ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Homogeneous bipartition based on multidimensional ranking

Michaël Aupetit

CEA-DAM - Département Analyse Surveillance, Environnement
Bruyères-le-Châtel, 91297 Arpajon Cedex - France

Abstract. We present an algorithm which partitions a data set in two parts with equal size and experimentally nearly the same distribution measured through the likelihood of a Parzen kernel density estimator. The generation of the partition takes $O(\frac{1}{2}N(N-1))$ operations (N number of data) and is 2 orders of magnitude faster than the state of the art.

1 Generating an equi-distributed bipartition

1.1 Problem and applications

We consider the problem of generating a **homogeneous partition** of a data set, that is a partition such that each part has the same distribution as the whole. We focus here on **bipartitions**, which are partitions containing two parts with equal size (the number N of data is even) (Figure 1a).

This problem has been studied under the name "data squashing" [2] to summarize massive data sets in a way which preserves statistical relationships among variables better than random sampling; in biostatistics to select patients for treatment and control groups [5]; and in Machine Learning as a way to improve the estimation of model complexity [7].

1.2 Inspiration from two-sample tests

To check whether two groups are drawn from the same distribution, one can use a **two-sample test**. Suppose a blue-red bipartition of the data. Several two-sample tests are based on building a **proximity graph** of the data[1] and counting the number of **mixed edges**, *i.e.* which have one red and one blue vertices. The higher this number, the higher the probability that both red and blue data are drawn from the same distribution. Then the null hypothesis that both parts are homogeneous is rejected if the number of mixed edges is too small. Friedman and Rafsky [3] proposed a test based on the Minimal Spanning tree. Recently, Rosenbaum proposed the "cross-matching" test [6], which uses a minimal distance non-bipartite matching [4].

The homogeneous bipartitionning problem can bring down to build a bipartition of the data which succeeds in passing a multivariate two-sample test. One way to do this is to generate several random bipartitions and keep the one

[1] A proximity graph connects two points if they are close to each other with respect to some measure of closeness. Example of such graphs are the Minimum Spanning tree, the Nearest Neighbor graph or the Delaunay graph.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

which succeeds in passing the test. This can take a long time and costs also much computation to perform the tests. We propose another way to solve the problem.

1.3 A solution based on Parzen estimators and likelihood

A way to measure how close are the distributions of both parts of a bipartition, is to build a generative model based on one of the parts, and to use this model to measure the likelihood of the other one. Then, the higher the likelihood that one part generated the other, the closer the probability density functions of both parts.

Consider a set $\underline{x}$ made of an even number N of points, one half in the blue set $\underline{x}_b \subset \underline{x}$ and the other half in the red set $\underline{x}_r = \underline{x} \setminus \underline{x}_b$. Suppose a Parzen density estimator[2] p^r based on gaussian kernels located at the red points:

$$\forall x \in \mathbb{R}^D,\ p^r(x, \sigma_r) = \frac{2}{N} \sum_{x_r \in \underline{x}_r} (2\pi\sigma_r^2)^{-\frac{D}{2}} e^{-\frac{(x-x_r)^2}{2\sigma_r^2}}$$

Similarly, suppose a Parzen density estimator based on the blue points $p^b(x, \sigma_b)$. Let's suppose the generative model based on the red points is used to explain the blue points. We wish to maximize the loglikelihood given by:

$$L_{\underline{x}_r}(\underline{x}_b, \sigma_r) = \sum_{x_b \in \underline{x}_b} \log(p^r(x_b, \sigma_r)) \tag{1}$$

The maximum likelihood Parzen estimator of the set $\underline{x}_b$ based on the complementary set $\underline{x}_r$ is obtained for the value σ_r^*:

$$L^*_{\underline{x}_r} = max_{\sigma_r > 0}(L_{\underline{x}_r}(\underline{x}_b, \sigma_r)) = L_{\underline{x}_r}(\underline{x}_b, \sigma_r^*)$$

The loglikelihood and maximum likelihood Parzen estimator of the red points given the blue points can be computed the same way reversing 'r' and 'b' in the above equations.

