



(12) 发明专利

(10) 授权公告号 CN 108351999 B

(45) 授权公告日 2021.11.16

(21) 申请号 201680066566.1

(22) 申请日 2016.11.03

(65) 同一申请的已公布的文献号
申请公布号 CN 108351999 A

(43) 申请公布日 2018.07.31

(30) 优先权数据
14/941,785 2015.11.16 US

(85) PCT国际申请进入国家阶段日
2018.05.15

(86) PCT国际申请的申请数据
PCT/US2016/060313 2016.11.03

(87) PCT国际申请的公布数据
W02017/087179 EN 2017.05.26

(73) 专利权人 甲骨文国际公司

地址 美国加利福尼亚

(72) 发明人 D·R·安德森 J·S·拜博
P·J·博安南

(74) 专利代理机构 中国贸促会专利商标事务所
有限公司 11038

代理人 王希

(51) Int.Cl.
G06Q 10/08 (2006.01)

(56) 对比文件
US 2015269617 A1, 2015.09.24
US 2014279294 A1, 2014.09.18
US 2013144757 A1, 2013.06.06

审查员 陈欢

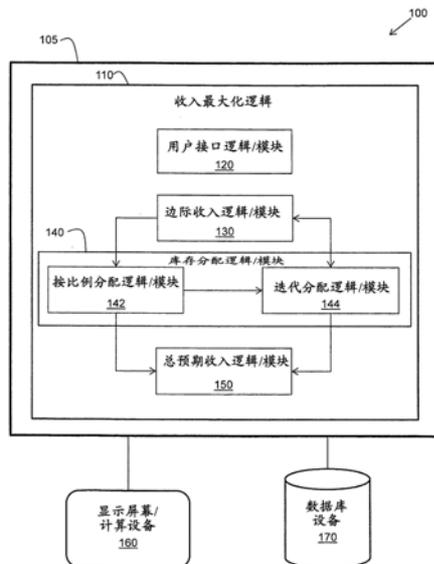
权利要求书3页 说明书13页 附图3页

(54) 发明名称

用于为零售商提供多渠道库存分配途径的系统和方法

(57) 摘要

公开了被配置为使零售项目的预期净收入最大化的系统、方法和其它实施例。在一个实施例中,读取与零售项目相关联的输入数据。可用于分发的零售项目的总库存量初始地跨销售零售项目的销售渠道分配,以形成如在库存分配数据结构中表示的多个被分配的库存量。试图均衡与如在边际收入数据结构中表示的与跨销售渠道的零售项目相关联的边际收入值。被分配的库存量被迭代地调整,并且边际收入值基于输入数据和销售渠道的调整后的被分配的库存量被迭代地更新。随着更新后的边际收入值跨销售渠道趋于均衡,预期净收入趋于最大化。



1. 一种由计算设备执行的计算机实现的用于跨多个销售渠道分配零售项目的总可用库存量的方法,其中所述计算设备至少包括用于执行来自存储器的指令的处理器,所述方法包括:

读取具有与零售项目相关联的输入数据的至少一个输入数据结构,其中所述输入数据包括销售零售项目的所述多个销售渠道中的每个销售渠道的(i)收入因子数据,其包括经由该销售渠道从零售项目的销售产生的预期净收入的生命周期值,(ii)统计需求数据,其包括基于该销售渠道处实际发生的不同需求水平的概率、零售项目在该销售渠道处被销售的可能性,和(iii)零售项目的当前库存水平数据;

基于所述至少一个输入数据结构,初始地跨所述多个销售渠道为零售项目分配总可用库存量,以形成在库存分配数据结构内表示的零售项目的针对所述多个销售渠道中的每个销售渠道的多个被分配的库存量;以及

通过执行以下迭代过程来跨所述多个销售渠道均衡与零售项目相关联的边际收入值,以最大化零售项目的预期净收入值,其中边际收入值包括由于销售渠道处的零售项目的被分配的库存量的增量变化而预期产生的附加增量收入量:

(i) 对于所述迭代过程的每次迭代,调整库存分配数据结构内的所述多个被分配的库存量同时保持所述总可用库存量不变,以及

(ii) 对于所述迭代过程的每次迭代,对于所述多个销售渠道中的每个销售渠道,至少部分地基于输入数据和经调整的所述多个被分配的库存量,生成边际收入数据结构内的更新后的边际收入值。

2. 如权利要求1所述的方法,还包括继续所述迭代过程,直到达到定义的最大迭代次数。

3. 如权利要求1所述的方法,还包括继续所述迭代过程,直到当前迭代与先前迭代之间的所述多个被分配的库存量的总变化的差异小于存储在数据字段中的阈值。

4. 如权利要求1所述的方法,还包括至少部分地基于在所述迭代过程完成之后的所述多个被分配的库存量,在数据字段中生成跨所述多个销售渠道的零售项目的总预期净收入值。

5. 如权利要求1所述的方法,还包括至少部分地基于紧挨在所述迭代过程开始之前的所述多个被分配的库存量,在数据字段中生成跨所述多个销售渠道的零售项目的总预期净收入值。

6. 如权利要求1所述的方法,还包括确定所述多个销售渠道中的每个销售渠道有资格接收与零售项目对应的库存。

7. 如权利要求1所述的方法,还包括至少部分地基于输入数据为所述多个销售渠道中的每个销售渠道确定在边际收入数据结构中表示的零售项目的初始边际收入值,其中跨所述多个销售渠道为零售项目初始分配总可用存货量与所述多个销售渠道中的每个销售渠道的初始边际收入值成比例地完成。

8. 如权利要求1所述的方法,还包括对于所述迭代过程的每次迭代,在数据字段中生成跨所述多个销售渠道的加权平均边际收入值,其中所述加权平均边际收入值通过总可用库存量进行加权。

9. 如权利要求8所述的方法,其中,所述迭代过程的每次迭代包括通过将所述多个销售

渠道中的每个销售渠道的边际收入值驱向所述加权平均边际收入值来调整所述多个被分配的库存量中的每个被分配的库存量。

10. 一种用于跨多个销售渠道分配零售项目的总可用库存量的计算系统,包括:

连接到存储器的处理器;

边际收入模块,所述边际收入模块包括存储在非瞬态计算机可读介质中并可由所述处理器执行的指令,所述指令使得所述处理器至少部分地基于输入数据来生成销售零售项目的多个销售渠道中的每个销售渠道的边际收入值,从而形成多个边际收入值,所述输入数据包括销售零售项目的所述多个销售渠道中的每个销售渠道的收入因子数据、统计需求数据和零售项目的库存量,其中收入因子数据包括经由该销售渠道从零售项目的销售产生的预期净收入的生命周期值,统计需求数据包括基于该销售渠道处实际发生的不同需求水平的概率、零售项目在该销售渠道处被销售的可能性;以及

库存分配模块,所述库存分配模块包括存储在非瞬态计算机可读介质中并可由所述处理器执行的指令,所述指令使得所述处理器试图通过以下来跨所述多个销售渠道均衡所述多个边际收入值,以最大化零售项目的预期净收入值:

(i) 基于所述输入数据,跨所述多个销售渠道为零售项目初始地分配总可用库存量以形成多个被分配的库存量,

(ii) 执行迭代过程以迭代地变换所述多个被分配的库存量同时保持所述总可用库存量不变,以及

(iii) 对于所述迭代过程的每次迭代,将所述多个被分配的库存量提供给所述边际收入模块,用于基于经变换的所述多个被分配的库存量更新地生成所述多个边际收入值,直到已满足迭代标准为止。

11. 如权利要求10所述的计算系统,其中所述库存分配模块被配置为通过确定已经达到定义的最大迭代次数来确定已满足所述迭代标准。

12. 如权利要求10所述的计算系统,其中所述库存分配模块被配置为通过确定当前迭代与先前迭代之间的所述多个被分配的库存量的总变化的差异小于阈值来确定已满足所述迭代标准。

13. 如权利要求10所述的计算系统,还包括总预期收入模块,所述总预期收入模块包括存储在非瞬态计算机可读介质中的指令,所述指令被配置为至少部分地基于已满足所述迭代标准之后的所述多个被分配的库存量生成跨所述多个销售渠道的零售项目的总预期净收入值。

