一种适用于汽车雷达的聚类算法研究和实现

Adam Yao *Processors FAE*

摘要

近年来，毫米波雷达在高级驾驶员辅助系统(ADAS)中的使用呈现了爆发式的增长。毫米波雷达在汽车上的应用包括舒适性功能和安全性功能。常规的雷达接收处理包括射频前端，基带信号处理和后处理算法。射频前端完成高频雷达接收信号的模拟域信号处理和数模转换，基带信号处理在零中频上完成雷达接收信号的数字信号处理和目标检测，而在目标检测之后的高层算法被统称为后处理算法。常用的后处理算法包括聚类(Clustering), 关联(Association)和跟踪(Tracking)等，它们对于提高系统的稳定性和感知周围的环境非常重要。随着目标尺寸，雷达发射系数(RCS)和检测算法的不同，一个物体在目标检测后可能产生从几个到几百个不同的反射点。聚类就是发掘这些反射点的内在结构，将他们分成若干簇(Cluster). 这些簇更接近真实的物体，因此先将雷达反射点聚类为簇再执行后面的高层算法更简洁，也更有效。本文比较分析了若干常用的聚类算法，从中选取了 DBSCAN(Density-Based Spatial Clustering of Application with Noise)作为适合汽车雷达的聚类算法。重点研究了这种算法的性能，参数敏感性，并提出了一种简单可行的参数调整方法。最后作者在 TI 的 AWR1642 上的 C674x DSP 完成了 DBSCAN 算法的优化，并给出了实际测试的 DSP 执行指令数。

修改记录

Version	Date	Author	Notes
1.0	July 22th 2017	Adam Yao	First release

目录

图

表

1 背景

近年来各种传感器技术在高级驾驶员辅助系统(ADAS)上的应用层出不穷，这些应用涵盖了从提高驾驶安全性到增加驾驶员的舒适度。同基于光学的摄像头和激光雷达技术相比，毫米波雷达在可接受的成本下可以获得极高的分辨率，同时不受雨，雾，雪和光线等周围环境的影响。同时雷达信号可以穿透汽车的保险杠，使得雷达探头可以被安装在保险杠后面，提高了车辆外观的整体美观度 (而无论是超声波探头还是摄像头，都需要露在车辆外部)。因此，毫米波雷达在 ADAS 领域被广泛采用，成为当前最流行的传感器技术之一。图 1 给出了雷达在 ADAS 上的一些典型应用。

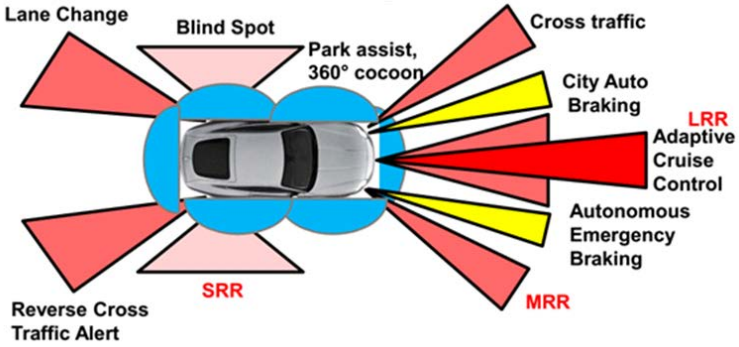

图 1. 雷达在 ADAS 上的典型应用

从汽车雷达的发展方向上看，传统的 24GHz 频段的雷达正逐步被 77GHz 频段的雷达所取代。从各国频率管理机构对未来汽车雷达的频段规划上看，76-77GHz 的频段已经被各国划分给车载远距离雷达(LRR)，77-81GHz 的频段在欧洲已经被划分给车载近距离雷达(SRR)，美国和中国也即将出台类似的频段划分。同时 77GHz 雷达相比 24GHz 雷达模块尺寸更小，干扰更低，分辨率更高。

