



(12)发明专利

(10)授权公告号 CN 106295671 B

(45)授权公告日 2020.03.03

(21)申请号 201510319922.3

(22)申请日 2015.06.11

(65)同一申请的已公布的文献号
申请公布号 CN 106295671 A

(43)申请公布日 2017.01.04

(73)专利权人 深圳市腾讯计算机系统有限公司
地址 518000 广东省深圳市南山区高新区
高新南一路飞亚达大厦5-10楼

(72)发明人 贺海军 孔蓓蓓 熊健 熊焰
杨剑鸣

(74)专利代理机构 深圳市深佳知识产权代理事
务所(普通合伙) 44285
代理人 王仲凯

(51)Int.Cl.
G06K 9/62(2006.01)

(56)对比文件

CN 104516900 A,2015.04.15,
CN 103679012 A,2014.03.26,
US 2013139163 A1,2013.05.30,

审查员 李亚楠

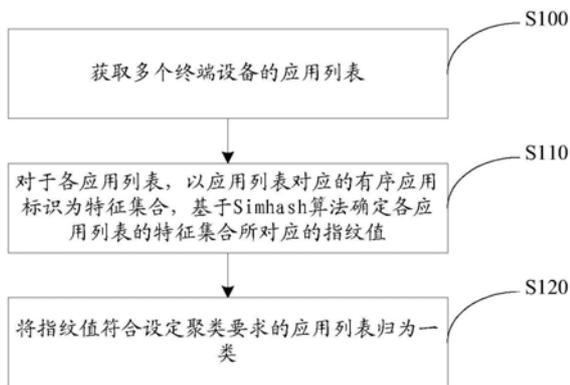
权利要求书2页 说明书10页 附图5页

(54)发明名称

一种应用列表聚类方法、装置及计算设备

(57)摘要

本发明实施例提供一种应用列表聚类方法、装置及计算设备,其中方法包括:获取多个终端设备的应用列表;对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;将指纹值符合设定聚类要求的应用列表归为一类。本发明可以较小的数据计算工作量,实现对应用列表进行聚类的目的。



1. 一种应用列表聚类方法,其特征在于,包括:
 - 获取多个终端设备的应用列表;
 - 对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;
 - 将指纹值符合设定聚类要求的应用列表归为一类;
 - 在将指纹值符合设定聚类要求的应用列表归为一类后,在相同的类中保留一个应用列表,去除所述类中的其他应用列表。
2. 根据权利要求1所述的应用列表聚类方法,其特征在于,所述基于Simhash算法确定各应用列表的特征集合所对应的指纹值包括:
 - 设置各应用列表对应的 f 维的向量 V ,及各应用列表对应的 f 位的数值 S ;
 - 对于各应用列表,采用hash算法确定各特征的 f 位指纹 b ,一个所述特征由应用列表的至少一个应用标识构成;
 - 对于各应用列表,根据各特征的 f 位指纹 b 调整应用列表对应的 f 维的向量 V ,得到的各应用列表对应的 f 维的向量 V 的最终值;
 - 根据各应用列表对应的 f 维的向量 V 的最终值,确定各应用列表对应的数值 S 的最终值,以各应用列表对应的数值 S 的最终值作为各应用列表对应的指纹值。
3. 根据权利要求2所述的应用列表聚类方法,其特征在于,所述设置各应用列表对应的 f 维的向量 V ,及各应用列表对应的 f 位的数值 S 包括:
 - 设置各应用列表对应的 f 维的向量 V ,且初始化各应用列表对应的向量 V 为0,及设置各应用列表对应的 f 位的数值 S ,且初始化各应用列表对应的数值 S 为0;
 - 所述对于各应用列表,根据各特征的 f 位指纹 b 调整应用列表对应的 f 维的向量 V ,得到的各应用列表对应的 f 维的向量 V 的最终值包括:
 - 对于各应用列表的第1至第 f 个特征,若第 i 个特征的指纹 b 的第 i 位为1, i 为1至 f 的整数,则在应用列表对应的向量 V 的第 i 个元素加上所述第 i 个特征对应的设定权重;若所述第 i 个特征的指纹 b 的第 i 位不为1,则在应用列表对应的向量 V 的第 i 个元素减去所述第 i 个特征对应的设定权重,得到各应用列表对应的 f 维的向量 V 的最终值。
4. 根据权利要求3所述的应用列表聚类方法,其特征在于,所述根据各应用列表对应的 f 维的向量 V 的最终值,确定各应用列表对应的数值 S 的最终值包括:
 - 若应用列表对应的向量 V 的最终值中第 i 个元素大于0,则将应用列表对应的数值 S 的第 i 位数值置为1;若应用列表对应的向量 V 的最终值中第 i 个元素不大于0,则将应用列表对应的数值 S 的第 i 位数值置为0,得到各应用列表对应的数值 S 的最终值。
5. 根据权利要求2-4任一项所述的应用列表聚类方法,其特征在于,一个应用列表的应用标识按照序数,依序划分为多个特征;其中,各特征包含至少一个应用标识,且各特征所包含的应用标识的个数相同。
6. 根据权利要求5所述的应用列表聚类方法,其特征在于,所述各特征包含多个应用标识,上一特征的最后一个应用标识,为下一特征的第一个应用标识。
7. 根据权利要求1所述的应用列表聚类方法,其特征在于,所述方法还包括:
 - 终端设备在确定各类所保留的应用列表后,若识别到获取所述终端设备的原始应用列表的指令,则根据所述指令进行hook,以将所述指令的返回结果调整为聚类后的任一类所

保留的应用列表；

返回调整后的所述指令的返回结果。

8. 一种应用列表聚类装置,其特征在于,包括:

列表获取模块,用于获取多个终端设备的应用列表;

指纹值确定模块,用于对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;

聚类模块,用于将指纹值符合设定聚类要求的应用列表归为一类;在将指纹值符合设定聚类要求的应用列表归为一类后,在相同的类中保留一个应用列表,去除所述类中的其他应用列表。