14. 如权利要求10所述的计算系统,其中所述边际收入模块被配置为对于所述迭代过程的每次迭代,生成跨所述多个销售渠道的加权平均边际收入值,其中所述加权平均边际收入值通过总可用库存量进行加权。

15. 如权利要求14所述的计算系统,其中所述库存分配模块被配置为对于所述迭代过程的每次迭代,通过将所述多个销售渠道中的每个销售渠道的边际收入值驱向所述加权平均边际收入值来变换所述多个被分配的库存量中的每个被分配的库存量。

16. 一种由计算设备执行的计算机实现的用于跨多个销售渠道分配零售项目的总可用库存量的方法,其中所述计算设备至少包括用于执行来自存储器的指令的处理器,所述方法包括:

由边际收入模块至少部分地基于输入数据来生成销售零售项目的多个销售渠道中的每个销售渠道的边际收入值,从而形成多个边际收入值,所述输入数据包括销售零售项目的多个销售渠道中的每个销售渠道的收入因子数据、统计需求数据和零售项目的库存量,其中收入因子数据包括经由该销售渠道从零售项目的销售产生的预期净收入的生命周期值,统计需求数据包括基于销售渠道处实际发生的不同需求水平的概率、零售项目在该销售渠道处被销售的可能性;以及

由库存分配模块试图通过以下来跨所述多个销售渠道均衡多个边际收入值,以最大化零售项目的预期净收入值:

(i) 基于所述输入数据,跨所述多个销售渠道为零售项目初始地分配总可用库存量以形成多个被分配的库存量,

(ii) 执行迭代过程以迭代地变换所述多个被分配的库存量同时保持所述总可用库存量不变,以及

(iii) 对于所述迭代过程的每次迭代,将所述多个被分配的库存量提供给所述边际收入模块,用于基于经变换的所述多个被分配的库存量更新地生成所述多个边际收入值,直到已满足迭代标准为止。

17. 如权利要求16所述的方法,其中确定已满足所述迭代标准包括确定已经达到定义的最大迭代次数。

18. 如权利要求16所述的方法,其中确定已满足所述迭代标准包括确定当前迭代与先前迭代之间的所述多个被分配的库存量的总变化的差异小于阈值。

19. 如权利要求16所述的方法,还包括由总预期收入模块至少部分地基于已满足所述迭代标准之后的所述多个被分配的库存量生成跨所述多个销售渠道的零售项目的总预期净收入值。

20. 如权利要求16所述的方法,还包括对于所述迭代过程的每次迭代,生成跨所述多个销售渠道的加权平均边际收入值,其中所述加权平均边际收入值通过总可用库存量进行加权。

21. 如权利要求20所述的方法,其中对于所述迭代过程的每次迭代,变换所述多个被分配的库存量中的每个被分配的库存量包括:将所述多个销售渠道中的每个销售渠道的边际收入值驱向所述加权平均边际收入值。

22. 一种其上存储有指令的非瞬态计算机可读介质,其中所述指令当由处理器执行时,使得所述处理器执行根据权利要求1-9和16-21中任一项所述的方法的操作。

23. 一种包括用于执行根据权利要求1-9和16-21中任一项所述的方法的操作的部件的装置。

24. 一种计算系统,包括
处理器;和

其上存储有指令的非瞬态计算机可读介质,其中所述指令当由所述处理器执行时,使得所述处理器执行根据权利要求1-9中任一项所述的方法的操作。

用于为零售商提供多渠道库存分配途径的系统和方法

背景技术

[0001] 如今,零售商经常购买固定数量的库存,并跨地点和渠道划分库存以销售库存。例如,拥有网站和一百家商店的零售商可能必须为购买的每种个别类型的零售项目确定要为网络销售预留多少库存以及向每家商店发送多少。传统上,零售行业侧重于基于需求分配库存单元。另外,传统上,库存在商店之间进行划分,使得每个地点的库存水平与其需求成比例。

[0002] 但是,一些零售商已经使用“优先级矩阵(priority matrix)”途径来分配库存。优先级矩阵可以用于根据指派给不同类型的需求和不同商店的优先级来分配库存。例如,优先级矩阵可以将最高优先级指派给延期未交定货(backorder)、将第二优先级指派给展示存货、以及将第三优先级指派给安全存货。因此,优先级矩阵将首先基于需求优先级并且其次基于商店优先级跨商店分配库存。

[0003] 因此,零售行业侧重于根据需求或针对不同需求目的跨商店分发库存。当大多数地点类似时,这种侧重需求的途径是有意义的。但是,由于零售商拥有多种渠道并广泛利用本地化,因此这些途径是不合适的。

附图说明

[0004] 结合在本说明书中并构成其一部分的附图图示了本公开的各种系统、方法和其它实施例。将认识到的是,图中所图示的元件边界(例如,方框、方框的组,或者其它形状)表示边界的一个实施例。在一些实施例中,一个元件可以被设计为多个元件,或者多个元件可以被设计为一个元件。在一些实施例中,示为另一个元件的内部组件的元件可以被实现为外部组件,并且反之亦然。此外,元件可能不是按比例绘制的。

[0005] 图1图示了计算机系统的实施例,其具有配置有包括库存分配逻辑的收入最大化逻辑的计算设备;

[0006] 图2图示了可以由图1的计算机系统的收入最大化逻辑执行的方法的一个实施例,该方法用于跨销售渠道为零售项目分配总可用存货,其导致零售项目的预期净收入最大化;以及

[0007] 图3图示了可以在其上实现计算系统的收入最大化逻辑的计算设备的一个实施例。

发明内容

[0008] 在本公开的一个实施例中,公开了由计算设备执行的计算机实现的方法,其中计算设备至少包括用于执行来自存储器的指令的处理器。该方法包括:读取具有与零售项目相关联的输入数据的至少一个输入数据结构,该输入数据结构包括销售零售项目的多个销售渠道中的每个销售渠道的收入因子数据、统计需求数据和当前库存水平数据;初始地跨多个销售渠道为零售项目分配总可用库存量,以形成在库存分配数据结构内表示的多个分配库存量;以及试图通过执行以下迭代过程来均衡与跨多个销售渠道的零售项目相关联的

边际收入值,以最大化零售项目的预期净收入值:(i)对于迭代过程的每次迭代,调整库存分配数据结构内的多个被分配的库存量,以及(ii)对于迭代过程的每次迭代,对于多个销售渠道中的每个销售渠道,至少部分地基于输入数据和经调整的多个被分配的库存量,在边际收入数据结构内生成更新后的边际收入值。

[0009] 在另一个实施例中,该方法还包括继续迭代过程,直到达到所定义的最大迭代次数。

[0010] 在另一个实施例中,该方法还包括继续迭代过程,直到当前迭代与先前迭代之间的多个被分配的库存量的总变化的差异小于存储在数据字段中的阈值。

[0011] 在另一个实施例中,该方法还包括至少部分地基于在迭代过程完成之后的多个被分配的库存量,在数据字段中生成跨多个销售渠道的零售项目的总预期净收入值。

[0012] 在另一个实施例中,该方法还包括至少部分地基于在迭代过程开始之前即刻的多个被分配的库存量,在数据字段中生成跨多个销售渠道的零售项目的总预期净收入值。

[0013] 在另一个实施例中,该方法还包括确定多个销售渠道中的每个销售渠道有资格接收与零售项目对应的库存。

[0014] 在另一个实施例中,该方法还包括至少部分地基于输入数据为多个销售渠道中的每个销售渠道确定在边际收入数据结构中表示的零售项目的初始边际收入值,其中跨多个销售渠道为零售项目初始分配总可用存货量与多个销售渠道中的每个销售渠道的初始边际收入值成比例地完成。

[0015] 在另一个实施例中,该方法还包括对于迭代过程的每次迭代,在数据字段中生成跨多个销售渠道的加权平均边际收入值,其中加权平均边际收入值通过总可用库存量进行加权。迭代过程的每次迭代包括通过将多个销售渠道中的每个销售渠道的边际收入值驱向加权平均边际收入值来调整多个被分配的库存量中的每个被分配的库存量。