TI 的 AWR1243/1443/1642 系列芯片是一款采用 45nm 低功耗 RF-CMOS 工艺的高集成单芯片 77GHz 雷达传感器片上系统 (SoC). 它专门为汽车雷达应用设计，符合 ASIL-B 安全等级。其中 AWR1642 在业界第一次在雷达传感器中集成了一个 600MHz 主频的 C674x DSP，可以在片内实现较复杂的雷达信号处理和高层算法。我们已经看到了有很多的资料介绍在 AWR1xxx 芯片上实现雷达的信号处理和目标检测，而本文将集中讨论雷达后处理中的一个重要模块-聚类。

2 信号链

线性快速频率调制连续波(Linear Fast FMCW) 是一种频率随时间快速线性变化(斜率在几十 MHz/us)的雷达波形。由于它可以提供较高的分辨率，同时有效地解决距离-速度模糊性的问题，正在成为主流的汽车雷达波形。TI 的 AWR1xxx 芯片选取了 Fast FMCW 作为支持的雷达波形。

参考文献[1]给出了 AWR1xxx 芯片上射频前端之后主要的信号链(如图 2 所示)。

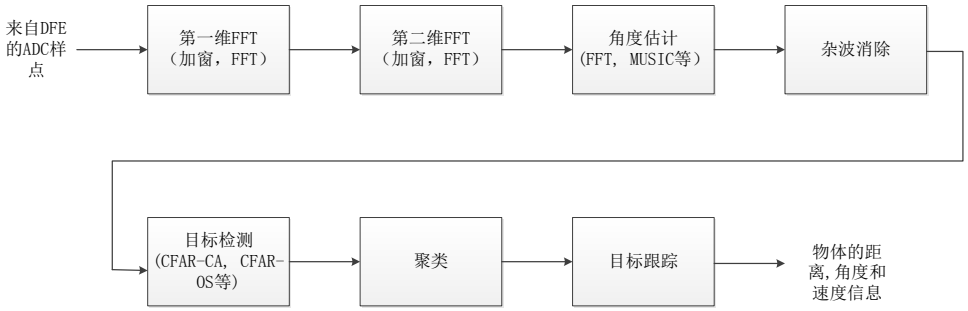

图 2. AWR1xxx 上的雷达信号链

首先，通过第一维和第二维 FFT 计算获得目标距离和速度的二维反射能量剖面(Profile)。接着，采用特定的角度估计算法(例如 FFT，MUSIC 等)来获得目标的角度剖面。之后，再通过特定的目标检测算法(例如 CFAR-CA, CFAR-OS 等)从噪声中检测获得反射点。随着目标的尺寸，雷达发射系数(RCS)和检测算法的不同，一个物体在目标检测后可能产生从几个到几百个不同的反射点。可以先通过聚类算法分析这些反射点的内部结构，将属于同一个物体的反射点归为一个簇，这样每一个检测到的物体都形成一个簇。再通过对聚类以后的簇进行目标跟踪和分类，可以获得可靠的物体的距离和移动速度。这些信息通过 CAN 总线发给车辆的中央处理单元来以实现不同的 ADAS 应用。

3 汽车雷达的聚类算法

聚类在无监督机器学习领域是一个非常热门的研究课题，近年来出现了许多的算法。现有的聚类算法可以大致分为原型聚类(Prototype-based clustering)，密度聚类 (Density-based clustering) 和层次聚类 (Hierarchical clustering).原型聚类通常假设聚类结构能通过一组原型刻画，在现实聚类任务中比较常用。通常情况下，算法先对原型进行初始化，然后对原型进行迭代更新求解。采用不同的原型表示，不同的求解方式，将产生不同的算法。常见的原型聚类算法有 k 均值方法 (k-means)，高斯混合聚类方法 (Mixture-of-Gaussian) 等在数据挖掘领域被广泛应用，但是这些算法都要求在聚类之前就确定输出的簇的数量。对于汽车雷达来说，也就是要求在聚类之前就确定目标的数量，这显然是无法做到的。层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。层次聚类可以采用“自下而上”的聚类策略，也可以采用“自上而下”的聚类策略。AGNES 方法 (AGglomerative NESting)是一种常用的“自下而上”的层次聚类算法。然而这种算法面临和原型聚类相同的问题，也需要在聚类之前确定输出的簇的数量，因此也无法直接应用到汽车雷达上。

