) \right\}.$$
**Theorem 2.** Suppose that the distribution $P_K$ has a finite second moment:

$$L_K \triangleq \sqrt{\mathbb{E}\|\omega\|^2} < \infty.$$

Let $X \subset \mathbb{R}^D$ be a compact domain with intrinsic (Assouad) dimension $d_X$. Then there exists a constant $C > 0$ independent of $D$ and $K$, such that, for any $\varepsilon, \delta \in (0, 1)$,

$$\sup_{x, y \in X} \left| \frac{1}{n} d_H(F^n(x), F^n(y)) - h_K(x - y) \right| \leq \delta$$

with probability at least $1 - \varepsilon$, provided

$$n \geq \max \left\{ \frac{C L_K d_X \text{diam}X}{\delta^2}, \frac{2}{\delta^2} \log \left( \frac{2}{\varepsilon} \right) \right\}.$$  

Thus, if $L_K \text{diam}X = \Theta(1)$, then $X$ can be embedded in a binary hypercube of dimension $\Omega(d_X)$ in a locality-preserving way.
Comparison to spectral hashing: synthetic results

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: synthetic results

Random Signs (this work)  
Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: synthetic results

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: synthetic results

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: synthetic results

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: synthetic results

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: synthetic results

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)

Synthetic data uniformly distributed in a 2D rectangle
Comparison to spectral hashing: real data

- Subset of LabelMe database: 13,000+ “training” images, 1,000 “test” images
- 320-dimensional GIST descriptors

Random Signs (this work)  Spectral Hashing (Weiss et al., 2008)
Query examples

Euclidean neighbors

32 bit code

Precision: 0.81
Query examples

Euclidean neighbors

512 bit code

Precision: 1.00
Query examples

Euclidean neighbors

32 bit code

Precision: 0.46
Query examples

Euclidean neighbors

512 bit code

Precision: 1.00
Query examples

Euclidean neighbors

32 bit code

Precision: 0.38
Query examples

Euclidean neighbors 512 bit code

Precision: 0.96
For more details, come see our poster T2 tonight.