预算分配算法任务文档

预算分配算法任务文档

一、任务背景

在广告系统中，每个广告活动（Campaign）都需要设置每日预算（Daily Budget）。这个预算会影响广告在一天中的曝光分布和整体表现。如果预算分配不合理，例如早上花光所有预算，后续流量高峰期将无法获得曝光，从而影响转化和销售。预算信息的介绍可参考广告官方文档（Daily budget overview）：https://advertising.amazon.com/help/GE5QEBS6QRJJAT3A

在实际运营中，我们往往面临两个问题：

1. **单个广告活动内部的小时级预算分配**：
   如何平滑地分配每日预算，使广告在一天中持续展示、覆盖高效时段并优化转化。
2. **多个广告活动之间的预算分配**：
   当总预算有限时（例如每天 100 美元），如何在多个广告活动间动态分配预算，使总体效果（如订单量、ROAS、ACoS）最优。

二、任务目标与层次

本项目分为两个层次任务，要求实习生独立完成并提交分析报告与实现代码。

🎯 任务一：单个广告活动的小时级预算分配

目标：
给定一个广告活动的每日预算（`daily_budget`），设计算法为其在 24 小时内分配预算 `budget_hour[h]`。

要求：

- 保证预算总和不超过每日预算；
- 避免早期预算耗尽；
- 结合历史表现（曝光、点击、转化、ROAS 等），在高效时段分配更多预算；
- 分配结果需平滑（相邻小时预算变化不大）。
- 充分需要考虑到周一到周天，每天的不同情况。只考虑接下来7天的预算分配情况，结合商品特点，结合是不是活动期间。

输入示例：

```
{
  "campaign_name": "Campaign_A",
  "historical_data": {
    "monday": [
      {
        "hour": 0,
        "impressions": 860,
        "clicks": 12,
        "spend": 5.8,
        "orders": 1,
        "revenue": 24.5,
        "ctr": 0.014,
        "cvr": 0.083,
        "cpc": 0.48,
        "acos": 0.237,
        "roas": 4.22
      },
      {
        "hour": 1,
        "impressions": 1275,
        "clicks": 18,
        "spend": 8.6,
        "orders": 2,
        "revenue": 45.2,
        "ctr": 0.014,
        "cvr": 0.111,
        "cpc": 0.48,
        "acos": 0.190,
        "roas": 5.26
      },
      // ... 2-23小时数据
    ],
    "tuesday": [
      // 周二24小时数据...
    ],
    "wednesday": [
      // 周三24小时数据...
    ],
    "thursday": [
      // 周四24小时数据...
    ],
    "friday": [
      // 周五24小时数据...
    ],
    "saturday": [
      // 周六24小时数据...
    ],
```

```
    "sunday": [
      // 周日24小时数据...
    ]
  },
  "constraints": {
    "min_daily_budget": 15,
    "max_daily_budget": 30,
    "weekly_budget_limit": 200,
    "target_acos": 0.25,
    "smoothness_factor": 0.3
  },
  "optimization_goals": {
    "primary": "maximize_revenue",
    "secondary": "minimize_acos",
    "constraint": "maintain_smooth_distribution"
  }
}
```