密度聚类(Density-based clustering)没有其它两种聚类的限制，不需要事先确定簇的数量。密度聚类假设簇的结构能通过目标点分布的紧密程度来确定。在密度聚类中，簇被认为是数据空间中目标点密集的区域，在簇之间出现的低密度的目标点被认为是噪声. 这些簇可以有任意的形状，并且簇内的目标点也可以任意分布。这一点和汽车雷达上的检测目标特性十分接近。汽车雷达对应同一个目标的检测点之间距离接近，并且这些点的密度分布是一定的（这个密度分布和物体的反射特性相关）.因为以上这些特性，密度聚类更加适合于汽车雷达的应用。

DBSCAN 算法是一种常用的密度聚类算法，将在下一节详细讨论这一算法的性能和实现。在这之前，先定义衡量聚类算法性能的指标。一个好的聚类结果的目标点应该具有高的簇内相似性和低的簇间相似性。在低维度数据集上，聚类性能的好坏通过可视化的数据分布图就可以直观的看出来，但是通过度量指标来定量地衡量聚类性能的好坏更加准确。参考文献[2] 给出了常用的聚类性能度量 DI (Dunn Index)，DBI (Davies-Bouldin Index)，这两种度量在原理上类似，本文选择 DI 作为分析聚类性能的度量指标。假设数据集$D = \{x_1, x_2, \dots, x_m\}$被划分成了$k$个簇$C = \{C_1, C_2, \dots, C_k\}$, DI 的定义如下：

$$DI = \min_{1 \le i \le k} \{ \min_{j \ne i} (\frac{d_{min}(C_i, C_j)}{\max\limits_{1 \le l \le k} diam\,(C_l)}) \}$$

上式中的分母对应的是所有簇内目标点之间的最大距离，分子对应的是所有簇间目标点之间的最小距离。显然同于同一个数据集，一个聚类结果对应的 DI 越大，聚类的性能越好。

4 DBSCAN

4.1 算法定义

DBSCAN 是一种著名的密度聚类算法，它基于一组“邻域” (neighborhood) 参数$(\varepsilon, MinPts)$来刻画样本分布的紧密程度。给定数据集$D = \{x_1, x_2, \dots, x_m\}$， 定义下面几个概念：

- ε-邻域：对 $x_j \in D$，其 ε-邻域包含样本集 D 中与 x_j 的距离不大于 ε 的样本，即 $N_\epsilon(x_j) = \{x_i \in D \mid dist(x_i, x_j) \le \varepsilon\}$;
- 核心对象(core object)：若x_j的 ε-邻域至少包括$MinPts$个样本，即$|N_\epsilon(x_j)| \ge MinPts$，则$x_j$是一个核心对象

DBSCAN 算法先任选数据集中的一个核心对象为种子 (seed)，将它的 ε-邻域中的所有样本加入一个簇，新加入的样本如果是核心对象，再将这个核心对象的 ε-邻域中的所有样本加入本簇。通过这种递归搜索，将所有密度相连的样本归入一个簇。如果此时数据集中还有未处理的核心对象，再重复上述的过程开始一个新簇的搜索。DBSCAN 算法的伪代码描述如图 3 所示。

输入：样本集$D = \{x_1, x_2, \ldots, x_m\}$;
　　邻域参数$(\varepsilon, MinPts)$.
过程：
初始化核心对象集合：$\Omega = \phi$
for j = 1,2,…,m do
　确定样本x_j的 ε-邻域$N_\varepsilon(x_j)$;
　if $\left|N_\varepsilon(x_j)\right| \geq MinPts$ then
　　将样本x_j加入核心对象集合：$\Omega = \Omega \cup \{x_j\}$
　end if
end for
初始化聚类簇数：$k = 0$
初始化未访问样本集合：$\Gamma = D$
while $\Omega \neq \phi$ do
　记录当前未访问样本集合：$\Gamma_{old} = \Gamma$;
　随机选取一个核心对象o $\in \Omega$，初始化队列$Q = \langle o \rangle$
　$\Gamma = \Gamma \setminus \{o\}$;
　while $Q \neq \Phi$ do
　　取出队列Q中的首个样本 q;
　　if $|N_\varepsilon(q)| \geq MinPts$ then
　　令Δ= $N_\varepsilon(q) \cap \Gamma$;
　　将Δ中的样本加入队列Q;
　　$\Gamma = \Gamma \setminus \Delta$;
　　end if
　end while
　$k = k + 1$, 生成聚类簇$C_k = \Gamma_{old} \setminus \Gamma$;
　$\Omega = \Omega \setminus C_k$
end while
输出：簇划分$C = \{C_1, C_2, \ldots, C_k\}$