9. 一种计算设备,其特征在于,包括权利要求8所述的应用列表聚类装置。

10. 一种计算设备,其特征在于,包括:存储器和处理器;

所述存储器用于存储计算机操作指令;

所述处理器用于执行所述存储器中存储的计算机操作指令;

所述计算机操作指令用于执行权利要求1-7任一项所述的应用列表聚类方法。

11. 一种存储介质,所述存储介质中存储有计算机操作指令,所述计算机操作指令用于执行权利要求1-7任一项所述的应用列表聚类方法。

一种应用列表聚类方法、装置及计算设备

技术领域

[0001] 本发明涉及信息分类技术领域,具体涉及一种应用列表聚类方法、装置及计算设备。

背景技术

[0002] 随着Android(安卓)、IOS等智能操作系统的普及,终端设备(如手机、平板电脑、笔记本电脑等用户设备)所装载的应用越来越多,为整理终端设备所装载的应用情况,应用列表的概念应运而生;应用列表表示的是终端设备上所装载的应用集合;每个应用可以有一个区分其他应用的应用标识(如应用名称等),将终端设备上所装载的所有应用的应用标识排序整理后,可形成一个有序的应用列表。

[0003] 出于统计各终端设备所装载的应用情况,结合各终端设备所装载的应用情况、确定终端设备的软硬件配置策略(如确定正研发的终端设备的软硬件配置参数)等目的,收集海量终端设备的应用列表,并对收集的海量终端设备的应用列表进行聚类,整理出海量终端设备的应用列表的类型,显得尤为必要。

[0004] 本发明的发明人研究发现:对所收集的海量终端设备的应用列表进行聚类的关键点在于,从所收集的海量终端设备的应用列表中确定出相似、甚至相同的应用列表,并将相似、甚至相同的应用列表归为一类;然而,由于终端设备的数量众多,且一个终端设备对应有一个自身的一个应用列表,这使得在对应用列表进行聚类的过程中,存在大规模的数据计算对象,及较大工作量的数据计算过程;因此,如何提供一种数据计算工作量较小的应用列表聚类方案,成为本领域技术人员需要考虑的问题。

发明内容

[0005] 有鉴于此,本发明实施例提供一种应用列表聚类方法、装置及计算设备,以使用较小的数据计算工作量,实现对应用列表进行聚类的目的。

[0006] 为实现上述目的,本发明实施例提供如下技术方案:

[0007] 一种应用列表聚类方法,包括:

[0008] 获取多个终端设备的应用列表;

[0009] 对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;

[0010] 将指纹值符合设定聚类要求的应用列表归为一类。

[0011] 本发明实施例还提供一种应用列表聚类装置,包括:

[0012] 列表获取模块,用于获取多个终端设备的应用列表;

[0013] 指纹值确定模块,用于对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;

[0014] 聚类模块,用于将指纹值符合设定聚类要求的应用列表归为一类。

[0015] 本发明实施例还提供一种计算设备,包括上述所述的应用列表聚类装置。

[0016] 基于上述技术方案,本发明实施例提供的应用列表聚类方法,可获取多个终端设备的应用列表;对于所获取的各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定出各应用列表的特征集合所对应的指纹值;从而将指纹值符合设定聚类要求的应用列表归为一类,实现对所获取的多个应用列表的聚类目的。本发明实施例采用Simhash算法,通过将应用列表的特征集合(特征集合由应用列表对应的有序应用标识构成)映射为指纹值,使得各应用列表的相似度可通过指纹值进行比对,进而实现了将指纹值符合设定聚类要求的应用列表归为一类的目的;在应用列表聚类的过程中,各应用列表的指纹值的确定及指纹值的比对,所涉及的计算工作量均较小,且Simhash算法可以较低的计算量和存储消耗实现,因此本发明实施例在实现对应用列表进行聚类的过程中,极大的降低了数据计算工作量,以较小的数据计算工作量,实现了对应用列表进行聚类的目的。

附图说明

[0017] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据提供的附图获得其他的附图。

[0018] 图1为本发明实施例提供的应用列表聚类方法的流程图;

[0019] 图2为本发明实施例提供的确定各应用列表的特征集合所对应的指纹值的方法流程图;

[0020] 图3为本发明实施例提供的确定各应用列表的特征集合所对应的指纹值的另一方法流程图;

[0021] 图4为本发明实施例提供的修改指令返回结果的方法流程图;

[0022] 图5为本发明实施例提供的应用列表聚类装置的结构框图;

[0023] 图6为本发明实施例提供的指纹值确定模块的结构框图;

[0024] 图7为本发明实施例提供的设置单元的结构框图;

[0025] 图8为本发明实施例提供的第一调整单元的结构框图;

[0026] 图9为本发明实施例提供的第二调整单元的结构框图;

[0027] 图10为本发明实施例提供的应用列表聚类装置的另一结构框图;

[0028] 图11为本发明实施例提供的应用列表聚类装置的再一结构框图;

[0029] 图12为本发明实施例提供的计算设备的硬件结构框图。

具体实施方式

[0030] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0031] 图1为本发明实施例提供的应用列表聚类方法的流程图,该方法可应用于具有数据处理能力的计算设备中,计算设备可以为服务器,也可以为终端设备;参照图1,该方法可以包括:

[0032] 步骤S100、获取多个终端设备的应用列表；

[0033] 可选的，若计算设备为服务器，本发明实施例可设置终端设备将各自的应用列表整理后上报至服务器，以使得服务器可收集到各终端设备的应用列表，实现多个终端设备的应用列表的获取；

[0034] 可选的，若计算设备为终端设备，则终端设备可在服务器收集到各终端设备的应用列表后，向服务器请求所收集到的各终端设备的应用列表，从而实现多个终端设备的应用列表的获取；

