

第 5 章

并行支持向量机的基本原理

前面几章的内容介绍了大数据预测的相关知识,而接下来的内容则是介绍大数据评价的相关方法以及具体实现过程。本章主要对进行大数据评价的并行支持向量机(Parallel Support Vector Machine,PSVM)的基本原理展开介绍,因此本章的内容将分为五节,分别为并行支持向量机概述、协同鸟群算法、支持向量机分类模型、并行支持向量机模型的构建以及知识扩展。

5.1 并行支持向量机概述

在对大数据进行分类时,最常用的机器学习算法为支持向量机,支持向量机的基本思想可以概括为通过非线性变换将输入空间映射到高维空间之后,求取一个最优的线性分类面从而将数据进行划分。在二维空间中这个线性分类面为一条直线;在三维空间中,这个线性分类面为一个平面,以此类推。

目前支持向量机已被证明在许多分类问题上具有较好的应用效果,例如 Edelman 将支持向量机应用于澳大利亚赛马比赛的小样本数据集中并对其中的 100 场比赛进行测试,结果显示支持向量机能够取得较好的分类结果;Takeda 等人针对支持向量机对异常值较为敏感的问题提出一种扩展健壮支持向量机,然后将其应用于金融风险度量,通过相关实验证明了在适当的参数设置下该模型具有更好的性能;Kim 等人针对在高度复杂的时间与空间上,实际实现的支持向量机分类效果无法达到理论上的预期水平这一问题,将 Boosting 算法与支持向量机相结合,然后将其应用于虹膜数据分类、手写数字识别等分类问题上,实验结果证明这一方法在分类精度上要优于单一支持向量机;Liu 等人利用支持向量机建立了城市土壤质量综合分类模型,并利用该模型对太原市土壤进行了分类评价,实验结果中,基于支持向量机的土壤质量模型能够较好地结合土壤数据,获得较好的分类

准确率。

支持向量机在近年来也得到了许多进展,例如 Suykens 等人在支持向量机的基础上提出了最小二乘支持向量机分类器,并以一个双螺旋基准分类问题为例验证了该分类器的有效性; Simon 等人提出一种基于支持向量机的主动学习算法,以此来避免使用随机选择的数据集,而是直接访问未被标记的实例并可以请求为其中一些实例添加标签,通过实验证明此方法可以显著减少归纳标准中需要标记的训练实例; Qi 等人在进行支持向量机特征权重分配时分别将加权特征消除算法与模糊加权特征消除算法应用到其中,以此来提高传统支持向量机的分类精度; Maali 等人则提出了一种提高支持向量机性能的新方法,此方法的主要目的是将更多的信息从训练阶段转移到测试阶段,且这些信息主要从训练阶段的错误分类中获得,实验结果表明,在不增加算法参数的情况下,这一方法可以显著提高支持向量机的分类性能。

虽然目前支持向量机在进行大数据分类时可以取得较好的效果,但在实际应用时仍存在一些问题,其中最主要的问题为其超参数的设置。支持向量机的超参数有许多,例如其分类模型原型形式和对偶形式中的惩罚系数 C , 其核函数中的参数 γ 等。这些参数直接影响模型最后的分类性能,因此在实验过程中需要采用交叉验证的方式选出最优的一组参数值,这一方式往往需要反复的实验,有时实验过后只能获得一个较优值而不是最优值,为了节省实验时间并获得更好的参数值,研究者通常将群智能优化算法与支持向量机相结合,使用群智能优化算法进行这些超参数的寻优过程。

并行支持向量机则继承了群智能优化算法与支持向量机相结合的思想,选用群智能优化算法中的鸟群算法(Bird Swarm Algorithm, BSA)进行支持向量机中超参数的寻优,在实际的建模过程中,以取值范围内不同数值的超参数组合作为鸟群算法的种群个体,以支持向量机的最终分类结果作为鸟群算法中每个个体的适应度值,将鸟群算法的个体寻优过程与支持向量机的训练过程相结合,并行展开多个支持向量机的分类训练,在训练结束后,将鸟群算法中表现最优的个体即最优参数组合赋值给支持向量机,以此作为最终的分类模型并展开大数据分类。

由于群智能优化算法本身仍具有一定的不确定性,因此在使用鸟群算法进行超参数寻优时容易出现陷入局部最优以及收敛速度较慢的问题,为了有效解决这一问题,在鸟群算法的基础上有针对性地在种群的寻优过程中为种群增加抱团行为以及接受准则,同时使用基于适应度差值比的位置更新方法来替代原有算法中的部分位置更新方法,以此提出了一种新的协同鸟群算法。最后使用协同鸟群算法替代传统的鸟群算法,将其与支持向量机相结合,组成并行支持向量机来实现大数据的分类过程。

5.2 协同鸟群算法

鸟群算法是由 Meng 等人于 2016 年提出的一种群智能优化算法,在研究者看来,自然界中的鸟类主要是以群居的方式进行生活,这些个体在群体中主要进行觅食与飞行等行为,而这些行为通常被认为是由分离、对齐和内聚等简单规则而产生的主要行为,通过这些简单的群体内行为,群体行为可以发展为更加复杂的运动与互动。

鸟类以群体的方式进行各种活动,这往往比以个体的形式进行活动所具有的生存优势与觅食效率更高。在鸟群进行觅食时,当种群中的某个个体在搜索范围内发现食物时,该个体会通过声音等方式将此信息传递给种群中的其他个体,以此来帮助其他个体进行觅食,同时在捕食的过程中,随时可能会出现威胁到个体的情况,例如捕食者的出现,因此个体也会经常抬头巡视周围的环境。上述两种状态即为鸟类觅食时的两种主要行为:捕食行为与保持警戒。研究表明,鸟类会在这两种状态中随机切换,一旦发现威胁到个体的情况,鸟类同样会通过声音的方式将信息传递给其他个体,当其他个体接收到信息后,整个种群会迅速飞离此环境来避免可能出现的威胁。由此可以看出,与以个体形式进行各种活动的鸟类相比,当鸟类以群体的方式进行各种活动时,个体之间的信息交流会更加频繁,这不仅可以让个体在觅食过程中减少自身受到捕食者威胁的风险,还可以增加个体进行觅食的时间以及觅食成功的概率,并且种群的规模越大,这种优势越能够体现。

当鸟群的种群规模达到一定程度,鸟群在觅食时将以一个包围圈的形式分布在觅食范围内,倘若此时出现捕食者并对鸟群进行攻击,捕食者的攻击目标更有可能是包围圈边缘的个体,而不是包围圈内部的个体,因此,在捕食过程中,鸟群个体将会更趋向于向它们自身所认为的鸟群中心进行移动,以此来避免受到捕食者的攻击,而这一移动行为一般在个体保持警戒的过程中进行。

通常鸟群在某个觅食范围内搜索完毕后或发现威胁的存在时,将会飞行至另一个觅食区域重新开始搜索,但是可以通过观察看到,在到达一个新的区域后并不是种群中的每一个个体都在寻找食物,往往存在这样一种现象,那就是部分个体会直接进食其他个体发现的食物,而不是通过自身的搜索获得,这些个体一般扮演者乞讨者的角色,而那些积极搜索食物的个体则扮演着生产者的角色。研究表明,鸟群中那些食物储量较低的个体通常是乞讨者,而那些食物储量较高的个体则是生产者。

由上述这些鸟群的群体行为可知,鸟类的群居生活给种群中的每个个体都带来更大的生存优势,每个个体都能从其他个体处获得一定的利益。而鸟群算法则是从上述这些群体行为中得以启发而设计出来的,它将鸟群的觅食过程与对客观问题的优化过程进行联系,将鸟群个体的觅食行为、保持警戒行为、飞行行为、生产行为、其他行为通过理想化的规则进行归纳,以此而得到的算法基本思路如下:

(1) 种群中所有个体通过信息共享来相互传递与觅食或威胁有关的信息,每个个体与整个种群都能记录和更新关于历史最佳捕食的信息;

(2) 每个个体在觅食的过程中会在觅食与保持警戒这两个状态中进行转换,同时这个转换具有随机性;

(3) 在保持警戒时,个体会试图向它所认为的鸟群中心进行移动,以此来避免自身受到捕食者的攻击,但此行为容易受到其他因素的影响,例如个体自身的位置信息;

(4) 当鸟群在某个区域的觅食结束后会以整个种群为单位进行飞行,从而到达一个新的觅食区域,通常每次飞行之间的间隔是固定的;

(5) 在鸟群飞行到达一个新的区域后,种群中的个体会分别扮演生产者与乞讨者的角色,种群中食物储备量最高的个体为生产者,食物储备量最低的个体为乞讨者,其他个体会随机选择一种角色;

(6) 扮演生产者角色的个体会积极地进行食物的搜索,而扮演乞讨者角色的个体则是在生产者发现食物后跟随其进行觅食。

按照最理想化的状态实施上述基本思路,从而得到鸟群算法的整个过程。

5.2.1 鸟群算法原理

鸟群算法的基本原理主要依托于鸟群的飞行行为、生产者行为、乞讨者行为、觅食行为以及保持警戒行为。下面将分别从这五方面进行介绍。

1. 飞行行为

由上述基本思路可以得知,鸟群在某个区域的觅食结束后会以整个种群为单位进行飞行来到达一个新的觅食区域,之后在新的区域种群个体将分别扮演生产者与跟随者,因此在鸟群算法中飞行行为主要用来判断鸟群个体是否要进行生产者与跟随者的相关行为。在算法寻优开始前会设置一个迭代周期,在每次达到预设的固定迭代周期时,种群中的个体将会按照自身的适应度值大小进行生产者与乞讨者两种角色的选择,而在其他迭代过程中,种群中的个体将随机选择觅食行为或警戒行为。

2. 生产者行为

在每次达到预设的固定迭代周期 FQ 时,整个种群将进行飞行行为来到达一个新的区域,此时个体将选择扮演生产者或乞讨者,在此算法中,每次迭代时种群中适应度值最高的个体将扮演生产者,适应度值最低的个体将扮演乞讨者,其他个体则会随机选择两种角色中的一种进行相关行为。当个体为生产者时,则使用下述公式进行自身的位置更新:

$$x_i^{t+1} = x_i^t + x_i^t * \text{randn}(0,1) \quad (5.1)$$

其中, x_i^t 为个体 i 在第 t 次迭代时的位置; x_i^{t+1} 为个体 i 在第 t 次迭代后的位置; $\text{randn}(0,1)$ 表示符合均值为 0、标准差为 1 的高斯分布的随机数。

3. 乞讨者行为

由前面的基本思路可知乞讨者主要在生产者发现食物后跟随其进行觅食,因此在其算法实现过程中通过下列公式进行乞讨者位置的更新:

$$x_i^{t+1} = x_i^t + (x_k^t - x_i^t) * \text{FL} * \text{rand}(0,1) \quad (5.2)$$

其中, x_k^t 为个体 k ($k \neq i$) 在第 t 次迭代时的位置; FL 表示乞讨者跟随生产者寻找食物的行为,取值范围为 $[0,2]$; $\text{rand}(0,1)$ 表示 $0 \sim 1$ 的随机数。

4. 觅食行为

当种群未进行飞行行为时,则会进行觅食行为或保持警戒,且种群中的所有个体将会在这两种状态中随机选择。在此将设定一个常量 p , p 的取值范围为 $0 \sim 1$,当每次迭代判断种群个体不进行飞行行为时,将针对每个个体生成一个 $0 \sim 1$ 的均匀分布的随机数,当此数小于常量 p 时,个体则进行觅食行为,否则个体保持警戒。当个体进行觅食行为时,将使用下列公式对个体的位置进行更新:

$$x_i^{t+1} = x_i^t + (p_i - x_i^t) * C * \text{rand} + (g_i - x_i^t) * S * \text{rand}(0,1) \quad (5.3)$$

其中, p_i 为个体 i 在当前迭代次数下的历史最优位置信息; g_i 为整个种群在当前迭代次数下的历史最优位置信息; C 为感知系数; S 为社会加速系数, rand 表示 $0 \sim 1$ 的均匀分布的随机数。

5. 保持警戒行为

当种群个体切换到保持警戒的状态时,由前面的基本思路可知个体试图向它所认为的鸟群中心进行移动,这一移动方式可通过下列公式进行描述:

$$x_i^{t+1} = x_i^t + A_1 * (\text{mean} - x_i^t) * \text{rand}(0,1) + A_2 * (p_k - x_i^t) * \text{rand}(-1,1) \quad (5.4)$$

$$A_1 = a_1 * \exp\left(-\frac{\text{pFit}_i}{\text{sumFit} + \epsilon} * N\right) \quad (5.5)$$

$$A_2 = a_2 * \exp\left(-\frac{\text{pFit}_i - \text{pFit}_k}{|\text{pFit}_i - \text{pFit}_k| + \epsilon} * N * \frac{\text{pFit}_i}{\text{sumFit} + \epsilon}\right) \quad (5.6)$$

其中, A_1 表示个体向种群中心移动的距离; A_2 表示个体向种群最优位置移动的距离; mean 表示整个种群的平均位置; $\text{rand}(0,1)$ 表示 $0 \sim 1$ 的随机数; $\text{rand}(-1,1)$ 表示 $-1 \sim 1$ 的随机数; p_k 表示个体 k ($k \neq i$) 在第 t 次迭代时的历史最优位置; a_1 与 a_2 均为 $0 \sim 1$ 的常量; pFit_i 为个体 i 在第 t 次迭代时的历史最优适应度值; sumFit 为整个种群在第 t 次迭代时的历史最优适应度值之和; N 为种群的个体数量; pFit_k 为个体 k ($k \neq i$) 在第 t 次迭代时的历史最优适应度值; ϵ 为避免分母部分为 0 而设置的极小值。

在公式(5.4)中,个体的位置更新主要分为两部分。前一部分表示当个体向它所认为的鸟群中心移动时,它不可避免地会受到来自种群中其他个体位置的影响,这主要是因为所有个体都想向中心位置移动,即所有个体之间也存在一定的竞争关系,因此使用整个种群的平均位置来表示其他个体对此个体带来的影响,而此影响不应过大,即 A_1 与 $\text{rand}(0,1)$ 的乘积需小于 1,具体的影响大小则通过个体适应度值的比较来确定。后一部分表示个体在移动的过程中直接受到某个特定的个体 k 的影响,当个体 k 的历史最优适应度值大于个体 i 的历史最优适应度值时, $A_2 > a_2$, 则个体 i 相较于个体 k 会受到更大的影响。另外,上述适应度的比较均基于算法用于最小化问题寻优时展开,适应度值越小,则个体的适应度越好。

此外对算法中存在的主要参数进行归纳,鸟群算法中的主要参数分别为:鸟群的飞行间隔 FQ ; 表示乞讨者跟随生产者寻找食物的行为的参数 FL ,取值范围为 $[0,2]$; 感知系数 C 与社会加速系数 S ,这两个参数一般均取值为 1.5; 保持警戒行为中的常量 a_1 与 a_2 ,这两个参数取值范围为 $0 \sim 1$,一般均取值为 0.5。

结合上述五种行为的基本原理可以了解到使用鸟群算法进行优化问题的求解时主要具有以下九个特性:

(1) 鸟群中的信息共享可以使算法在迭代过程中保存和更新个体的历史最优位置信息与最优适应度值以及整个种群的历史最优位置信息与最优适应度值;

(2) 鸟群个体通过对位置进行移动从而在目标区域内搜索猎物,此搜索区域可以为

多维,而不仅仅是二维,即求解的优化问题可以是多维的;

(3) 飞行间隔 FQ 控制个体通过何种方式进行位置的移动;

(4) 飞行行为中生产者通过随机方式进行位置移动,扩大了种群的搜索范围;

(5) 非飞行行为时个体通过向种群内部表现更优的位置信息进行学习来进行位置移动,增加了种群内部信息的利用率;

(6) 感知系数 C 和社会加速系数 S 可以帮助算法平衡外部信息的探索与内部信息的利用这两个过程;

(7) 将适应度值的比较加入保持警戒过程中,位置移动增加了个体的自适应性;

(8) 当 A_1 较大时,个体的位置移动受到种群整体的影响较大,当 A_2 较大时,个体的位置移动受到种群中某个个体的影响较大;

(9) 鸟群算法中主要有两个参数需要进行调整,这两个参数分别为飞行间隔 FQ 与参数 FL 。

使用鸟群算法对优化问题进行求解时的具体步骤可以归纳如下:

(1) 设置种群的个体数量、飞行间隔 FQ 、表示乞讨者跟随生产者寻找食物的行为的参数 FL 、感知系数 C 与社会加速系数 S 以及保持警戒行为中的常量 a_1 与 a_2 ;

(2) 以鸟群个体的位置信息作为待优化问题的解,根据待优化问题的解的范围,随机初始化种群所有个体的位置信息;

(3) 根据待求解问题,计算种群中每个个体的适应度值,将其作为每个个体的历史最优适应度值,将此时对应的位置信息作为每个个体的历史最优位置信息,之后对种群个体的适应度值进行比较,将最高适应度值作为种群的历史最优适应度值,将其对应的位置信息作为种群的历史最优位置信息;

(4) 根据飞行间隔 FQ 判断个体是否进行飞行行为,若为飞行行为,则为每个个体分配生产者以及乞讨者的身份,适应度值最高的个体将扮演生产者,适应度值最低的个体将扮演乞讨者,其他个体则会随机选择两种角色中的一种,之后根据相关公式进行位置更新;

