Federal State Autonomous Educational Institution
for the Higher Education
National Research University "Higher School of Economics"
Faculty of Mathematics

Minasian Levon Lermentovich

Impact of Randomized Features on Linear Models

Bachelor's thesis

Degree programme: bachelor's educational programme "Mathematics"

Scientific supervisor:
Evgeny Andreevich Sokolov

Moscow 2021

Kernel Trick

Consider the classification task with training set (X, Y). We will use an ordinary linear classifier

$$a(x) = \text{sgn}(w^T x + b)$$

Direct application of an SVM method leads to the following optimization problem:

$$\frac{1}{2} w^T w + C \cdot \sum_{i=1}^{N} \xi_i \to \min_{w, \xi_i} \tag{1}$$

$$\text{w.r.t.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \tag{2}$$

On practice almost always data is not linearly separable and therefore such a linear model will perform poorly.

There is an idea to treat our current data as projection of a set $\mathbb{X}$ of a higher dimensional vector space V on a subspace U of this space: $X = \text{proj}_U \mathbb{X}$. While the projection of $\mathbb{X}$ on U is not linearly separable, it may happen that $\mathbb{X}$ itself is perfectly divisible in V.

The problem here is that in order to define V s.t. the data $\mathbb{X}$ may be easily separated by a linear classifier we need to add hundreds of new non-linear feature axes to the original space. This will dramatically increase computational complexity of our optimization problem.

Definition 1.
Function $k : X \times X \to \mathbb{R}$ is called kernel if it is symmetric and positive-definite.

Theorem 1 (Mercer)**.**
Suppose that $k : X \times X \to \mathbb{R}$ is a kernel function.

Then there is an injective map $\varphi : \mathbb{R}^d \to V$ s.t.

$$k(x, y) = \langle \varphi(x), \varphi(y) \rangle$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product in the vector space V (possibly infinite-dimensional).

It can be shown that our optimization problem (1) could be converted into this one???:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_j \alpha_k y_j y_k \cdot x_j^T x_k \to \max_{\alpha} \tag{3}$$

$$\text{w.r.t.} \quad 0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i y_i \tag{4}$$

which as you can see depends only on the dot products $x_j^T x_k$ between the objects from sample.

It follows from the above theorem that by defining an arbitrary kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ we implicitly map our data to a higher dimensional vector space with $\varphi : X \to V$, $\mathbb{X} = \varphi(X)$

We can use this fact to deal with the optimization problem (3) in slightly distinct manner.

Let's write down (3) formulated for objects of $\mathbb{X}$:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_j \alpha_k y_j y_k \cdot \langle \varphi(x_j), \varphi(x_k) \rangle \to \max_{\alpha}$$

which is obviously the same as

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_j \alpha_k y_j y_k \cdot k(x_j, x_k) \to \max_{\alpha}$$

since $\langle \varphi(x), \varphi(y) \rangle = k(x, y)$

And here we actually solved our computational complexity problem: the matrix $K = (k(x_i, x_j))_{i,j=1}^{N}$ (which will be used for computations) is always $N \times N$ size independently of $\dim V$.

Randomized Features

Downsides of Kernel Models

While we managed to make our computational complexity independent of $\dim V$, we actually very heavily tied it with the number N of objects in sample. Thus, it will be hard to learn the classifier on large training sets.

Ordinary linear inference methods works well with large datasets. It would be nice if we were able to map our data into *lower dimensional* space so that the inner product in that space approximate the kernel value. Thus, we would use both efficiency of ordinary inference methods and power of implicitly defined by kernel features.

It turns out, that we actually can define such a map.

Theorem 2 (Bochner)**.**
Continuous function $f : \mathbb{R}^d \to \mathbb{R}$ is positive-definite iff there is a finite non-negative Borel measure μ on $\mathbb{R}^d$ such that

$$f(x) = \int_{\mathbb{R}^d} \exp(i\omega^T x) d\mu(\omega)$$

i.e. $f(x)$ is an inverse Fourier transform of a non-negative measure μ.

Definition 2.
Kernel $k(x, y)$ is called shift-invariant if $k(x, y) = g(x - y)$ for some positive definite function $g : \mathbb{R}^d \to \mathbb{R}$.

We will write $k(x, y) = k(x - y)$ emphasizing that the function g in the definition above is related to the kernel $k(x, y)$.

It follows from the Bochner's theorem that if shift-invariant kernel $k(x-y)$ is normalized as needed, then it's Fourier transform $p(\omega)$ represents a PDF(Probability Density Function).

Thus, denoting with $p(w)$ a Fourier transform of $k(x - y)$, we get:

$$k(x - y) = \int_{\mathbb{R}^d} p(\omega) \exp(i\omega^T (x - y)) d\omega = \mathbb{E}_\omega \Big[\exp(iw^T (x - y)) \Big] \qquad (5)$$

and we can conclude that $\exp(iw^T(x - y))$ is a reasonable approximation of $k(x, y)$ when $\omega \sim p(\omega)$.

Since $p(\omega)$ and $k(x - y)$ are real-valued we can conclude that the imaginary part of an integral (5) is 0 and therefore we can replace $\exp(-i\omega^T(x - y))$ with correspondent cosines: $\exp(i\omega^T(x - y)) = cos(\omega^T(x - y))$.

Taking now $z(x) = \sqrt{2}/D \cdot \big[cos(\omega_1^T x + b_1), \ldots, cos(\omega_D^T x + b_D)\big]$ where each $\omega_j \in \mathbb{R}^d$ was drawn from $p(w)$ and b_j from $\text{Uniform}(0, 2\pi)$ we get the desired map $z(x)$ into lower dimensional space $\mathbb{R}^D$ which will approximate the kernel since $\mathbb{E}\big[z(x)^T z(y)\big] = k(x, y)$

See the implementation here: https

References

[1] Evgeny Sokolov. Kernels in Machine Learning. *Machine Learning on CS HSE 2020*

[2] Ali Rahimi and Ben Recht. Random Features for Large-Scale Kernel Machines.

[3] Michael I. Jordan and Anat Capsi(scribe). Soft Margin SVM. *CS281B/Stat241B: Advanced Topics in Learning & Decision Making*