[0016] 在另一个实施例中,公开了一种包括连接到存储器的至少一个处理器的计算系统。该系统还包括边际收入模块,其包括存储在非瞬态计算机可读介质中并可由处理器执行的指令,以使得处理器至少部分地基于与零售项目相关联的收入因子数据、统计需求数据和库存量来生成其中销售零售项目的多个销售渠道中的每个销售渠道的边际收入值,从而形成多个边际收入值;以及库存分配模块,其包括存储在非瞬态计算机可读介质中并可由处理器执行的指令,以使得处理器试图通过以下来均衡跨多个销售渠道的多个边际收入值以最大化零售项目的预期净收入值:(i)跨多个销售渠道为零售项目初始地分配总可用库存量以形成多个被分配的库存量,(ii)执行迭代过程以迭代地变换多个被分配的库存量,以及(iii)对于迭代过程的每次迭代,将多个被分配的库存量提供给边际收入模块,直到满足迭代标准为止,用于更新地生成多个边际收入值。

[0017] 在计算系统的另一个实施例中,库存分配模块被配置为通过确定已经达到所定义的最大迭代次数来确定已满足迭代标准。

[0018] 在计算系统的另一个实施例中,库存分配模块被配置为通过确定当前迭代与先前迭代之间的多个被分配的库存量的总变化中的差异小于阈值来确定已满足迭代标准。

[0019] 在另一个实施例中,计算系统还包括总预期收入模块,其包括存储在非瞬态计算机可读介质中的指令,被配置为至少部分地基于满足迭代标准之后的多个被分配的库存量生成跨多个销售渠道的零售项目的总预期净收入值。

[0020] 在计算系统的另一个实施例中, 边际收入模块被配置为对于迭代过程的每次迭代, 生成跨多个销售渠道的加权平均边际收入值, 其中加权平均边际收入值通过总可用库存量进行加权。在另一个实施例中, 库存分配模块被配置为对于迭代过程的每次迭代, 通过将多个销售渠道中的每个销售渠道的边际收入值驱向加权平均边际收入值来变换多个被分配的库存量中的每个被分配的库存量。

具体实施方式

[0021] 公开了计算机化的系统、方法和其它实施例, 该计算机化的系统、方法和其它实施例用于确定如何跨多个销售渠道分配零售项目的库存, 以试图最大化预期由零售项目的预期销售产生的收入。考虑新的数据类型, 包括交叉销售机会、特定于客户的每单位平均收入 (ARPU)、商店级价格以及特定于渠道的运输成本。跨销售渠道的统计需求模式的差异也被考虑在内。示例实施例在本文关于计算机化零售管理进行讨论, 其中考虑了收入因子数据、统计需求数据和当前库存水平。

[0022] 在一个实施例中, 公开了收入最大化逻辑, 其被配置为考虑感兴趣的数学关系。数学关系指示, 当零售项目的库存以使得零售项目的边际收入值跨销售渠道被均衡的方式跨销售渠道进行分配时, 零售项目的预期收入可以被最大化。本文描述了迭代途径, 其允许系统导向库存的收入最大化分配。

[0023] 关于各种实施例使用了以下术语。

[0024] 如本文使用的, 术语“项目”或“零售项目”是指要在销售环境中销售、购买和/或返回的商品。

[0025] 如本文使用的, 术语“销售渠道”或“地点”可以指销售项目的实体商店或地点, 或者指经由其销售项目的在线商店。术语“销售渠道”和“地点”在本文中可互换使用。

[0026] 如本文使用的, 术语“净收入因子”是指经由销售渠道从零售项目的销售中产生的预期净收入的生命周期值。净收入因子取决于参数, 诸如, 例如, 在销售渠道处的零售项目的价格、在销售渠道处的交叉销售机会以及在销售渠道处的零售项目的全部每单位成本。交叉销售机会包括附加实体商品的预期销售和服务的预期销售, 所述服务例如电信实体的网络服务合同或电子零售商的服务计划。

[0027] 如本文使用的, 术语“统计需求”是指例如由需求分布函数 (即, 实际发生的不同需求水平的概率) 表示的零售项目要被销售的可能性。

[0028] 如本文使用的, 术语“总可用库存”是指可跨多个销售渠道分布 (分配) 的零售项目的单位的数量。

[0029] 如本文使用的, 术语“边际收入”是指由于销售渠道处的零售项目的库存量的增量变化而产生的附加增量的预期收入量。

[0030] 图1图示了计算机系统100的一个实施例, 其具有配置有收入最大化逻辑110的计算设备105。例如, 在一个实施例中, 收入最大化逻辑110可以是配置为预测和管理销售、促销、以及各个零售地点的零售项目的库存的较大计算机应用的一部分。收入最大化逻辑110被配置为使分析数据以分配零售项目的库存以便最大化预期收入的过程计算机化。在一个实施例中, 软件和计算设备105可以被配置为与基于云的联网系统、软件即服务 (SaaS) 体系架构或其它类型的计算解决方案一起操作或者被实现为基于云的联网系统、软件即服

务(SaaS)体系架构或其它类型的计算解决方案。

[0031] 零售商面临的一个问题是,跨销售渠道分配固定数量的库存以在需求不确定的情况下在特定时间段内产生最多的净收入。该问题在数学上可以被描述为在给定需求模式和收入因子的情况下,受限于可用库存,最大化在一段时间内销售的预期净收入。该问题可以被表达如下:

$$[0032] \quad \max_{v_1, v_2, \dots, v_n} \sum_{i=1}^N r_i * \left[\int_0^{z_i+v_i} x f_i(x) dx + \int_{z_i+v_i}^{\infty} (z_i + v_i) f_i(x) dx \right],$$

[0033] 受限于 $\sum_{i=1}^N v_i = V$,

[0034] 并且对于所有的 $i, v_i \geq 0$ 。

[0035] 其中:

[0036] N 是分配库存的地点销售渠道的总数,

[0037] i 是地点的索引,

[0038] z_i 是在地点 i 处现有的初始库存,

[0039] V 是该时段期间可用的库存总量,

[0040] v_i 是要分配给地点 i 的库存,

[0041] r_i 是地点 i 的净收入因子,以及

[0042] f_i 是地点 i 在时段期间的需求分布函数。

[0043] 注意的是,地点 i 的净收入因子是在该地点处从销售项目产生的预期净收入的生命周期值。净收入因子取决于,例如,该地点处的项目的价格、该地点处的交叉销售机会以及该地点处的全部每单位成本。交叉销售机会包括附加实体商品的预期销售和服务的预期销售,该服务诸如电信实体的网络服务合同或电子零售商的服务计划。净收入因子还考虑了由于特定于地点的物流成本而导致的商品预期单位成本的差异。

[0044] 一个实施例考虑了两个因子,这两个因子使得与需求成比例地分配库存是次优的。这两个因子是在每单位预期净收入方面的不同以及在相对需求变化性方面的不同。本文的途径通过开发迭代算法来解决该问题,该算法找到满足最优性条件的库存水平。最优性条件考虑了售出的每个单位的总净收入的差异和需求模式的差异。最优性条件适用于任何连续的需求分布。该途径还计算了与按比例分配相比,将由最优分配导致的收入的预期增加。

[0045] 电信行业提供了良好的示例来说明本文描述的分配途径如何可以在增加收入方面提供显著的益处。电信公司可能有许多销售渠道,包括许多实体商店以及在线展示。此外,电信公司可能每天处理数千个单位的例如特定类型的移动电话。使库存分配过程计算机化可能是有机会以最大化收入的方式分配库存的唯一可行方法。

[0046] 在电信行业,客户价值绑定(customer value banding)的技术通常用于将客户群分割成共享类似品质的组。例如,一个绑定组可能包括富有且旅途较多的商务人士,其倾向于使用昂贵的智能电话和相关配件。另一个绑定组可能包括倾向于使用价格较便宜的智能电话和配件的青少年女孩。此外,某些销售渠道可能与一个绑定组高度相关,并且其它销售渠道可能与另一个绑定组高度相关。

[0047] 在一个实施例中,当试图分配可用库存以最大化跨销售渠道的收入时,考虑这些客户价值集绑定。例如,在一个实施例中,客户价值绑定被考虑为净收入因子的一部分。在

另一个实施例中,可以引入考虑顾客价值绑定的单独因子。因此,电信公司可以在对简单地展示基本需求的销售渠道进行库存分配之前优先对服务最高绑定客户的销售渠道进行库存分配。以这种方式,客户影响被卷入解决方案空间,并且提供了显著更好的机会来最大化收入。

