A note on kernelizing the smallest enclosing ball for machine learning

Frank Nielsen
Frank.Nielsen@acm.org

September 27, 2017

Abstract

This note describes how to kernelize Badoiu and Clarkson's algorithm [1] to compute approximations of the smallest enclosing balls in the feature space induced by a kernel.

This column is also available in pdf: filename `kernelCoreSetMEB.pdf`

1 Smallest enclosing ball and coresets

Let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be a finite point set. In $\mathbb{R}^d$, the Smallest Enclosing Ball (SEB) $\mathrm{SEB}(\mathcal{P})$ with radius $r(\mathrm{SEB}(\mathcal{P}))$ is fully determined by s points of $\mathcal{P}$ lying on the boundary sphere [15, 9], with $2 \leq s \leq d+1$ (assuming general position with no $d+2$ cospherical points). Computing efficiently the SEB in finite dimensional space has been thoroughly investigated in computational geometry [5].

A $(1+\epsilon)$-approximation of the SEB is a ball covering $\mathcal{P}$ with radius $(1+\epsilon)r(\mathrm{SEB}(\mathcal{P}))$. A simple iterative approximation algorithm [1] (BC algorithm) proceeds iteratively as follows: Set $c^{(0)} = p_1$ and update the current center as

$$c^{(i+1)} = \frac{i}{i+1}c^{(i)} + \frac{1}{i+1}f_i,$$

where f_i denotes the farthest point of $c^{(i)}$ in $\mathcal{P}$ (in case of ties choose any arbitrary farthest point). To get a $(1+\epsilon)$-approximation of the SEB, one needs to perform $\lceil \frac{1}{\epsilon^2} \rceil$ iterations [1], so that this simple heuristic yields a $O(\frac{dn}{\epsilon^2})$-time approximation algorithm. Moreover, this algorithm proves that there exist *coresets* [1, 10] of size $O(\frac{1}{\epsilon^2})$: That is, a subset $\mathcal{C} \subset \mathcal{P}$ such that $r(\mathrm{SEB}(\mathcal{C})) \leq (1+\epsilon)r(\mathrm{SEB}(\mathcal{P}))$ and $\mathcal{P} \subset (1+\epsilon)\mathrm{SEB}(\mathcal{C})$. Optimal coresets for balls are known to be of size $\lceil \frac{1}{\epsilon} \rceil$, see [2] and [8, 2] for more efficient coreset algorithms. Notice that the size of a coreset for the SEB is both independent of the dimension d and the number of source points n.

2 Smallest enclosing ball in feature space

In machine learning, one is interested in defining the data domain [13]. For example, this is useful for anomaly detection that can be performed by checking whether a new point belongs to the domain (inlier) or not (outlier). The Support Vector Data Description [13, 14] (SVDD) defines the domain of data by computing an enclosing ball in the feature space $\mathcal{F}$ induced by a given kernel $k(\cdot,\cdot)$ (e.g., polynomial or Gaussian kernels). The Support Vector Clustering [3] (SVC) further builds on the enclosing feature ball to retrieve clustering in the data space.

Let $k(\cdot,\cdot)$ be a kernel [12] so that $k(x,y) = \langle \Phi(x), \Phi(y) \rangle$ (kernel trick) for a feature map $\Phi(x) : \mathbb{R}^d \to \mathbb{R}^D$, where $\langle \cdot,\cdot \rangle$ denotes the inner product in the Hilbert feature space $\mathbb{F}$. Denote by $\mathcal{F} = \{\phi_1, \ldots, \phi_n\}$ the corresponding feature vectors in $\mathbb{F}$, with $\phi_i = \Phi(p_i)$. SVDD (and SVC) needs to compute $\mathrm{SEB}(\mathcal{F})$.

We can kernelize the BC algorithm [1] by maintaining an *implicit representation* of the *feature center* $\varphi = \sum_{i=1}^n \alpha_i \phi_i$ where $\alpha \in \Delta_n$ is a normalized unit positive weight vector (with Δ_n denoting the $(n-1)$-dimensional probability simplex). The distance between the feature center $\varphi = \sum_i \alpha_i \phi(p_i) = \sum_i \alpha_i \phi_i$ and a feature point $\phi(p)$ is calculated as follows:

$$\begin{aligned} \|\varphi - \phi(p)\|^2 &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \langle \phi_i, \phi_j \rangle - 2 \sum_{i=1}^n \alpha_i \langle \phi_i, \phi(p) \rangle + \langle \phi(p), \phi(p) \rangle, & (1) \\ &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(p_i, p_j) - 2 \sum_i \alpha_i k(p_i, p) + k(p,p). & (2) \end{aligned}$$

Therefore at iteration i, the farthest distance of the current center φ_i to a point of $\mathcal{P}$ in the feature space can be computed using the implicit feature center representation: $\max_{j \in [d]} \|\varphi_i - \phi(p_j)\|$. Denote by f_i the *index* of the farthest point in $\mathbb{F}$. Then we update the implicit representation of the feature center by updating the weight vector α as follows:

$$\alpha^{(i+1)} = \frac{i}{i+1} \alpha^{(i)} + \frac{1}{i+1} e_{f_i},$$

where e_l denotes the hot unit vector (all coordinates set to zero except the l-th one set to one).

