Random features for large-scale kernel machines

Rahimi, Recht

(NIPS 2007)

October 23, 2012

Introduction

In kernel methods, learned functions take the form

$$f(x) = \sum_i \alpha_i k(x, x_i) = \sum_i \alpha_i \langle \phi(x), \phi(x_i) \rangle_{\mathcal{H}}$$

for training points x_i.

1. Advantage: can work with infinite feature spaces.
2. Disadvantage: need to store all the training points.

Introduction

In kernel methods, learned functions take the form

$$f(x) = \sum_i \alpha_i k(x, x_i) = \sum_i \alpha_i \langle \phi(x), \phi(x_i) \rangle_{\mathcal{H}}$$

for training points x_i.

1. Advantage: can work with infinite feature spaces.
2. Disadvantage: need to store all the training points.

Ways to get around this:

1. Throw points away (incomplete Cholesky, sparse methods,...)
2. This paper: finite random feature spaces

Method 1: Fourier space

Bochner's theorem: a **continous** kernel $k(\mathbf{x}-\mathbf{y})$ on $\Re^d$ is positive definite iff

$$k(\mathbf{x}-\mathbf{y}) = \int_{\Re^d} p(\omega) e^{i\omega^\top(\mathbf{x}-\mathbf{y})} d\omega$$

for a **probability measure** $p(\omega)$ (actually a finite non-negative Borel measure: prob. measure with appropriate normalization)

Method 1: Fourier space

Bochner's theorem: a **continous** kernel $k(\mathbf{x}-\mathbf{y})$ on $\Re^d$ is positive definite iff

$$k(\mathbf{x}-\mathbf{y}) = \int_{\Re^d} p(\omega) e^{i\omega^\top(\mathbf{x}-\mathbf{y})} d\omega$$

for a **probability measure** $p(\omega)$ (actually a finite non-negative Borel measure: prob. measure with appropriate normalization)

Define $\zeta_\omega := e^{i\omega^\top \mathbf{x}}$. Then

$$\begin{aligned} k(\mathbf{x}-\mathbf{y}) &= \mathbb{E}_\omega\left[\left(e^{i\omega^\top \mathbf{x}}\right)\left(e^{i\omega^\top \mathbf{y}}\right)^*\right] \\ &= \mathbb{E}_\omega(\cos(\omega^\top(\mathbf{x}-\mathbf{y}))) + i\underbrace{\mathbb{E}_\omega(\sin(\omega^\top(\mathbf{x}-\mathbf{y})))}_{=0}. \end{aligned}$$

Method 1: Fourier space

Because $k(\mathbf{x}-\mathbf{y})$ is real and $p(\omega)$ is real, can replace this with cosine features:

$$z_{\omega,b}(\mathbf{x}) = \sqrt{2}\cos\left(\omega^\top \mathbf{x} + b\right)$$

where b uniform on $[0, 2\pi)$

Then

$$k(\mathbf{x}-\mathbf{y}) = \mathbb{E}_{\omega,b}\left[z_{\omega,b}(\mathbf{x}) z_{\omega,b}(\mathbf{y})\right]$$

Method 1: Fourier space

Because $k(\mathbf{x}-\mathbf{y})$ is real and $p(\omega)$ is real, can replace this with cosine features:

$$z_{\omega,b}(\mathbf{x}) = \sqrt{2}\cos\left(\omega^\top \mathbf{x} + b\right)$$

where b uniform on $[0, 2\pi)$

Then

$$k(\mathbf{x}-\mathbf{y}) = \mathbb{E}_{\omega,b}\left[z_{\omega,b}(\mathbf{x}) z_{\omega,b}(\mathbf{y})\right]$$

Proof:

$$2\cos(\omega^\top \mathbf{x} + b)\cos(\omega^\top \mathbf{y} + b) = \underbrace{\cos(\omega^\top(\mathbf{x}+\mathbf{y}) + 2b)}_{\text{expectation zero}} + \cos(\omega^\top(\mathbf{x}-\mathbf{y}))$$

Method 1: Fourier space

Generate D random features to decrease variance. Then

$$k(\mathbf{x}-\mathbf{y}) \approx \frac{1}{D}\sum_{j=1}^{D} z_{\omega,b}^{(j)}(\mathbf{x}) z_{\omega,b}^{(j)}(\mathbf{y}).$$

Method 1: Fourier space

Generate D random features to decrease variance. Then

$$k(\mathbf{x}-\mathbf{y}) \approx \frac{1}{D}\sum_{j=1}^{D} z^{(j)}_{\omega,b}(\mathbf{x})\, z^{(j)}_{\omega,b}(\mathbf{y}).$$

Convergence result:

Claim 1 (Uniform convergence of Fourier features)**.** *Let $\mathcal{M}$ be a compact subs eter* $\operatorname{diam}(\mathcal{M})$*. Then, for the mapping* $\mathbf{z}$ *defined in Algorithm 1, we have*

$$\Pr\left[\sup_{x,y\in\mathcal{M}} |\mathbf{z}(\mathbf{x})'\mathbf{z}(\mathbf{y}) - k(\mathbf{x},\mathbf{y})| \geq \epsilon\right] \leq 2^8 \left(\frac{\sigma_p \operatorname{diam}(\mathcal{M})}{\epsilon}\right)^2 \exp\left(-\right.$$

where $\sigma_p^2 \equiv E_p[\omega'\omega]$ *is the second moment of the Fourier transfo ther,* $\sup_{x,y\in\mathcal{M}} |\mathbf{z}(\mathbf{x})'\mathbf{z}(\mathbf{y}) - k(\mathbf{y},\mathbf{x})| \leq \epsilon$ *with any constant probab* $\Omega\left(\frac{d}{\epsilon^2}\log\frac{\sigma_p \operatorname{diam}(\mathcal{M})}{\epsilon}\right)$.