[0048] 最优性条件将更多库存发送给具有更高预期净收入因子的商店。基于定价或历史交叉销售的差异,通常可以预期某些地点产生更多的每销售单位的净收入。例如,在具有更多商业用户的中心商务区中销售电话手机的商店可能具有比其它区域的商店更高的服务每用户平均收入(ARPU)。将更多的手机交付给产生更高的每单位平均收入的商店将有助于将手机分配给最重视他们的客户,并将有助于增加零售商的收入。

[0049] 此外,最优性条件将更多的库存发送到具有较低相对销售变化性的商店。增加具有较低相对销售变化性的商店中的库存增加了收入,因为在库存太低而无法满足预期需求的情况下,这些商店将比具有较高销售变化性的商店更有可能销售它们接收到的库存。跨商店的相对销售变化性的差异是常见的。通常,较大的商店与较小的商店相比具有更低的相对销售变化性,并且应该与较小的商店相比接收相对更多的库存。对于多渠道零售商而言,网络渠道可以具有极大商店的需求特性,与实体商店相比具有低得多的需求变化性。

[0050] 根据一个实施例,用于实现优化条件的算法使用正态分布来对需求模式建模。正态分布使用平均销售和销售的标准偏差拟合到项目/地点级别需求。该算法通过迭代搜索不同的库存分配水平来求解优化条件。在每个迭代步骤,估算附加库存单位的边际收入的加权平均值,并且然后计算实现该边际收入的每个销售渠道的库存水平。加权平均边际收入值通过总可用库存量进行加权。

[0051] 对于某些销售渠道而言,计算出的库存水平可能为负值,这意味着不应该将任何库存发送到那个销售渠道。该途径的完整迭代对跨先前两个边际收入均衡步骤的库存进行平均。在单个步骤缓慢地朝最佳分配振荡的情况下,该平均可以有助于算法的加速。

[0052] 在一个实施例中,当迭代之间的库存水平的差异变得足够小时,或者当达到最大迭代次数时,算法停止搜索改进的解。在找到最优分配之后,该算法计算与按比例分配相比,使用最优分配的益处。知道这些益处可以有助于零售商理解与按比例分配相比使用最优分配的重要性。

[0053] 参考图1,在一个实施例中,收入最大化逻辑/模块110在计算设备105上实现,并且包括用于实现收入最大化逻辑/模块110的各种功能方面的逻辑。在一个实施例中,收入最大化逻辑/模块110包括用户接口逻辑/模块120、边际收入逻辑/模块130、库存分配逻辑/模块140和总预期收入逻辑/模块150。在一个实施例中,库存分配逻辑/模块140包括按比例分配逻辑/模块142和迭代分配逻辑/模块144。

[0054] 计算机系统100还包括可操作地连接到计算设备105的显示屏幕160。根据一个实施例,显示屏幕160被实现为显示用户与由用户接口逻辑120生成的用于查看和更新与最优库存分配相关联的信息的图形用户界面(GUI)交互的视图并且促进这种交互。图形用户界面可以与收入最大化应用相关联,并且用户接口逻辑120可以被配置为生成图形用户界面。在一个实施例中,收入最大化逻辑110是由许多客户端设备/用户访问的集中式服务器端应用。因此,显示屏幕160可以表示允许用户经由联网计算机通信从收入最大化逻辑110访问和接收服务的多个计算设备/终端。

[0055] 在一个实施例中,计算机系统100还包括至少一个数据库设备170,其可操作地连接到计算设备105和/或经由网络连接访问数据库设备170的网络接口。例如,在一个实施例中,数据库设备170可操作地连接到用户接口逻辑120。根据一个实施例,数据库设备170被配置为存储和管理与数据库系统(例如,计算机化的零售管理应用)中的收入最大化逻辑110相关联的数据结构(例如,收入因子数据、统计需求数据和库存水平数据的记录)。

[0056] 其它实施例可以提供不同的逻辑或逻辑组合,其提供与图1的收入最大化逻辑110相同或类似的功能。在一个实施例中,收入最大化逻辑110是可执行应用,包括被配置为执行逻辑的功能的算法和/或程序模块。应用被存储在非瞬态计算机存储介质中。即,在一个实施例中,收入最大化逻辑110的逻辑被实现为存储在计算机可读介质上的指令的模块。

[0057] 返回参考图1的收入最大化逻辑110的逻辑,在一个实施例中,用户接口逻辑120被配置为生成图形用户界面(GUI)以促进与收入最大化逻辑110的用户交互。例如,用户接口逻辑120包括程序代码,该程序代码基于所实现的界面的图形设计来生成图形用户界面并使得图形用户界面被显示。响应于经由GUI的用户动作和选择,可以操纵零售项目的收入最大化记录和参数的相关联方面。

[0058] 例如,在一个实施例中,用户接口逻辑120被配置为响应于用户动作来促进接收输入和读取数据。例如,用户接口逻辑120可以促进与跨多个销售渠道销售的零售项目相关联的零售数据(例如,收入因子数据、统计需求数据、当前库存水平数据)的选择和读取。零售数据可以驻留在与收入最大化应用(例如,收入最大化逻辑110)相关联(并且可经由图形用户界面由收入最大化应用访问)的至少一个数据结构中(例如,在数据库设备170内)。零售项目的零售数据(在可用时)可以经由网络通信被访问。零售项目的预期净收入的最大化可以至少部分地基于零售数据。

[0059] 此外,用户接口逻辑120被配置为经由显示屏幕160上的图形用户界面促进输出数据的输出和显示。输出数据可以包括例如库存分配数据、执行的迭代次数、最终迭代的分配中的变化、以及最终分配的预期收入。根据各种其它实施例,其它类型的输出数据也是可能的。

[0060] 再次参考图1,在一个实施例中,边际收入逻辑130被配置为为销售零售项目的多个销售渠道中的每个销售渠道生成边际收入值。因此,边际收入逻辑130形成多个边际收入值。边际收入值基于例如收入因子数据、统计需求数据和与零售项目相关联的库存量来生成。根据一个实施例,用于最大化零售项目的预期收入的最优性条件是,零售项目的所有销售渠道的边际收入是均衡的。这种均衡可以被表达为:

[0061] $\text{marginal_revenue}_i = \text{marginal_revenue}_j$, 或者

[0062] $r_i * [1 - F_i(z_i + v_i)] = r_j * [1 - F_j(z_j + v_j)]$,

[0063] 对于所有销售渠道*i*和*j*,其中 F_i 是销售渠道*i*处的概率密度函数 f_i (统计需求)的累积分布函数(CDF), r_i 是销售渠道*i*的净收入因子, z_i 是在销售渠道*i*处的零售项目的当前库存水平,以及 v_i 是要分配给销售渠道*i*的新库存量。

[0064] 当内部解不可行时,不能为所有销售渠道求解正库存值下的最优性条件。在这种情况下,不应该将库存量分配给具有比接收正库存量的销售渠道的值小的最优性条件值的销售渠道。

[0065] 在一个实施例中,库存分配逻辑140包括按比例分配逻辑142和迭代分配逻辑144。

按比例分配逻辑142被配置为与每个销售渠道的初始边际收入值成比例地为跨多个销售渠道的零售项目分配总可用存货量。根据一个实施例,每个销售渠道的初始边际收入值可以由边际收入逻辑130基于每个销售渠道处的零售项目的收入因子数据、统计需求数据和当前库存水平数据来确定。

[0066] 在一个实施例中,迭代分配逻辑144被配置为试图均衡跨多个销售渠道的多个边际收入值,以最大化零售项目的预期净收入值。例如,如上所述,根据一个实施例,按比例分配逻辑142可以初始地为零售项目跨多个销售渠道分配总可用库存量以形成多个被分配的库存量。然后,迭代分配逻辑144可以执行迭代过程以迭代地变换或调整多个被分配的库存量,同时维持总可用库存量。对于迭代过程的每次迭代,迭代分配逻辑144将多个被分配的库存量提供给边际收入逻辑130,使得边际收入逻辑130可以更新边际收入值。

[0067] 理想情况下,迭代过程可以继续,直到跨多个销售渠道的边际收入值均衡为止。但是,实际上,实现绝对平衡并非总是可能。根据一个实施例,继续迭代过程直到满足如由迭代分配逻辑144所确定的至少一个迭代标准。例如,迭代过程可以继续,直到达到所定义的最大迭代次数。替代地,迭代过程可以继续,直到当前迭代与先前迭代之间的多个被分配的库存量的总变化的差异小于阈值。