Observe that at iteration i, at most $i+1$ coordinates of α are non-zero (sparse implicit representation), so that the maximum distance of the center to the point set $\mathcal{P}$ can be computed via Eq. 2 in $O(ni^2)$. Thus it follows that the kernelized BC algorithm costs overall $O(\frac{dn}{\epsilon^4})$-time. The proof of the approximation quality of the BC algorithm relies on the Pythagoras's theorem [6, 7] that holds in finite-dimensional Hilbert spaces. Although we used an implicit feature map Φ (e.g., Gaussian kernel feature map), we can approximate feature maps by finite-dimensional feature maps using the randomized Fourier feature maps $\tilde{\Phi}$ [11].

This note is closely related to the work [4] where the authors compute a feature SEB for each class of data (points having the same label), and perform classification using the Voronoi diagrams defined on the feature (approximated) circumcenters.

Notice that the choice of the kernel for SVDD/SVC is important since the feature SEB has at most $D+1$ support vectors (without outliers) in general position, where D is the dimension of the feature space. Thus for a polynomial kernel, the number of support vectors is bounded (and so is the number of clusters retrieved using SVC). Another byproduct of the kernelized BC algorithm is that it proves that the feature circumcenter is contained in the convex hull of the feature vectors (since α encodes a convex combination of the feature vectors).

3 Some kernel examples

The feature map of the polynomial kernel $k_P(x,y) = \left(\sum_{i=1}^d x_i y_i + c \right)^a$ (with $c \geq 0$) is finite-dimensional ($\mathbb{F} = \mathbb{R}^D$ with $D = \binom{d+a}{a}$). For $a = 2$, we get the explicit feature map:

$$\Phi_P(x) = \left(x_d^2, \ldots, x_1^2, \sqrt{2} x_d x_{d-1}, \ldots, \sqrt{2} x_d x_1, \sqrt{2} x_{d-1} x_{d-2}, \ldots, \sqrt{2} x_{d-1} x_1, \ldots, \sqrt{2} x_2 x_1, \sqrt{2c} x_d, \ldots, \sqrt{2c} x_1, c \right).$$

That is, a 2D polynomial kernel k_P induces a 6D feature map $\Phi_P(x)$.

The feature map $\Phi_G(x)$ of the Gaussian kernel $k_G(x,y) = \exp(-\gamma \|x-y\|^2)$ (Radial Basis Function, RBF) is infinite-dimensional ($D = \infty$):

$$\Phi_G(x) = \exp(-\gamma x^2) \left(\sqrt{\frac{(2\gamma)^i}{i!}} x^i \right)_{i \in \{0,1,\ldots\}}.$$

References

[1] Mihai Badoiu and Kenneth L Clarkson. Smaller core-sets for balls. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 801–802. Society for Industrial and Applied Mathematics, 2003.

[2] Mihai Bădoiu and Kenneth L Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14–22, 2008.

[3] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[4] Yaroslav Bulatov, Sachin Jambawalikar, Piyush Kumar, and Saurabh Sethia. Hand recognition using geometric classifiers. *Biometric Authentication*, pages 1–29, 2004.

[5] Bernd Gärtner. Fast and robust smallest enclosing balls. *Algorithms-ESA99*, pages 693–693, 1999.

[6] Richard V Kadison. The pythagorean theorem: I. the finite case. *Proceedings of the National Academy of Sciences*, 99(7):4178–4184, 2002.

[7] Richard V Kadison. The pythagorean theorem: II. the infinite discrete case. *Proceedings of the National Academy of Sciences*, 99(8):5217–5222, 2002.

[8] Piyush Kumar, Joseph SB Mitchell, E Alper Yildirim, and E Alper Yıldırım. Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. In *ALENEX), Lecture Notes Comput. Sci.* Citeseer, 2003.

[9] Frank Nielsen and Richard Nock. On the smallest enclosing information disk. *Information Processing Letters*, 105(3):93–97, 2008.

[10] Richard Nock and Frank Nielsen. Fitting the smallest enclosing Bregman ball. In *ECML*, pages 649–656. Springer, 2005.

[11] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[12] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[13] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11):1191–1199, 1999.

[14] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[15] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). *New results and new trends in computer science*, pages 359–370, 1991.