(5) 若不为飞行行为,则根据常量 p 与随机数的比较判断个体是进行觅食行为还是保持警戒,并对位置信息进行更新;

(6) 计算每个个体更新后的位置信息以及对应的适应度值,对个体的历史最优适应度值、历史最优位置信息以及种群的历史最优适应度值、历史最优位置信息进行更新;

(7) 根据预设的迭代次数重复步骤(4)~步骤(6),当达到最大迭代次数时停止迭代过程,输出种群的历史最优位置信息,此位置信息即为算法优化后获得的问题最优解。

按照上述过程对鸟群算法进行归纳,如算法 5.1 所示。

算法 5.1 鸟群算法

- 1: 初始化鸟群种群中每个个体的位置信息
- 2: 设置种群数量 N 、飞行间隔 FQ 、常量 p 、参数 FL 、 C 、 S 、 a_1 与 a_2
- 3: 根据待求解问题计算每个个体的适应度值
- 4: 确定种群的历史最优位置相关信息以及个体历史最优位置相关信息

```
5: while(未满足停止迭代条件)do
6:   if 达到飞行间隔
7:     将种群个体分为生产者与乞讨者
8:     for  $i = 1: N$ 
9:       if  $i$  为生产者
10:        生产行为
11:       else
12:        乞讨行为
13:       end if
14:     end for
15:   else
16:     for  $i = 1: N$ 
17:       if  $\text{rand}(0,1) < p$ 
18:        觅食行为
19:       else
20:        保持警戒
21:       end if
22:     end for
23:   计算位置更新后的适应度值
24:   更新种群的历史最优位置相关信息以及个体历史最优位置相关信息
25: end while
26: 输出种群的历史最优位置信息
```

5.2.2 抱团行为

在鸟群算法中,当鸟群个体进行大部分行为时,其位置移动主要是朝着种群中保存的较高适应度相关位置方向进行移动,例如在觅食行为中,个体进行位置移动时主要对个体历史最优位置信息以及种群历史最优位置信息进行学习,因此容易导致种群的位置多样性降低,易陷入局部最优。而在自然界中存在一种现象,那就是当鸟群种群规模较大时,鸟群在进行飞行或觅食行为的过程中,鸟群中的个体会根据彼此之间的距离划分为多个小团体,个体的行为主要受到小团体的影响,一旦再次进行飞行或觅食行为时,鸟群内每个个体的位置发生变化,小团体的个体组成也会随之改变,在这种方式下每个个体之间的信息交换则会更加随机,受到这一现象的启发,协同鸟群算法在鸟群算法的基础上为种群个体增加了抱团行为。

鸟群的抱团行为主要发生在每次迭代时判断种群是否进行飞行行为之前,此时整个种群将会随机分配成若干小团体,此行为的执行步骤如下:

- (1) 随机在整个种群中选择若干个体作为小团体的中心点,选择的个体的总数即为小团体的总数;
- (2) 分别计算每个个体与每个小团体中心点的空间距离,按照就近优先原则等量依次将个体分配至每个小团体;
- (3) 在之后的觅食行为、警戒行为以及乞讨者行为中,每个个体对小团体中的最优个

体进行位置学习迁移,例如觅食行为的更新公式中需要对整个种群的历史最优位置信息进行学习,而在新的更新公式中使用小团体中的历史最优位置信息替换种群的历史最优位置信息,其他行为的位置更新公式的修改方式与之相同。

(4) 在对每个个体的位置、个体历史最优位置与种群历史最优位置更新完毕后,对每个小团体进行重组,即重新随机确定若干中心点,并根据个体与中心点的空间距离确定新的小团体个体构成,通过比较新的小团体中每个个体的历史最优位置信息来确定新的小团体历史最优位置信息。

为鸟群增加抱团行为,首先可以帮助鸟群在位置移动时获得更多方向上的选择,随着小团体数量的增加,鸟群可选择的飞行方向也随之增加,朝着不同方向的飞行过程也是种群朝着周围位置发散的过程,相较于鸟群算法中种群朝着一个方向的收敛过程,这一方式可以扩大种群的搜索范围,提高种群的多样性。而在每次飞行或觅食行为时均进行抱团行为,旨在调整每次迭代时个体的飞行方向,避免种群中的个体一直朝着同一方向移动而陷入局部最优。另外,在觅食行为等位置更新公式中,个体位置移动的距离受到被学习个体的适应度值的影响,抱团行为可以使个体在每次迭代中选择不同的个体进行学习。若被学习个体的适应度值较之前有所增加,则可以加快个体朝向更好位置的移动速度以提高其适应度值;若被学习个体的适应度值较之前降低,移动速度降低,则可以通过移动方向的改变来增加发现更多较优解的概率。

5.2.3 基于适应度差值比的位置更新方式

在个体进行位置学习移动时,参数 FL 代表乞讨者跟随生产者寻找食物,该值越大则学习的量越多,越小则学习的量越少。传统鸟群算法中将 FL 设为固定值,但是在实际的学习过程中,除了种群中适应度值最高的个体外,其他的生产者均为随机选择的,因此每个被学习个体的位置好坏无法得到保证,倘若学习位置较差个体的学习量与学习位置较好个体的学习一致,则会直接影响整个算法的收敛速度,因此采用固定值 FL 是不合理的,每个乞讨者在跟随生产者进行位置移动时具体的学习量应该视生产者的适应度值大小而定。

基于适应度差值比的正弦变换位置更新将正弦变换作为 FL 的变化方式,之后分别计算学习个体与被学习个体间的适应度差值和最优个体与最差个体间的适应度差值,并将两者的比值作为 FL 变化的自变量,其具体的计算公式如下:

$$FL_i = (FL_{\max} - FL_{\min}) + FL_{\min} \cdot \sin\left(\frac{\pi |f_k - f_i|}{2(f_{\max} - f_{\min})}\right) \quad (5.7)$$

其中, FL_{\max} 与 FL_{\min} 为 0~2 的常数, FL_{\max} 大于 FL_{\min} , 确保了 FL 的变化范围与原算法中的取值范围相同; f_{\max} 为小团体中的最高适应度值; f_{\min} 为小团体中的最小适应度值; f_k 为被学习的个体 k 的适应度值; f_i 为学习个体 i 的适应度值。假设 FL_{\max} 取值为 2, FL_{\min} 取值为 1, 则 FL 随适应度差值比的变化过程如图 5.1 所示。

在图 5.1 中 FL 的大小主要受被学习个体适应度值的影响且主要存在两种情况:一种是学习个体的适应度值低于被学习个体,此时 FL 随被学习个体适应度值的增大而增大,个体学习较优位置的量也随之增加,以此向更好的位置移动;另一种是学习个体的适

应度值高于被学习个体,此时 FL 随被学习个体适应度值的减小而增大,个体向周围区域探索的量也随之增加,以此发现更多可能存在的较优解以避免陷入局部最优。

另外由图 5.1 中的曲线变化情况可以看到,在前期适应度差值比较小时,FL 具有较快的增长速度以及较大的增加幅度,主要是因为此时学习个体与被学习个体之间的适应度值较为接近,个体之间的学习不一定能够帮助其获得更好的适应度值,通过 FL 的快速增长,可以使个体快速获得较大的学习的量,以此来避免适应度差值较小对个体位置学习带来的影响。而随着适应度差值比逐渐接近 1,学习个体与被学习个体之间的适应度差值也逐渐接近种群中的最优个体与最差个体之间的适应度差值,此时 FL 已取得一个较高值,可以满足低适应度个体学习量的需求以及高适应度个体向周围探索的需求,因此 FL 无须再进行增加,而是令其增长速度与增加幅度逐渐减小至 0。

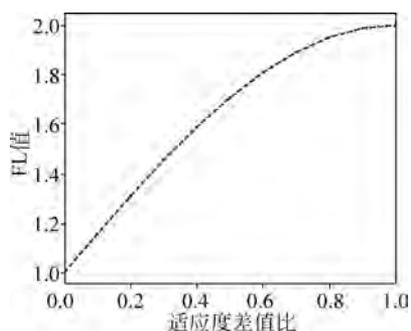


图 5.1 FL 随适应度差值比的变化过程

5.2.4 接受准则

由于鸟群算法在其每次迭代中的位置更新属于一种随机搜索的行为,因此在每次位置更新时,即使个体向种群信息交流中适应度更高的位置信息进行学习,也无法保证更新后的位置信息较更新前更好,个体的适应度值仍可能较更新之前有所降低,而这一种情况的出现意味着算法有可能已经陷入了局部最优,这将会直接影响到算法的收敛速度。为了能够有效应对这一情况并提高算法的收敛速度,协同鸟群算法将接受准则加入其算法步骤中,以此来提高整个算法的收敛速度。

接受准则主要用于在每次种群个体进行位置更新后判断是否接受此次位置更新,若接受,则保持此次更新状态;若不接受,则将位置信息恢复到上一次迭代更新时的状态。在接受准则中,主要涉及的变量为判断个体是否接受此次位置更新的概率 P ,其具体的计算公式如下:

$$P = \begin{cases} \exp\left(-\frac{T}{3}\right), & f_{\text{new}} - f_{\text{old}} \leq 0 \\ 1, & f_{\text{new}} - f_{\text{old}} > 0 \end{cases} \quad (5.8)$$

其中, f_{new} 为个体在位置更新之后计算得到的适应度值; f_{old} 为个体在位置更新之前的适应度值; T 表示个体在迭代过程中连续出现适应度降低或保持不变的次数,其初始值为 0。当开始连续出现更新后的适应度值降低或保持不变的情况时,对 T 值按照迭代次数进行累加;当出现位置更新后适应度值较更新前增加的情况时, T 值将变为其初始值 0。

图 5.2 给出了在连续出现适应度降低或保持不变的情况下,概率 P 随次数 T 的变化曲线。由图 5.2 可以看到,随着次数 T 的增加,概率 P 在不断减小,但其减小的速度是在不断变化的。在连续出现上述情况的次数较少时,概率 P 具有较大的减小速度,这主要

是因为前期概率 P 保持在一个较大值,此时仍有很大的概率去接受位置更新之后的状态,较大的减小速度可以有效避免陷入局部最优情况的出现,而当连续出现上述情况的次数开始增多后,概率 P 已经逐渐降低至一个较小值,其接受位置更新之后的状态的可能性较小,因此使其减小速度逐渐降低直至概率 P 降低至 0。

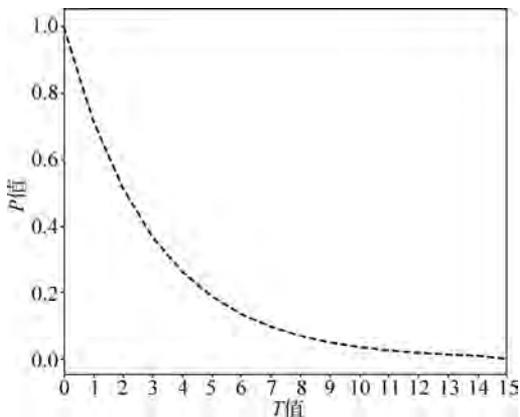


图 5.2 概率 P 随次数 T 的变化曲线

加入接受准则后,此算法的步骤调整如下:

(1) 每次迭代后,分别计算种群中每个个体的适应度值,并将位置更新后的适应度值与位置更新前的适应度值进行比较;

(2) 当个体出现适应度值降低或保持不变的情况时,将此次情况累加到个体对应的 T 值,当个体出现适应度值增加的情况时,将 T 值变为其初始值 0;

(3) 按照适应度值比较的结果分别对每个个体使用公式(5.8)计算其接受此次更新状态的概率;

(4) 依次对每个个体生成一个 $0 \sim 1$ 的随机数,倘若该数值小于或等于其对应的概率,个体则接受更新之后的位置状态;倘若该数值大于其对应的概率,个体则恢复到上一次迭代更新时的位置状态。

5.2.5 协同鸟群算法运行机制

将上述抱团行为、基于适应度差值比的位置更新方式、接受准则与鸟群算法相结合,即为协同鸟群算法(Collaborative Bird Swarm Algorithm, CBSA)。由前面对这三种改进的介绍可以得知,协同鸟群算法除了具有鸟群算法的八个特性外,还下列几种特性:

(1) 在每次判断是否进行飞行行为之前为种群个体增加抱团行为,使种群个体的学习对象在整个迭代过程中不断改变,进一步增加了种群个体的多样性;

(2) 抱团行为随机选取种群中的个体作为小团体的中心点,一定程度上扩大了种群的搜索范围;

(3) 基于适应度差值比的位置更新方式主要以个体自身适应度值与被学习个体适应度值的大小比较作为依据来确定个体位置更新的量,具有自适应性;

(4) 基于适应度差值比的位置更新方式更符合实际情况,即当被学习个体的适应度

值相对较高时,则向其学习更多的位置信息,当被学习个体的适应度值相对较低时,则向其学习较少的位置信息;

(5) 以正弦曲线变化的 FL 值可以加快算法的收敛速度;

(6) 接受准则在有效避免算法陷入局部最优的同时,加快了算法向全局最优收敛的速度;

(7) 概率 P 的变化方式有助于个体在迭代过程中对最优解进行搜索;

(8) 除原算法中的参数外,此算法中主要增加了 FL_{\max} 与 FL_{\min} 、连续出现适应度降低或保持不变的次数 T 以及概率 P 这几个参数,其中 FL_{\max} 与 FL_{\min} 的取值为原有 FL 值的取值范围最大值与最小值,次数 T 通过统计次数获得,概率 P 由次数 T 计算得来,因此无新增参数需要调整。

将鸟群算法的算法特性与上述八个特性相结合,可以将协同鸟群算法的算法步骤归纳总结如下:

(1) 设置种群的个体数量、小团体数量、飞行间隔 FQ 、表示乞讨者跟随生产者寻找食物的行为的参数 FL_{\max} 与 FL_{\min} 、感知系数 C 与社会加速系数 S 、保持警戒行为中的常量 a_1 与 a_2 以及统计连续出现适应度降低或保持不变的参数 T ;

(2) 以鸟群个体的位置信息作为待优化问题的解,根据待优化问题的解的搜索范围,随机初始化种群所有个体的位置信息;

(3) 根据待求解问题,计算种群中每个个体的适应度值,将其作为每个个体的历史最优适应度值,将此时对应的位置信息作为每个个体的历史最优位置信息,之后对种群个体的适应度值进行比较,将最高适应度值作为种群的历史最优适应度值,将其对应的位置信息作为种群的历史最优位置信息;

(4) 按照预先设置的小团体数量随机在整个种群中选择对应数量的个体作为小团体的中心点;

(5) 分别计算每个个体与每个小团体中心点的空间距离,按照就近优先原则等量依次将个体分配至每个小团体,之后通过小团体内的个体适应度值比较确定小团体的历史最优适应度值;

(6) 根据飞行间隔 FQ 判断个体是否进行飞行行为,若为飞行行为,则为每个个体分配生产者以及乞讨者的身份,适应度值最高的个体将扮演生产者,适应度值最低的个体将扮演乞讨者,其他个体则会随机选择两种角色中的一种;

(7) 生产者使用原算法中的更新方式进行位置更新,乞讨者使用基于适应度差值比的位置更新方式进行位置更新;

(8) 若不为飞行行为,则根据常量 p 与随机数的比较判断个体是进行觅食行为还是保持警戒,并使用相关公式对位置信息进行更新;

(9) 计算每个个体更新后的位置信息以及对应的适应度值,比较更新前后的适应度值同时对每个个体对应的参数 T 进行更新;

(10) 通过参数 T 计算得出每个个体接受此次更新状态的概率 P ;

(11) 依次对每个个体生成一个 $0\sim 1$ 的随机数,倘若该数值小于或等于其对应的概率 P ,个体则接受更新之后的位置状态,倘若该数值大于其对应的概率 P ,个体则恢复到

上一次迭代更新时的位置状态；

(12) 对个体的历史最优适应度值、历史最优位置信息以及种群的历史最优适应度值、历史最优位置信息进行更新；

(13) 根据预设的迭代次数重复步骤(4)~步骤(12),当达到最大迭代次数时停止迭代过程,输出种群的历史最优位置信息,此位置信息即为算法优化后获得的问题最优解。

按照上述过程对协同鸟群算法进行归纳,如算法 5.2 所示。

算法 5.2 协同鸟群算法

```

1:  初始化鸟种群中每个个体的位置信息
2:  设置种群数量  $N$ 、飞行间隔  $FQ$ 、常量  $p$ , 参数  $FL_{max}$ 、 $FL_{min}$ 、 $C$ 、 $S$ 、 $a_1$ 、 $a_2$  以及  $T$ 
3:  根据待求解问题计算每个个体的适应度值
4:  确定种群的历史最优位置相关信息以及个体历史最优位置相关信息
5:  while(未满足停止迭代条件)do
6:      对种群进行小团体划分,确定小团体历史最优位置相关信息
7:      if 达到飞行间隔
8:          将种群个体分为生产者与乞讨者
9:          for  $i = 1: N$ 
10:             if  $i$  为生产者
11:                 生产行为
12:             else
13:                 乞讨行为
14:             end if
15:          end for
16:      else
17:          for  $i = 1: N$ 
18:             if  $\text{rand}(0,1) < p$ 
19:                 觅食行为
20:             else
21:                 保持警戒
22:             end if
23:          end for
24:      计算位置更新后的适应度值
25:      for  $i = 1: N$ 
26:          更新参数  $T$ , 计算概率  $P$ 
27:          if  $\text{rand}(0,1) \leq P$ 
28:              接受更新后的位置信息
29:          else
30:              回到更新前的位置状态
31:          更新种群的历史最优位置相关信息以及个体历史最优位置相关信息
32:      end while
33:  输出种群的历史最优位置信息