[0035] 可选的，本发明实施例也可使用第三方工具或应用检测各终端设备的软硬件接入环境属性，从而基于该第三方工具或应用的检测，可以累积获得多个终端设备的应用列表。

[0036] 步骤S110、对于各应用列表，以应用列表对应的有序应用标识为特征集合，基于Simhash算法确定各应用列表的特征集合所对应的指纹值；

[0037] 其中，Simhash算法是在大文本重复识别中使用的一个方法，该方法主要是通过将对对象的特征集合映射为一个固定长度的指纹(如签名)，将对象之间的相似度的度量转化为指纹的汉明距离，通过这样的方式，从而以较低的计算和存储的消耗，实现对象之间的相似度的衡量；

[0038] 具体在应用列表的相似度比对上，本发明实施例可以应用列表对应的有序应用标识为特征集合，从而基于Simhash算法将各应用列表的所述特征集合映射为一个固定长度的指纹，并且得出各应用列表的指纹值；进而通过指纹值的比对，实现应用列表的相似度的比对；

[0039] 此处需要注意的是，应用列表的特征集合由应用列表中有有序应用标识构成，本发明实施例可将应用列表的应用标识按照序数，依序划分为多个特征，其中，一个特征包含至少一个应用标识，各特征所包含的应用标识的个数相同；通过上述划分出的多个特征可构成应用列表的特征集合；

[0040] 如一个应用列表具有a1,a2,a3至a10共10个应用，以一个特征包含一个应用标识为例，则按照a1,a2,a3至a10的序数，可将a1的应用标识视为第一个特征，将a2的应用标识视为第二个特征，以此类推，确定出10个特征，构成该应用列表的特征集合；

[0041] 若一个特征包含多个应用标识，则本发明实施例可设置上一特征的最后一个应用标识，为下一特征的第一个应用标识；即第N个特征的最后一个应用标识，为第N+1个特征的第一个应用标识；

[0042] 如一个应用列表具有a1,a2,a3至a10共10个应用，以一个特征包含二个应用标识为例，则按照a1,a2,a3至a10的序数，可将a1,a2的应用标识划分为第一个特征，a2,a3的应用标识划分为第二个特征，a3,a4的应用标识划分为第三个特征，a4,a5的应用标识划分为第四个特征，以此类推，确定出9个特征，构成该应用列表的特征集合。

[0043] 步骤S120、将指纹值符合设定聚类要求的应用列表归为一类。

[0044] 可选的，符合设定聚类要求的指纹值可以为数值完全相同的指纹值，也可以为指纹值的差值在设定差值范围内的指纹值；

[0045] 以应用列表共有8个，分别为Y1~Y8为例，若Y1、Y2、Y5的指纹值相同，Y3、Y7的指纹值相同，Y4、Y6、Y8的指纹值相同，则可将应用列表Y1、Y2、Y5归为一类，将应用列表Y3、Y7归为一类，将应用列表Y4、Y6、Y8归为一类，实现对多个终端设备的应用列表的聚类。

[0046] 本发明实施例提供的应用列表聚类方法,可获取多个终端设备的应用列表;对于所获取的各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定出各应用列表的特征集合所对应的指纹值;从而将指纹值符合设定聚类要求的应用列表归为一类,实现对所获取的多个应用列表的聚类目的。本发明实施例采用Simhash算法,通过将应用列表的特征集合(特征集合由应用列表对应的有序应用标识构成)映射为指纹值,使得各应用列表的相似度可通过指纹值进行比对,进而实现了将指纹值符合设定聚类要求的应用列表归为一类的目的;在应用列表聚类的过程中,各应用列表的指纹值的确定及指纹值的比对,所涉及的计算工作量均较小,且Simhash算法可以较低的计算量和存储消耗实现,因此本发明实施例在实现对应用列表进行聚类的过程中,极大的降低了数据计算工作量,以较小的数据计算工作量,实现了对应用列表进行聚类的目的。

[0047] 可选的,在本发明实施例中,图1所示步骤S100中可由计算设备的接口收集终端设备的应用列表后,导入计算设备内的处理器等处理芯片中;同时,S110和步骤S120可由计算设备内的处理器等处理芯片实现。

[0048] 可选的,基于Simhash算法确定各应用列表的特征集合所对应的指纹值的方式可如图2示出,参照图2,该方法可以包括:

[0049] 步骤S200、设置各应用列表对应的 f 维的向量 V ,及各应用列表对应的 f 位的数值 S ;

[0050] 可选的,在设置各应用列表对应的 f 维的向量 V ,及各应用列表对应的 f 位的数值 S 后,可统一各应用列表初始的向量 V 及数值 S ,以使得各应用列表初始的向量 V 和数值 S 处于同一基础上;具体的,本发明实施例可初始化各应用列表对应的向量 V 为0,并初始化各应用列表对应的数值 S 为0;

[0051] 显然,也可初始化各应用列表对应的向量 V 统一为其他数值,初始化各应用列表对应的数值 S 统一为其他数值,仅需保证各应用列表初始化后的向量 V 是相同的,且各应用列表初始化后的数值 S 是相同的即可。

[0052] 步骤S210、对于各应用列表,采用hash算法确定各特征的 f 位指纹 b ,一个所述特征由应用列表的至少一个应用标识构成;

[0053] 可选的,在特征的选取上,本发明实施例可将应用列表的应用标识按照序数,依序划分为多个特征;其中,一个特征包含至少一个应用标识,且各特征所包含的应用标识的个数相同;通过上述划分出的多个特征可构成应用列表的特征集合。

[0054] 步骤S220、对于各应用列表,根据各特征的 f 位指纹 b 调整应用列表对应的 f 维的向量 V ,得到的各应用列表对应的 f 维的向量 V 的最终值;

[0055] 可选的,根据各特征的 f 位指纹 b 调整应用列表对应的 f 维的向量 V 的方式的主要原则是:通过特征在应用列表中的序位,调整向量 V 中相应序位的元素的值,使得应用列表的向量 V 中的各元素,可与应用列表中序数相应的特征匹配上;基于此原则,根据各特征的 f 位指纹 b 调整应用列表对应的 f 维的向量 V 的方式可以有多种,本发明实施例并不作限制。

