Table of contents

Master seminar: Solving localization problem in first person computer games with deep learning

Yauheni Selivonchyk[1]

[1]Student of Master Computer Science,
University of Bonn

April 24, 2017

Support vector machines

Support vector machines generalize well given sufficient amount of training data.
General SVM algorithm solves next empirical risk optimization problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \qquad \alpha_i > 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{1}$$

Where $k(x, y)$ is kernel function:

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle \tag{2}$$

Support vector machines. Continued

Whenever mapping $\Phi(x)$ of x into Hilbert space is explicitly known:

$$\Phi : \mathcal{X} \to \mathcal{V} \tag{3}$$

and $\mathcal{V}$ is of relatively low dimension D.
Training and evaluation time can then be reduced from $O(n^2)$ and $O(n)$ to $O(nD)$ and $O(D)$ respectively.

Can we find such a mapping for every function $k(x, y)$?

Random Fourier features

Solution provided by Rahimi and Rechts (2007) suggests relatively low-dimensional mapping for stationary kernels:

$$k_s(z) = k_s(x - y) = k(x, y) \tag{4}$$

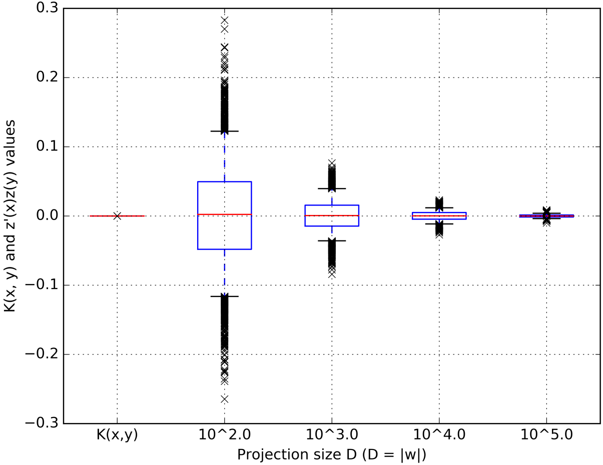
By Bochner theorem (1956) any positive semi-definite function as $k_s(z)$ can be expressed as:

$$k_s(x - y) = \int_{\mathbb{R}^d} p(\omega) cos \omega (x - y) d\omega \tag{5}$$

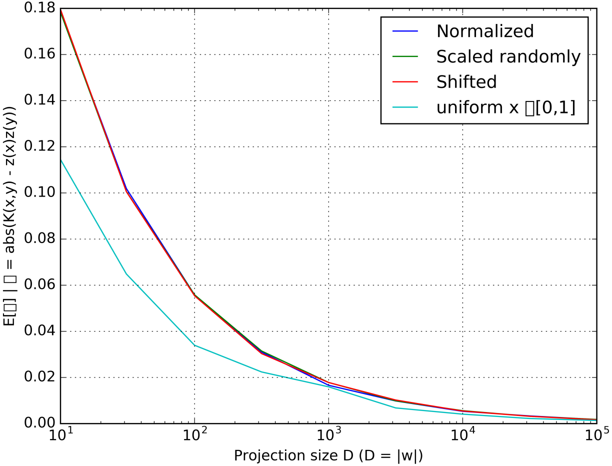
where $p(\omega)$ is a finite measure. Therefore we can estimate $k_s(z)$ as:

$$k_s(x - y) = E_\omega[cos \omega (x - y)] \approx z_\omega(x) z_\omega(y)^* \tag{6}$$

