IEEE DCASE 2017
Task 1: Acoustic Scene Classification
Using
Shift-Invariant Kernels and Random Features

Abelino Jimenez, Benjamin Elizalde, Bhiksha Raj
Carnegie Mellon University
Acoustic Scene Classification

Input

Acoustic Scene Classification System

Output

Office
Examples of acoustic scenes

- Bus - traveling by bus in the city (vehicle)
- Cafe / Restaurant - small cafe/restaurant (indoor)
- Car - driving or traveling as a passenger, in the city (vehicle)
- City center (outdoor)
- Forest path (outdoor)
- Grocery store - medium size grocery store (indoor)
- Home (indoor)
- Lakeside beach (outdoor)
- Library (indoor)
- Metro station (indoor)
- Office - multiple persons, typical work day (indoor)
- Residential area (outdoor)
- Train (traveling, vehicle)
- Tram (traveling, vehicle)
- Urban park (outdoor)
Outline

Introduction and Method

Experiments and Results

Ongoing work
Some state of the art approaches combine large-dimensional features and SVMs

(Geiger, 2013) Large-scale audio feature extraction and SVM for acoustic scene classification. 6,553 dim

(Metze, 2014) Improved Audio Features for Large-scale Multimedia Event Detection. 4,096 dim

(Rakotomamonjy, 2016) Enriched Supervised Feature Learning for Acoustic Scene Classification. 2,000 dim

(Zhang, 2017) Learning Audio Sequence Representations for Acoustic Event Classification. 6,373 dim

(Arandjelovic, 2017) Look, Listen and Learn. 6,144 dim
SVMs may employ nonlinear functions, but the computation complexity increases.

Kernel Trick

\[ \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \quad \forall i, j \]

Kernel matrix complexity:
Training $O(k^2n)$, Testing $O(kn)$

\[ \phi : \mathbb{R}^n \rightarrow \mathbb{R}^q \quad (n << q) \]

$k = \text{number of samples}$

$n = \text{dimensionality of features}$
Random Features

Consist of mapping the input/original features to a randomized lower-dimensional feature space.

Then, the RFs are passed to a linear SVM to approximate a nonlinear SVM.

(Rahimi, 2008) “Random features for large-scale kernel machines”
Function to compute Random Features

\[ \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \approx \langle \Phi_{RF}(\mathbf{x}_i), \Phi_{RF}(\mathbf{x}_j) \rangle \quad \forall i, j \]

\[ \Phi_{RF}(\mathbf{x}) = \sqrt{\frac{2}{M}} \cos (\mathbf{Wx + b}) \]

- **Input features**
- **Fixed value**
- **Kernel dependent**
  - \( \mathbf{W} : \text{MxN} \)
- **Uniform distribution**
  - \( \mathbf{b} : \text{Mx1} \)
Random Features for shift-invariant kernels

\[ K(x_1 + z, x_2 + z) = K(x_1, x_2) \]

**Gaussian**

\[ w_{ij} \sim \mathcal{N}(0, 2\gamma) \]

\[ K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||_2^2) \]

**Laplace**

\[ w_{ij} \sim \text{Cauchy}(0, \gamma) \]

\[ K(x_1, x_2) = \exp(-\gamma ||x_1 - x_2||_1) \]

**Cauchy**

\[ w_{ij} \sim \text{Laplace}(0, \gamma) \]

\[ K(x_1, x_2) = \prod_{i=1}^{N} \frac{1}{1 + \gamma^2 (x_{1i} - x_{2i})^2} \]
SVM prediction with Random Features

\[ f(x_{\text{test}}) = \sum_i \alpha_i y_i K(x_i, x_{\text{test}}) + \delta \]

keep \{\alpha_i, y_i, x_i, \delta\}

\[ f(x_{\text{test}}) = \omega^\top \Phi_{RF}(x_{\text{test}}) + \delta \]

keep \{\omega, \text{seed}, \delta\}
Outline

Introduction and Method

Experiments and Results

Ongoing work
Compare performance with/without Random Features

- Compute input features
- Compute random features
- Train/Test linear SVM
- Acoustic Scene Classification
- Train/Test non-linear SVM
- Acoustic Scene Classification
Experimental setup

Audio: 3-5 minutes duration from 15 scenes (e.g. bus, park, library)

Input features: 6,553 dims, cepstral, spectral, energy related, voicing, functionals (Geiger, 2013)