[0056] 步骤S230、根据各应用列表对应的 f 维的向量 V 的最终值,确定各应用列表对应的数值 S 的最终值,以各应用列表对应的数值 S 的最终值作为各应用列表对应的指纹值。

[0057] 可选的,根据各应用列表对应的 f 维的向量 V 的最终值调整对应的数值 S 的主要原则是:通过调整后的向量 V 中各序位元素的值,调整数值 S 中与元素序位相同的位置的数值,从而使得应用列表的数值 S 中的各位置的数值,可与应用列表的最终值的向量 V 中序数相应

的元素值匹配上,进而使得应用列表对应的指纹值,能够准确的表示出应用列表中有应用标识的特征;基于此原则,根据各应用列表对应的f维的向量V的最终值,确定各应用列表对应的数值S的最终值的方式可以有多种,本发明实施例并不作限制。

[0058] 可选的,图3示出了本发明实施例提供的确定各应用列表的特征集合所对应的指纹值的另一方法流程图,参照图3,该方法可以包括:

[0059] 步骤S300、设置各应用列表对应的f维的向量V,且初始化各应用列表对应的向量V为0,及设置各应用列表对应的f位的数值S,且初始化各应用列表对应的数值S为0;

[0060] 可选的,步骤S300可以认为是图2所示步骤S200的一种具体实现方式。

[0061] 步骤S310、对于各应用列表,采用hash算法确定各特征的f位指纹b,一个所述特征由应用列表的至少一个应用标识构成;

[0062] 步骤S320、对于各应用列表的第1至第f个特征,若第i个特征的指纹b的第i位为1, i为1至f的整数,则在应用列表对应的向量V的第i个元素加上所述第i个特征对应的设定权重;若所述第i个特征的指纹b的第i位不为1,则在应用列表对应的向量V的第i个元素减去所述第i个特征对应的设定权重,得到各应用列表对应的f维的向量V的最终值;

[0063] 可选的,步骤S320可以认为是图2所示步骤S220的一种具体实现;

[0064] 可选的,本发明实施例可将特征在应用列表中的序位,与向量V的元素的序位匹配上,从而通过相同序位的特征的值调整向量V中的相应元素的值;具体的,若应用列表的第1至第f个特征中的任一特征T在应用列表中的序位为第i位,且该特征T的指纹b的第i位为1,则可将应用列表对应的向量V的第i个元素加上该特征T对应的设定权重(设定权重可选为1,具体可视实际情况设定),否则,若该特征T的指纹b的第i位不为1,则可将应用列表对应的向量V的第i个元素减去该特征T对应的设定权重;

[0065] 对于应用列表的第1至第f个特征,执行上述调整相应向量V的元素值的过程,可得到应用列表对应的f维的向量V的最终值。

[0066] 步骤S330、若应用列表对应的向量V的最终值中第i个元素大于0,则将应用列表对应的数值S的第i位数值置为1;若应用列表对应的向量V的最终值中第i个元素不大于0,则将应用列表对应的数值S的第i位数值置为0,得到各应用列表对应的数值S的最终值,以各应用列表对应的数值S的最终值作为各应用列表对应的指纹值。

[0067] 在确定了各应用列表对应的向量V的最终值后,可根据应用列表对应的向量V的最终值中各元素的值,调整应用列表对应的数值S中相应序位的数值;具体的,对于向量V中的任一元素K, K的序位为第i位,若该第i位元素K的值大于0,则应用列表对应的数值S的第i位数值置为1,否则,若该第i位元素K的值不大于0,则应用列表对应的数值S的第i位数值置为0;

[0068] 对于应用列表得到最终值的向量V的各元素,执行上述调整相应的数值S的各位置的数值的过程,可得到应用列表对应的数值S的最终值。

[0069] 可选的,不同终端设备的应用列表,会存在一定数量的共同应用;若将一个应用标识作为一个特征,并采用这种方式基于SimHash算法确定应用列表的特征集合所对应的指纹值,将导致指纹值重复的情况很多,而实际上这些存在一定数量的共同应用的应用列表还是有一定区别的;因此在本发明实施例中,一个特征(即构成应用列表的特征集合的基础)可以包含多个应用标识;

[0070] 即本发明实施例可将应用列表的应用标识按照序数,依序划分为多个特征;其中,各特征包含多个应用标识,各特征所包含的应用标识的个数相同,且第N个特征的最后一个应用标识,为第N+1特征的第一个应用标识;

[0071] 以二元应用名称作为一个特征(一个特征包含应用列表中依序排列的两个应用标识,应用名称为应用标识的一种)为例,假设应用列表中的应用名称排列如下:

[0072] “android,com.android.backupconfirm,com.android.bluetooth,com.android.calendar,com.android.certinstaller,com.android.contacts,com.android.defcontainer”

[0073] 则在本发明实施例中,可得到如下特征:

[0074] “android,com.android.backupconfirm”,“com.android.backupconfirm,com.android.bluetooth”,“com.android.bluetooth,com.android.calendar”
“com.android.calendar,com.android.certinstaller”,“com.android.certinstaller,com.android.contacts”,“com.android.contacts,com.android.defcontainer”;

[0075] 可以看出,本发明实施例在确定各特征时,是按照各特征包含个数相同的多个应用标识,且上一特征的最后一个应用标识,为下一特征的第一个应用标识的原则,将应用列表的应用标识按照序数,依序划分为多个特征。

[0076] 可选的,在将指纹值符合设定聚类要求的应用列表归为一类后,对于各类,本发明实施例可仅保留一个应用列表作为该类的应用列表的标准,而去除该类中的其他应用列表;具体的,本发明可在将指纹值符合设定聚类要求的应用列表归为一类后,在相同的类中保留一个应用列表,去除所述类中的其他应用列表;