We propose that an optimal equi-distributed bipartition P^* of $\underline{x}$ should be one which maximizes the joint loglikelihood of both blue and red σ-optimal models:

$$P^*_{br} = \arg \max_{P_{br} \in \underline{P}} \left(L^*_{\underline{x}_b} + L^*_{\underline{x}_r} \right) = \arg \max_{P_{br} \in \underline{P}} (L^*_{br}) \tag{2}$$

where $\underline{P} = \{(\underline{x}_b, \underline{x}_r) \in \underline{x}^2 |\ \underline{x}_b \cap \underline{x}_r = \emptyset,\ |\underline{x}_b| = |\underline{x}_r| = \frac{N}{2}\}$ is the set of all possible bipartitions.

For each possible bipartition $P_{br} \in \underline{P}$, we need to find the optimal σ_b^* and σ_r^* of the corresponding Parzen estimators. The number of bipartitions is the number of way to take $N/2$ elements from the set of N data: ${}_{N/2}C_N$. This number is huge for large N and so the direct maximization of L^*_{br} is not tractable. That's why we study different ways to obtain an approximate solution to this optimization problem.

[2]Parzen density estimators can model multimodal and multidimensional density functions.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

2 Approximate solutions

2.1 A hill-climbing method

We can use a hill-climbing method to find a suboptimal bipartitionning by starting from an initial partition, finding the neighborhood partition[3] which increases the likelihood L^*_{br} and iterating from this locally optimal partition until no increase occurs (Figure 1d). In this case, $\frac{N^2}{4}$ neighbors need to be checked at each iteration, and a check takes $O(N^2)$ operations to compute the likelihood, times the number T of σ values tested. The number of iterations is bounded up by the diameter of the exploration space which is $\frac{N}{2}$ (maximum number of permutations needed to pass from a bipartition to another one) so the locally optimal bipartition attainable from the initial one, will be reached in no more than $\frac{N}{2}$ iterations. Therefore this approximation algorithm takes $O(TN^5)$ at worst to get a local optimum, but we cannot tell how close this optimum is to the global one.

We propose hereafter to consider two other approximate solutions: one is based on the minimal distance non-bipartite matching algorithm; the other is based on a multidimensional ranking algorithm.

2.2 Towards better solutions

We observe that near optimal bipartitions obtained with the above iterative algorithm (Figure 1d), tend to spatially alternate blue and red points. This corresponds to partitions where the sum of the distances between blue points (resp. red) and their nearest red point (resp. blue), is small.

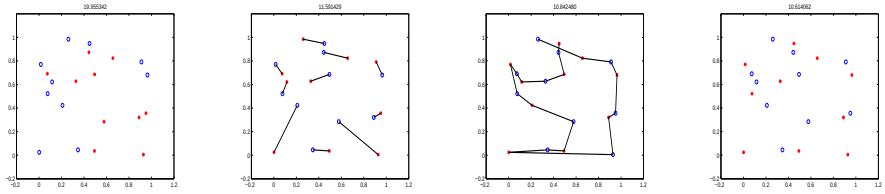
Such partitions are more likely to succeed two-sample tests based on proximity graphs, as these graphs tend to connect relatively close points, and a high number of mixed blue-red pairs increases the probability of success.

Why such bipartitions get higher likelihood of being homogeneous can be explained by the fact that two identical distributions should provide nearly the same number of points in each area of the space, whatever small this area is. When coloring in red and blue, pairs of nearest neighbors, we insure that this is the case for the smallest areas which contain only two points. So this tends to make both distributions the same, even at small scale. This can be seen also observing the likelihood expression (1) in Parzen estimators. The smaller the distance between a reference point and a data point, the higher the contribution of the exponential term in the sum used to compute the likelihood.

This leads us to study algorithms which build partitions tending to minimize the distance between nearest neighbors blue-red pairs, without ever computing the likelihood at each step, so bringing the computation complexity down by two order of magnitude ($O(N^2)$). It appears these algorithms provide near optimal partitions anyway.