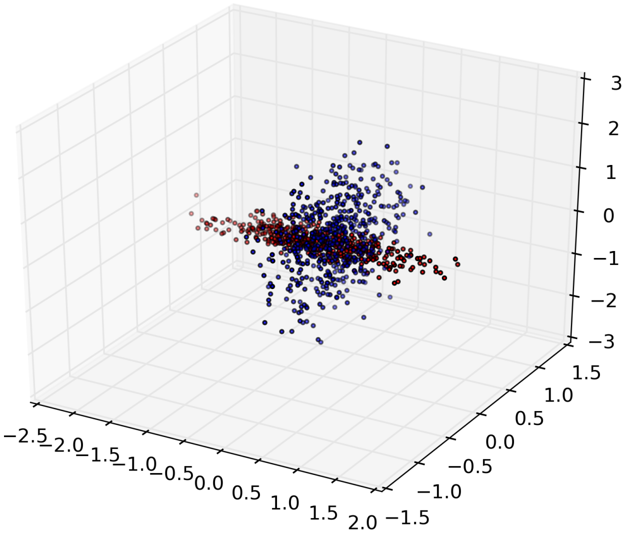
Kernel function approximation rate



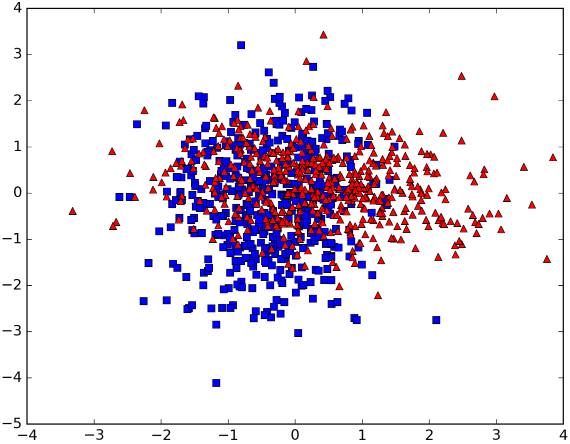
Qualities of input space



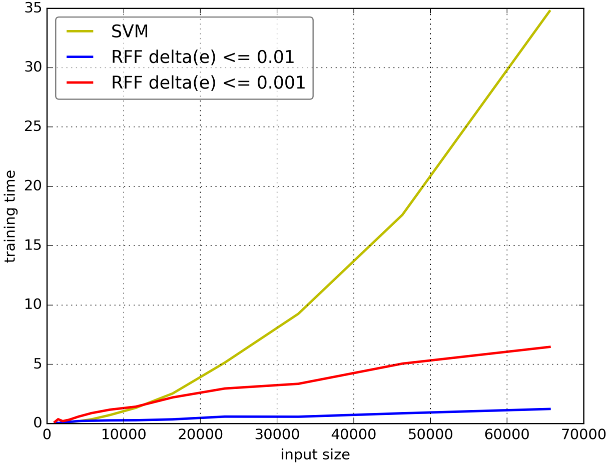
LinearSVM using RFF vs SVM. Input



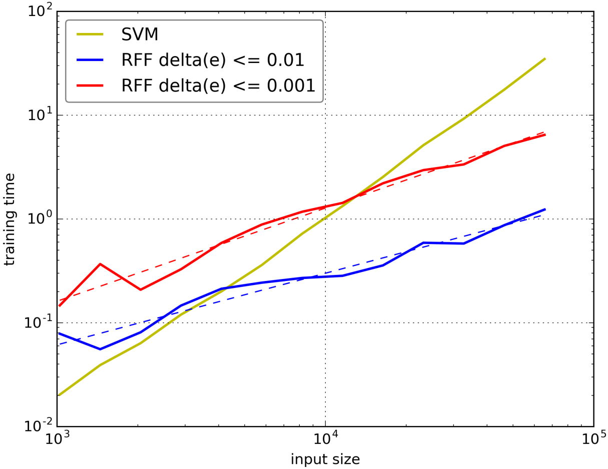
LinearSVM using RFF vs SVM. Input



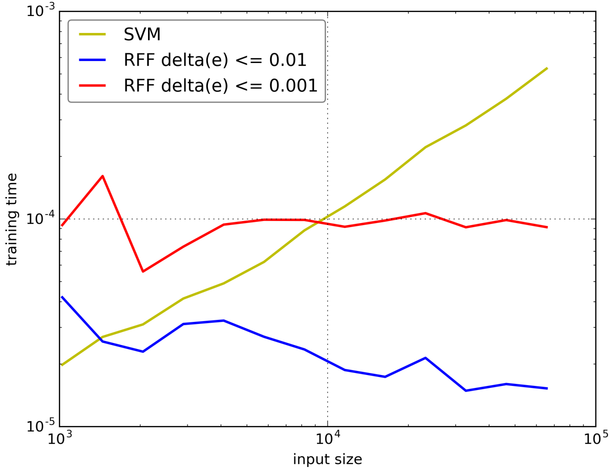
LinearSVM using RFF vs SVM. Running time



LinearSVM using RFF vs SVM. Running time

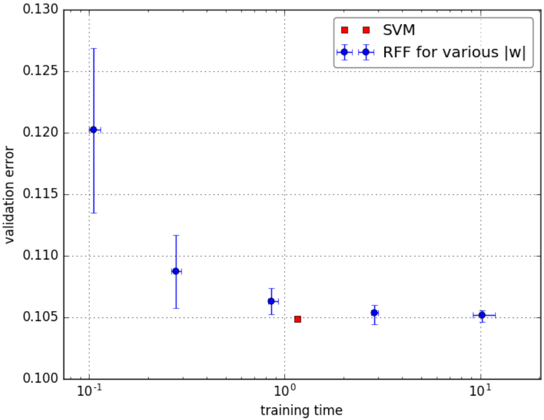


LinearSVM using RFF vs SVM. Evaluation time



Time/Accuracy tradeoff of SVM approximation

D=[100, 320, 1000, 3200, 10000], N=10000



Conclusion

- Experimental data confirms theoretical finding by Rahimi and Recht. Some inputs show even greater approximation speed than stated in the paper
- Approximating linearSVM maintains linear complexity in input size for a fixed approximation error
- General SVM algorithm has superior accuracy and it should be chosen when possible

SVM

Primal form:

$$\min_{\omega,b} \sum_{i=1}^{n} L(x_i, y_i) + \lambda ||w||^2 \tag{7}$$

$$L(x,t) \text{ - loss funciton, and } \lambda\epsilon(0,\infty) \tag{8}$$

Dual form:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{9}$$

$$\alpha_i > 0 \text{ for } \forall i \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{10}$$

RFF

$$
\begin{aligned}
&k_s(x-y) = \int_{\mathbb{R}^d} p(\omega) e^{j\omega^T(x-y)} d\omega = \int_{\mathbb{R}^d} p(\omega) cos\omega(x-y) d\omega \\
&k_s(x-y) = E_\omega[\zeta_\omega(x)\zeta_\omega(y)^*], \text{ where } \omega \text{ are drawn from } p. \\
&k_s(x-y) = E_\omega[cos\omega(x-y)] \approx z_\omega(x) z_\omega(y)^* \\
&z_\omega(x) = \sqrt{2} cos(\omega' x + b), \text{ where } b\epsilon[0, 2\pi]
\end{aligned}
\tag{11}
$$