[0077] 如将应用列表Y1、Y2、Y5归为一类后,可在Y1、Y2、Y5这一类中仅保留应用列表Y1,而去除应用列表Y2、Y5。

[0078] 可选的,本发明实施例提供的应用列表聚类方法可应用于终端设备中,以使得终端设备在相同的类中保留一个应用列表后,确定出各类所保留的应用列表;显然,本发明实施例提供的应用列表聚类方法也可应用于服务器中,由服务器将各类所保留的应用列表发送至终端设备,使得终端设备确定出各类所保留的应用列表;

[0079] 在终端设备确定各类所保留的应用列表的基础上,为避免病毒程序对终端设备本地应用列表的窃取,或者躲避对终端设备的某种检测,在本发明实施例中,终端设备可在识别到获取所述终端设备的原始应用列表的指令时,将所述指令的返回结果调整为聚类后的任一类所保留的应用列表,返回调整后的所述指令的返回结果,从而对所述指令的返回结果进行干扰,达到隐藏终端设备本地的应用列表的目的,以避免病毒程序对终端设备本地应用列表的窃取,或者躲避对终端设备的某种检测;

[0080] 图4示出了本发明实施例提供的修改指令返回结果的方法流程图,图4所示方法可应用于终端设备,参照图4,该方法可以包括:

[0081] 步骤S400、终端设备在确定各类所保留的应用列表后,若识别到获取所述终端设备的原始应用列表的指令,则根据所述指令进行hook,以将所述指令的返回结果调整为聚类后的任一类所保留的应用列表;

[0082] 其中,hook又称为钩子,实质上是一个处理消息的程序段,通过系统调用可将hook挂入系统中;

[0083] 可选的,步骤S400的触发前提可以是所述指令由外部的输入触发,如外部病毒程序触发所述指令,从而使得终端设备识别到获取所述终端设备的原始应用列表的指令;或者,检测到外部设备对终端设备的某种检测,从而使得终端设备识别到获取所述终端设备的原始应用列表的指令;

[0084] 显然,为实现对终端设备本地的原始应用列表的隐藏的目的,本发明实施例也可在只要识别到获取所述终端设备的原始应用列表的指令时,即根据所述指令进行hook;

[0085] 可选的,调整结果中所述聚类后的任一类所保留的应用列表,可以为与终端设备本地原始的应用列表不同的任一类所保留的应用列表。

[0086] 步骤S410、返回调整后的所述指令的返回结果。

[0087] 在具体应用中,本发明实施例可针对PackageManger管理器中的getInstalledApplications函数进行hook,当识别到获取所述终端设备的原始应用列表的指令时,所述指令将调用getInstalledApplications函数获取所述终端设备本地原始的应用列表;本发明实施例可在执行了getInstalledApplications函数的调用,在返回getInstalledApplications函数的返回结果之前,先调用afterHookedMethod函数,在afterHookedMethod函数中将getInstalledApplications函数的返回结果修改为,终端设备所确定的聚类后的任一类所保留的应用列表。

[0088] 可选的,在本发明实施例中,图4所示步骤S400和步骤S410可由终端设备内的处理器等处理芯片实现。

[0089] 可选的,图4所示方法可应用于躲避刷下载量检测的场景,从而在识别到获取所述终端设备的原始应用列表的指令时,修改所述指令的返回结果为聚类后的任一类所保留的应用列表,以使得所述终端设备返回的应用列表与真实用户相一致,从而使得终端设备不容易被检测工具识别为存在刷下载量的行为;显然,躲避刷下载量检测仅有本发明实施例的一个应用,本发明实施例也可应用于对病毒程序的干扰等场景。

[0090] 本发明实施例在实现对应用列表进行聚类的过程中,极大的降低了数据计算工作量,以较小的数据计算工作量,实现了对应用列表进行聚类的目的。

[0091] 下面对本发明实施例提供的应用列表聚类装置进行介绍,下文描述的应用列表聚类装置可与上文描述的应用列表聚类方法相互对应参照。

[0092] 图5为本发明实施例提供的应用列表聚类装置的结构框图,参照图5,该应用列表聚类装置可以包括:

[0093] 列表获取模块100,用于获取多个终端设备的应用列表;

[0094] 指纹值确定模块200,用于对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;

[0095] 聚类模块300,用于将指纹值符合设定聚类要求的应用列表归为一类。

[0096] 可选的,图6示出了本发明实施例提供的指纹值确定模块200的一种可选结构,参照图6,指纹值确定模块200可以包括:

[0097] 设置单元210,用于设置各应用列表对应的f维的向量V,及各应用列表对应的f位的数值S;

[0098] 指纹确定单元220,用于对于各应用列表,采用hash算法确定各特征的f位指纹b,一个所述特征由应用列表的至少一个应用标识构成;

[0099] 第一调整单元230,用于对于各应用列表,根据各特征的f位指纹b调整应用列表对应的f维的向量V,得到的各应用列表对应的f维的向量V的最终值;

[0100] 第二调整单元240,用于根据各应用列表对应的f维的向量V的最终值,确定各应用列表对应的数值S的最终值,以各应用列表对应的数值S的最终值作为各应用列表对应的指纹值。

[0101] 可选的,图7示出了本发明实施例提供的设置单元210的一种可选结构,参照图7,设置单元210可以包括:

[0102] 初始设置子单元211,用于设置各应用列表对应的f维的向量V,且初始化各应用列表对应的向量V为0,及设置各应用列表对应的f位的数值S,且初始化各应用列表对应的数值S为0。

[0103] 可选的,图8示出了本发明实施例提供的第一调整单元230的一种可选结构,参照图8,第一调整单元230可以包括:

[0104] 元素值调整子单元231,用于对于各应用列表的第1至第f个特征,若第i个特征的指纹b的第i位为1,i为1至f的整数,则在应用列表对应的向量V的第i个元素加上所述第i个特征对应的设定权重;若所述第i个特征的指纹b的第i位不为1,则在应用列表对应的向量V的第i个元素减去所述第i个特征对应的设定权重,得到各应用列表对应的f维的向量V的最终值。