图 3. DBSCAN 算法描述

4.2 参数敏感性

DBSCAN 算法有两个输入参数，$MinPts$ 定义为一个簇的邻域中的最少点数，而ϵ定义为邻域的半径。DBSCAN 算法的性能对这两个参数非常敏感，因此如何设置算法的入参对聚类的效果非常重要。不合适的入参设置将会产生错误的聚类结果。图 4 给出了一个在不同入参下 DBSCAN 聚类结果的例子。在子图(1)的原始的数据集中我们很容易发现存在 5 个簇，子图(2)的聚类结果输出了 5 个簇，比较接近真实的场景。子图(3)因为选取了过大的$MinPts$将原本比较稀疏的原始数据集中右上角的簇错误地分成了 2 个簇，而子图(4)因为选取了过小的ϵ导致聚类结果中出现了过多的簇划分，并且错误地将原始数据集中右上角的簇当成了噪声。同样通过对子图(2),(3),(4)的聚类结果计算的 DI 分别为 0.8323, 0.1234 和 0.1145，同样说明子图(2)的聚类效果比较好，而子图(3) 和(4)的聚类效果不太令人满意。

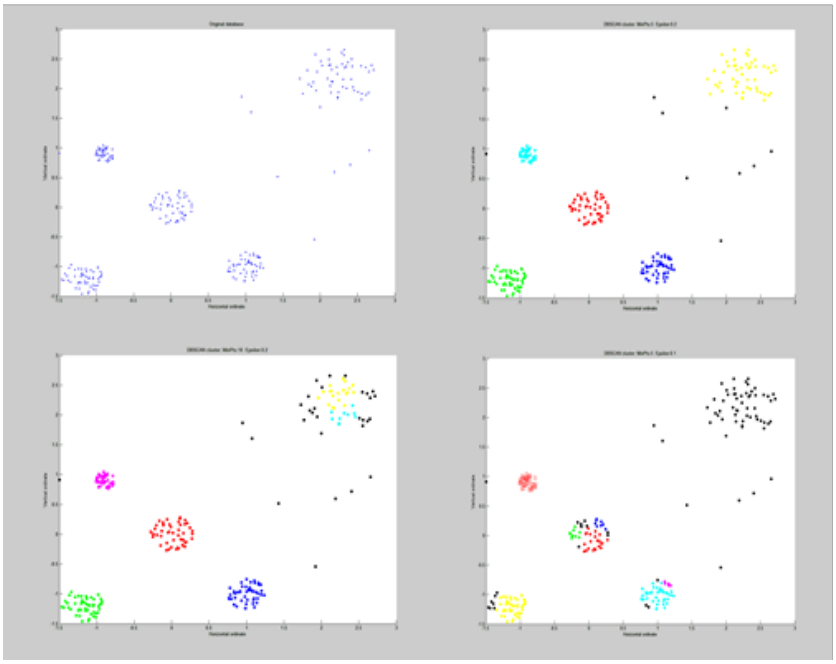

图 4. DBSCAN 参数敏感性 (1)原始数据(左上角), (2)MinPts=5, ε=0.2 下聚类结果(右上角), (3)MinPts=10, ε=0.2 下聚类结果(左下角), (4)MinPts=5, ε=0.1 下聚类结果(右下角)

4.3 参数调整方法

大部分关于 DBSCAN 的文献都讨论的是算法本身，而很少涉及如何为算法选择合适的参数，而从上一节的介绍中可以看到参数的选取对聚类结果非常重要。汽车雷达使用 DBSCAN 时需要根据探测场景和目标检测算法的性能对参数有针对性地进行一些调整。实践中，通常选取一些典型的测试场景，对 DBSCAN 的入参进行调教以获得期望的聚类效果，最后对不同测试场景下的理想参数进行综合来确定车上实际使用的参数。这里将提出一种创新的方法针对一个特定测试场景，确定最优的$MinPts$和ϵ.