输出示例：

Code block

```
{
  "campaign_name": "Campaign_A",
  "weekly_budget_strategy": {
    "monday": {
      "daily_budget": 18,
      "hourly_allocation": {
        "0": 0.4, "1": 0.4, "2": 0.3, "3": 0.3, "4": 0.4, "5": 0.5,
        "6": 0.7, "7": 1.0, "8": 1.5, "9": 1.8, "10": 1.6, "11": 1.4,
        "12": 1.8, "13": 1.7, "14": 1.5, "15": 1.3, "16": 1.2, "17": 1.4,
        "18": 1.6, "19": 1.8, "20": 1.5, "21": 1.2, "22": 0.9, "23": 0.6
      },
      "total": 18.0
    },
    "tuesday": {
      "daily_budget": 20,
      "hourly_allocation": {
        "0": 0.4, "1": 0.4, "2": 0.3, "3": 0.3, "4": 0.4, "5": 0.5,
        "6": 0.8, "7": 1.2, "8": 1.6, "9": 1.9, "10": 1.7, "11": 1.5,
        "12": 1.9, "13": 1.8, "14": 1.6, "15": 1.4, "16": 1.3, "17": 1.5,
        "18": 1.7, "19": 1.9, "20": 1.6, "21": 1.3, "22": 1.0, "23": 0.7
      },
      "total": 20.0
    },
    "wednesday": {
```

```
        "daily_budget": 22,
        "hourly_allocation": {
          "0": 0.5, "1": 0.5, "2": 0.4, "3": 0.4, "4": 0.5, "5": 0.6,
          "6": 0.9, "7": 1.3, "8": 1.8, "9": 2.1, "10": 1.9, "11": 1.7,
          "12": 2.1, "13": 2.0, "14": 1.8, "15": 1.6, "16": 1.5, "17": 1.7,
          "18": 1.9, "19": 2.1, "20": 1.8, "21": 1.5, "22": 1.2, "23": 0.9
        },
        "total": 22.0
      },
      "thursday": {
        "daily_budget": 21,
        "hourly_allocation": {
          "0": 0.5, "1": 0.5, "2": 0.4, "3": 0.4, "4": 0.5, "5": 0.6,
          "6": 0.9, "7": 1.3, "8": 1.7, "9": 2.0, "10": 1.8, "11": 1.6,
          "12": 2.0, "13": 1.9, "14": 1.7, "15": 1.5, "16": 1.4, "17": 1.6,
          "18": 1.8, "19": 2.0, "20": 1.7, "21": 1.4, "22": 1.1, "23": 0.8
        },
        "total": 21.0
      },
      "friday": {
        "daily_budget": 25,
        "hourly_allocation": {
          "0": 0.6, "1": 0.6, "2": 0.5, "3": 0.5, "4": 0.6, "5": 0.7,
          "6": 1.0, "7": 1.4, "8": 1.9, "9": 2.2, "10": 2.0, "11": 1.8,
          "12": 2.2, "13": 2.1, "14": 1.9, "15": 1.7, "16": 1.6, "17": 1.8,
          "18": 2.0, "19": 2.2, "20": 1.9, "21": 1.6, "22": 1.3, "23": 1.0
        },
        "total": 25.0
      },
      "saturday": {
        "daily_budget": 28,
        "hourly_allocation": {
          "0": 0.8, "1": 0.8, "2": 0.7, "3": 0.7, "4": 0.8, "5": 0.9,
          "6": 1.0, "7": 1.1, "8": 1.2, "9": 1.4, "10": 1.6, "11": 1.8,
          "12": 2.0, "13": 2.1, "14": 2.0, "15": 1.9, "16": 1.8, "17": 1.9,
          "18": 2.0, "19": 2.1, "20": 2.0, "21": 1.7, "22": 1.4, "23": 1.1
        },
        "total": 28.0
      },
      "sunday": {
        "daily_budget": 24,
        "hourly_allocation": {
          "0": 0.7, "1": 0.7, "2": 0.6, "3": 0.6, "4": 0.7, "5": 0.8,
          "6": 0.9, "7": 1.0, "8": 1.2, "9": 1.5, "10": 1.7, "11": 1.9,
          "12": 2.1, "13": 2.0, "14": 1.9, "15": 1.8, "16": 1.7, "17": 1.8,
          "18": 1.9, "19": 1.8, "20": 1.6, "21": 1.4, "22": 1.1, "23": 0.9
        },
```

```
        "total": 24.0
      }
    },
    "weekly_total": 158,
    "summary": {
      "weekly_budget_distribution": {
        "monday": "11.4%", "tuesday": "12.7%", "wednesday": "13.9%",
        "thursday": "13.3%", "friday": "15.8%", "saturday": "17.7%", "sunday":
"15.2%"
      },
      "peak_performance_days": ["saturday", "friday", "sunday"],
      "recommended_weekly_budget": 158,
      "efficiency_ranking": ["saturday", "wednesday", "friday", "sunday",
"thursday", "tuesday", "monday"]
    }
}
```

评估指标：

- 平滑度指标（相邻小时差的方差）；
- 预算利用率；
- 模拟下的预计收益或订单数；
- 高频时段覆盖率。

🚀 任务二：多个广告活动的预算分配优化

目标：

当你有总预算 100 美元、10 个广告活动时，如何根据它们的历史表现和预测效果，将预算最优地分配给各活动。

要求：

- 总预算不超过 100；
- 各活动预算不超过其上限（`daily_budget`）；
- 优化目标：最大化整体收入、或最小化平均 ACoS；
- 支持动态更新：能根据历史数据迭代优化分配策略。