Random Features: approximate Gaussian, Laplacian, Cauchy

Classifier: SVM with Gaussian, Laplacian, Cauchy

Metric: Accuracy
Nonlinear SVM outperformed MLP baseline

<table>
<thead>
<tr>
<th>Acoustic scene</th>
<th>Baseline</th>
<th>Gaussian Kernel</th>
<th>Laplacian Kernel</th>
<th>Cauchy Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beach</td>
<td>75.3 %</td>
<td>78.8 %</td>
<td>77.2 %</td>
<td>77.9 %</td>
</tr>
<tr>
<td>Bus</td>
<td>71.8 %</td>
<td>93.6 %</td>
<td>92.0 %</td>
<td>92.3 %</td>
</tr>
<tr>
<td>Cafe/Restaurant</td>
<td>57.7 %</td>
<td>76.9 %</td>
<td>82.7 %</td>
<td>78.5 %</td>
</tr>
<tr>
<td>Car</td>
<td>97.1 %</td>
<td>94.9 %</td>
<td>94.2 %</td>
<td>95.5 %</td>
</tr>
<tr>
<td>City center</td>
<td>90.7 %</td>
<td>91.0 %</td>
<td>92.3 %</td>
<td>89.4 %</td>
</tr>
<tr>
<td>Forest path</td>
<td>79.5 %</td>
<td>89.1 %</td>
<td>85.9 %</td>
<td>87.2 %</td>
</tr>
<tr>
<td>Grocery store</td>
<td>58.7 %</td>
<td>74.7 %</td>
<td>74.7 %</td>
<td>74.0 %</td>
</tr>
<tr>
<td>Home</td>
<td>68.6 %</td>
<td>66.3 %</td>
<td>67.3 %</td>
<td>66.3 %</td>
</tr>
<tr>
<td>Library</td>
<td>57.1 %</td>
<td>65.7 %</td>
<td>58.3 %</td>
<td>65.1 %</td>
</tr>
<tr>
<td>Metro station</td>
<td>91.7 %</td>
<td>82.7 %</td>
<td>83.7 %</td>
<td>83.3 %</td>
</tr>
<tr>
<td>Office</td>
<td>99.7 %</td>
<td>89.7 %</td>
<td>92.9 %</td>
<td>90.4 %</td>
</tr>
<tr>
<td>Park</td>
<td>70.2 %</td>
<td>65.1 %</td>
<td>61.5 %</td>
<td>60.9 %</td>
</tr>
<tr>
<td>Residential area</td>
<td>64.1 %</td>
<td>65.7 %</td>
<td>68.3 %</td>
<td>63.5 %</td>
</tr>
<tr>
<td>Train</td>
<td>58.0 %</td>
<td>57.7 %</td>
<td>65.7 %</td>
<td>61.9 %</td>
</tr>
<tr>
<td>Tram</td>
<td>81.7 %</td>
<td>82.7 %</td>
<td>84.3 %</td>
<td>81.7 %</td>
</tr>
<tr>
<td><strong>Overall</strong></td>
<td><strong>74.8 %</strong></td>
<td><strong>78.3 %</strong></td>
<td><strong>78.8 %</strong></td>
<td><strong>77.9 %</strong></td>
</tr>
</tbody>
</table>
For RFs, as $M$ increases, performance approximates nonlinear kernels.
For RFs, as M increases, performance approximates nonlinear kernels

<table>
<thead>
<tr>
<th>Dimensionality</th>
<th>Gaussian Kernel</th>
<th>Laplacian Kernel</th>
<th>Cauchy Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>6,553; $&gt;2^{12}$</td>
<td>78.3 %</td>
<td>78.8 %</td>
<td>77.9 %</td>
</tr>
</tbody>
</table>
For RFs, as $M$ increases, performance approximates nonlinear kernels.

<table>
<thead>
<tr>
<th>Dimensionality</th>
<th>Gaussian Kernel</th>
<th>Laplacian Kernel</th>
<th>Cauchy Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>6,553; $&gt;2^{12}$</td>
<td>78.3 %</td>
<td>78.8 %</td>
<td>77.9 %</td>
</tr>
<tr>
<td>4,096; $2^{12}$</td>
<td>77.2 %</td>
<td>75.8 %</td>
<td>76.9 %</td>
</tr>
</tbody>
</table>
Outline

Introduction and Method

Experiments and Results

Ongoing work
Discretize random features (Hashing) to change real-valued vectors into bits

\[ K(x, y) = \langle \phi(x), \phi(y) \rangle \approx \langle H(x), H(y) \rangle \]

Hashing Trick

Abelino Jimenez, Benjamin Elizalde, Bhiksha Raj, “Acoustic Scene Classification Using Discrete Random Hashing for Laplacian Kernel Machines”, in submission to ICASSP 2018
Discretize random features (Hashing) to change real-valued vectors into bits

\[ K(x, y) = \langle \phi(x), \phi(y) \rangle \approx \langle H(x), H(y) \rangle \]

Hashing Trick

\[ H_{A,U}(x) = \frac{1}{\sqrt{M}} h(Ax + U) \]

Abelino Jimenez, Benjamin Elizalde, Bhiksha Raj, “Acoustic Scene Classification Using Discrete Random Hashing for Laplacian Kernel Machines”, in submission to ICASSP 2018
Reduces representation up to six orders of magnitude

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th># of Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCASE Challenge</td>
<td>74.8%</td>
<td></td>
</tr>
<tr>
<td>Laplacian Kernel</td>
<td>78.6%</td>
<td>$&gt; 2^{18}$</td>
</tr>
<tr>
<td>Random features $M = 2^{12}$</td>
<td>75.8%</td>
<td>$2^{18}$</td>
</tr>
<tr>
<td>Hashing $M = 2^{12}$</td>
<td>75.2%</td>
<td>$2^{12}$</td>
</tr>
</tbody>
</table>
Summary

Random features with linear SVM approximates well nonlinear SVM.

RFs reduced dimensionality by 37% with minimal loss of performance.

Hashing can also reduce storage up to 6 orders of magnitude.

Allows bit-based operations (XOR for similarity)

Speeds up transmissions and processing (self-training, boosting)