[0105] 可选的,图9示出了本发明实施例提供的第二调整单元240的一种可选结构,参照图9,第二调整单元240可以包括:

[0106] 数值调整子单元241,用于若应用列表对应的向量V的最终值中第i个元素大于0,则将应用列表对应的数值S的第i位数值置为1;若应用列表对应的向量V的最终值中第i个元素不大于0,则将应用列表对应的数值S的第i位数值置为0,得到各应用列表对应的数值S的最终值。

[0107] 可选的,在本发明实施例中,可设置一个应用列表的应用标识按照序数,依序划分为多个特征;其中,各特征包含至少一个应用标识,且各特征所包含的应用标识的个数相同;

[0108] 可选的,若所述各特征包含多个应用标识,则可设置上一特征的最后一个应用标识,为下一特征的第一个应用标识。

[0109] 可选的,图10示出了本发明实施例提供的应用列表聚类装置的另一结构框图,结合图5和图10所示,该装置还可以包括:

[0110] 保留模块400,用于在将指纹值符合设定聚类要求的应用列表归为一类后,在相同的类中保留一个应用列表,去除所述类中的其他应用列表。

[0111] 可选的,本发明实施例提供的应用列表聚类装置可应用于终端设备中,当终端设备存在干扰病毒程序对终端设备本地的原始应用列表的获取目的,或者提交与真实用户相一致的应用列表,以躲避刷下载量的检测,等隐藏本地应用列表的目的时,本发明实施例可在终端设备识别到获取所述终端设备的原始应用列表的指令时,对所述指令的返回结果进行修改;对应的,图11示出了本发明实施例提供的应用列表聚类装置的再一结构框图,结合图10和图11所示,该装置还可以包括:

[0112] Hook模块500,用于终端设备在确定各类所保留的应用列表后,若识别到获取所述

终端设备的原始应用列表的指令,则根据所述指令进行hook,以将所述指令的返回结果调整为聚类后的任一类所保留的应用列表;

[0113] 结果返回模块600,用于返回调整后的所述指令的返回结果。

[0114] 本发明实施例提供的应用列表聚类装置,在实现对应用列表进行聚类的过程中,极大的降低了数据计算工作量,以较小的数据计算工作量,实现了对应用列表进行聚类的目的。

[0115] 本发明实施例还提供一种计算设备,该计算设备可以为服务器,由服务器实现对应用列表的聚类;该计算设备也可以为终端设备,由终端设备实现对应用列表的聚类,及本地应用列表的隐藏(通过在识别到获取终端设备的原始应用列表的指令时,修改所述指令的返回结果实现);在本发明实施例中,该计算设备可以包括上述所述的应用列表聚类装置。

[0116] 本发明实施例还提供有计算设备的硬件结构框图,图12为本发明实施例提供的计算设备的硬件结构框图,参照图12,该计算设备可以包括:处理器1,通信接口2,存储器3和通信总线4;

[0117] 其中处理器1、通信接口2、存储器3通过通信总线4完成相互间的通信;

[0118] 可选的,通信接口2可以为通信模块的接口,如GSM模块的接口;

[0119] 处理器1,用于执行程序;

[0120] 存储器3,用于存放程序;

[0121] 程序可以包括程序代码,所述程序代码包括计算机操作指令。

[0122] 处理器1可能是一个中央处理器CPU,或者是特定集成电路ASIC(Application Specific Integrated Circuit),或者是被配置成实施本发明实施例的一个或多个集成电路。

[0123] 存储器3可能包含高速RAM存储器,也可能还包括非易失性存储器(non-volatile memory),例如至少一个磁盘存储器。

[0124] 其中,程序可具体用于:

[0125] 获取多个终端设备的应用列表;

[0126] 对于各应用列表,以应用列表对应的有序应用标识为特征集合,基于Simhash算法确定各应用列表的特征集合所对应的指纹值;

[0127] 将指纹值符合设定聚类要求的应用列表归为一类。

[0128] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0129] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0130] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0131] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本发明。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明的精神或范围的情况下,在其它实施例中实现。因此,本发明将不会被限制于本文所示的这些实施例,而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