$MinPts$是区别于噪声的一个簇内的最少点数。在理论上噪声和它的邻点之间的距离要大于目标点和它的邻点之间的距离。如果定义点p 和它的第k个最近的邻点之间的k近邻距离为k_neighbor_dis(p)，理论上噪声点的k近邻距离要大于目标点的k近邻距离。如果对一个数据集中的所有点分别计算k近邻距离并进行从大到小排序，k近邻距离较大的一些点对应的是噪声，而k近邻距离较小的一些点对应的是簇中的目标点。另外如果分别计算一个数据集中所有点的 1 近邻，2 近邻，...k近邻距离，并对近邻距离最大的 n 个点的近邻距离分别进行平均，会发现这个平均值随着k的增加而增加，但是平均值的增量随着k的增加而降低。会发现一个特定的k，超过这个k后，$k+1$近邻距离，$k+2$近邻距离...会越来越接近。从理论上讲，这个k值接近于合理的$MinPts$值，因为对于近邻距离最大的 n 个点(主要是噪声点)，其k近邻，$k+1$近邻， $k+2$近邻距离趋向接近，也就是对应的k近邻，$k+1$近邻，$k+2$近邻更加接近于簇中的目标点。基于这种思想，设计了下面的$MinPts$搜索算法：

1) 计算数据集D中所有样点的 1 近邻，2 近邻，…n 近邻距离，并对所有的近邻距离从大到小进行排序
2) 选定一个 n 值，从 1)中得到的 1 近邻，2 近邻，…n 近邻距离选择出最大的 n 个值，并求平均
3) 计算 2)中得到的相邻两个平均值之间的差值，找到一个k值使得k近邻的平均值大于$k+1$近邻的平均值，并且k近邻的平均值减去$k+1$近邻的平均值最小
4) 定义$MinPts = k - 1$

图 5 中画出了对于图 4 数据集中最大的 3 个$k+1$近邻和k近邻的平均值之间的差值(例如图中第一个柱状图对应的是 3 个最大的 2 近邻距离平均值减去 3 个最大的 1 近邻距离平均值的差值)。从这张图中我们可以发现$k=6$下有最小的近邻距离增量，因此建议的$MinPts$设置为 5

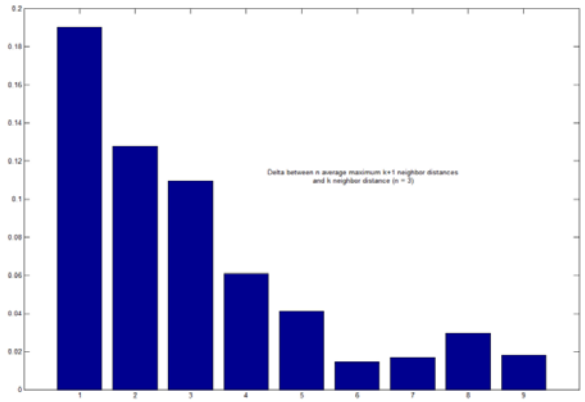

图 5. 图 4 数据集的 k+1 近邻和 k 近邻距离中最大 n 个数值的平均值之差(n=3)

ϵ决定了聚类时邻域搜索的半径，DBSCAN 算法中一个点的ϵ-邻域中的邻点的数目大于$MinPts$的时候被定义为核心节点，并且这个点和所有它的ϵ-邻域中的点都被归入同一个簇。通常一个数据集越密集，聚类时选取的ϵ应该越小。因此在调整ϵ参数的时候，需要分析数据集的密度结构。参考文献[3]介绍了一种 OPTICS (Ordering Points To Identify the Clustering Structure) 算法可以获得一个数据集的密度信息。作者基于 OPTICS 算法进行了一些改进，提出了一种更有效的ϵ搜索的算法。作者设计的ϵ搜索算法分为两步：第一步通过改进的 OPTICS 算法对数据集的密度信息进行分析，得到一个大致的ϵ搜索区间，第二步在第一步获得的ϵ搜索区中，按照一定的步长尝试进行 DBSCAN 聚类，对聚类的结果分别计算 DI 进行评估，从而选择出最优的ϵ.