输入示例：

Code block

```
{
```

```
  "total_budget": 100,
  "optimization_target": "maximize_revenue", // 或 "minimize_acos"
  "historical_data": [
    {
      "date": "2025/10/21",
      "hour": 0,
      "campaign_name": "Campaign_A",
      "daily_budget": 20,
      "impressions": 860,
      "clicks": 12,
      "spend": 5.8,
      "orders": 1,
      "revenue": 24.5,
      "ctr": 0.014,
      "cvr": 0.083,
      "cpc": 0.48,
      "acos": 0.237,
      "roas": 4.22
    },
    {
      "date": "2025/10/21",
      "hour": 1,
      "campaign_name": "Campaign_A",
      "daily_budget": 20,
      "impressions": 1275,
      "clicks": 18,
      "spend": 8.6,
      "orders": 2,
      "revenue": 45.2,
      "ctr": 0.014,
      "cvr": 0.111,
      "cpc": 0.48,
      "acos": 0.190,
      "roas": 5.26
    },
    // ... 更多小时数据
    {
      "date": "2025/10/21",
      "hour": 0,
      "campaign_name": "Campaign_B",
      "daily_budget": 25,
      "impressions": 1482,
      "clicks": 22,
      "spend": 10.5,
      "orders": 3,
      "revenue": 52.8,
      "ctr": 0.015,
```

```
        "cvr": 0.136,
        "cpc": 0.48,
        "acos": 0.199,
        "roas": 5.03
      }
      // ... 其他活动和小时的数据
    ]
}
```

输出示例：

Code block

```
{
  "total_budget": 100,
  "optimization_target": "maximize_revenue",
  "allocation": {
    "Campaign_A": 15.2,
    "Campaign_B": 12.8,
    "Campaign_C": 9.5,
    "Campaign_D": 11.3,
    "Campaign_E": 8.7,
    "Campaign_F": 10.1,
    "Campaign_G": 13.5,
    "Campaign_H": 7.9,
    "Campaign_I": 6.4,
    "Campaign_J": 4.6
  },
  "allocation_rationale": {
    "Campaign_A": "high_roas_stable",
    "Campaign_B": "good_efficiency_growing",
    "Campaign_C": "moderate_performance",
    "Campaign_J": "low_efficiency_high_acos"
  },
  "expected_outcomes": {
    "total_revenue": 458.3,
    "average_acos": 0.23,
    "overall_roas": 4.58
  },
  "adjustment_recommendations": {
    "increase_budget": ["Campaign_A", "Campaign_G"],
    "decrease_budget": ["Campaign_J", "Campaign_I"],
    "monitor_closely": ["Campaign_B", "Campaign_D"]
  },
  "next_update_time": "2025-01-15T10:00:00"
```

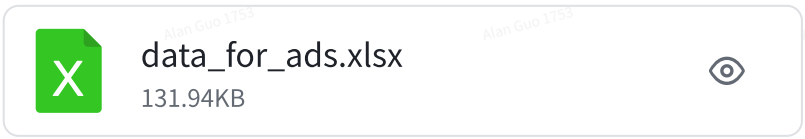
```
33    }
```

评估指标：

- 总订单量 / 收益；
- 总体 ROAS、平均 ACoS；
- 分配稳定性；
- 不同策略下的结果对比。

三、示例数据说明

已为你准备好一份 7 天 × 24 小时 × 10 个广告活动的历史表现样本



字段定义：

字段名	含义	示例
date	日期	2025/10/30
hour	小时（0-23）	15
campaign_name	广告活动名称	Campaign_A
daily_budget	每日预算上限	20
impressions	曝光量	1450
clicks	点击量	37
spend	当小时花费	18.42
orders	订单数	3
revenue	销售额	78.5
ctr	点击率	0.0255
cvr	转化率	0.081
cpc	平均点击花费	0.48
acos	广告花费占比	0.23
roas	广告投资回报率	4.2

四、交付内容

1. **Python 脚本**
    - 含数据处理、算法实现；
2. **报告（PDF 或 Markdown）**
    - 包含任务思路、方法、结果分析与结论；
3. **结果样例文件**
    - 输出的预算分配结果 JSON/CSV。

五、加分项（Optional）

- 对广告活动进行聚类，识别相似表现模式；
- 引入预测模型（如 XGBoost、LSTM）预测未来小时收益；
- 将算法封装成自动化预算优化模块。
- 你可以使用任何的统计模型或机器学习模型，你也可以自己模拟更多的历史表现数据，以便使用复杂的模型进行训练。

六、参考资源

- Amazon Advertising Help - Daily budget overview
- Scipy Optimization
- CVXPY Documentation
- Multi-Armed Bandit Overview (UCB, Thompson Sampling)