```

5.2.6 对比实验

由于协同鸟群算法是一种在原有算法基础上进行改进而提出的新算法,因此在将其应用到并行支持向量机之前需要对其算法性能进行验证。为了验证此算法的有效性,选择十种不同的基准函数对协同鸟群算法、鸟群算法、萤火虫算法(Firefly Algorithms, FA)以及磷虾群算法(Krill Herd Algorithm, KHA)进行测试,后面三种算法主要作为对比算法与协同鸟群算法的性能进行比较。

所使用的十种测试函数与第3章中对自适应动态灰狼优化算法进行算法性能测试时所使用的测试函数完全相同。其中,前五种为单峰函数,即在所考虑的范围区间内只有一个严格局部极值(峰值);后五种为多峰函数,即在所考虑的范围区间内有多个局部极值(峰值),它们在目标范围内搜索最小值,且目标最小值为0,另外将每个测试函数的维度都设为20。

在参数设置方面,将上述四种算法的种群数量都设为30,将它们的算法迭代次数都设为100,每个算法具体的参数设置如下:协同鸟群算法与鸟群算法中将飞行间隔FQ设为3,感知系数C与社会加速系数S均设为1.5,保持警戒行为中的常量 a_1 与 a_2 均设为0.5;鸟群算法中表示乞讨者跟随生产者寻找食物的行为的参数FL设为1.5;协同鸟群算法中 FL_{max} 与 FL_{min} 分别设为2和1;萤火虫算法中初始吸引度值 β_0 设为1,传播介质对光的吸收系数 γ 设为1,步长的扰动因子 α 设为0.2;磷虾群算法中诱导速度最大值 N_{max} 设为0.01,觅食速度 V_f 设为0.02,惯性权重 ω_f 、 ω_n 以及参数 C_t 的初始值分别设为0.9、0.9与0.5,且它们均会在迭代过程中随着迭代次数的增加线性减小至0.1。

对这四种算法的性能进行比较主要通过以下指标展开,分别是在迭代过程中每种算法的种群最优适应度值的变化情况、迭代之后四种算法种群个体候选解的最优适应度值、适应度值的平均值以及适应度值的标准差、Friedman检验与Nemenyi后续检验以及四种算法迭代相同次数所需的运行时间。

首先分别统计了每种算法在每种测试函数下进行迭代训练时的种群历史最优适应度值的变化情况,并将其进行绘图,依据适应度值的差别来比较这四种算法在十种基准函数上的收敛速度与收敛精度,统计结果如图5.3所示。由图5.3可以看到,除测试函数Schwefels P2.22、Dixon-Price以及Levy外,其他测试函数下所有算法在迭代初期时的种群历史最优适应度值是比较接近的,之后随着迭代次数的增加,算法之间的性能差别逐渐得以体现,不同算法之间的种群历史最优适应度值差值逐渐增大。另外,由单峰函数与多峰函数的曲线比较可以看到,鸟群算法与协同鸟群算法在除Schwefels函数外的测试函数上均能在迭代前期获得较大的收敛速度,同时其迭代过程中种群历史最优适应度值的下降幅度也较大,而萤火虫算法与磷虾群算法在所有单峰函数上可以取得相同的效果,但是在多峰函数上其收敛速度与最优适应度值下降幅度均相对较小,特别是在测试函数Schwefels与Ackley上,通过图片比较仅能观察到其收敛速度与最优适应度值下降幅度发生了较为细微的变化。

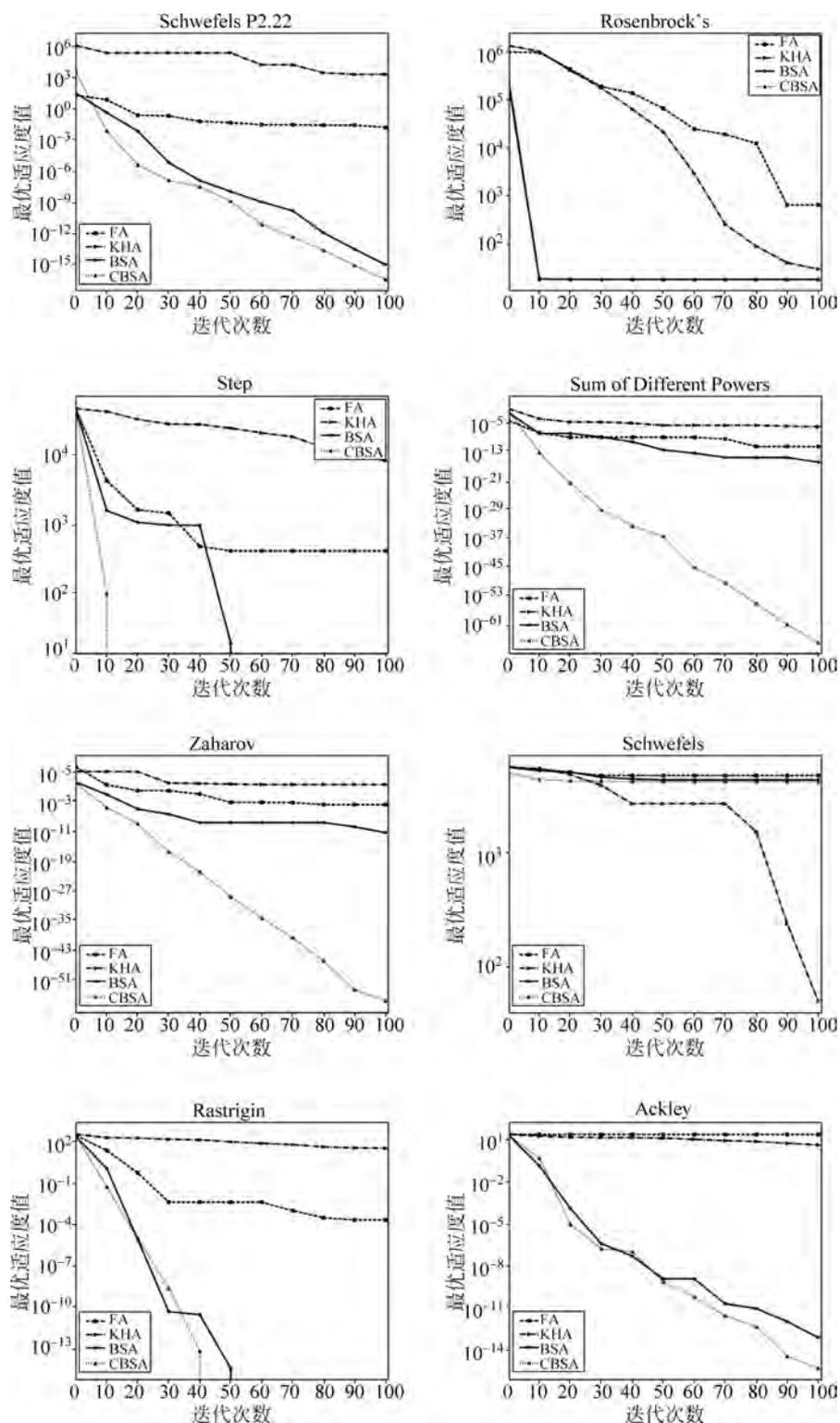


图 5.3 四种算法在十种基准函数上的最优适应度值的变化情况

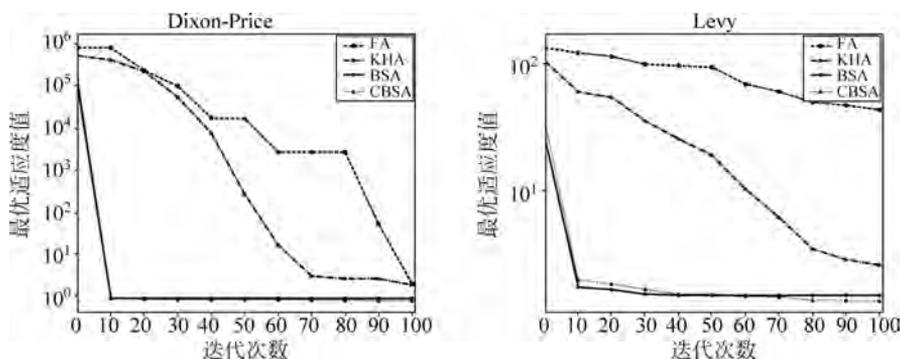


图 5.3 (续)

比较不同测试函数下代表不同算法种群历史最优适应度值曲线可以看到,在除 Schwefels 以外的测试函数上,代表鸟群算法与协同鸟群算法的曲线均处于代表萤火虫算法与磷虾群算法的曲线之下,其中代表协同鸟群算法的曲线位于代表鸟群算法的曲线之下,代表萤火虫算法与磷虾群算法的曲线保持在一个较为接近的位置,由于所使用的测试函数均为在目标范围内搜索最小值,因此曲线越低代表算法搜索的适应度值越好,由此可以得知鸟群算法与协同鸟群算法的收敛速度与收敛精度相对另外两种算法要更好一些,萤火虫算法与磷虾群算法的收敛速度与收敛精度相对较为接近。而在测试函数 Schwefels 中,代表磷虾群算法的曲线位于代表另外三种算法曲线之下,另外三条曲线之间相对较为接近,说明在此测试函数上磷虾群算法的收敛速度与收敛精度要优于另外三种算法,另外三种算法的收敛速度与收敛精度较为接近,这一比较情况与其他测试函数上的比较情况差别较大,因此单从此对比图难以做出相关结论,还需要参考其他对比指标。

第二个对比指标对四种算法在十种测试函数上迭代训练结束后其种群的历史最优适应度值、适应度值的平均值以及适应度值的标准差,统计结果如表 5.1 所示。由于算法对优化问题寻优后输出的种群历史最优位置信息即为问题的解,因此种群历史最优适应度值代表了算法的寻优能力。比较这十个测试函数上不同算法的最优适应度值可以看到,除测试函数 Schwefels 外,鸟群算法与协同鸟群算法的最优适应度值均小于另外两种算法,其中在测试函数 Step 与 Rastrigin 上均达到了目标范围内的最小值 0,而在其他算法上协同鸟群算法的最优适应度值均小于鸟群算法。在比较另外两种算法的最优适应度值时可以看到,九种测试函数中有五种测试函数上萤火虫算法的最优适应度值小于磷虾群算法,四种测试函数上磷虾群算法的最优适应度值小于萤火虫算法。由此可以得知,使用算法进行测试函数最小值寻优时,协同鸟群算法能够在大部分情况下具有最好的寻优结果,其次是鸟群算法,而磷虾群算法与萤火虫算法的寻优结果较为接近。这一结果与图 5.3 中的历史最优适应度值比较情况是符合的。

表 5.1 四种算法在十个基准函数上的优化结果对比

基准函数	算法	最优适应度值	适应度平均值	适应度标准差
Schwefels P2.22	FA	2.58E-02	5.69E+01	2.15E+01
	KHA	3.33E+03	3.59E+09	1.84E+10
	BSA	1.47E-15	2.04E-12	2.57E-12
	CBSA	5.56E-17	2.76E-14	5.91E-14
Rosenbrock's	FA	7.73E+02	2.76E+06	5.11E+06
	KHA	3.18E+01	8.11E+01	3.89E+01
	BSA	1.89E+01	1.89E+01	1.10E-01
	CBSA	1.89E+01	1.90E+01	1.71E-01
Step	FA	2.86E+02	5.28E+03	1.04E+04
	KHA	7.01E+03	1.38E+04	4.36E+03
	BSA	0.00E+00	4.81E+04	6.22E-04
	CBSA	0.00E+00	3.33E-02	1.79E-01
Sum of Different Powers	FA	1.09E-11	4.07E+00	6.93E-00
	KHA	3.56E-06	8.36E-03	2.81E-02
	BSA	5.64E-16	3.46E-00	6.44E-00
	CBSA	1.56E-65	8.65E-61	1.87E-60
Zaharov	FA	9.15E-04	4.62E+10	1.48E+11
	KHA	2.13E+02	1.91E+08	9.21E+08
	BSA	2.91E-11	1.89E+10	5.99E+10
	CBSA	3.07E-56	1.21E-51	3.62E-51
Schwefels	FA	5.57E+03	8.29E+03	8.72E+02
	KHA	5.42E+01	3.96E+03	2.71E+03
	BSA	5.11E+03	7.29E+03	8.77E+02
	CBSA	4.89E+03	6.39E+03	1.56E+03
Rastrigin	FA	2.18E-04	2.43E+01	2.13E+00
	KHA	2.87E+01	5.13E+01	1.29E+01
	BSA	0.00E+00	2.18E-10	7.56E-10
	CBSA	0.00E+00	2.72E-15	8.23E-15
Ackley	FA	1.01E+01	1.97E+01	4.90E-01
	KHA	3.54E+00	4.41E+00	4.64E-01
	BSA	6.65E-11	4.00E-05	7.22E-05
	CBSA	4.22E-14	2.70E-10	6.50E-10
Dixon-Price	FA	2.15E+00	1.11E+06	1.89E+06
	KHA	2.04E+00	7.78E+00	4.09E+00
	BSA	9.83E-01	2.87E+06	3.49E+06
	CBSA	8.66E-01	2.84E+05	1.44E+06
Levy	FA	4.38E+01	2.32E+02	1.93E+02
	KHA	2.68E+01	3.08E+02	2.84E-01
	BSA	1.56E+00	3.36E+02	4.12E-02
	CBSA	1.41E+00	1.71E+00	4.11E-01

适应度平均值与适应度标准差代表了算法迭代结束后种群中所有个体位置信息的分布情况,对比十种测试函数上的这两个指标可以看到,不同的算法在不同的测试函数上表现均不同,除测试函数 Step、Sum of Different Powers 以及 Zaharov 外,鸟群算法与协同鸟群算法在这两个指标上的统计结果较为接近,这主要是因为协同鸟群算法是在鸟群算法的基础上进行改进,因此它们的算法搜索机制是相同的,而另外两种算法在大部分测试函数上这两个指标的统计结果均小于协同鸟群算法与鸟群算法,说明其迭代结束后种群中不同个体的位置信息分布情况相对较差,这可能是由于萤火虫算法与磷虾群算法在位置移动过程中的位置更新方式更倾向于随机搜索,种群内的信息交流情况相对鸟群算法与协同鸟群算法来说更差一些。综合四种算法在十种测试函数上的历史最优适应度值变化情况以及训练结束后种群个体的最优适应度值、适应度平均值以及适应度标准差可以初步得知,四种算法中协同鸟群算法的寻优能力较好,鸟群算法次之,萤火虫算法与磷虾群算法的寻优能力较为接近。

在第 4 章对多个模型进行性能比较时介绍了一种多模型性能对比评价方法 Friedman 检验与 Nemenyi 后续检验,同时也将其扩展应用到了多种算法的性能对比上,因此为了更好地了解这四种算法之间的性能差别,将使用 Friedman 检验与 Nemenyi 后续检验对这四种算法的性能进行比较。由于这四种算法是在十种测试函数上进行最小化寻优,且最能体现其性能好坏的指标为种群历史最优适应度值,因此将以此指标作为参考依据并对不同测试函数下的算法性能进行排序,具体排序结果如表 5.2 所示。

表 5.2 算法性能比较排序

测试函数	FA	KHA	BSA	CBSA
Schwefels P2.22	3	4	2	1
Rosenbrock's	4	3	2	1
Step	3	4	1.5	1.5
Sum of Different Powers	3	4	2	1
Zaharov	3	4	2	1
Schwefels	4	1	3	2
Rastrigin	3	4	1.5	1.5
Ackley	4	3	2	1
Dixon-Price	4	3	2	1
Levy	4	3	2	1
平均序值	3.5	3.3	2	1.3

首先对 Friedman 检验中的检验变量 τ_F 进行计算,可以得到其计算结果为 22.6901,当对比算法数量为 4、测试函数数量为 10 时,由显著度为 0.05 时的常用检验值表可以得知临界值为 2.960,此临界值小于计算结果,因此可以得知这四种算法之间存在显著差别。之后使用 Nemenyi 后续检验对这四种算法的临界值域进行计算,从而更进一步区分这四种算法,计算过程中变量 q_α 选用对比算法数量为 4、测试函数数量为 10 时的常用值 3.164,根据最终计算结果得到的 Friedman 检验图如图 5.4 所示。

在对 Friedman 检验图进行观察时,主要需要比较代表不同算法的直线之间是否存

在交叠,若存在交叠则算法性能之间不存在显著差别,反之则存在显著差别。由图 5.4 可以看到,代表协同鸟群算法的直线与代表鸟群算法的直线之间存在交叠,与代表磷虾群算法以及萤火虫算法的直线之间不存在交叠,因此协同鸟群算法的性能与鸟群算法的性能之间不存在显著差别,与磷虾群算法以及萤火虫算法的算法性能存在显著差别,而代表鸟群算法、磷虾群算法以及萤火虫算法的直线之间存在交叠,即表示这三种算法之间不存在显著差别。之后对这四种算法的平均序值进行比较,可以看到四种算法的平均