[0068] 在一个实施例中,总预期收入逻辑150被配置为至少部分地基于多个被分配的库存量来生成跨多个销售渠道的零售项目的总预期净收入值。总预期净收入值可以在满足迭代标准之后(即,一旦确定最终的被分配的存货量时)生成。替代地或附加地,可以在迭代过程开始之前立即生成总预期净收入值(即,基于零售项目的总可用存货量的初始分配)。

[0069] 以这种方式,收入最大化逻辑110(例如,实现为更大计算机应用的一部分)可以跨多个销售渠道分配零售项目的总可用存货量,以便最大化零售项目在一段时间内的预期净收入。因此,使用这种收入最大化逻辑110,零售商可以更智能地分配零售项目的可用库存。当已知存在产品短缺并且零售商正在尽可能好地度过短缺时段时,以这种方式智能地分配可用库存的能力在短期内可能变得尤为重要。

[0070] 图2图示了用于分配零售项目的可用库存以使预期净收入最大化的计算机实现的方法200的一个实施例。方法200描述了收入最大化逻辑110的操作,并且被实现为由图1的收入最大化逻辑110执行,或者由配置有方法200的算法的计算设备执行。例如,在一个实施例中,方法200由配置为执行计算机应用的计算设备实现。计算机应用被配置为处理电子形式的数据并且包括存储的执行方法200和/或其等价物的功能的可执行指令。

[0071] 将从以下观点来描述方法200:对于在不同地点(例如,销售渠道)销售的项目(例如,零售项目),为该项目新获取的库存在不同时间变得可用,并且将跨各种销售渠道进行分配。不是简单地基于例如每个销售渠道处的需求简单地分配新的库存,而是采取试图最大化预期净收入的更聪明的途径。方法200假定某些类型的零售数据可用(例如,来自数据库设备)于处理。零售数据可以包括例如销售零售项目的每个销售渠道的收入因子数据、统计需求数据和当前库存水平数据。

[0072] 在启动方法200之后,在方框210处,将零售数据输入(例如,读取或加载到)收入最大化逻辑110的输入数据结构。根据一个实施例,具有与零售项目相关联的输入数据的至少一个数据结构被读取。输入数据可以包括销售零售项目的多个销售渠道中的每个销售渠道的收入因子数据、统计需求数据和当前库存水平数据。更特别地,输入数据可以包括现有库

存的地点级别信息、预期生命周期净收入、平均销售额和销售额的标准偏差。

[0073] 此外,输入数据还可以包括要分配的总可用库存量、要运行的最大迭代次数以及用于停止迭代过程的变化阈值。本文假定总可用库存量是正数。此外,根据一个实施例,已被确定为作为迭代过程的收敛标准表现良好的值包括最大迭代次数为十(10)的值和变化阈值为0.1%的值。根据一个实施例,参考图1,零售数据可以由用户接口逻辑120促进的收入最大化逻辑110从数据库设备170读取。例如,用户接口逻辑120可以寻址数据库设备170的存储器以从存储在数据库设备170的存储器中的数据结构读取零售数据。用户接口逻辑120然后可以寻址计算设备105的存储器并将零售数据存储存储在计算设备105的存储器中。

[0074] 在方框220处,识别有资格接收新库存的销售渠道。根据一个实施例,如果销售渠道具有正的平均销售额、正的销售标准偏差和正的生命周期收入值,那么它们有资格接收库存。对于上面刚刚提到的三个量值中的任何一个,具有零或负值的销售渠道将不会经历分配过程。所有其它销售渠道将经历分配过程。根据一个实施例,用户接口逻辑120被配置为确定哪些销售渠道有资格接收库存。

[0075] 在方框230处,总可用存货量最初跨有资格的销售渠道与边际收入成比例地被分配,以形成在库存分配数据结构内表示的多个被分配的库存量。在初始库存水平 z_i ,每个销售渠道 i 的初始边际收入被确定为 $r_i * [1 - F_i(z_i)]$,其中 r_i 是生命周期收入因子并且 $F_i(z_i)$ 是当需求处于初始库存水平时,销售渠道的正态分布的CDF的值。根据一个实施例,初始分配由按比例分配逻辑142执行。

[0076] 总可用库存量在有资格地点之间与初始边际收入值成比例地划分,使得销售渠道 i 的分配库存水平 v_i 被表示为:

$$v_i = V * r_i * [1 - F_i(z_i)] / \sum_{j \in E} r_j * [1 - F_j(z_j)],$$

[0078] 其中 E 是有资格的销售渠道的集合。根据其它实施例,可以根据一些其它标准(不与边际收入成比例)跨有资格的销售渠道初始地分配总可用库存量。

[0079] 在方框240处,开始试图更新总可用库存量的跨销售渠道的分配,使得边际收入被均衡(即,使用边际收入均衡技术)的迭代过程。在方框250处,关于跨销售渠道的分配量的总变化是否小于变化阈值和/或是否已达到最大迭代次数进行检查。在方框250上进行详细说明,变化阈值可以被计算为 $\sum_{i \in E} |v_{ic} - v_{ip}| / V$,其中 v_{ic} 是地点 i 的当前库存分配,并且 v_{ip} 是地点 i 的先前库存分配。方框240和250构成试图使均衡所得到的边际收入值的跨销售渠道的分配量收敛的迭代过程。再次,均衡跨销售渠道的边际收入值使得零售项目的预期净收入最大化。根据一个实施例,方框240和250由迭代分配逻辑144与边际收入逻辑130合作来执行。

[0080] 对方框240进行详细说明,方框240使用在方框230中确定的初始库存水平或从先前迭代确定的库存水平来均衡边际收入。根据一个实施例,方框240针对有资格的销售渠道(地点)根据以下子步骤(i 至 v_i)进行:

[0081] i. 给定其当前分配 $z_i + v_i$,将每个地点 i 的边际收入计算为:如果 $v_i = 0$,则 $MR_i = 0$,否则, $MR_i = r_i * [1 - F_i(z_i + v_i)]$ 。

[0082] ii. 计算跨所有有资格地点的加权平均边际收入为:

$$[0083] \quad \overline{MR} = \frac{\sum_{j \in E} (z_j + v_j) * MR_j}{\sum_{j \in E} (z_j + v_j)}$$

[0084] iii. 计算每个地点*i*的查找值, 其为 $1 - \overline{MR}/r_i$ 。

[0085] iv. 计算对于每个地点*i*将实现 \overline{MR} 的总库存水平 t_i 。如果查找值小于或等于0, 则将 t_i 定义为 z_i 。如果查找值为正数, 则将 t_i 定义为 z_i 和在查找值处地点*i*的正态需求分布的倒数的最大值。

[0086] v. 将要分配给地点*i*的初始库存定义为 q_i , 其中 $q_i = t_i - z_i$ 。

[0087] vi. 与子步骤v的输出成比例地重新调整库存, 使得地点*i*接收等于 $V * q_i / \sum_{j \in E} q_j = v_i$ 的库存。

[0088] 使用前一次迭代的输出库存分配作为初始库存分配, 子步骤i至vi被重复地迭代, 并有效地将每个销售渠道的边际收入值驱向迭代过程的每次迭代的加权平均边际收入值。此外, 在一个实施例, 迭代过程对当前迭代和先前迭代的输出进行平均以便在库存水平在最佳值周围振荡的情况下加速收敛。

[0089] 总之, 对于方框240和250, 试图跨多个有资格销售渠道均衡与零售项目相关联的边际收入值, 以使零售项目的预期净收入值最大化。执行迭代过程, 其调整(变换)迭代过程的每次迭代的被分配的库存量。在迭代过程的每次迭代中, 至少部分地基于经调整的当前被分配的库存量, 为每个有资格的销售渠道生成更新后的边际收入值。

[0090] 在方框260处, 至少部分地基于迭代过程完成之后所得到的被分配的库存量, 生成跨销售渠道的零售项目的第一总预期净收入值。此外, 至少部分地基于与初始边际收入成比例的初始被分配的库存量, 生成跨销售渠道的零售项目的第二总预期净收入值。根据一个实施例, 方框260由总预期收入逻辑150执行。