[3]Two partitions A and B with equal size, are neighbors if one blue point in A is red in B and one red point in A is blue in B, all the other points keeping the same color in A and B.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.



(a) $-L^* = 19.95$ Random $O(N)$ (b) $-L^* = 11.59$ MDNM $O(N^3)$ (c) $-L^* = 10.84$ PMR $O(N^2)$ (d) $-L^* = 10.61$ Iterative $O(N^5)$

Fig. 1: Bipartitions and their negative loglikelihood on the same set of points.

2.3 The minimal distance non-bipartite matching

A combinatorial optimization algorithm has been proposed by Galil [4] to extract pairs of vertices in a weighted undirected graph such that the sum of the weights of the pairs is maximum. It takes $O(N^3)$ operations where N is the number of vertices. This algorithm can be used to get pairs of points close to each other by using negative pairwise distances for the weights, a convenient bipartition is then obtained by coloring the vertices of a pair in blue and red (Figure 1b). This algorithm, called **MDNM** (minimum distance nonbipartite matching), is used to build a proximity graph in the cross-matching test [6]. The complete graph of the data weighted by their interpoint distances is used, which requires $O(N^2)$ additional computation. As this algorithm remains costly, we propose to look for a graph whose coloring is easier.

2.4 Our method: bipartition based on multidimensional ranking

2.4.1 A multidimensional ranking algorithm

We consider a set $\underline{x}$ of N points in $\mathbb{R}^D$. The following algorithm proposed in [1] is originally used for clustering purpose, and provides a chain of points whose index in the chain can be viewed as a rank order: Select a starting point x_1 and put it in E. Let $E = \{x_1, \ldots, x_k\}$ be the set of k points explored at iteration k, with respective ranks $1, \ldots, k$. At iteration $k+1$, select the nearest neighbor x_{k+1} of x_k among the $N-k$ remaining points unexplored $U = \underline{x} \setminus E$, and append it to E.

2.4.2 Getting the partition

Here we exploit the parity of the rank obtained from the previous algorithm to get the two partitions P_r and P_b (see algorithm on figure 2). One will contain the points whose rank is even, the other one the points whose rank is odd. We call this algorithm **PMR** (Parity Multidimensional Ranking) (Figure 1c).

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

```
k ← 1;
P_b ← indstart; P_r ← ∅;
E ← indstart;
U ← {1, ..., N} \ indstart
indcur ← indstart;
WHILE U ≠ ∅ DO
    indcur ← arg min_{i∈U}(dist(x_i, x_indcur))
    E ← E ∪ indcur;
    U ← U \ indcur;
    IF k is even
    THEN P_b ← P_b ∪ indcur;
    ELSE P_r ← P_r ∪ indcur;
    ENDIF
    k ← k + 1;
ENDWHILE
```

Fig. 2: PMR partitionning algorithm : a starting point *indstart* and a distance measure *dist* are provided. E and U are the sets of explored and unexplored points respectively. A chain of points is built incrementally appending to the end of the chain, the nearest point out of it. The parity of the rank of each point in the chain is used to select the partition P_b or P_r it belongs to.

2.4.3 Complexity

At the k^{th} pass in the "while" loop ($k = 1, \ldots, N-1$), it remains to compute the distance to the $N-k$ unexplored points, which requires $O(\frac{1}{2}DN(N-1))$ operations. This is 3 order of magnitude less than the iterative approach. All the algorithms scale linearly with D which only appears in the distance function.

3 Bipartition of a multidimensional uniform distribution

We draw at random sets of $N \in \{10; 50\}$ data points with a uniform density on the unit D-cube, for $D \in \{3; 5; 10; 20; 50\}$. For each (N, D) pairs, we draw 10 different sets. For each draw, we compute the joint loglikelihood (2) of the following bipartitions: N random bipartitions; 1 MDNM bipartition; N PMR bipartitions starting each at one of the N data points; 1 optimized MDNM bipartition which consists in running the incremental algorithm starting from the MDNM partition; 1 optimized PMR bipartition running the incremental algorithm starting from the best PMR partition among the N obtained. Figure 3 shows the computing time and the average of the loglikelihood for each (N, D) pairs. The PMR algorithm is implemented in C. The MDNM is generated thanks to the Galil code [4]. Both algorithms are provided with an $N.N$ distance matrix although the PMR does not need to compute all the possible pairwise distances. The time is counted only for the partitionning stage of the algorithms, not for the generating of the distance matrix.