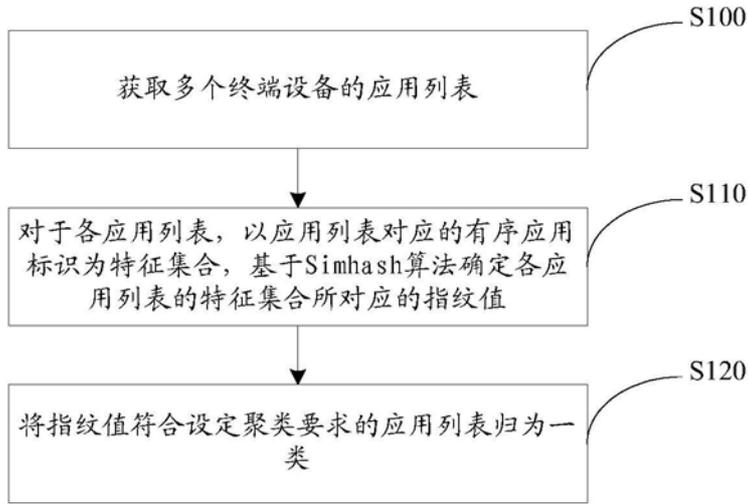


图1

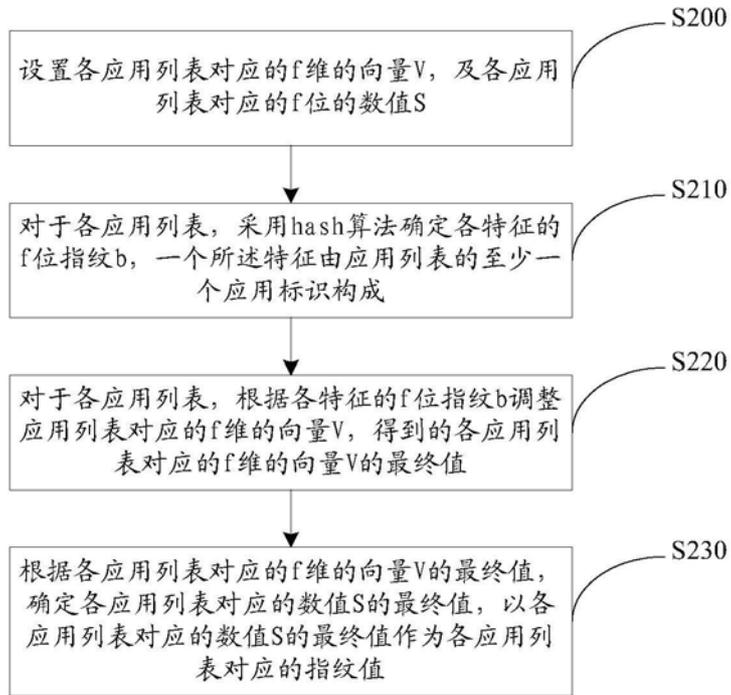


图2

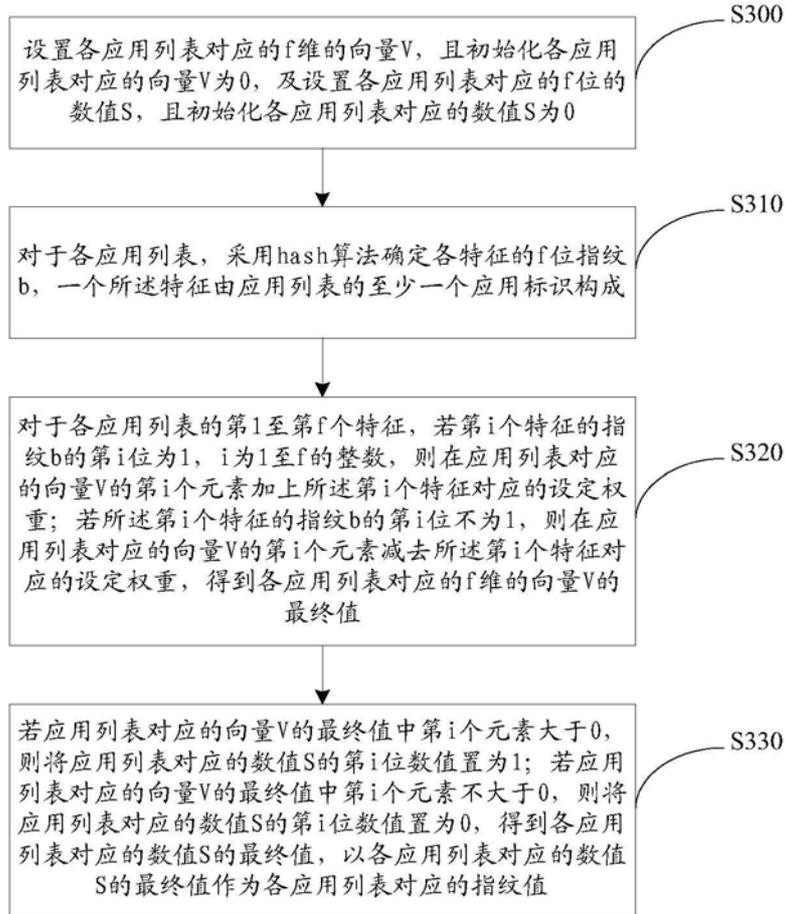


图3

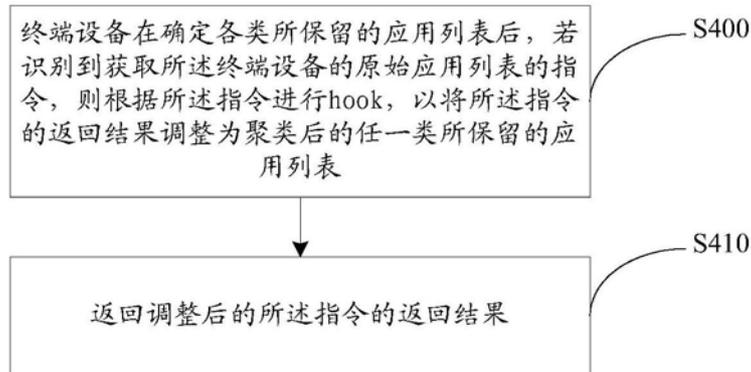


图4

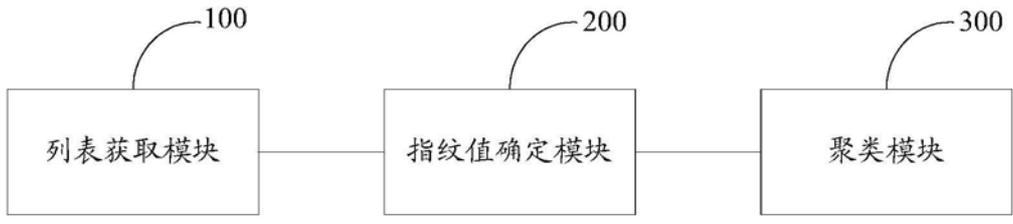


图5

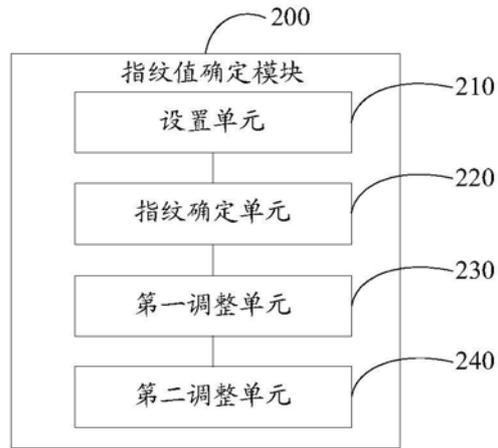


图6