[0091] 根据一个实施例, 在方框260处, 预期净收入被计算为生命周期收入乘以预期销售额减去预期损失销售额。预期销售额在方框210中输入。预期损失销售额使用查找值来计算。预期损失销售额取决于与库存水平平均值的标准偏差的数量。对于地点*i*, 与平均值的标准偏差的数量被计算为 $(z_i + v_i - \mu_i) / \sigma_i$ 。使用特殊的查找表来估计预期损失销售额, 并且最终的损失销售额估计是在查找表中找到的值乘以标准偏差 σ_i 。

[0092] 在方框270处, 输出最终分配结果(例如, 输出到输出数据结构)。输出包括获得的最终分配、按比例分配的分配、运行的迭代次数、最终迭代的分配的变化以及最终分配和按比例分配的预期收入。根据一个实施例, 方框270由用户接口逻辑120执行。例如, 用户接口逻辑120可以寻址计算设备105的存储器并且将最终分配结果存储到存储在计算设备105的存储器中的输出数据结构。

[0093] 以这种方式, 收入最大化和库存分配系统可以使用这个信息来为零售项目跨销售渠道最优地分发总可用库存量。由于收入最优问题的现实表达得到解决, 因此对于零售商而言, 净收入可以显著增加。可以通过将库存转移到具有更可靠需求的地点来改善客户服务。通过将库存转移到具有更高价格或更高交叉销售机会的地点可以实现更高的收入。

[0094] 已经描述了用于最大化跨多个销售渠道销售的零售项目的净收入的系统、方法和其它实施例。边际收入逻辑被配置为至少部分地基于与零售项目相关联的收入因子数据、统计需求数据以及库存量为每个销售渠道生成边际收入值, 从而形成多个边际收入值。库

存分配逻辑被配置为试图均衡跨销售渠道的多个边际收入值,以最大化零售项目的预期净收入值。库存分配逻辑初始地为跨销售渠道的零售项目分配总可用库存量以形成多个分配库存量。库存分配逻辑然后执行迭代地变换多个被分配的库存量的迭代过程。对于迭代过程的每次迭代,将多个被分配的库存量提供给边际收入逻辑,使得边际收入逻辑可以更新多个边际收入值。执行迭代过程直到满足迭代标准,此时多个边际收入值被均衡(或几乎如此)。

[0095] 计算设备实施例

[0096] 图3图示了用本文描述的示例系统和方法中的一个或多个和/或等同物配置和/或编程的示例计算设备。图3图示了计算设备的一个示例实施例,在该计算设备上可以实现收入最大化逻辑的实施例,以最大化由零售项目的销售产生的收入。示例计算设备可以是计算机300,其包括处理器302、存储器304以及通过总线308可操作地连接的输入/输出端口310。

[0097] 在一个示例中,计算机300可以包括如本文所公开的配置有编程算法的收入最大化逻辑330(对应于图1的收入最大化逻辑110),以迭代地调整零售项目的跨销售渠道的被分配的库存量,直到跨销售渠道的对应的边际收入值被均衡(或几乎如此)。在不同的示例中,逻辑330可以用硬件、具有存储的指令的非瞬态计算机可读介质、固件和/或其组合来实现。虽然逻辑330被图示为附连到总线308的硬件组件,但是应该认识到的是,在其它实施例中,逻辑330可以在处理器302中实现,存储在存储器304中或存储在盘306中。

[0098] 在一个实施例中,逻辑330或计算机300是用于执行所描述的动作的装置(例如,结构:硬件、非瞬态计算机可读介质、固件)。在一些实施例中,计算设备可以是在云计算系统中操作的服务器、以软件即服务(SaaS)体系架构配置的服务器、智能电话、笔记本电脑、平板计算设备等。

[0099] 例如,该装置可以被实现为ASIC,该ASIC被编程以促进由零售项目的销售所产生的收入最大化。该装置还可以被实现为存储的计算机可执行指令,该计算机可执行指令作为临时存储在存储器304中并且然后由处理器302执行的数据316呈现给计算机300。

[0100] 逻辑330还可以提供用于促进由零售项目的销售所产生的收入的最大化的装置(例如,硬件、存储可执行指令的非瞬态计算机可读介质、固件)。

[0101] 一般地描述计算机300的示例配置,处理器302可以是各种各样的处理器,包括双微处理器和其它多处理器体系架构。存储器304可以包括易失性存储器和/或非易失性存储器。非易失性存储器可以包括例如ROM、PROM等等。易失性存储器可以包括例如RAM、SRAM、DRAM等等。

[0102] 存储盘306可以经由例如输入/输出接口(例如,卡、设备)318和输入/输出端口310可操作地连接到计算机300。盘306可以是例如磁盘驱动器、固态盘驱动器、软盘驱动器、带驱动器、Zip驱动器、闪存卡、存储器棒等。此外,盘306可以是CD-ROM驱动器、CD-R驱动器、CD-RW驱动器、DVD ROM等等。例如,存储器304可以存储过程314和/或数据316。盘306和/或存储器304可以存储控制和分配计算机300的资源的操作系统。

[0103] 计算机300可以经由I/O接口318和输入/输出端口310与输入/输出设备交互。输入/输出设备可以是例如键盘、麦克风、指向和选择设备、相机、视频卡、显示器、盘306、网络设备320等等。输入/输出端口310可以包括例如串行端口、并行端口和USB端口。

[0104] 计算机300可以在网络环境中操作并且因此可以经由I/O接口318和/或I/O端口310连接到网络设备320。通过网络设备320,计算机300可以与网络进行交互。通过网络,计算机300可以在逻辑上连接到远程计算机。计算机300可以与之交互的网络包括但不限于LAN、WAN和其它网络。

[0105] 定义和其它实施例

[0106] 在另一个实施例中,所描述的方法和/或它们的等同物可以用计算机可执行指令来实现。因此,在一个实施例中,非瞬态计算机可读/存储介质被配置为具有存储的算法/可执行应用的计算机可执行指令,当该指令由(一个或多个)机器执行时使该(一个或多个)机器(和/或相关联的部件)执行所述方法。示例机器包括但不限于处理器、计算机、在云计算系统中操作的服务器、用软件即服务(SaaS)体系架构配置的服务器、智能电话等等。在一个实施例中,计算设备用被配置为执行任何所公开的方法的一个或多个可执行算法来实现。

[0107] 在一个或多个实施例中,所公开的方法或它们的等同物由以下任一项执行:被配置为执行所述方法的计算机硬件;或体现在非瞬态计算机可读介质中包括被配置为执行所述方法的可执行算法的计算机软件。

[0108] 虽然为了简化说明的目的,图中图示的方法被示出和描述为算法的一系列方框,但是应该认识到的是,这些方法不受方框的顺序的限制。一些方框可以以与所示出和描述的不同的顺序出现和/或与其它方框同时出现。而且,可以使用比全部图示的方框少的方框来实现示例方法。方框可以被组合或分成多个动作/部件。此外,附加的和/或替代的方法可以采用未在方框中图示的附加动作。

[0109] 以下包括本文所采用的所选择术语的定义。定义包括属于术语的范围并且可以用于实现的部件的各种示例和/或形式。示例并不旨在是限制性的。术语的单数和复数形式都可以在定义之内。

[0110] 对“一个实施例”、“实施例”、“一个示例”、“示例”等的引用指示如此描述的(一个或多个)实施例或(一个或多个)示例可以包括特定的特征、结构、特性、性质、元素或限制,但并非每个实施例或示例都必须包括该特定的特征、结构、特性、性质、元素或限制。此外,重复使用短语“在一个实施例中”不一定指代相同的实施例,但是可以指代相同的实施例。

[0111] ASIC:专用集成电路。

[0112] CD:紧凑型盘。

[0113] CD-R:CD可记录的。

[0114] CD-RW:CD可重写的。

[0115] DVD:数字多功能盘和/或数字视频盘。

[0116] HTTP:超文本传输协议。

[0117] LAN:局域网。

[0118] RAM:随机存取存储器。

[0119] DRAM:动态RAM。

[0120] SRAM:同步RAM。

[0121] ROM:只读存储器。

[0122] PROM:可编程ROM。

[0123] EPROM:可擦除PROM。

[0124] EEPROM:电可擦除PROM。

[0125] USB:通用串行总线。

[0126] WAN:广域网。

[0127] “可操作的连接”或实体通过其“可操作地连接”的连接是其中可以发送和/或接收信号、物理通信和/或逻辑通信的连接。可操作的连接可以包括物理接口、电接口和/或数据接口。可操作的连接可以包括足以允许可操作的控制的接口和/或连接的不同组合。例如，可以可操作地连接两个实体以将信号直接或通过一个或多个中间实体(例如，处理器、操作系统、逻辑，非瞬态计算机可读介质)彼此通信。可操作的连接可以包括生成数据并将数据存储存储在存储器中的一个实体以及经由例如指令控制从存储器中检索该数据的另一个实体。逻辑和/或物理通信通道可以用于创建可操作的连接。