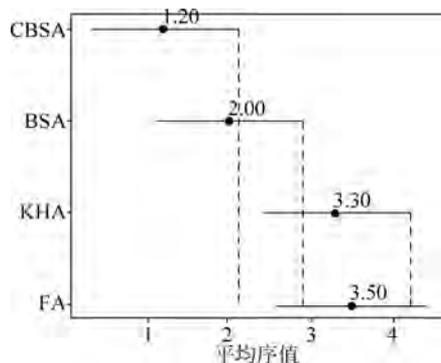


图 5.4 Friedman 检验图

序值从小到大排序依次为协同鸟群算法、鸟群算法、磷虾群算法以及萤火虫算法。将平均序值排序结果与直线之间的交叠情况进行结合可以得出结论:使用这四种算法在十种测试函数上进行最小寻优时,协同鸟群算法的算法性能最好,鸟群算法次之,磷虾群算法的算法性能略优于萤火虫算法。

在实际使用算法对优化问题进行求解时,除了将算法性能作为选择算法的参考依据外,还要考虑算法的运行效率,即算法在求解问题所需的运行时间,因此最后统计了每个算法在不同测试函数下迭代训练 100 次所需的运行时间,统计结果如表 5.3 所示。

表 5.3 四种算法在十种基准函数上的运行时间

单位: s

基准函数	算法	运行时间	基准函数	算法	运行时间
Schwefels P2, 22	FA	5.2791	Schwefels	FA	4.2071
	KHA	2.3762		KHA	2.7341
	BSA	0.1548		BSA	0.3255
	CBSA	0.2286		CBSA	0.5228
Rosenbrock's	FA	6.9841	Rastrigin	FA	6.4157
	KHA	3.0019		KHA	2.5876
	BSA	0.2494		BSA	0.4075
	CBSA	0.5422		CBSA	0.7256
Step	FA	5.8287	Ackley	FA	6.1711
	KHA	2.7082		KHA	2.8651
	BSA	0.2001		BSA	0.3993
	CBSA	0.2447		CBSA	0.4832
Sum of Different Powers	FA	5.6044	Dixon-Price	FA	2.9337
	KHA	2.7415		KHA	2.2786
	BSA	0.3351		BSA	0.1794
	CBSA	0.5807		CBSA	0.3311
Zaharov	FA	6.5607	Levy	FA	6.5919
	KHA	2.4252		KHA	3.0937
	BSA	0.1981		BSA	0.5526
	CBSA	0.6611		CBSA	0.7265

由表 5.3 可以看到,不同算法之间的运行时间差别较大,但是在不同的测试函数上的表现基本一致,萤火虫算法所需的运行时间最长,磷虾群算法次之,鸟群算法与协同鸟群算法所需的运行时间相对较短,其中萤火虫算法的运行时间达到鸟群算法的三十倍以上,磷虾群算法的运行时间达到鸟群算法的二十倍以上,因此从算法效率上来看,鸟群算法与协同鸟群算法的算法效率要优于萤火虫算法与磷虾群算法。而在比较鸟群算法与协同鸟群算法时从十种基准函数上的运行时间可以看到,协同鸟群算法的运行时间要略长于鸟群算法,这两种算法从算法复杂度上来看是相同的,其不同点主要表现在协同鸟群算法为鸟群在每次判断是否进行飞行行为之前增加了抱团行为,同时使用接受准则对每次迭代前后的适应度值进行比较,这两个步骤均是针对种群中的每个个体进行,因此在运行时间上会有所增加,增加的部分是合理的,相对于整个运行时间来说,增加的时间是可以接受的。因此可以得出结论:在使用这四种优化算法进行十种测试函数的最小寻优过程中,鸟群算法与协同鸟群算法的运行效率相对较高,其中协同鸟群算法的改进部分并未使其原有效率降低。

5.3 支持向量机分类模型

5.3.1 概述

20 世纪 90 年代中期,随着统计学习理论的不断发展和成熟,支持向量机作为一种基于统计学习理论的新型机器学习方法出现在大家的视野中。人类在学习中通过对已知事实的分析和对规则的总结,进而预测不可能发生的事实,这种能力称为泛化能力。在机器学习问题中,泛化能力也很重要。研究者期望他们提出的方法能够通过已知数据的学习,发现数据之间的内在联系,从而预测未知的事物或因素。这也是机器学习研究的主要内容。如何从大量的数据中学习有用且正确的规则在机器学习研究中起着极其重要的作用。在此背景之下,数据挖掘以及数据分析技术应运而生,机器学习在各行各业得到了广泛应用和迅速发展。统计学在机器学习从最初到现在的发展中起着基础性的作用,但传统的统计主要集中在渐近理论上,即样本的性质趋于无穷大。

在实际情况中面对的样本数量通常有限,当样本数有限时,原模型的泛化能力较差。随着 1958 年 Rosenblatt 最早的机器学习模型 perceptron 问世,机器学习的数学研究时代已经到来。此后不久,随着 BP 技术的发展,机器学习的研究进入了一个新的阶段,即神经网络时代。在神经网络时代,第一次神经网络模拟考试得到了迅速而全面的发展。但是,神经网络有其自身的局限性:它的实现过程主要依赖于人们的主观意识和先验知识,而不是建立在严格的数学理论基础之上,其运行过程更像一个黑箱,因此,神经网络的理论分析比较困难。另外,如上所述,神经网络研究的是样本数趋于无穷大的情况,但在实际问题中,样本数往往是有限的,神经网络的过学习问题就是一个典型的例子。当样本数据有限时,学习能力强的学习模型泛化能力较差。在这种背景下,Vapnik 自 20 世纪 70 年代开始从事统计学习理论的研究。统计学习理论是专门研究小样本情况下机器学习规律的基础理论。它为了解决有限样本学习问题提供了一种新的思路,可以解决许多难以解决的问题。由于支持向量机良好的数学特性,人们开始关注该方法,其成为继神经网络之

后的一个新的研究热点。目前,国内外学者在支持向量机的目标函数、模型、算法及应用等方面做了大量的研究。

支持向量机是一种有监督学习模型,主要应用于二分类问题,其模型的基础是定义在特征空间上的最大化间隔分类(Maximal Margin Classification)方法。在对一维数据进行二分类时,通常倾向于找到两个类别的边界数据点,使用两个边界数据点的中间点作为分类阈值。阈值与边界点的距离称为间隔(Margin)。当选取两个边界数据点的中间点作为阈值时,间隔最大,这种使间隔最大来确定阈值的方法就是最大化间隔分类。使用最大化间隔分类方法使支持向量机与感知机区别开来,感知机的思想是让所有误分类的点到超平面的距离和最小。与一维时一样,在对二维数据进行二分类时,通常寻找一条直线作为分类阈值。在对三维数据进行二分类时,则需要找到一个平面,而在面对更高维的数据进行二分类时,就需要找到一个超平面。在实际应用中,不管数据的维度是多少,以上提到的点、直线、平面、超平面都可以统一称为超平面(Hyperplane)。

上面提到的数据是严格线性可分的,通过间隔最大化,使支持向量机与感知机区别开来。间隔最大化有两种类型:硬间隔最大化和软间隔最大化。硬间隔最大化可以训练得到线性可分支持向量机。由于数据不可避免会有异常值的存在,当异常值分布在类别边界时,数据不再是严格线性可分的,硬间隔最大化分类会受到异常值的影响而造成分类错误。为了减小最大化间隔分类方法受异常值的影响,可以引入松弛变量,通过软间隔最大化以及交叉验证寻找分类阈值,训练得到线性支持向量机。位于间隔边界上的数据点就是支持向量(Support Vector)。最后,在面对非线性数据时,无法通过硬间隔或者软间隔最大化得出分类阈值,一般的解决方法是使用核函数(Kernel Function)将数据映射到更高维的空间,再寻找超平面进行分类,这时训练得到的则是非线性支持向量机。可见,支持向量机既可以支持线性数据的分类,也可以支持非线性数据的分类。

支持向量机分类模型具有坚实的理论基础,适合用于解决小样本非线性的高维度问题,在使用支持向量机进行分类前不需要对数据分布做任何假设。因此,当拿到一个数据集不确定其是何分布时,可以尝试使用支持向量机模型进行分类。

5.3.2 统计学习原理

1. 函数间隔与几何间隔

因为支持向量机的分类策略就是找到一个分割数据的超平面,并使间隔最大化,所以在推导分类函数之前需要对函数间隔和几何间隔进行区分。

在划分超平面固定为 $\omega^T x + b = 0$ 时, $|\omega^T x + b|$ 表示点 x 到超平面的相对距离。通过观察 $|\omega^T x + b|$ 和 y 是否同号,可以判断分类是否正确。这里引入函数间隔的概念,定义函数间隔 l' 表达式为:

$$l' = y(\omega^T x + b) \quad (5.9)$$

其中, l' 是感知机模型里面的误分类点到超平面距离的分子。对于数据集中的 m 个样本点,每个样本对应的 m 个函数间隔中的最小值,就是整个数据集的函数间隔。

函数间隔并不能反映点到超平面的距离,在感知机模型中,当分子成比例的增长时,

分母也成倍增长。为了统度量,给法向量 ω 定义一个约束条件,就能够得到几何间隔 l ,几何间隔计算表达式为:

$$l = \frac{y(\omega^T x + b)}{\|\omega\|} = \frac{l'}{\|\omega\|} \quad (5.10)$$

几何间隔表示的其实是点到超平面的真实距离,几何间隔主要被应用于感知机模型。

2. 支持向量与划分超平面模型

以二分类为例,给定数据集 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, 类别空间 $y = \{-1, 1\}$ 。

首先,假设数据集 S 的特征空间是二维空间,给定数据集 S 中某个样本点 x' 。图 5.5 用直角坐标系展示了划分超平面和样本点的情况,其中圆点表示样本点的类别 $y = -1$, 方点表示样本点的类别 $y = 1$ 。则样本点 x' 在直角坐标系中的二维特征坐标可以表示为 (x'_1, x'_2) 。此时的划分超平面应该是一条直线,直线方程表达式如公式(5.11)所示。

$$\omega_1 x_1 + \omega_2 x_2 + b = 0 \quad (5.11)$$

图 5.5 为使用直角坐标系表示二维特征空间的划分超平面。

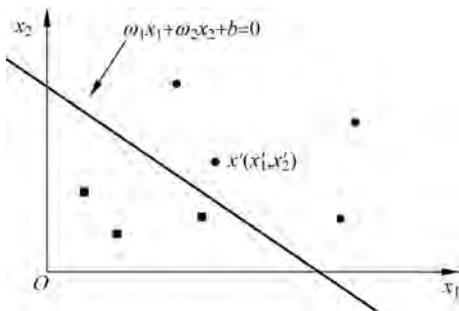


图 5.5 使用直角坐标系表示二维特征空间的划分超平面

根据距离公式可以得到该样本点到该划分超平面的距离 l , 计算表达式如下所示:

$$l = \frac{|\omega_1 x'_1 + \omega_2 x'_2 + b|}{\sqrt{\omega_1^2 + \omega_2^2}} \quad (5.12)$$

若希望该划分超平面可以正确分类,则对于样本 $(x_i, y_i) \in S$, 应该满足当 $\omega_1 x_{1i} + \omega_2 x_{2i} + b > 0$ 时,类别 $y_i = 1$; 当 $\omega_1 x_{1i} + \omega_2 x_{2i} + b < 0$ 时,类别 $y_i = -1$ 。可以令

$$\begin{cases} \omega_1 x_{1i} + \omega_2 x_{2i} + b \geq 1, & y_i = 1 \\ \omega_1 x_{1i} + \omega_2 x_{2i} + b \leq -1, & y_i = -1 \end{cases} \quad (5.13)$$

把公式(5.9)由二维空间推广到一般,划分超平面可以通过以下方程进行描述:

$$\omega^T x + b = 0 \quad (5.14)$$

其中, $\omega = (\omega_1; \omega_2; \omega_3; \dots; \omega_d)$ 为法向量,决定了超平面的方向; b 为位移项,决定了超平面与原点之间的距离。划分超平面可以由法向量 ω 和 b 确定。

因为希望该划分超平面可以正确分类,所以有样本 $(x_i, y_i) \in S$, 应该满足当 $\omega^T x_i + b > 0$ 时,类别 $y_i = 1$; 当 $\omega^T x_i + b < 0$ 时,类别 $y_i = -1$ 。可以令

$$\begin{cases} \boldsymbol{\omega}^T x_i + b \geq 1, & y_i = 1 \\ \boldsymbol{\omega}^T x_i + b \leq -1, & y_i = -1 \end{cases} \quad (5.15)$$

因为支持向量机的基本模型是最大化间隔分类,所以需要得出间隔的计算表达式。由图 5.5 可以看出,样本点 x' 是类别属于 $y = -1$ 的点中距离划分超平面最近的点,该点就是支持向量机中的支持向量。同理,在类别属于 $y = 1$ 的点中,距离划分超平面最近的点也是支持向量。两个不同类别的支持向量到划分超平面的距离之和就是间隔。通过公式(5.12)可以推导出间隔的计算表达式如公式(5.16)所示。

$$r = \frac{2}{\|\boldsymbol{\omega}\|} \quad (5.16)$$

最大化间隔的表达式可以写为:

$$\max_{\boldsymbol{\omega}, b} \frac{2}{\|\boldsymbol{\omega}\|} \quad (5.17)$$

且上式需要满足约束 $y_i(\boldsymbol{\omega}^T x_i + b) \geq 1, i = 1, 2, \dots, n$ 。对上式求倒数可以把最大化问题转换为求最小化问题,表达式可以转换为:

$$\min_{\boldsymbol{\omega}, b} \frac{1}{2} \|\boldsymbol{\omega}\|^2 \quad (5.18)$$

其中,对 $\|\boldsymbol{\omega}\|$ 求平方,去掉根号方便后续计算;同样地,该式也需要满足约束条件 $y_i(\boldsymbol{\omega}^T x_i + b) \geq 1, i = 1, 2, \dots, n$ 。

至此,就可以通过对公式(5.18)求解得出划分超平面所对应的模型。可以看到公式(5.18)的目标函数是二次的,且关于 $\boldsymbol{\omega}$ 的约束条件是线性约束,因此这个问题是一个受约束的二次型规划(Quadratic Programming, QP)问题。所以可以利用拉格朗日乘子法解决约束最优问题。

3. 拉格朗日对偶性质

因为本问题的特殊结构,可以利用拉格朗日的对偶(Lagrange Duality)性质将其转换为对偶变量(Dual Variable)的优化问题。也就是,通过求解与原问题等价的对偶问题(Dual Problem)而得出原始问题的最优解。这就是线性可分条件下支持向量机的对偶算法。这种算法的优点有两点:第一是对偶问题往往更容易求解;第二是核函数可以自然地引入到线性分类问题中,进而推广到非线性分类问题中。引入拉格朗日乘子(Lagrange Multiplier)法,给原始目标函数的每一个约束条件都加上一个拉格朗日乘子,将有约束的原始目标函数转换为无约束的新构造的拉格朗日函数,得到表达式:

$$L(\boldsymbol{\omega}, b, \alpha) = \frac{1}{2} \|\boldsymbol{\omega}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\boldsymbol{\omega}^T x_i + b) - 1) \quad (5.19)$$

其中, α_i 为拉格朗日乘子,且 $\alpha_i \geq 0, i = 1, 2, \dots, n$; 令 $\theta(\boldsymbol{\omega}) = \max_{\alpha_i \geq 0} L(\boldsymbol{\omega}, b, \alpha)$, 根据之前的约束条件 $y_i(\boldsymbol{\omega}^T x_i + b) \geq 1, i = 1, 2, \dots, n$ 可知,公式(5.19)中第二项为非负项。如果样本点不在可行解区域内,即不满足约束条件时,应该得到 $y_i(\boldsymbol{\omega}^T x_i + b) < 1$ 。此时,如果将 α 设置为无穷大,则显然有 $\theta(\boldsymbol{\omega})$ 也为无穷大。而当所有约束条件都满足时, $\theta(\boldsymbol{\omega})$ 则

为原函数 $\frac{1}{2} \|\omega\|^2$ 。因此,在要求约束条件得到满足的情况下最小化 $\frac{1}{2} \|\omega\|^2$,实际上等价于直接最小化 $\theta(\omega)$,当然,此处也有约束条件,就是 $\alpha_i \geq 0$,且 $i=1,2,\dots,n$ 。

最终,得到目标函数为:

$$\min_{\omega, b} \theta(\omega) = \min_{\omega, b} \max_{\alpha_i \geq 0} L(\omega, b, \alpha) = p^* \quad (5.20)$$

因为在求最大化的过程中,约束条件有不等式,不方便求解,所以可以利用拉格朗日函数的对偶性,交换最大化和最小化的先后顺序,可以得到:

$$\max_{\alpha_i \geq 0} \min_{\omega, b} L(\omega, b, \alpha) = d^* \quad (5.21)$$

交换顺序之后的新问题是原问题的对偶问题,要使 $d^* = p^*$,首先需要该问题是一个凸优化问题,通过对公式(5.19)的观察可以确定该问题是凸优化问题。其次,由 Kuhn Tucker 定理可知,该问题的最优解还需要满足 KKT(Karush-Kuhn-Tucker)条件,KKT条件的要求如下所示:

$$\begin{cases} \alpha_i \geq 0 \\ y_i(\omega^T x_i + b) - 1 \geq 0 \\ \alpha_i y_i(\omega^T x_i + b) - 1 = 0 \end{cases} \quad (5.22)$$