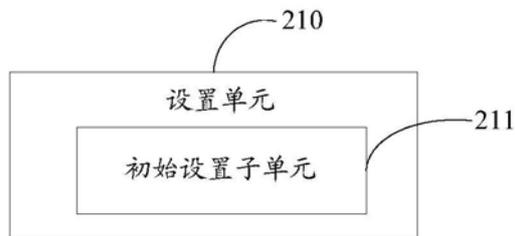


图7

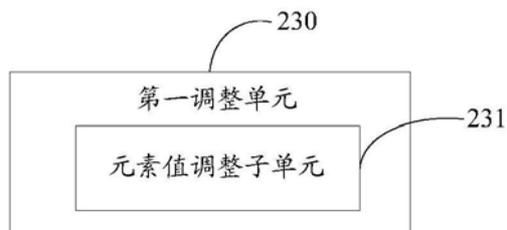


图8

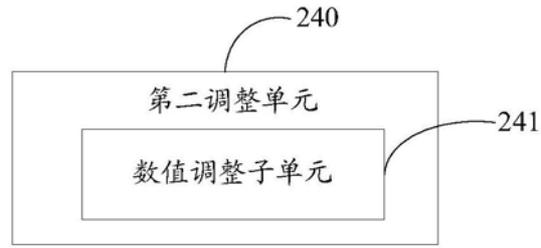


图9

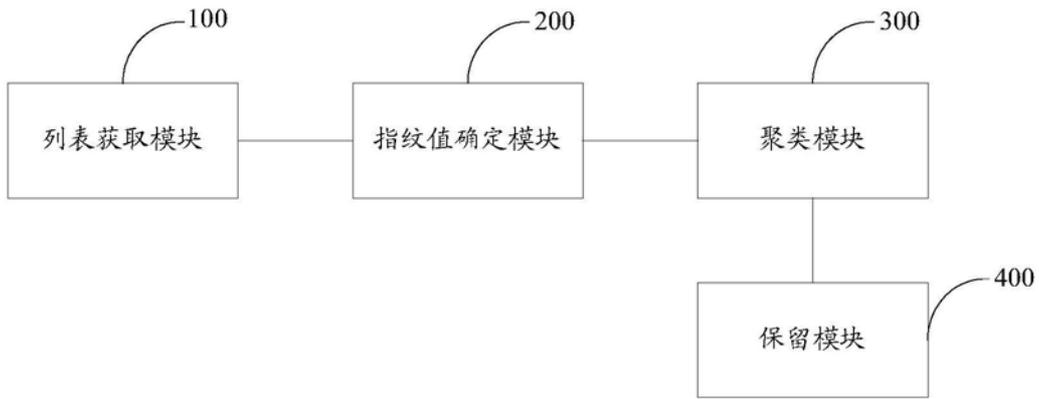


图10

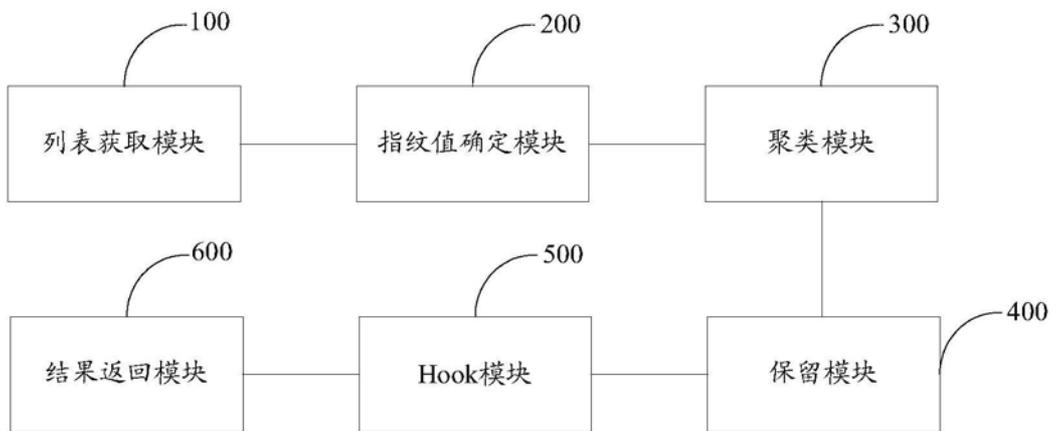


图11

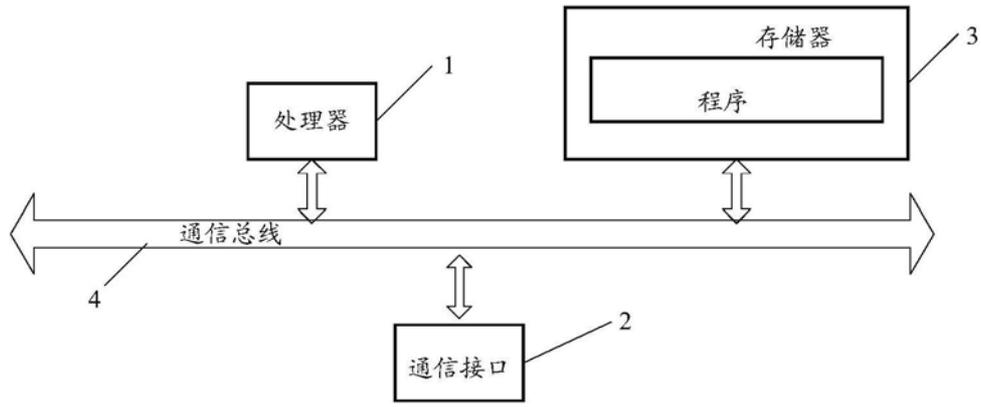


图12