[0128] 如本文使用的，“数据结构”是计算系统中存储在存储器、存储设备或其它计算机化系统中的数据的组织。数据结构可以是例如数据字段、数据文件、数据阵列、数据记录、数据库、数据表、图表、树、链表等中的任何一个。数据结构可以由许多其它数据结构形成以及包含许多其它数据结构(例如，数据库包括许多数据记录)。根据其它实施例，数据结构的其它示例也是可能的。

[0129] 如本文所使用的，“计算机可读介质”或“计算机存储介质”是指存储被配置为当被执行时执行所公开的功能中的一个或多个功能的指令和/或数据的非瞬态介质。计算机可读介质可以采取包括但不限于非易失性介质和易失性介质的形式。非易失性介质可以包括例如光盘、磁盘等。易失性介质可以包括例如半导体存储器、动态存储器等。计算机可读介质的常见形式可以包括但不限于软盘、柔性盘、硬盘、磁带、其它磁性介质、专用集成电路(ASIC)、可编程逻辑器件、紧凑型盘(CD)、其它光学介质、随机存取存储器(RAM)、只读存储器(ROM)、存储器芯片或卡、存储器棒、固态存储设备(SSD)、闪存驱动器、以及计算机、处理器或其它电子设备可以利用其工作的其它介质。如果每种类型的媒体在一个实施例中被选择用于实现，则其可以包括被配置为执行所公开的和/或所要求保护的功能中的一个或多个功能的算法的存储指令。

[0130] 如本文所使用的，“逻辑”表示利用计算机或电气硬件、具有存储的可执行应用或程序模块的指令的非瞬态介质和/或这些的组合实现的部件，以执行如本文公开的任何功能或动作，和/或使得来自另一逻辑、方法和/或系统的功能或动作如本文所公开的那样被执行。等效逻辑可以包括固件、利用算法编程的微处理器、离散逻辑(例如，ASIC)、至少一个电路、模拟电路、数字电路、编程的逻辑器件、包含算法的指令的存储器设备等，其中任何一个可以被配置为执行所公开的功能中的一个或多个功能。在一个实施例中，逻辑可以包括一个或多个门、门的组合、或者被配置为执行所公开的功能中的一个或多个能够的其它电路部件。在描述多个逻辑的情况下，有可能将多个逻辑合并到一个逻辑中。类似地，在描述单个逻辑的情况下，有可能在多个逻辑之间分配那单个逻辑。在一个实施例中，这些逻辑中的一个或多个是与执行所公开的和/或所要求保护的功能相关联的对应结构。选择实现哪种类型的逻辑可以基于期望的系统条件或规范。例如，如果考虑更高的速度，则将选择硬件来实现功能。如果考虑更低的成本，则将选择存储的指令/可执行应用来实现功能。

[0131] 如本文所使用的，“用户”包括但不限于一个或多个人、计算机或其它设备、或者这些的组合。

[0132] 虽然已经相当详细地图示和描述了所公开的实施例,但并不旨在将所附权利要求的范围限制或以任何方式限定到这样的细节。当然,不可能为了描述主题的各个方面而描述部件或方法的每种预期的组合。因此,本公开不限于所示出和描述的特定细节或说明性示例。因此,本公开旨在涵盖落入满足法定主题要求的所附权利要求的范围内的变更、修改和变化。

[0133] 就术语“包含”在具体实施方式或权利要求书中被采用的程度而言,它旨在以类似于当术语“包括”作为过渡词在权利要求中被采用时所解释的方式是包含性的。

[0134] 就术语“或”在具体实施方式或权利要求书中被采用的程度而言(例如,A或B),其旨在意味着“A或B或两者”。当申请人旨在指示“仅A或B但不是两者”时,那么将使用短语“仅A或B但不是两者”。因此,术语“或”在本文的使用是包含性的,而不是排他性使用。

[0135] 就短语“A、B和C中的一个或多个”在本文被使用的程度而言,(例如,被配置为存储A、B和C中的一个或多个的数据存储库)其旨在传达可能性A、B、C、AB、AC、BC和/ABC的集合(例如,数据存储库可以仅存储A、仅存储B、仅存储C、存储A&B、存储A&C、存储B&C和/或存储A&B&C)。并不旨在要求A中的一个、B中的一个和C中的一个。当申请人旨在指示“A中的至少一个、B中的至少一个和C中的至少一个”时,那么将使用短语“A中的至少一个、B中的至少一个和C中的至少一个”。