为了求解得到该对偶问题的具体形式,首先需要令 $L(\omega, b, \alpha)$ 对 ω 和 b 求偏导,令 $\frac{\partial L}{\partial \omega} = 0, \frac{\partial L}{\partial b} = 0$,可得:

$$\omega = \sum_{i=1}^m \alpha_i y_i x_i \quad (5.23)$$

$$\sum_{i=1}^m \alpha_i y_i = 0 \quad (5.24)$$

将公式(5.23)与公式(5.24)代入拉格朗日目标函数 $L(\omega, b, \alpha)$,即公式(5.19)中得到:

$$\begin{aligned} L(\omega, b, \alpha) &= \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j - \\ & b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \end{aligned} \quad (5.25)$$

化简后写为:

$$L(\omega, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (5.26)$$

此时该函数只有一个变量,即拉格朗日乘数 α 。由此可以得到,对 $\min_{\omega, b} L(\omega, b, \alpha)$ 求以 α 为变量的极大值,就是直接求极大。

$$\max_{\alpha_i \geq 0} \min_{\omega, b} L(\omega, b, \alpha) = \max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (5.27)$$

该表达式需要满足两个约束,即 $\alpha_i \geq 0, i=1,2,\dots,n$,且 $\sum_{i=1}^n \alpha_i y_i = 0$ 。给上式加上负

号再次转为求极小值问题:

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^n \alpha_i \quad (5.28)$$

该表达式仍然需要满足以上两个约束 $\alpha_i \geq 0, i=1, 2, \dots, n$, 且 $\sum_{i=1}^n \alpha_i y_i = 0$ 。

对于这种形式的问题,只需要利用序列最小优化(Sequential Minimal Optimization, SMO)算法即可求解出拉格朗日乘子 α , 并且通过 α 求出 ω 和 b , 最后完成对划分超平面方程的求解。SMO 算法的具体思路在 5.3.4 节介绍。

4. 软间隔与松弛变量

上面通过计算得到的划分超平面方程,是以数据集的数据严格线性可分为前提以及硬间隔最大化求来的。当数据集的数据不是严格线性可分时,需要使用软间隔最大化的方式求划分超平面方程。软间隔放宽了约束条件,允许部分样本点不满足约束条件 $y_i(\omega^T x_i + b) \geq 1$, 其中 $i=1, 2, \dots, n$, 对原优化问题采用 hinge 损失, 并且引入松弛变量, 可得到公式(5.29):

$$\min_{\omega, b, \xi} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i \quad (5.29)$$

约束条件为: $y_i(\omega^T x_i + b) \geq 1 - \xi_i$, 且 $\xi_i \geq 0, i=1, 2, \dots, n$ 。其中, ξ_i 为松弛变量, $\xi_i = \max(0, 1 - y_i(\omega^T x_i + b))$, 即 hinge 损失函数。针对数据集中每一个样本都有一个对应的松弛变量, 用来表示这个样本不满足约束的程度。 $C > 0$, 且 C 是一个常数, 被称为惩罚参数。 C 值越大, 对分类错误情况下的惩罚越大, C 值越小, 对分类错误情况下的惩罚越小。 C 值在实际应用中需要通过调参来选择。

软间隔最大化与硬间隔最大化计算划分超平面目标函数的方式一样, 也是通过拉格朗日乘子法, 给原始目标函数的每一个约束条件都加上一个拉格朗日乘子, 将有约束的原始目标函数转换为无约束的新构造的拉格朗日函数, 其计算表达式如下所示:

$$L(\omega, b, \xi, \alpha, \mu) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i [y_i(\omega^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^n \xi_i \mu_i \quad (5.30)$$

其中, μ_i 和 α_i 都为拉格朗日乘子, 且 $\alpha_i \geq 0, \mu_i \geq 0, i=1, 2, \dots, n$ 。

最终, 得到待优化的目标函数表达式为:

$$\min_{\omega, b, \xi} \max_{\alpha_i \geq 0, \mu_i \geq 0} L(\omega, b, \xi, \alpha, \mu) \quad (5.31)$$

这个待优化目标函数也需要满足 KKT 条件, 同样因为求最大化的过程中, 约束条件有不等式, 不方便求解, 所以利用拉格朗日函数的对偶性, 交换最大化和最小化的先后顺序, 可以得到公式(5.32):

$$\max_{\alpha_i \geq 0, \mu_i \geq 0} \min_{\omega, b, \xi} L(\omega, b, \xi, \alpha, \mu) \quad (5.32)$$

为了求解得到该对偶问题的具体形式, 首先, 需要求待优化目标函数对 ω 、 b 、 ξ 的极

小值,然后求待优化目标函数对于 α, μ 的极大值。对 ω, b, ξ 求偏导,令 $\frac{\partial L}{\partial \omega} = 0, \frac{\partial L}{\partial b} = 0, \frac{\partial L}{\partial \xi} = 0$ 分别得到表达式:

$$\omega = \sum_{i=1}^n \alpha_i y_i x_i \quad (5.33)$$

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (5.34)$$

$$C - \alpha_i - \mu_i = 0 \quad (5.35)$$

把公式(5.33)、公式(5.34)以及公式(5.35)代入待优化目标,并化简可以得出:

$$L(\omega, b, \xi, \alpha, \mu) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (5.36)$$

从公式(5.36)可以看出其与硬间隔最大化的划分超平面目标函数没有差别,区别在于约束条件,即 $\alpha_i \geq 0, i=1, 2, \dots, n, \mu_i \geq 0, i=1, 2, \dots, n, \sum_{i=1}^n \alpha_i y_i = 0$ 且 $C - \alpha_i - \mu_i = 0$ 。

对约束条件进行化简,即 $0 \leq \alpha_i \leq C$, 且 $\sum_{i=1}^n \alpha_i y_i = 0$ 。给公式(5.36)加上负号再次转换为求极小值问题:

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^n \alpha_i \quad (5.37)$$

其中,约束条件为 $0 \leq \alpha_i \leq C$, 且 $\sum_{i=1}^n \alpha_i y_i = 0$ 。

最后,同样是用 SMO 算法求解出软间隔最大化目标函数的拉格朗日乘数 α , 并且通过 α 求出 ω 和 b 来完成对划分超平面方程的求解。

5.3.3 核函数

在 5.3.2 节中假设样本数据集全部都是线性可分的,虽然使用软间隔最大化的问题的样本数据集不是严格的线性可分,但大部分数据集还是可分的。然而,在实际任务中很可能遇到以下情况,即不存在一个能够正确划分两个类别的样本的超平面。对于这种类型的问题,可以将样本从原始空间映射到一个更高维的特征空间中,使得样本在这个特征空间中线性可分。从数学上可以证明,如果原始空间的维数是有限的,也就是样本属性数是有限的,则一定会存在一个高维特征空间使数据集可分。把样本从原始空间映射到一个更高维空间所用到的方法就是核函数方法。

令 $\phi(x)$ 表示将 x 映射后的特征向量,于是,在特征空间中划分超平面所对应的方程可表示为:

$$f(x) = \omega^T \phi(x) + b \quad (5.38)$$

其中, ω 和 b 为参数。类似公式(5.18),有:

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 \quad (5.39)$$

其中,约束条件为 $y_i(\omega^T \phi(x_i) + b) \geq 1, i = 1, 2, \dots, n$ 。写出其对偶问题,可得到计算公式(5.40):

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \quad (5.40)$$

约束条件为: $\sum_{i=1}^n \alpha_i y_i = 0$, 且 $\alpha_i \geq 0$ 。

由于在线性支持向量机学习的对偶问题里对公式(5.38)进行求解时,目标函数和分类决策函数都要计算 $\phi(x_i)^T \phi(x_j)$,也就是计算样本 x_i 和样本 x_j 映射到特征空间之后的内积。因此不需要显式地指定非线性变换,当数据集维度非常高时,要计算出内积是非常困难的。所以可以使用核函数来替换内积来计算,则有计算公式如下:

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (5.41)$$

通过函数 $k(\cdot, \cdot)$ 可以计算得到样本 x_i 和样本 x_j 映射到特征空间之后的内积。

即可以针对公式(5.39),用 $k(x_i, x_j)$ 代替 $\phi(x_i)^T \phi(x_j)$,得到针对非线性问题的划分超平面方程:

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_{i=1}^n \alpha_i \quad (5.42)$$

约束条件为: $\sum_{i=1}^n \alpha_i y_i = 0$, 且 $\alpha_i \geq 0$ 。

关于核函数有如下定理:令 \mathcal{X} 为输入空间,对于任意数据 $x_i, x_j \in \mathcal{X}, i, j = 1, 2, \dots, n$ 。 $k(x_i, x_j)$ 定义在 $\mathcal{X} \cdot \mathcal{X}$ 上的对称函数。核矩阵(Kernel Matrix) \mathbf{K} 是半正定矩阵:

$$\mathbf{K} = \begin{pmatrix} k(x_1, x_1) & \mathbf{L} & k(x_1, x_n) \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ k(x_n, x_1) & \mathbf{L} & k(x_n, x_n) \end{pmatrix} \quad (5.43)$$

该定理表明,只要一个对称函数所对应的和矩阵半正定,它能够作为核函数使用。下面对常用的几种核函数分别做介绍。

1. 线性核函数

线性核函数(Linear Kernel Function)是形式最为简单的核函数,其表达式如下所示:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \quad (5.44)$$

2. 多项式核函数

多项式核函数(Polynomial Kernel Function)表达式如下所示:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^d \quad (5.45)$$

其中, $d=1$ 时表示线性核函数; $d>1$ 表示多项式的次数。

3. 径向基核函数

径向基核函数(Radial Basis Function, RBF)也称为高斯核函数(Gaussian Kernel

Function), 其表达式如下所示:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (5.46)$$

其中, $\sigma > 0$ 表示高斯核的带宽。

4. 拉普拉斯核函数

拉普拉斯核函数的表达式如下所示:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma}\right) \quad (5.47)$$

其中, $\sigma > 0$ 表示参数。

5. Sigmoid 核函数

Sigmoid 核函数(Sigmoid Kernel Function)的表达式如下所示:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^T \mathbf{x}_j + \theta) \quad (5.48)$$

其中, \tanh 为双曲正切函数, $\theta < 0, \beta > 0$ 。

核函数的选择对支持向量机分类性能有很大影响, 如果选择了不合适的核函数, 可能会因为把数据样本映射到不合适的特征空间而导致分类结果欠佳。核函数的选择不局限于上述提到的几种, 上述几种核函数在日常情况下使用得较多。比如, 在自然语言处理领域的文本分类中, 常使用线性核函数。在不清楚数据的详细信息时, 可以尝试使用径向基核函数。

5.3.4 分类过程

支持向量机的主要特点如下: 第一, 支持向量机基于数学理论, 通过逐步推导和计算, 克服了传统神经网络学习依赖经验和启发式先验成分的特点, 并且都有公式可寻。第二, 支持向量机结合了经验风险最小化和结构风险最小化的原理来寻找目标函数, 避免了训练过程中的过拟合问题, 提高了模型在新的测试数据集上的泛化能力。第三, 支持向量机的分割超平面只与样本边界上的支持向量机点有关, 而且这些点比所有样本点都少。因此, 在使用支持向量机进行数据挖掘时, 与其他机器学习算法相比, 支持向量机可以大大降低计算复杂度, 减少维数灾难的影响。第四, 面对不同类型的问题, 模型选择的好坏直接影响支持向量机的分类性能。模型选择主要是指支持向量机中模型判断因素的选择和参数的搜索方法。根据不同的具体问题, 需要选择符合问题实际情况的相应模型。在使用支持向量机算法进行数据挖掘时, 主要包括两个步骤: 训练和预测。训练主要是对已知类别的样本点进行监督训练。预测是根据训练好的模型对新的未知类别的数据进行计算, 从而对数据的类别进行预测。结果预测的准确性与支持向量机的训练过程密切相关。支持向量机的训练方法主要有增量算法、分解算法、块算法和 SMO 算法, 其中 SMO 算法是目前求解实际问题中应用最广泛的优化方法。

其中, 增量算法是在利用支持向量机算法进行训练时, 如果此时将新的样本点数据输

入进来,则不需要把新的样本点数据和原始数据混合起来,从头开始训练,而是通过修改或删除新的样本点与原来的模型之间有关联的部分完成训练,其他没有关联的部分则不进行修改或删除。另外利用该算法进行训练时不是一次性就能完成的,而是通过不断增加数据集、不断进行迭代优化来完成的。

块算法通常采用迭代策略逐步地减去非支持的向量,从而达到改变训练过程中数据集大小的目的。其具体实施方法为,将一个较为复杂的二次规划问题拆分为若干个小的二次规划问题,再进行求解,并把矩阵中的拉格朗日乘子为零的所有行和列进行删除。当支持向量的样本数目远远小于总的训练样本数目时,块算法可以提高训练效率。利用块算法在数据上进行训练可以快速地得到一个用于分类或回归预测的模型。

分解算法是求解大规模二次规划问题最有效、最常用的方法。分解算法的工作原理是将一个二次规划问题分解为一系列小的二次规划子问题进行后续的迭代计算。使用该算法进行分类训练时,需要将训练数据集分为两部分:一部分是工作集;另一部分是非工作集。主要是在工作集上进行训练,因此需要用尽可能少的工作集来保证训练效果,这就需要一种良好的划分方法,将训练数据集有效地划分为两部分,使工作集尽可能小,并包含支持向量机的所有样本点。

SMO 算法是一种启发式的算法,该算法的基本原则是假如所有变量的解都能达到最优化问题的 KKT 条件,那么最优化问题的解就可以通过计算求得。因为 KKT 条件是求解最优化问题的充要条件,如果并非所有变量都满足这些条件,那么就可以选择两个变量,固定其他变量,单独针对这两个变量构建一个二次规划问题,而这个问题对应这两个变量的解,是近似原始问题的解。通过这种方式迭代地把原二次规划问题划分为一些子二次规划问题进行求解,从而达到计算原始二次规划问题的目的。注意,子问题两个变量中只有一个是自由变量,另一个由等式约束确定。把 SMO 算法应用于求解支持向量机的划分超平面模型,就是先固定 α 之外的所有参数,再求以 α 为变量的极值。由于存在约束,如果固定 α 之外的其他变量,则 α 可以由其他变量计算得出。因此,SMO 每次固定其他的参数,选择两个合适的变量 α_i 和 α_j 。合适的变量必须满足两个条件:第一个条件是 α_i 和 α_j 的值必须要在 α 分隔边界之外;第二个条件是 α_i 和 α_j 都不在边界上。如此一来,在参数初始化后,SMO 算法不断执行以下两个步骤直至收敛:

- (1) 选取一对需要更新的变量 α_i 和 α_j ;
- (2) 固定 α_i 和 α_j 以外的参数,求解公式(5.27)获得更新后的 α_i 和 α_j 。

在支持向量机分类模型的构建过程中,使用诸如 SMO 算法等优化算法对公式(5.28)或(5.37)求解后,得出拉格朗日乘子 α^* 和法向量 ω^* 、位移项 b^* 的值后,便可以计算出硬间隔最大化线性支持向量机的最优划分超平面函数。而最优分类函数表达式为:

$$f(x) = \text{sign}(\omega^* \cdot x + b^*) = \left[\text{sign} \left(\sum_{i,j=1}^n \alpha_i^* y_i x_i^T x_j + b^* \right) \right] \quad (5.49)$$

硬间隔最大化线性支持向量机算法如算法 5.3 所示。

算法 5.3 硬间隔最大化线性支持向量机算法

- 1: 输入: 训练数据集 $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$; 类别空间 $y_i = \{-1, 1\}, i = 1, 2, \dots, n$
 2: 构造划分超平面目标函数: 见公式(5.42), 目标函数满足约束 $\alpha_i \geq 0, i = 1, \dots, n$, 且

$$\sum_{i=1}^n \alpha_i y_i = 0。$$

- 3: 使用 SMO 算法求出划分超平面目标函数最小时的 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)^T$
 4: 通过 α^* 向量值计算出法向量 $\omega^* = \sum_{i=1}^n \alpha_i^* y_i x_i = 0$
 5: 找出所有 S 个支持向量, 也就是满足条件 $\alpha_s > 0$ 的对应的样本 (x_s, y_s) , 通过 $y_s \left(\sum_{i=1}^n \alpha_i y_i x_i^T x_s + b \right) = 1$, 计算出每个样本 (x_s, y_s) 所对应的 b_s^* 对应的平均值, 即为最终的 $b^* = \frac{1}{s} \sum_{i=1}^s b_s^*$
 6: 计算出划分超平面函数 $f(x) = \text{sign}(\omega^* \cdot x + b^*)$, 得出分类决策函数并输出

软间隔最大化线性支持向量机算法如算法 5.4 所示。

算法 5.4 软间隔最大化线性支持向量机算法

- 1: 输入: 训练数据集 $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$; 类别空间 $y_i = \{-1, 1\}, i = 1, 2, \dots, n$
 2: 选择一个惩罚系数 $C > 0$, 构造划分超平面目标函数: 见公式(5.42), 目标函数满足约束

$$0 \leq \alpha_i \leq C, i = 1, \dots, n, \text{ 且 } \sum_{i=1}^n \alpha_i y_i = 0$$