作者给出的改进的 OPTICS 算法的伪代码描述如图 6 所示。对于图 4 中的数据集，采用改进的 OPTICS 算法对样本重新排序后输出的包含可达距离的曲线如图 7 所示。图中的峰值突起对应的是一个簇中样本和其前序样本(通常是另一个簇中的样本) 之间的可达距离。从图中可以直观地发现，簇中样本之间的可达距离（对应曲线的底部）要远远小于簇间样本之间的可达距离（对应曲线的峰值突起），同时也能直观的发现数据集中簇中样本之间和簇间样本之间的可达距离的大致范围。可以设置ϵ搜索的大致范围在曲线的最低峰值和底部平均值之间，比如在图 7 中可以设置ϵ的搜索范围为 0.1 到 0.6 之间。在 0.1 到 0.6 之间选择搜索步长为 0.1，对数据集进行 DBSCAN 聚类，并计算聚类结果的 DI，可以得到表 1 中对应的 DI 数值。从表 1 中可以发现，$\epsilon = 0.1$时，DI 最小，而对于其他的ϵ值，DI 较大且取值相同，也就是说采用这些ϵ值的聚类效果都比较理想。综合本节前面的内容，对于图 4 数据集，采用$\text{MinPts} = 5$，$\varepsilon = 0.2 - 0.6$是一个比较合理的 DBSCAN 算法入参，这与图 4 本身显示的聚类效果一致。

表 1. 图 4 数据集在 0.1 到 0.6 区间 DBSCAN 结果 DI 指标比较

ε	0.1	0.2	0.3	0.4	0.5	0.6
DI	0.1145	0.8323	0.8323	0.8323	0.8323	0.8323

输入：样本集$D = \{x_1, x_2, \ldots, x_m\}$;
　　邻域参数 $MinPts$.
过程：
1: 初始化一个队列$Seedlist = \Phi$
2: 初始化一个队列$Orderlist = \Phi$
3: 将样本集D中的样本依次放入队列$Seedlist$, $Seedlist = \langle x_1, x_2, \ldots, x_m \rangle$
4: 得到队列$Seedlist$中的样本数m
5: for j = 1,2,…,m do
6:　计算队列$Seedlist$中样本j的k近邻距离$k_neighbor_dist(j)$，其中$k = MinPts$;
7:　初始化样本j的可达距离$Reachable_dist(j) = k_neighbor_dist(j)$;
8: end for
9: while $Seedlist \neq \Phi$ do
10:　取出队列$Seedlist$头部的样本p;
11:　将p放入队列$Orderlist$的尾部，并记录它的可达距离$Reachable_dist(p)$;
12:　更新当前队列$Seedlist$中的样本数$m = m - 1$;
13:　for $j = 1,2,\ldots m$ do
14:　计算队列$Seedlist$中的样本j和样本p之间的距离$dist(j,p)$;
15:　更新队列$Seedlist$中的样本j的可达距离$Reachable_dist(j) = \max(dist(j,p), k_neighbor_dist(j))$;
16:　end for
17:　对队列$Seedlist$中的样本按可达距离从小到大排序
18: end while
输出：队列$Orderlist$，以及队列中每个样本对应的可达距离