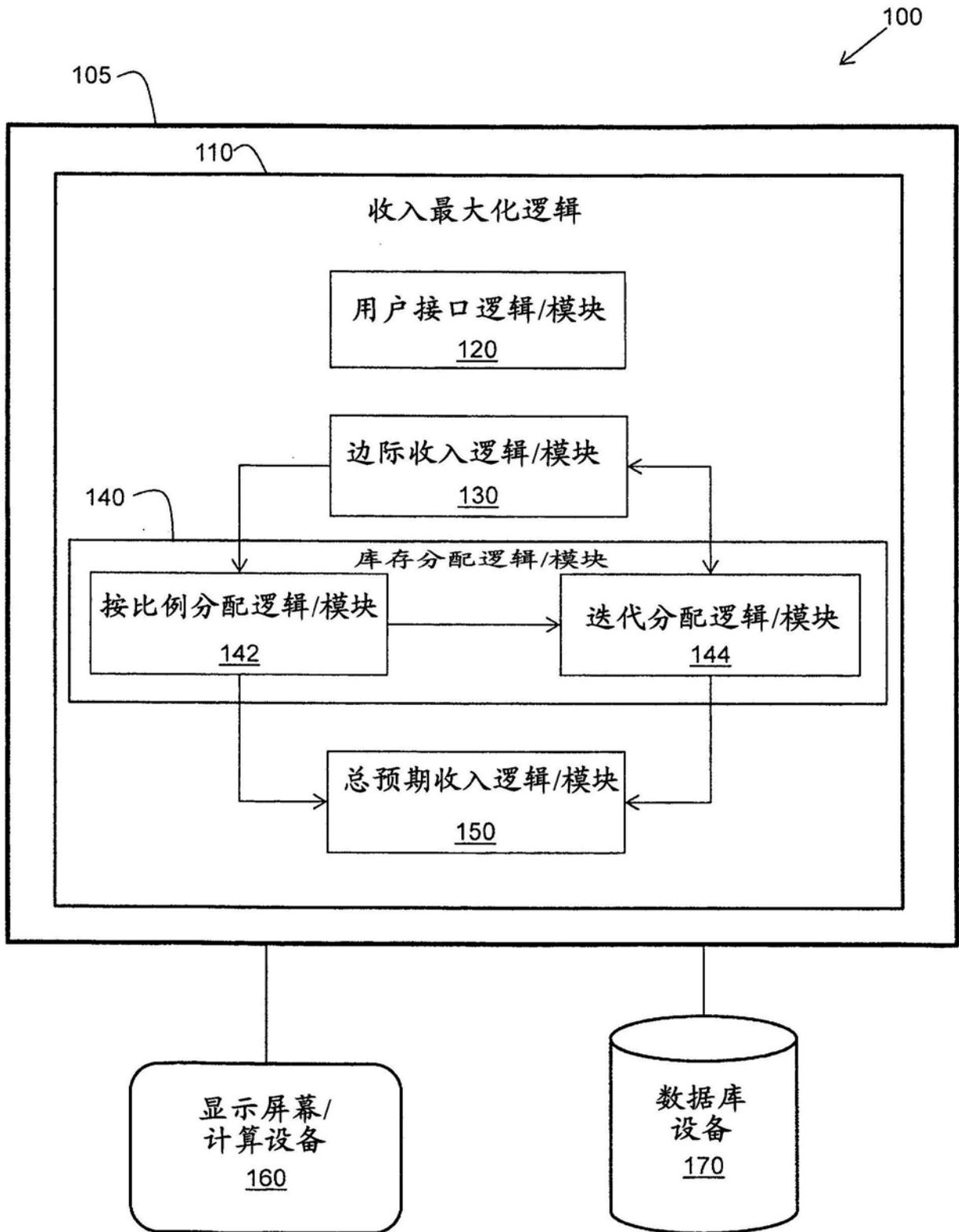


图1

200

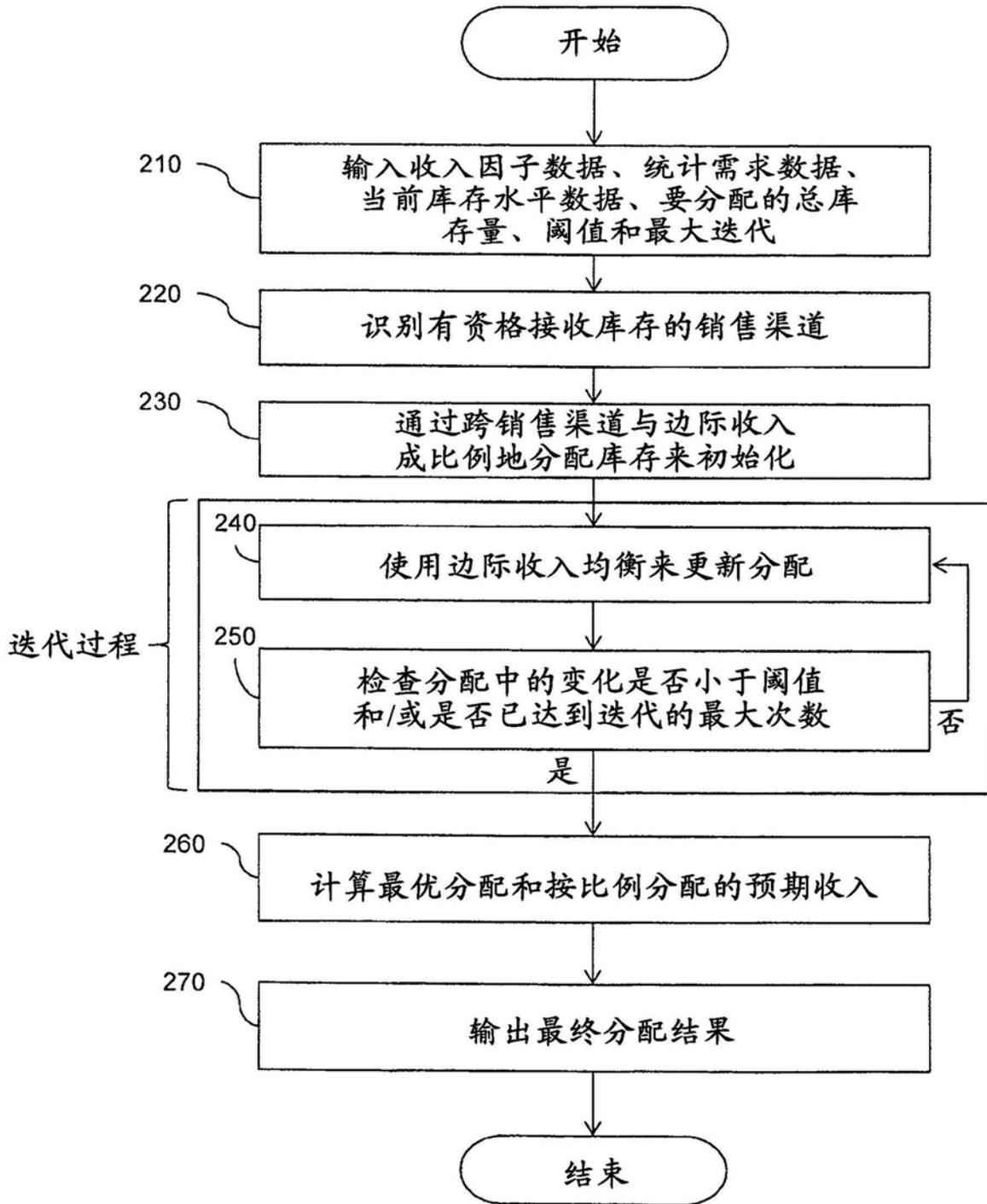


图2

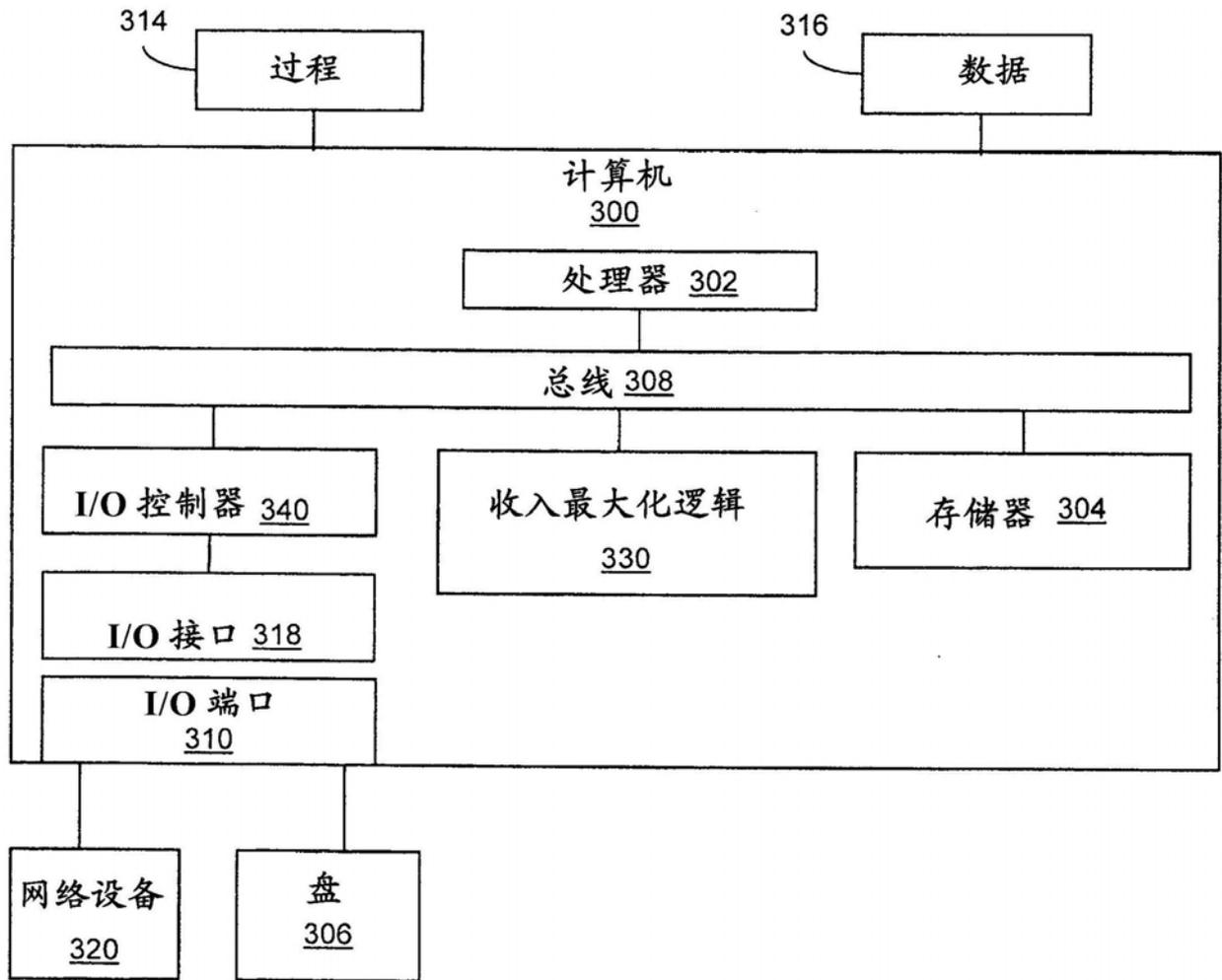


图3