- 3: 使用 SMO 算法求出划分超平面目标函数最小时的 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)^T$
 4: 通过 α^* 向量值计算出法向量 $\omega^* = \sum_{i=1}^n \alpha_i^* y_i x_i = 0$
 5: 找出所有 S 个支持向量, 也就是满足条件 $\alpha_s > 0$ 的对应的样本 (x_s, y_s) , 通过 $y_s \left(\sum_{i=1}^n \alpha_i y_i x_i^T x_s + b \right) = 1$, 计算出每个样本 (x_s, y_s) 所对应的 b_s^* 对应的平均值, 即为最终的 $b^* = \frac{1}{s} \sum_{i=1}^s b_s^*$
 6: 计算出划分超平面函数 $f(x) = \text{sign}(\omega^* \cdot x + b^*)$, 得出分类决策函数并输出

非线性支持向量机的最优划分超平面的最优分类函数表达式为:

$$f(x) = \text{sign}(\omega^T \phi(x) + b) = \left[\text{sign} \left(\sum_{i,j=1}^n \alpha_i^* y_i k(x_i, x_j) + b^* \right) \right] \quad (5.50)$$

使用径向基核函数的非线性支持向量机算法如算法 5.5 所示。

算法 5.5 非线性支持向量机(使用径向基核函数)算法

- 1: 输入: 训练数据集 $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$; 类别空间 $y_i = \{-1, 1\}, i = 1, 2, \dots, n$
 - 2: 选择一个惩罚系数 $C > 0$, 与径向基核函数构造划分超平面目标函数: 见公式(5.42), 目标函数满足约束 $0 \leq \alpha_i \leq C, i = 1, 2, \dots, n$, 且 $\sum_{i=1}^n \alpha_i y_i = 0$ 。
 - 3: 使用 SMO 算法求出划分超平面目标函数最小时的 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)^T$
 - 4: 通过 α^* 向量值计算出法向量 $\omega^* = \sum_{i=1}^n \alpha_i^* y_i x_i = 0$
 - 5: 计算出 $b^* = y_j - \sum_{i=1}^n \alpha_i^* y_i k(x_i, x_j)$
 - 6: 计算出划分超平面函数 $f(x) = \text{sign}\left(\sum_{i,j=1}^n \alpha_i^* y_i k(x_i, x_j) + b^*\right)$, 得出分类决策函数并输出
-

5.4 并行支持向量机模型的构建

通过 5.3 节对支持向量机中相关原理的介绍, 可以将使用支持向量机进行大数据分类的建模步骤总结如下。

(1) 将需要进行分类的相关数据转换为支持向量机所支持的数据格式, 即每一条数据由两部分组成: 一部分为数据的类属性标签, 若为二分类问题则标签为 0 与 1, 若为多分类问题则对标签数量按照此形式进行增加; 另一部分则为类属性标签对应的多种属性值。

(2) 对实验数据进行训练数据集与测试数据集的划分, 一般在比较实验结果时通过 K 折交叉验证方式获得多个测试数据集下的结果并取平均值, 因此进行实验数据划分时应根据数据量的大小按照 K 折交叉验证方式进行划分。

(3) 使用归一化方式对数据中的属性值部分进行归一化处理, 以避免较大的属性值对较小属性值的支配影响, 归一化方式一般选择最大最小归一化处理。

(4) 对支持向量机的核函数进行选择, 支持向量机可选择得出核函数主要有线性核函数、多项式核函数以及高斯核函数, 一般优先选择高斯核函数, 另外还可通过实验比较的方式进行选择。

(5) 在不同的核函数中存在不同参数, 在对参数值进行设置时可通过交叉验证的方式进行确定。

(6) 使用设置了最佳参数值的支持向量机进行训练数据集的训练过程。

(7) 使用测试数据集对训练好的支持向量机模型进行测试, 通过各种性能比较指标确定此模型的性能是否达到预设的期望值。

(8) 当模型的性能达到预设的期望值时, 支持向量机分类模型完成构建, 之后可以将其应用到其他数据集进行大数据分类。

在上述步骤中, 支持向量机分类模型的建模复杂点主要体现在三方面: 首先是在实

验前需要对数据集按照分类要求进行数据格式的转换；然后在对支持向量机核函数进行选择时通过比较实验确定最适合的核函数；最后则是进行核函数参数值确定时所需要的交叉验证方式，其中核函数的选择以及核函数参数值将直接影响模型最后的分类结果。

由于核函数主要从线性核函数、多项式核函数以及高斯核函数这三种中进行选择，因此在实验阶段可以通过对比实验来选择最适合的核函数，但核函数中的参数取值范围较大，难以通过简单的对比实验来确定，倘若使用交叉验证的方式进行获取，可能获得的参数值为局部最优值，非目标范围内的最优值，而目前的研究中并未有一种公认的方法可以用于核函数参数值的确定。

通过前面对群智能优化算法的介绍可以得知，群智能优化算法主要用于在目标范围内进行问题最优解的搜索，而支持向量机核函数参数值的确定正好可以视作目标范围内最优核函数参数值的搜索过程，因此将群智能优化算法与支持向量机核函数参数值寻优相结合可以很好地解决核函数参数值难以确定的问题。前面已经通过算法性能对比实验证明了协同鸟群算法用于优化问题求解的有效性，因此将协同鸟群算法与支持向量机相结合，以此构建并行支持向量机模型并用于大数据分类。

结合协同鸟群算法的相关原理以及支持向量机的建模步骤，可以将并行支持向量机模型的建模步骤归纳总结如下。

(1) 将需要进行分类的相关数据转换为支持向量机所支持的数据格式，即每一条数据由两部分组成：一部分为数据的类属性标签，若为二分类问题则标签为 0 与 1，若为多分类问题则对标签数量按照此形式进行增加；另一部分则为类属性标签对应的多种属性值。

(2) 对实验数据进行训练数据集与测试数据集的划分，一般在比较实验结果时通过 K 折交叉验证方式获得多个测试数据集下的结果并取平均值，因此进行实验数据划分时应根据数据量的大小按照 K 折交叉验证方式进行划分。

(3) 使用归一化方式对数据中的属性值部分进行归一化处理，以避免较大的属性值对较小属性值的支配影响，归一化方式一般选择最大最小归一化处理。

(4) 对支持向量机的核函数进行选择，支持向量机可选择核函数主要有线性核函数、多项式核函数以及高斯核函数，一般优先选择高斯核函数，另外还可通过实验比较的方式进行选择。

(5) 设置鸟群种群的个体数量、小团体数量、飞行间隔 FQ 、表示乞讨者跟随生产者寻找食物的行为的参数 FL_{\max} 与 FL_{\min} 、感知系数 C 与社会加速系数 S 、保持警戒行为中的常量 a_1 与 a_2 以及统计连续出现适应度降低或保持不变的参数 T 。

(6) 将支持向量机核函数中的参数作为协同鸟群算法中种群个体的位置形式，若参数数量为 1，则种群个体位置信息为一维；若参数数量为 2，则种群个体位置信息为二维；以此类推，在目标范围内随机初始化所有个体的位置信息。

(7) 以基于不同参数值的支持向量机对训练数据集进行分类训练后的误差值作为不同种群个体的适应度值，对基于不同参数值的支持向量机进行训练，从而得到其对应的适应度值，适应度值越低，则对应个体的位置信息越好。

(8) 按照预先设置的小团体数量随机在整个种群中选择对应数量的个体作为小团体

的中心点,之后按照协同鸟群算法中的小团体分配原则依次将个体分配至小团体,通过小团体内的个体适应度值比较确定小团体的历史最优适应度值。

(9) 根据飞行间隔 FQ 判断个体是否进行飞行行为,若为飞行行为,则为每个个体分配生产者以及乞讨者的身份,之后每个个体都按照其身份对应的位置更新公式进行位置更新。

(10) 若不为飞行行为,则根据相关准则判断个体是进行觅食行为还是保持警戒,并使用相关公式对位置信息进行更新。

(11) 计算每个个体更新后的位置信息以及对应的适应度值,比较更新前后的适应度值同时对每个个体对应的参数 T 进行更新。

(12) 根据参数 T 计算得出每个个体接受此次更新状态的概率 P ,之后按照概率 P 判断个体是否接受此次更新并进行相关操作。

(13) 对个体的历史最优适应度值、历史最优位置信息以及种群的历史最优适应度值、历史最优位置信息进行更新。

(14) 根据预设的迭代次数重复步骤(8)~步骤(13),当达到最大迭代次数时停止迭代过程,输出种群的历史最优位置信息,此位置信息即为支持向量机进行分类的最优核函数参数值或核函数参数组合。

(15) 使用设置了最佳参数值的支持向量机进行训练数据集的训练过程。

(16) 使用测试数据集对训练好的支持向量机模型进行测试。

(17) 并行支持向量机分类模型完成构建,之后可以将其应用到其他同类型数据集进行大数据分类。

并行支持向量机分类模型的流程如图 5.6 所示。

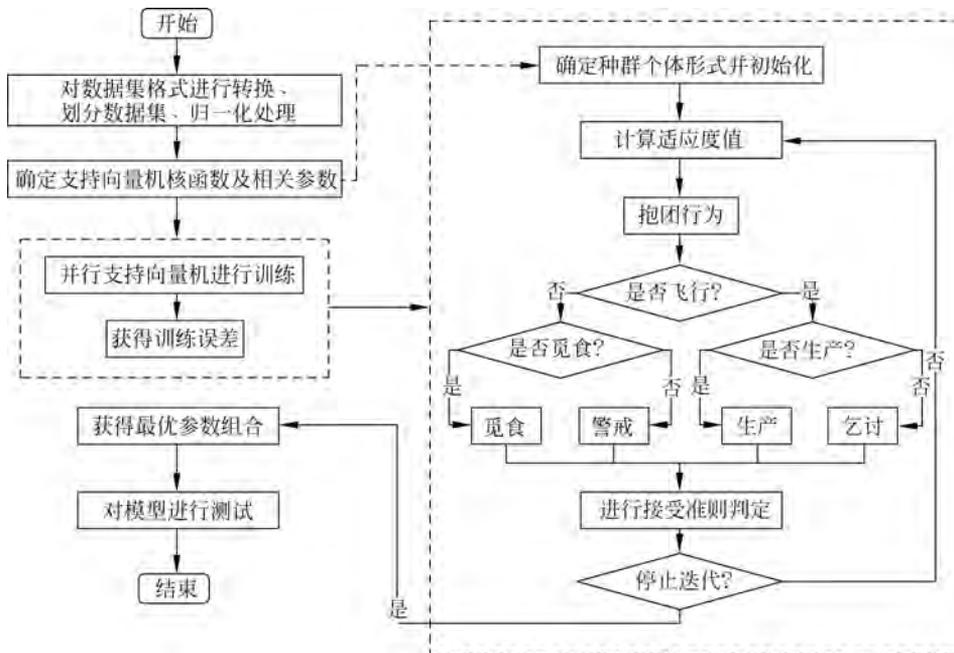


图 5.6 并行支持向量机分类模型的流程

按照上述并行支持向量机的相关建模步骤,可以将并行支持向量机的特性总结如下:

(1) 由于不同类型的数据集之间数据分布特征存在较大区别,因此在对不同类型的数据集进行分类之前,需要通过对比确定最为适合的核函数;

(2) 支持向量机核函数中的参数数量较少,因此相较于大部分优化问题,此寻优过程较为简单;

(3) 在使用协同鸟群算法进行参数寻优时,需要同时对种群数量个支持向量机进行训练数据集的训练过程,这也是并行支持向量机并行性的主要体现;

(4) 相较于使用交叉验证等传统方法确定核函数参数值,并行支持向量机同时对多个支持向量机进行训练,可以节省大量训练时间;

(5) 并行支持向量机使用协同鸟群算法进行目标范围内的核函数参数搜索过程,能够更好地发现目标范围内的全局最优值;

(6) 对传统支持向量机的训练过程结束后,由于参数值的不确定性,需要判断其分类性能是否符合预期,而并行支持向量机的训练过程结束后可直接获得符合预期的参数值;

(7) 对并行支持向量机进行训练后,可直接获得一个具有较好性能的大数据分类模型,并将其应用到其他同类型数据集进行大数据分类;

(8) 并行支持向量机的模型复杂度主要体现在使用协同鸟群算法进行参数寻优,因此其复杂度与协同鸟群算法相同;

(9) 在参数设置上,并行支持向量机只增加了协同鸟群算法中的相关参数,此外并未新增其他的参数。

5.5 知识扩展

在对协同鸟群算法进行算法性能验证时使用了萤火虫算法与磷虾群算法作为对比算法,因此本节将专门针对这两种算法的基本原理及其实现步骤展开相关介绍,同时针对前面章节所涉及的实验中使用到的几种群智能优化算法,展开它们各自算法特性的对比分析。

5.5.1 萤火虫算法

萤火虫算法是由 Xin-She Yang 于 2010 年提出的一种群智能优化算法,在自然界中,萤火虫之间通过自身发光来吸引异性前来交配以及吸引猎物进行捕猎,而该算法主要仿照自然界中萤火虫之间受彼此的亮度而相互吸引的行为来进行目标范围内的寻优过程。

自然界中的萤火虫数量大约有两千多种,其中大多数萤火虫均会产生短暂而有节奏的闪光,不同的萤火虫物种之间具有不同的闪光时长与频率。萤火虫闪光的最基本功能主要包括两方面:一个是对异性萤火虫进行吸引,从而达到交配的目的;另一个则是对潜在猎物的吸引,进而达到捕食的目的。通常来说,闪光的节奏、速度以及时间长度,构成了某个物种的萤火虫独有的信号,因此只有同一个物种的萤火虫才会被彼此之间的闪光所吸引。

萤火虫的光源较弱,一般只有在黑夜才能够被观察到,而光在空气中进行传播时,其

强度大小受到观察者距离光源距离大小的影响,即光的强度大小与距离的平方成反比,当对同一个光源进行观察时,增加观察者与光源之间的距离,则能被观察到的光亮会随着距离的增加变得越来越弱,这个因素使得萤火虫之间只能在有限的距离内被彼此所吸引。将上述萤火虫之间的吸引行为进行理想化,萤火虫算法的基本思想可以归纳如下:

(1) 种群中所有萤火虫都是雌雄同体的,即种群中的每只萤火虫都会被其他萤火虫所吸引,与它们的性别无关;

(2) 萤火虫的发光行为假定在黑夜,发光亮度只与不同萤火虫之间的距离有关,不受任何其他因素的影响;

(3) 萤火虫彼此之间相互吸引,吸引力的强弱受它们各自亮度的影响,亮度较弱的个体会被吸引着向亮度较强的个体移动;

(4) 具体吸引力的大小与萤火虫之间的距离相关,随着距离的增加,吸引力也会随之减小;

(5) 萤火虫的亮度大小类比为萤火虫个体的适应度值,即当对优化问题进行求解时,个体的适应度值越大,萤火虫的亮度越大,对其他个体的吸引力也越大。

在上述基本思想中,主要涉及的概念为吸引力和个体间的位置移动。下面将分别进行介绍。

1. 吸引力

在萤火虫算法中,每个萤火虫的位置代表了一个待求问题的可行解,而萤火虫的亮度表示该萤火虫位置的适应度值,亮度越高的萤火虫个体在解空间内的位置越好。在解空间内,每个萤火虫会向着亮度比自己高的萤火虫飞行来搜寻更优的位置,亮度越大对其他的萤火虫的吸引度越大。同时,萤火虫之间光的传播介质会吸收光,降低光的亮度,影响光的传播,所以萤火虫之间的吸引度会随着空间距离成反比,即两只萤火虫之间的吸引度会随着这两只萤火虫之间距离的增大而减小。基于此,使用相互吸引度公式对萤火虫之间的吸引力进行计算。

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (5.51)$$

其中, β_0 为初始吸引度值,即两只萤火虫之间的距离为 0 时的吸引度,初始 β_0 取值为 1; γ 为传播介质对光的吸收系数; r 为两只萤火虫之间的空间距离。

在计算两只萤火虫之间的空间距离时,采用下列公式:

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (5.52)$$

其中, d 表示位置信息的具体位置; x_{ik} 与 x_{jk} 分别表示个体 i 与个体 j 在第 k 个维度的具体位置信息。

2. 个体间的位置移动

受到吸引力的影响,种群中的每只萤火虫均会被亮度比自己大的萤火虫所吸引,进而向其所在的方向进行移动,而依次向所有亮度更大的个体移动完之后的位置才是萤火虫

的最终确定位置,具体的位置更新公式如下。

$$x_i^{t+1} = x_i^t + \beta(x_i^t - x_j^t) + \alpha * \text{rand}() \quad (5.53)$$

其中, x_i^t 为个体 i 在第 t 次迭代前的位置信息; x_j^t 为个体 j 在第 t 次迭代前的位置信息; x_i^{t+1} 为个体 i 在第 t 次迭代后的位置信息; β 为个体 j 对个体 i 的吸引度; α 为步长的扰动因子,一般为 $0 \sim 1$ 的随机数; $\text{rand}()$ 是 $-0.5 \sim 0.5$ 的满足均匀分布的随机数。由于萤火虫是朝向亮度比自己高的萤火虫进行飞行,因此在上述公式中个体 j 的适应度值高于个体 i 的适应度值。