The PMR algorithm shows better homogeneity than the $MDNM$ and is two order of magnitude faster. Both PMR and $MDNM$ partitions are improved by the iterative algorithm, showing they provide only suboptimal solutions. Results with more data will be presented during the conference.

			D				
	N	$Partition$	3	5	10	20	50
$-L^*$	10	$rand$	8.97 ± 3.0	16.8 ± 2.7	37.0 ± 4.4	82.4 ± 4.6	247 ± 26
		$MDNM$	6.21	13.5	31.3	77.4	234
		$MDNMopt$	4.16	11.4	29.0	73.5	222
		PMR	4.71 ± 0.47	12.4 ± 0.68	50.6 ± 1.2	75.1 ± 1.7	235 ± 17
		$PMRopt$	4.16	11.4	29.0	73.6	222
$-L^*$	50	$rand$	31.5 ± 5.6	66.1 ± 6.7	163 ± 7.8	367 ± 10	1060 ± 110
		$MDNM$	33.8	67.7	164	366	1070
		$MDNMopt$	8.37	35.9	127	322	964
		PMR	11.8 ± 1.2	39.9 ± 1.7	134 ± 2.3	331 ± 3.1	1010 ± 120
		$PMRopt$	8.17	35.5	127	321	964
$time(ms)$	10	$MDNM$	28.9 ± 3.8				
		PMR	0.321 ± 0.087				
$time(ms)$	50	$MDNM$	329 ± 22				
		PMR	1.16 ± 0.086				

Fig. 3: $xx \pm yy$: xx is the average negative loglikelihood over all the experiments at (N, D) (the lower xx, the higher the homogeneity), and yy is the average of the 10 standard deviations computed over the N tests for the random and PMR partitions. Computing time does not depend on the dimension.

4 Conclusion

We proposed an efficient algorithm to get a homogeneous bipartition from unlabelled data, whose complexity is $O(N^2)$ while the best heuristic costs $O(N^3)$. As highlighted in [7], we foresee such homogeneous bipartitions could be usefull for cross validation purpose in Machine Learning: only one draw of such a partition could replace K draws of random K-folds, leading to learn only twice over $N/2$ data, rather than K times over N/K data, with lower bias and variance expected in the estimate of the model complexity. This would be interesting when the complexity of the learning algorithm dominates the complexity of the partitionning. It could also be possible to get homogeneous parts with different sizes by slightly modifying the algorithm. At last, another issue would be taking into account the output variable in a supervised setting.

References

[1] C. Dematteï, N. Molinari and J.-P. Daurès, *Computational Statistics & Data Analysis*, 51(8):3931-3945, Elsevier, may 2007.

[2] W. DuMouchel and D. K. Agarwal, *I*nt. Conf. on Knowledge discovery and data mining, 511-516, ACM, New York, NY, USA, 2003.

[3] J.H. Friedman and L. C. Rafsky, *A*nn. Statist., 7, 697717, 1979.

[4] Z. Galil, *A*CM Comput. Surv., 18, 2338, 1986. (Available from http://elib.zib.de/pub/Packages/mathprog/matching/weighted/index.html)

[5] R. Greevy, B. Lu, J. H. Silber and P. R. Rosenbaum, *B*iostatistics, 5(2):263-275, Oxford University Press, 2004.

[6] P. R. Rosenbaum, *J. of Royal Stat. Soc., Series B* 67(4):515530, 2005.

[7] M. Vasquez, S. Janaqi, *Metaheuristics International Conference (MIC'01)*, Porto, Portugal, July 16-20, 2001.