Method 2: randomly shifted grid

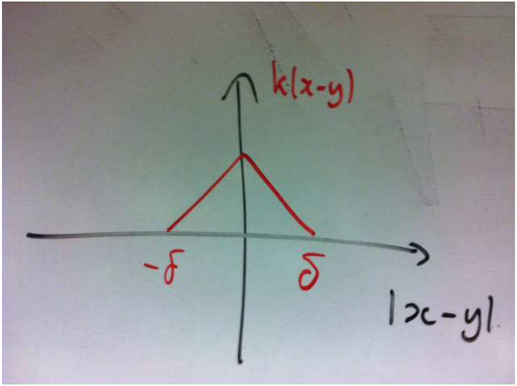

Figure: Kernel $k_{\mathrm{hat}}(x-y) = \max\left(0, 1 - \frac{|x-y|}{\delta}\right)$.

Method 2: randomly shifted grid

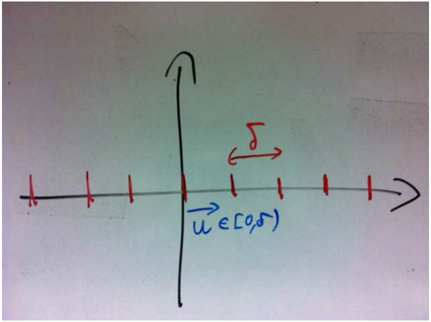

Figure: Randomly shifted grid. $u \sim \mathcal{U}(0, \delta)$.

Probability of x, y falling in the same bin:

$$\Pr_u(\hat{x} = \hat{y} \mid \delta) = k_{\text{hat}}(x - y) \qquad \hat{x} = \left\lfloor \frac{x - u}{\delta} \right\rfloor .$$

Method 2: randomly shifted grid

As before, take distributions over features to get more advanced kernels:

$$k(x, y) = \int_0^\infty k_{\mathrm{hat}}(x, y; \delta) p(\delta) d\delta.$$

Given a kernel, how to compute $p(\delta)$?

Method 2: randomly shifted grid

As before, take distributions over features to get more advanced kernels:

$$k(x, y) = \int_0^\infty k_{\mathrm{hat}}(x, y; \delta) p(\delta) d\delta.$$

Given a kernel, how to compute $p(\delta)$?

$$\begin{aligned} k(|x - y|) &=: k(\Delta) \\ &= \int_0^\infty \max\left(0, 1 - \frac{\Delta}{\delta}\right) p(\delta) d\delta \\ &= \int_\Delta^\infty p(\delta) d\delta - \Delta \int_\Delta^\infty \frac{p(\delta)}{\delta} d\delta. \end{aligned}$$

Take 2nd derivative wrt Δ:

$$\frac{d^2 k}{d\Delta^2} = \frac{p(\Delta)}{\Delta} \quad \Longrightarrow \quad p(\Delta) = \Delta \frac{d^2 k}{d\Delta^2}$$

Method 2: randomly shifted grid

Example:

$$k_{\mathrm{lap}} = \exp\left(-\left|x - y\right|\right) = \exp\left(-\Delta\right)$$

then $p(\delta) = \delta \exp(-\delta)$ (Gamma distribution).
Note: for a Gaussian, $p(\delta)$ not a valid prob. density.

Method 2: randomly shifted grid

Example:

$$k_{\mathrm{lap}} = \exp\left(-\left|x-y\right|\right) = \exp\left(-\Delta\right)$$

then $p(\delta) = \delta \exp(-\delta)$ (Gamma distribution).
Note: for a Gaussian, $p(\delta)$ not a valid prob. density.

Reduce variance by averaging over P independent grids (u, δ).

Method 2: randomly shifted grid

Example:

$$k_{\rm lap} = \exp\left(-\left|x - y\right|\right) = \exp\left(-\Delta\right)$$

then $p(\delta) = \delta \exp(-\delta)$ (Gamma distribution).
Note: for a Gaussian, $p(\delta)$ not a valid prob. density.

Reduce variance by averaging over P independent grids (u, δ).

Multiple dimensions: use independent grids in each dimension, and

$$k(\mathbf{x} - \mathbf{y}) = \prod_{k=1}^{m} k_m(x^m - y^m).$$

The feature is an m-dimensional binary tensor with a single one at coordinate $\left[\ \left\lfloor \frac{x_1 - u_1}{\delta_1} \right\rfloor \quad \ldots \quad \left\lfloor \frac{x_m - u_m}{\delta_m} \right\rfloor\ \right]$

Method 2: randomly shifted grid

In practice: use a hash of the binary vector as a feature map.

Convergence result:

Results

Interpretation: for data where interpolation is needed, use Fourier kernels. For data where "memorization" is needed, use binning features.
Caveat: the Gaussian kernel was used for Fourier+LS, the Laplace for Binning+LS