对上述两个概念中存在的参数进行归纳,萤火虫算法中主要参数分别为:初始吸引度值 β_0 ,一般取值为 1;传播介质对光的吸收系数 γ ,一般取值为 1;步长的扰动因子 α ,一般为 $0 \sim 1$ 的随机数。结合上述相关思想与概念,可以将使用萤火虫算法进行优化问题求解时的特性总结如下。

(1) 萤火虫个体通过被其他萤火虫吸引从而在目标范围内对位置进行移动,此目标区域可以为多维,而不仅仅是二维,即求解的优化问题可以是多维的。

(2) 萤火虫亮度的大小由其适应度值的大小决定,即个体的位置信息越好,亮度越大。

(3) 萤火虫相互吸引力的大小只与彼此之间的空间距离相关,不受其他因素的影响。

(4) 萤火虫在进行位置移动时主要分为两部分:一部分为向其他个体的位置信息进行学习,通过此过程可以有效利用种群内部的位置信息;另一部分为受步长扰动因子的影响进行随机位移,以此扩大向外部进行随机搜索的范围。

(5) 步长扰动因子 α 的取值大小将直接决定算法的外部信息探索能力。

(6) 由于每个萤火虫个体会记录自身的亮度信息,因此整个种群的历史最优位置信息与最优适应度值会在迭代过程中进行保存与更新。

(7) 萤火虫算法中主要可进行调整的参数为步长的扰动因子 α 。

使用萤火虫算法对优化问题进行求解时的具体步骤可以归纳如下。

(1) 设置种群的个体数量、初始吸引度值 β_0 、传播介质对光的吸收系数 γ 以及步长的扰动因子 α 。

(2) 以萤火虫个体的位置信息作为待优化问题的解,以萤火虫个体的亮度作为解对应的适应度值,根据待优化问题的解的范围,随机初始化种群所有个体的位置信息。

(3) 根据待求解问题,计算种群中每个个体的适应度值,之后对种群个体的适应度值进行比较,将最高适应度值作为种群的历史最优适应度值,将其对应的位置信息作为种群的历史最优位置信息。

(4) 依次将每个个体与其他个体进行适应度值的比较,同时计算个体之间的空间距离,按照适应度值低的个体被适应度高的个体吸引的原则,确定每个个体分别被种群内哪些个体吸引并根据空间距离计算出每个个体受到的所有吸引力。

(5) 每个个体依次向所有适应度值比它高的个体位置方向进行移动,按照相关公式进行位置更新。

(6) 计算每个个体更新后的适应度值,对种群的历史最优适应度值、历史最优位置信息进行更新。

(7) 根据预设的迭代次数重复步骤(4)~步骤(6),当达到最大迭代次数时停止迭代过程,输出种群的历史最优位置信息,此位置信息即为算法优化后获得的问题最优解。

萤火虫算法的流程如图 5.7 所示。

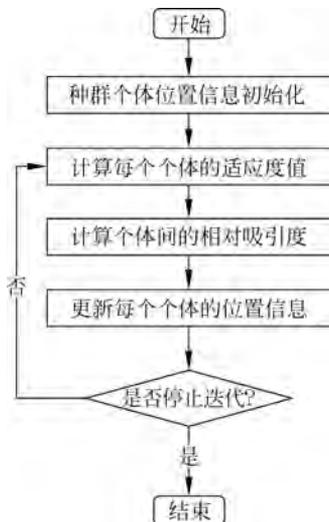


图 5.7 萤火虫算法的流程

按照上述流程对萤火虫算法进行归纳,如算法 5.6 所示。

算法 5.6 萤火虫算法

- 1: 初始化萤火虫种群中每个个体的位置信息
- 2: 设置种群数量 N 、参数 β_0 、 γ 与 α
- 3: 根据待求解问题计算每个个体的适应度值
- 4: 确定种群的历史最优位置信息以及历史最优适应度值
- 5: while(未满足停止迭代条件)do
- 6: for $i=1:N$
- 7: for $j=1:N$
- 8: 计算与其他个体间的空间距离、相对吸引力
- 9: 对位置信息进行更新
- 10: end for
- 11: end for
- 12: 计算位置更新后的适应度值
- 13: 更新种群的历史最优位置信息以及历史最优适应度值
- 14: end while
- 15: 输出种群的历史最优位置信息

下面以萤火虫算法对 Dixon-Price 测试函数进行目标范围内的最小值寻优为例进行说明,此测试函数为多峰函数,计算公式如公式(5.54)所示,其取值范围为 $[-10,10]$,取值范围内的理想最优解为 0,将其搜索的空间维度设为 20。

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2 \quad (5.54)$$

在参数设置方面,将萤火虫算法的种群数量设置为 50,初始吸引度值 β_0 设为 1,传播介质对光的吸收系数 γ 设为 1,步长的扰动因子 α 设为 1,迭代次数设为 500。

主要的寻优过程:首先在此函数的取值范围 $[-10, 10]$ 内随机生成 50 个 20 维数组作为种群个体的位置信息,将每个数组依次使用上述公式计算适应度值,确定适应度值最高的个体作为种群历史最优个体,保存该个体的位置信息以及适应度值;然后依次进行每个个体与种群中其他个体的适应度值比较,计算它们的空间距离并根据空间距离求得相互吸引度;之后根据相互吸引度依次对每个个体的位置进行更新,计算它们的新的适应度值,更新种群历史最优个体相关信息;最后重复上述适应度值比较及位置移动相关步骤,当迭代次数达到 500 时,停止算法迭代,输出此时的种群历史最优个体适应度值及其位置信息。

在优化过程中,种群个体的历史最优适应度值的大小随迭代次数的变化情况如图 5.8 所示。可以看到在整个迭代过程中,适应度值的主要变化阶段为迭代开始到迭代 50 次左右,此时历史最优适应度值具有较快的下降速度,说明此萤火虫算法在迭代前期能够具有一个较快的收敛速度,而在迭代 100 次之后,历史最优适应度值保持在一个相对稳定的状态,其中在迭代 300 次左右发生了一次较为明显的小幅变化,结合最优算法得到的历史最优适应度值为 0.2498。可以推测,在迭代 100 次到迭代 300 次之间,算法可能陷入了局部最优,但在迭代 300 次时跳出了此局部最优,之后获得了一个更好的寻优值。

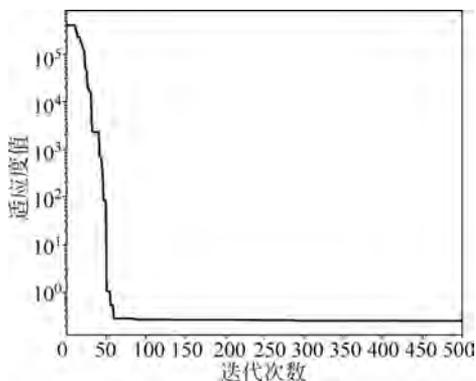


图 5.8 历史最优适应度值的变化情况

5.5.2 磷虾群算法

磷虾群算法是由 Gandomi 等人于 2012 年提出的一种种群智能优化算法,该算法主要对磷虾个体的聚集行为进行模拟,模拟的过程中,磷虾个体的移动行为主要受三方面的影响,分别为其他个体的存在引起的移动、觅食活动引起的移动以及随机扩散。

磷虾群算法中的磷虾主要指的是南极磷虾,它是目前研究得最好的海洋动物之一,研究者在对其观察时发现,这个物种的一个主要特征在于它们能够形成一个大的群体,磷虾群是这个物种的基本组成单位。当磷虾群的捕食者对磷虾群进行攻击时,它们会先将磷虾个体移走,从而降低磷虾群密度,而被捕食后磷虾群的形成主要有两个目标,分别为增加磷虾密度和获得食物。基于上述两个目标,磷虾群算法将食物作为目标最佳解决方案,将高密度作为磷虾个体在算法寻优过程中的移动趋势,让磷虾个体分别受到高种群密度趋势的吸引(其他个体的存在)、觅食活动、随机扩散这三方面的影响而进行自身的位置移动,除此之外,还将遗传算法中的交叉、变异这两个遗传算子加入到算法思想中以增加算

法的多样性。下面将分别就磷虾个体位置移动的几种影响因素展开相关介绍。

1. 磷虾个体主要移动行为对其造成的综合影响

由于磷虾群个体的移动主要朝着提高种群密度和觅食两个目的进行,因此将捕食者刚对磷虾群进行攻击后的分散状作为种群的初始状态,之后随着时间的流逝,磷虾个体开始进行位置移动,其位置移动主要受其他个体的存在引起的移动、觅食活动引起的移动以及随机扩散三方面的影响,因此可将使用拉格朗日模型将其移动距离与这三个因素之间的关系归纳如下:

$$\frac{dx_i}{dt} = N_i + F_i + D_i \quad (5.55)$$

其中, N_i 是由其他磷虾个体引起的运动; F_i 是觅食运动; D_i 是磷虾个体的随机物理扩散。

2. 其他个体引起的移动

一般在磷虾的移动过程中,主要受到一定范围内的其他个体的影响、种群中位置信息最优的个体的影响以及在上次移动中的影响惯性,因此可以将其他个体引起的移动使用下列公式进行归纳:

$$N_i^{\text{new}} = N_{\max} \alpha_i + \omega_n N_i^{\text{old}} \quad (5.56)$$

其中, N_i^{new} 为第 i 个个体在本次移动过程中受到其他个体影响而进行的运动; N_i^{old} 为第 i 个个体在上次移动过程中受到其他个体影响而进行的运动; ω_n 为惯性权重,取值范围为 $0 \sim 1$; N_{\max} 表示最大诱导速度; α_i 表示第 i 个个体受到一定范围内的其他个体以及种群中位置信息最优的个体的影响。其计算公式为:

$$\alpha_i = \alpha_i^{\text{local}} + \alpha_i^{\text{target}} \quad (5.57)$$

当磷虾群受到一定范围内其他个体的影响时,主要伴随着吸引或排斥运动。此影响的计算公式分别如下:

$$\alpha_i^{\text{local}} = \sum_{j=1}^M f_{ij} x_{ij} \quad (5.58)$$

$$x_{ij} = \frac{x_j - x_i}{x_j - x_i + \epsilon} \quad (5.59)$$

$$f_{ij} = \frac{f_i - f_j}{f_{\text{worst}} - f_{\text{best}}} \quad (5.60)$$

其中, x_i 与 x_j 分别表示第 i 个个体以及对其造成影响的第 j 个个体的位置信息; f_i 与 f_j 分别对应它们的适应度值; f_{worst} 与 f_{best} 分别为种群中最低的适应度值与最高的适应度值; ϵ 为避免分母为 0 而设置的极小值; M 为个体 i 周围能对其造成影响的个体总数。由上述公式可知,个体受到周围个体的影响主要由两个因素决定:一个是不同个体之间的距离;另一个是不同个体之间的适应值比较。当周围个体较第 i 个个体的适应度高,则 f_{ij} 为正值,个体将朝向周围个体移动;反之则周围个体远离此个体移动。

由于磷虾个体的位置是在不断变化的,因此在每一次移动位置后都需要对能够影响

其移动的个体进行确定,判定的依据为个体之间的空间距离,公式如下:

$$d_{si} = \frac{1}{5N} \sum_{j=1}^N x_j - x_i \quad (5.61)$$

其中, d_{si} 为磷虾之间的感应距离; N 为磷虾群的个体总数。若种群中某个个体与个体 i 之间的空间距离小于此感应距离,则个体 i 将会受到此个体的影响。

当磷虾个体受到种群中适应度值最高的个体影响时,此影响的计算公式如下:

$$\alpha_i^{\text{target}} = C^{\text{best}} f_{\text{ibest}} x_{\text{ibest}} \quad (5.62)$$

其中, f_{ibest} 与 x_{ibest} 的计算方式与个体受到周围某个个体影响的计算方式一致; C^{best} 表示适应度最高的个体对第 i 个个体进行影响的有效系数。其计算公式如下:

$$C^{\text{best}} = 2(\text{rand} + I/I_{\text{max}}) \quad (5.63)$$

其中, rand 为 $0 \sim 1$ 的随机数; I 为当前迭代次数; I_{max} 为预设的最大迭代次数,此有效系数随着迭代次数的增加而不断增大。

3. 觅食移动

磷虾的觅食移动主要由两个因素进行诱导:一个是食物所在的位置;另一个则是先前关于食物位置的相关经验。可以将觅食移动用下列公式进行描述:

$$F_i^{\text{new}} = V_f \beta_i + \omega_f F_i^{\text{old}} \quad (5.64)$$

其中, F_i^{new} 与 F_i^{old} 分别为个体 i 在本次移动与上次移动中的觅食移动量; ω_f 为惯性权重,取值范围为 $0 \sim 1$; V_f 为觅食速度;而受到食物的影响, β_i 主要由两部分组成,一部分为当前食物的吸引力,另一部分则为第 i 个个体的历史最优位置信息的影响,其计算公式为:

$$\beta_i = \beta_i^{\text{food}} + \beta_i^{\text{best}} \quad (5.65)$$

通常食物的具体位置是无法确定的,但可以通过每次迭代过程中个体的位置及其适应度值对食物的位置进行估计,估计的方式如下:

$$x^{\text{food}} = \frac{\sum_{i=1}^N \frac{1}{f_i} x_i}{\sum_{i=1}^N \frac{1}{f_i}} \quad (5.66)$$

通过估计食物的位置计算出其对应的适应度值,然后通过下列公式计算得出当前食物对个体的吸引力:

$$\beta_i^{\text{food}} = C^{\text{food}} f_{\text{ifood}} x_{\text{ifood}} \quad (5.67)$$

其中, f_{ifood} 与 x_{ifood} 的计算方式与个体受到周围某个个体影响的计算方式一致;而 C^{food} 表示食物影响系数。由于随着时间的流逝,食物对个体的影响将逐渐减小,因此食物系数取值是在不断变化的,其计算公式如下:

$$C^{\text{food}} = 2(\text{rand} - I/I_{\text{max}}) \quad (5.68)$$

一般磷虾群会被食物逐渐吸引,从而到达食物所在的位置,而由前面的定义可知食物所在的位置即为全局最优解,因此将食物影响系数按照上述方式进行设置有助于种群进行全局收敛。

第 i 个个体的历史最优位置信息对此个体位置移动的影响主要通过下列公式进行描述:

$$\beta_i^{\text{best}} = f_{\text{ibest}} x_{\text{ibest}} \quad (5.69)$$

其中, f_{ibest} 与 x_{ibest} 的计算方式与个体受到周围某个个体影响的计算方式一致。

4. 物理扩散移动

在磷虾个体进行物理扩散移动时,主要是基于最大扩散速度以及随机的扩散方向来进行描述的,其计算公式为:

$$D_i = D^{\max} \delta \quad (5.70)$$

其中, D^{\max} 为最大扩散速度; δ 为随机方向矢量,该矢量的数组为 $-1 \sim 1$ 的随机值。在磷虾个体进行物理扩散行为时,其随机方向矢量不受其他因素影响,但其最大扩散速度会随着迭代次数的增加而在不断减小,这主要是因为当磷虾个体的位置信息越好,则越不需要通过物理扩散进行位置移动,随着迭代次数的增加,个体的位置信息也在逐渐变好,则扩散速度相应逐渐变小。结合最大扩散速度不断减小的原理,可以磷虾个体物理扩散移动的计算公式总结如下:

$$D_i = D^{\max} (1 - I/I_{\max}) \delta \quad (5.71)$$

5. 磷虾个体位置移动

结合磷虾个体主要移动行为对其造成的综合影响,可将其在每次迭代中的位置移动公式归纳如下:

$$x_i(t + \Delta t) = x_i(t) + \Delta t \frac{dx_i}{dt_i} \quad (5.72)$$

式中, Δt 为速度的比例因子,因此该值的设置主要与优化问题的搜索空间直接相关,一般可以使用下列公式对其进行定义:

$$\Delta t = C_t \sum_{j=1}^V (UB_j - LB_j) \quad (5.73)$$

其中, V 表示优化问题中的变量总数; UB_j 与 LB_j 表示第 j 个变量的上下界; C_t 为 $0 \sim 2$ 的常数。

6. 遗传算子

为了提高算法的性能,磷虾群算法将遗传算法中的交叉操作与变异操作引入其中,以此来提高种群的多样性。交叉操作主要由交叉概率 Cr 进行控制,对个体位置信息的交叉操作可以使用下列公式进行描述:

$$x_{i,m} = \begin{cases} x_{r,m}, & \text{rand}_{r,m} < Cr \\ x_{i,m}, & \text{其他} \end{cases} \quad (5.74)$$

$$Cr = 0.2 f_{\text{ibest}} \quad (5.75)$$

上述交叉过程可以归纳为首先根据个体 i 当前的适应度值与历史最优适应度值的相

关情况计算个体 i 的交叉概率,然后针对个体 i 的位置信息中的每个变量随机生成一个 $0\sim 1$ 的均匀分布的随机数,当此数小于个体 i 的交叉概率时,随机选择种群中的一个个体的对应变量则作为个体 i 的新变量,反之,则该变量保持不变。

