Sketching Techniques Comparison

This file provides the comparison of four sketching techniques investigated in the article *Metric Embedding into the Hamming space with the n-Simplex Projection* [8]. Please see the second page. The sketching techniques transform an arbitrary given metric space to the Hamming space that approximates the distances in the original space. The bit-string *sketches* are then exploited to speed-up the searching for close objects in the original metric space. We considered the following sketching techniques:

GHP_50 technique [7] uses λ pairs of reference objects (*pivots*), that define λ instances of the *Generalized Hyperplane Partitioning* (GHP) [9] of the dataset S. Therefore, each GHP instance splits the dataset into two parts according to the closer pivot, and these parts define values of one bit of all sketches $sk(o), o \in S$. The pivots are selected to produce balanced and low correlated bits [7]: (1) an initial set of pivots $P_{sup} \in D$ is selected in random, (2) the balance of the GHP is evaluated for all pivot pairs using a sample set T of S, (3) set P_{bal} is formed by pivot pairs that divide T into parts balanced to at least 45 % to 55 %, and corresponding sketches sk_{bal} are created, (4) the correlation matrix M with absolute values of the Pearson correlation coefficient is evaluated for all pairs of bits of sketches sk_{bal}, and (5) a heuristic is applied to select rows and columns of M which form its sub-matrix with low values and size $\lambda \times \lambda$. (6) Finally, the λ pivot *pairs* that produce the corresponding low correlated bits define sketches $sk(o), o \in S$.

BP_50 uses the *Ball Partitioning* (BP) instead of the GHP [7]. BP uses one pivot and a radius to split data into two parts, that again define the values in one bit of sketches $sk(o), o \in S$. Pivots are selected again via a random set of pivots P_{sup}, for which we evaluate radii dividing the sample set T into halves. The same heuristic as in case of the technique GHP_50 is than employed to select λ pivots that produces low correlated bits.

tPCA_50 is a simple sketching technique surprisingly well approximating the Euclidean spaces [3, 5, 6, 4, 1]. It uses the *Principal Component Analysis* (*PCA*) to shrink the original vectors, which are then rotated using a random matrix and binarized by the thresholding. The i-th bit of sketch $sk(o)$ thus expresses whether the i-th value in the shortened vector is bigger then the median computed on a sample set T. If sketches longer than the original vectors are desired, we propose to apply the PCA and to rotate transformed vectors using independent random matrices. Then we concatenate corresponding binarized vectors.

tNSP_50 is a sketching technique that is applicable to all metric spaces with the n-point property. It uses the n-Simplex projection to transform the data objects into λ-dimensional vectors, which are then randomly rotated and binarized by thresholding. The main steps used for computing sketches of length λ are: (1) Random selection of λ pivots. (2) Computation of the base simplex. (3) n-Simplex projection of all the data objects. (4) Random rotation of the projected data. (5) Binarization by thresholding of each dimension using the median value evaluated on a sample set T of S.

The following table, compare sketching approaches in terms of the floating point operations *Flops* and number of distance computations required to learn the transformation, and to transform each object from the metric space to the Hamming space. The remaining parameters are the following. λ: the length of sketches; P_{sup}: the set of randomly selected pivots; P_{bal}: the set of all pivots pairs $(p_1, p_2), p_1, p_2 \in P_{sup}$ that produce balance bits, T: a sample set of the data.

Table 1: Sketching techniques comparison.

	BP_50	**GHP_50**	**tPCA_50**	**tNSP_50**
Applicability	Metric Space	Metric Space	Euclidean Vector Space	Metric Space with n-point property
Cost of learning transformation	(i) $\lvert P_{sup}\rvert \cdot \lvert T\rvert$ *distance computations* to get radii dividing sample set into halves; (ii) $O(\lvert P_{sup}\rvert\,\lambda + \lvert P_{sup}\rvert^2\ const)$ *flops* to select a subset of λ pivots producing low correlated bits	(i) $\lvert P_{sup}\rvert\,\lvert T\rvert$ *distance computations* to get pivot pairs that produce sketches with balanced bits; (ii) $O(\lvert P_{bal}\rvert\,\lambda + \lvert P_{bal}\rvert^2\ const)$ *flops* to select a subset of λ pivots producing low correlated bits	(i) $O(2\lvert T\rvert\,m^2 + 11\lvert T\rvert^3 + 2m\,\lvert T\rvert)$ *flops* to get the PCA matrix learned on a sample set; (ii) $O(\lambda^2 m)$ *flops* to multiply the PCA matrix by a random rotation matrix; (iii) $O(\lvert T\rvert m\lambda + \lambda\lvert T\rvert\ \log\lvert T\rvert)$ *flops* to compute median values of rotated shortened vector	(i) $\lambda(\lambda-1)/2$ *distance computations* between λ randomly selected pivots (ii) $O(\lambda^3)$ *flops* to get the vertices of the base simplex (iii) $\lambda\lvert T\rvert$ *dist. computation* + $O(\lvert T\rvert\lambda^2 + \lambda\lvert T\rvert\log\lvert T\rvert)$ *flops* to compute median values of rotated shortened vector for a sample set T
Cost of Object-to-sketch transformation	λ *distance computations*	2λ *distance computations*	$O(\lambda\,m)$ *flops*	λ *distance computations* + $O(\lambda^2)$ *flops*
Parameters Used in Experiments	$\lvert T\rvert = 20{,}000$; $\lvert P_{sup}\rvert = 512$; $const = 40{,}000$	$\lvert T\rvert = 20{,}000$; $\lvert P_{sup}\rvert = 512$; $const = 40{,}000$ $\lvert P_{bal}\rvert = 15{,}000$	$\lvert T\rvert = 35{,}000$	$\lvert T\rvert = 35{,}000$
The recall on SQFD dataset $\lambda = 192$; $c = 0.1\,\%$	0.65	0.81	*not applicable*	0.88
The recall on SIFT dataset $\lambda = 192$, $c = 0.2\,\%$; $m = 128$; $d = \ell_2$	0.58	0.69	0.79	0.77

⋆ The Singular Value Decomposition (SVD) over the centred data is considered to make the PCA matrix. Centring the vectors costs $2m\lvert T\rvert$ flops, SVD costs $O(2\lvert T\rvert m^2 + 11\lvert T\rvert^3)$ *flops*, assuming $\lvert T\rvert > m$ and the efficient R-SVD algorithm [2, p. 293].

References

[1] Cao, Y., Qi, H., Zhou, W., Kato, J., Li, K., Liu, X., Gui, J.: Binary hashing for approximate nearest neighbor search on big data:A survey. IEEE Access **6**, 2039–2054 (2018)

[2] Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische mathematik **14**(5), 403–420 (1970)

[3] Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2916–2929 (2013)

[4] Gordo, A., Perronnin, F., Gong, Y., Lazebnik, S.: Asymmetric distances for binary embeddings. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 33–47 (2014)

[5] Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of CVPR 2010. pp. 3304–3311. IEEE (2010)

[6] Mic, V., Novak, D., Vadicamo, L., Zezula, P.: Selecting sketches for similarity search. In: Proceedings of ADBIS 2018. pp. 127–141 (2018)

[7] Mic, V., Novak, D., Zezula, P.: Designing sketches for similarity filtering. In: Proceedings of IEEE ICDM Workshops. pp. 655–662 (Dec 2016)

[8] Vadicamo, L., Mic, V., Falchi, F., Zezula, P.: Metric embedding into the hamming space with the n-simplex projection. In: Proceedings of SISAP 2019 (2019)

[9] Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity search: the metric space approach, vol. 32. Springer Science & Business Media (2006)