图 6. 改进的 OPTICS 算法描述

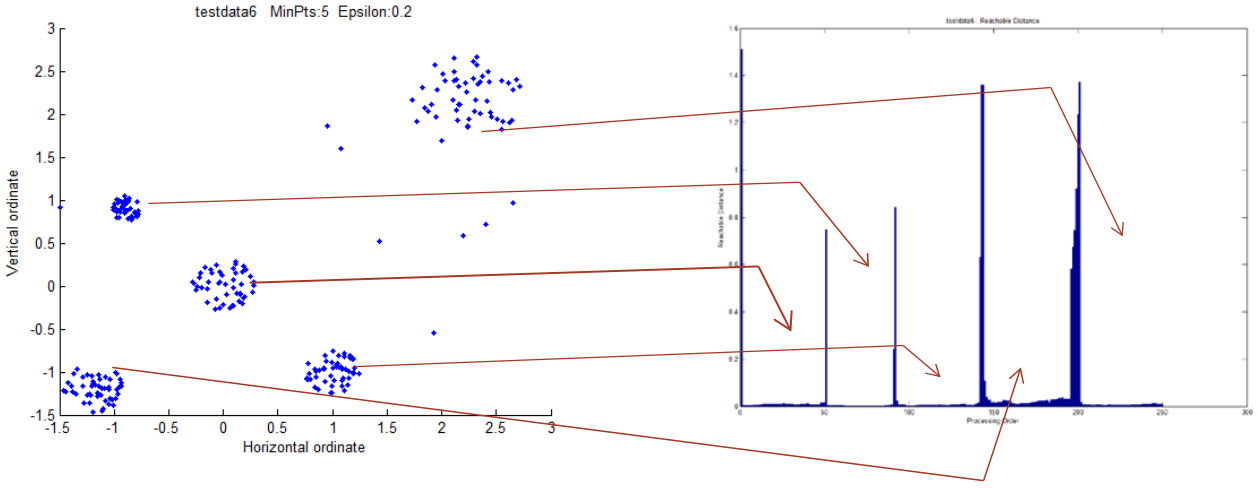

图 7. 图 4 数据集采用改进的 OPTICS 算法排序输出的可达距离

4.4 DSP 实现和优化

作者在 TI 的 AWR1642 中的 C674x DSP 上完成了 DBSCAN 算法的实现和优化，在 C674x DSP 上测试的算法执行指令数如表 2 所示。从表 2 我们可以发现对于常见的包含 100-200 个目标点的数据集的聚类，C674x 的执行指令数在 50,000 到 100,000 之间，对应在 AWR1642 上的处理时间在 0.08 到 0.16us 之间。经过优化之后的 DBSCAN 算法在 AWR1642 上的实时性能够满足常见汽车雷达应用的要求。

表 2. DBSCAN 在 C674x DSP 的执行指令数

ID	目标数	簇数	指令数	ID	目标数	簇数	指令数
1	30	1	9,994	9	240	6	131,235
2	60	2	24,269	10	240	6	128,171
3	90	3	33,573	11	240	6	137,731
4	120	3	51,677	12	270	6	162,143
5	120	5	46,349	13	300	6	196,556
6	150	3	63,735	14	330	7	248,553
7	180	4	88,503	15	360	8	276,876
8	210	5	116,462	16	400	9	284,454

5 总结

本文比较研究了常用的聚类算法，提出了使用 DBSCAN 做为汽车雷达的聚类算法。接着深入研究了 DBSCAN 算法的性能，参数敏感性，并提出了一种定量的参数调整方法，解决了算法在汽车雷达应用上的主要困难。最后本文完成了基于 TI AWR1642 上的 C674x DSP 的算法实现和优化，并给出了算法的执行指令数。

参考文献

1. Sriram Murali and Pankaj Gupta, "FMCW Radar System Overview, Session #1: FMCW Radar Signal Processing", TI internal document, 2015

2.周志华,"机器学习"， 清华大学出版社，2016， 202-217 页

3.Mihael Ankerst, Markus M. Breunig, Hans-Peter Krieegel and Jorg Sander, "OPTICS: Ordering Points To Identify the Clustering Structure", in SIG-MOD, 1999, pp49-60,

有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的 应用； 如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和 判断， 且应全权负责并确保 应用的安全性， 以及您的 应用 （包括应用中使用的所有 TI 产品）） 应符合所有适用的法律法规及其他相关要求。你就您的 应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何 应用前， 您将彻底测试该等 应用 和该等应用所用 TI 产品的 功能。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的 应用时， 才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法理授予您任何TI知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等产权包括但不限于任何专利权、版权、屏蔽作品权或与使用TI产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默认的保证或陈述，包括但不限于对准确性或完整性、产权保证、无屡发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (http://www.ti.com/sc/docs/stdterms.htm)、评估模块和样品 (http://www.ti.com/sc/docs/sampterms.htm) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122