变异操作主要由变异概率 Mu 进行控制,对个体位置信息的变异操作可以使用下列公式进行描述:

$$x_{i,m} = \begin{cases} x_{\text{gbest},m} + \mu(x_{p,m} - x_{q,m}), & \text{rand}_{r,m} < Cr \\ x_{i,m}, & \text{其他} \end{cases} \quad (5.76)$$

$$Mu = 0.05/f_{\text{ibest}} \quad (5.77)$$

其中, $x_{\text{gbest},m}$ 表示种群历史最优位置信息的第 m 个变量; $x_{p,m}$ 与 $x_{q,m}$ 表示随机选取的两个个体的第 m 个变量; μ 表示 $0\sim 1$ 的随机数。

上述变异过程可以归纳为首先根据个体 i 当前的适应度值与历史最优适应度值的相关情况计算个体 i 的变异概率,然后针对个体 i 的位置信息中的每个变量随机生成一个 $0\sim 1$ 的均匀分布的随机数,当此数小于个体 i 的变异概率时,对该变量进行变异操作,反之则该变量保持不变。

在上述磷虾群算法原理中主要的参数分别为:最大诱导速度 N_{max} ; 惯性权重 ω_n , 取值范围为 $0\sim 1$; 觅食速度 V_f ; 惯性权重 ω_f , 取值范围为 $0\sim 1$; 食物影响系数 C^{food} , 该值与迭代次数与迭代周期相关; 最大扩散速度 D^{max} , 一般选取 $0.002\sim 0.01$ 的均匀分布的随机数; 参数 C_t , 一般为 $0\sim 2$ 的常数; 交叉概率 Cr 与变异概率 Mu , 均与当前个体的适应度值直接相关。通常, 惯性权重 ω_f 、 ω_n 以及参数 C_t 的初始值会分别设为 0.9 、 0.9 与 0.5 , 且它们均会在迭代过程中随着迭代次数的增加线性减小至 0.1 。结合上述相关算法原理, 可以将使用磷虾群算法进行优化问题求解时的特性总结如下:

- (1) 磷虾群个体在移动过程中提高种群密度, 同时搜索食物, 此移动范围不仅可以为二维, 还可以为多维, 即求解的优化问题可以是多维的;
- (2) 磷虾群理想的食物所在区域即为目标范围内的最佳位置信息;
- (3) 磷虾个体的位置移动受到三方面的影响, 以此形式进行位置移动可以扩大对食物的搜索范围;
- (4) 磷虾群个体在受到其周围个体的影响移动时, 周围个体的适应度值越高, 磷虾群个体向其移动位置的量越大, 通过此方式可以快速提高个体的适应度值;
- (5) 磷虾群个体在受到其他个体的影响移动时考虑到了种群最优位置信息对其造成的影响, 通过向种群最优位置进行移动可以提高算法收敛到全局最优的能力;
- (6) 磷虾群个体受到食物的吸引度会随着迭代次数的增加而不断减小, 可以有效避免在迭代后期陷入局部最优;
- (7) 磷虾群个体的觅食移动方式中增加了对个体历史最优位置的影响, 以此提高了种群内部信息的利用率;
- (8) 在磷虾个体进行物理随机扩散移动时, 移动的速度随着迭代次数的增加而不断减小, 以减小后期出现个体位置移动后适应度值降低的概率;
- (9) 将变异算法加入磷虾群算法中, 可以有效提高种群个体位置信息的多样性, 避免

个体朝着一个方向进行移动,有利于提高搜索到全局最优解的概率;

(10) 此算法中主要进行调节的参数为最大诱导速度 N_{\max} 和觅食速度 V_f 。

使用磷虾群算法对优化问题进行求解时的具体步骤可以归纳如下:

(1) 设置种群的个体数量、最大诱导速度 N_{\max} 、惯性权重 ω_n 初始值、觅食速度 V_f 、惯性权重 ω_f 初始值和参数 C_t 初始值;

(2) 将磷虾群个体的位置信息作为待优化问题的解,根据待优化问题的解的范围,随机初始化种群所有个体的位置信息;

(3) 根据待求解问题,计算种群中每个个体的适应度值,将其作为每个个体的历史最优适应度值,将此时对应的位置信息作为每个个体的历史最优位置信息,之后对种群个体的适应度值进行比较,将最高适应度值作为种群的历史最优适应度值,将其对应的位置信息作为种群的历史最优位置信息;

(4) 根据当前迭代次数及迭代周期对惯性权重 ω_n 、 ω_f 、参数 C_t 、食物影响系数 C^{food} 以及种群最优个体影响系数 C^{best} 进行更新;

(5) 依次对每个磷虾个体计算其他个体引起的移动量、觅食行为的移动量以及物理随机扩散的移动量;

(6) 按照上述三种移动量对个体的位置信息进行更新;

(7) 针对每个个体的适应度值和个体的历史最优适应度值计算其交叉概率与变异概率;

(8) 遍历每个个体位置信息中的每个变量,按照个体的交叉概率与变异概率对所有变量进行交叉与变异操作;

(9) 计算每个个体进行交叉与变异操作后的位置信息和对应的适应度值,对个体的历史最优适应度值、历史最优位置信息和种群的历史最优适应度值、历史最优位置信息进行更新;

(10) 根据预设的迭代次数重复步骤(4)~步骤(9),当达到最大迭代次数时停止迭代过程,输出种群的历史最优位置信息,此位置信息即为算法优化后获得的问题最优解。

按照上述过程对磷虾群算法进行归纳,如算法 5.7 所示。

算法 5.7 磷虾群算法

- 1: 初始化鸟群种群中每个个体的位置信息
- 2: 设置种群数量 N 、最大诱导速度 N_{\max} 、觅食速度 V_f 、参数 ω_n 、 ω_f 以及 C_t 初始值
- 3: 根据待求解问题计算每个个体的适应度值
- 4: 确定种群的历史最优位置相关信息以及个体历史最优位置相关信息
- 5: while(未满足停止迭代条件) do
- 6: 更新惯性权重 ω_n 、 ω_f 、参数 C_t 、最优个体影响系数 C^{best} 以及食物影响系数 C^{food}
- 7: for $i = 1: N$
- 8: 计算其他个体引起的移动量
- 9: 计算觅食行为的移动量
- 10: 计算物理随机扩散的移动量
- 11: 更新位置信息

- 12: 交叉与变异操作
- 13: end for
- 14: 计算位置更新后的适应度值
- 15: 更新种群的历史最优位置相关信息以及个体历史最优位置相关信息
- 16: end while
- 17: 输出种群的历史最优位置信息

磷虾群算法的算法流程如图 5.9 所示。

下面以磷虾群算法对 Schwefels 测试函数进行目标范围内的最小值寻优为例进行说明,该测试函数的计算公式如下所示,其取值范围为 $[-500, 500]$,取值范围内的理想最优解为 0,将其搜索的空间维度设为 20。

$$f(x) = 418.9829n - \sum_{j=1}^n x_j \sin(\sqrt{|x_j|}) \quad (5.78)$$

在参数设置方面将种群数量设为 50,诱导速度最大值 N_{\max} 设为 0.01,觅食速度 V_f 设为 0.02,惯性权重 ω_f 、 ω_n 以及参数 C_i 的初始值分别设为 0.9、0.9 与 0.5,且它们均会在迭代过程中随着迭代次数的增加线性减小至 0.1,迭代次数设为 500。

主要的寻优过程如下:首先在此函数的取值范围 $[-500, 500]$ 内随机生成 50 个 20 维数组作为磷虾群中每个个体的位置信息,将每个数组依次使用上述公式计算适应度值,并将其作为每个个体的历史最优位置信息与适应度值,通过比较选出适应度值最高的个体作为种群历史最优个体,保存该个体的位置信息以及适应度值;然后分别计算其他个体引起的移动量、觅食行为的移动量以及物理随机扩散的移动量,以此对每个个体进行位置更新;之后对每个个体进行交叉与变异操作,然后计算它们新的适应度值,更新个体历史最优信息以及种群历史最优信息;最后重复上述位置移动与交叉变异操作,当达到最大迭代次数 500 时停止迭代,输出此时的种群历史最优个体适应度值及其位置信息。

在优化过程中,种群的历史最优适应度值的大小变化情况如图 5.10 所示。

由图 5.10 可以看到,在整个迭代的过程中,虽然历史最优适应度值在不断减小,但其减小速度是在不断改变的。在迭代开始到迭代 200 次时,历史最优适应度值由接近 10^4 下降到 10,降幅较大且减速较快,这主要是因为,在迭代前期,控制算法全局搜索能力的惯性权重 ω_n 、 ω_f 、参数 C_i 、食物影响系数 C^{food} 均处于一个较大值状态,此时个体能够以较快的速度在搜索范围内搜索到相对较好的值,而当迭代次数由 200 增加到 300 这一阶段,磷虾群个体基本均处于一个相对较好的位置,因此算法开始注重局部搜索,历史最优适应度值的下降速度也相对较低,在迭代次数到达 300 次之后,历史最优适应度值基本保持不变,这可能是因为算法已经搜索到了一个较好的寻优值,最后算法迭代结束得到的最优寻

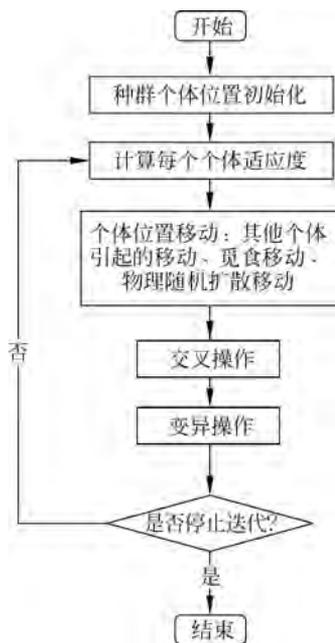


图 5.9 磷虾群算法的流程

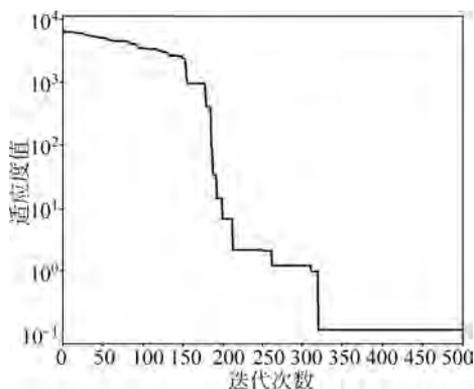


图 5.10 历史最优适应度值的变化情况

优值为 0.1029, 与理想最优解 0 较为接近。

5.5.3 算法特性对比分析

目前关于群智能优化算法性能的研究主要集中在对局部寻优能力与全局寻优能力的平衡上, 通过全局寻优可以帮助种群在目标区域扩大搜索范围, 发现更多可能存在的较优解, 通过局部寻优可以充分提高种群内部的信息利用率, 提高向现有搜索最优位置移动的速度, 而由于不同算法寻优方式的不同, 其局部寻优能力与全局寻优能力也完全不同, 因此本节将分别从算法的局部寻优能力、全局寻优能力、超参数数量以及时间复杂度这四方面, 对前面的章节中作为对比算法使用的灰狼优化算法、遗传算法、粒子群优化算法、鸟群算法、萤火虫算法以及磷虾群算法展开对比介绍。

这六种算法在这四方面的对比如表 5.4 所示。在算法的局部寻优能力方面, 这六种算法均具有相对较好的局部寻优能力, 其中粒子群优化算法与磷虾群算法的局部寻优能力是可以调节的, 在粒子群优化算法中, 当其惯性因子 w 选取较大值时, 个体对自身历史位置信息继承较多, 导致局部寻优能力相对较弱, 而当惯性因子 w 选取较小值时, 个体的位置更新主要受到自身历史最优位置信息以及种群历史最优位置信息的影响, 因此局部寻优能力相对较强; 在磷虾群算法中, 性权重 ω_n 、 ω_f 、参数 C_t 、食物影响系数 C^{food} 以及最优个体影响系数 C^{best} 是随着迭代次数不断变化的, 前四个参数随着迭代次数的增加而不断减小, 最后一个参数则随迭代次数的增加而不断增加, 这使得在迭代前期个体在位置移动时对历史更新量的继承较多, 因而具有较弱的局部寻优能力, 而到了迭代后期个体更多的是朝向种群当前最优位置信息进行移动, 因而能够更好地搜索局部区域。

在算法的全局寻优能力方面这六种算法的差别较大, 这主要是因为提出时间的不同, 研究者对全局寻优能力的考量也有所区别。遗传算法与粒子群优化算法的提出时间相对较早, 因此这两种算法实现全局寻优能力的方式相对较为简单, 其中遗传算法通过交叉与变异操作来实现, 而粒子群优化算法则通过对过去移动位置信息的继承来实现。萤火虫算法与磷虾群算法的提出时间在六种算法中处于中间阶段, 与前两种算法相比, 萤火虫算法中的步长扰动因子使位置移动具有更多的随机性, 磷虾群算法也加入物理随机扩散移

表 5.4 算法寻优能力对比

算 法	局部寻优能力	全局寻优能力	超参数数量	时间复杂度
遗传算法	通过选择操作选取种群中适应度较高的个体来获得较好的局部寻优能力	主要受交叉算法与变异算子的影响	2	$O(N^2)$
粒子群优化算法	由惯性因子 w 决定。该值较大时,局部寻优能力弱;反之,局部寻优能力强	较弱,主要通过过去移动位置信息的继承来获得较好的全局寻优能力	3	$O(N^2)$
灰狼优化算法	较强,个体均朝向位置信息最好的灰狼 α 、 β 和 δ 移动,可迅速收敛	受收敛因子 a 的影响。迭代前期 $a > 1$ 使得 $ A > 1$,全局搜索能力强;迭代后期 $a < 1$ 使得 $ A < 1$,全局搜索能力弱	1	$O(N^2)$
萤火虫算法	较强,个体受多个亮度强于自己的个体吸引而移动,可迅速收敛	主要受到步长的扰动因子影响,具有一定的随机性	3	$O(N^3)$
磷虾群算法	较强,个体在多种位置移动方式中均受到适应度值更高的位置信息的影响而移动	通过物理扩散移动来提高个体搜索能力,加入交叉操作、变异操作来提高种群的多样性	5	$O(N^2)$
鸟群算法	较强,觅食行为与警戒行为中分别通过学习个体自身历史最优、种群历史最优、种群中心位置以及其他个体历史最优来进行移动,收敛速度快	通过飞行间隔 FQ 的设置以平衡全局搜索与局部搜索的能力,飞行行为中的生产者行为可以对自身周围区域进行探索以增加种群的多样性	6	$O(N^2)$

动这一算法步骤,因此它们的全局寻优能力有所增强。而灰狼优化算法与鸟群算法的提出时间在这六种算法中最晚,因此这两种算法对全局寻优能力的考量也更多,其中灰狼优化算法通过收敛因子 a 的设定来进行全局寻优与局部寻优能力的平衡,鸟群算法则是通过设置飞行间隔 FQ,让个体通过不同的位置移动方式来平衡这两种能力。

超参数数量与时间复杂度主要与算法中的位置更新方式直接相关,磷虾群算法与鸟群算法中的个体位置更新方式相对于另外四种算法更为复杂,因此它们的超参数数量也相对多一些;而在时间复杂度方面,除萤火虫算法以外的其他算法主要是在迭代周期内依次对每个个体按照位置更新方式进行更新,因此它们的时间复杂度相同,均为 $O(N^2)$,而萤火虫算法在对其位置进行更新时需要学习所有亮度高于自身的个体位置信息,因此增加了一个遍历过程,导致算法复杂度相对更高一些。

由前面的算法介绍以及性能比较实验可以得知,不同的算法均存在各自的优缺点,且它们在不同优化问题上的表现也存在一定的区别,因此在实际的使用过程中不能够单凭算法某一方面的指标来进行选择,最好的方式是针对实际的优化问题选择多种不同优化算法进行相关实验,通过结果的对比来选择表现最好的优化算法。

5.6 本章小结

本章主要介绍了一种用于大数据分类的并行支持向量机模型的基本原理,此并行支持向量机主要将协同鸟群算法与支持向量机相结合,使用协同鸟群算法对支持向量机中核函数的相关参数进行优化,因此本章内容主要由四部分组成,分别为协同鸟群算法相关原理、支持向量机分类模型、并行支持向量机的构建以及知识扩展。

在协同鸟群算法相关部分,首先针对协同鸟群算法的基础算法——鸟群算法展开相关介绍,包括其算法原理与算法步骤等,然后针对原始算法中存在的相关问题,分别加入抱团行为、基于适应度差值比的位置更新方式以及接受准则来优化算法的性能,最后将协同鸟群算法与鸟群算法、萤火虫算法、磷虾群算法进行比较算法优化性能的比较,实验结果证明了无论是算法收敛速度还是收敛精度,协同鸟群算法较对比算法均具有更好的表现。

在支持向量机分类模型部分,主要通过支持向量机的统计学原理、核函数以及分类过程的实现等内容展开介绍来更好地理解使用支持向量机进行大数据分类的原理及过程。

之后对并行支持向量机的构建进行说明时,主要围绕使用并行支持向量机进行大数据分类的具体建模过程进行说明,包括并行支持向量机的核函数选择以及如何使用协同鸟群算法对核函数参数进行优化等,最后给出了具体的建模步骤及流程图。

最后在知识扩展部分分别对前面算法性能对比实验所使用到的萤火虫算法以及磷虾群算法的算法原理展开介绍,同时对使用这两个算法进行最小化问题寻优的具体过程做详细说明,之后以表格的形式对前面章节所使用到的六种群智能优化算法的寻优能力进行了对比分析。