### 7. 灰色预测 + 马尔可夫链案例：农产品月销量预测

**问题背景**：某农产品批发市场需预测某水果未来 6 个月的月销量，历史数据仅 12 个月（样本量小），且受季节、天气等随机因素影响，销量波动大（如夏季销量骤增，冬季骤减）。

**数据**：

* 过去 12 个月销量（吨）：120/110/130/150/200/250/230/180/160/140/130/120。

**要求**：用灰色预测 GM (1,1) 捕捉趋势，用马尔可夫链修正随机波动，输出未来 6 个月预测值及误差范围，对比单一灰色预测的精度提升。

### 7. 灰色预测 + 马尔可夫链代码：农产品月销量预测

|  |
| --- |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  # 1. 数据准备（过去12个月销量，单位：吨）  sales = np.array([120, 110, 130, 150, 200, 250, 230, 180, 160, 140, 130, 120])  months = np.arange(1, 13)  n = len(sales)  # 2. 灰色预测GM(1,1)  def grey\_prediction(data, forecast\_steps):  # 累加生成序列  x1 = np.cumsum(data)  # 构造矩阵B和向量Y  B = np.zeros((n-1, 2))  Y = np.zeros((n-1, 1))  for i in range(n-1):  B[i, 0] = -(x1[i] + x1[i+1]) / 2  B[i, 1] = 1  Y[i, 0] = data[i+1]  # 求解参数a, b  B\_T = B.T  a, b = np.dot(np.dot(np.linalg.inv(np.dot(B\_T, B)), B\_T), Y)  # 预测模型  x1\_pred = (data[0] - b/a) \* np.exp(-a \* np.arange(n + forecast\_steps)) + b/a  # 还原预测值  x0\_pred = np.zeros(n + forecast\_steps)  x0\_pred[0] = data[0]  for i in range(1, n + forecast\_steps):  x0\_pred[i] = x1\_pred[i] - x1\_pred[i-1]  return x0\_pred[:n], x0\_pred[n:] # 拟合值和预测值  # 灰色预测未来6个月  grey\_fit, grey\_forecast = grey\_prediction(sales, 6)  # 3. 马尔可夫链修正  def markov\_correction(observed, predicted, forecast\_steps):  # 计算残差  residuals = observed - predicted[:len(observed)]  # 划分状态（3个状态：低、中、高）  states = 3  bins = np.percentile(residuals, [i\*100/states for i in range(1, states)])  # 状态转移矩阵  trans\_matrix = np.zeros((states, states))  prev\_state = None  for r in residuals:  state = np.digitize(r, bins)  if prev\_state is not None:  trans\_matrix[prev\_state, state] += 1  prev\_state = state  # 归一化转移矩阵  trans\_matrix = trans\_matrix / np.sum(trans\_matrix, axis=1, keepdims=True)    # 预测残差状态  last\_residual = residuals[-1]  last\_state = np.digitize(last\_residual, bins)  forecast\_states = []  for \_ in range(forecast\_steps):  next\_state = np.random.choice(states, p=trans\_matrix[last\_state])  forecast\_states.append(next\_state)  last\_state = next\_state    # 生成修正残差（取对应状态的均值）  state\_residuals = [residuals[np.digitize(residuals, bins) == i] for i in range(states)]  state\_means = [np.mean(sr) for sr in state\_residuals]  forecast\_residuals = [state\_means[s] for s in forecast\_states]  return forecast\_residuals  # 修正预测结果  residuals\_correction = markov\_correction(sales, grey\_fit, 6)  final\_forecast = grey\_forecast + residuals\_correction  # 4. 结果展示  plt.figure(figsize=(12, 6))  plt.plot(months, sales, 'bo-', label='历史销量')  plt.plot(months, grey\_fit, 'r--', label='灰色拟合')  forecast\_months = np.arange(13, 13+6)  plt.plot(forecast\_months, grey\_forecast, 'g--', label='灰色预测')  plt.plot(forecast\_months, final\_forecast, 'mo-', label='马尔可夫修正预测')  plt.xlabel('月份')  plt.ylabel('销量（吨）')  plt.title('农产品月销量预测')  plt.legend()  plt.grid(True)  plt.show()  # 输出预测值  result = pd.DataFrame({  '月份': [f'第{i}月' for i in forecast\_months],  '灰色预测值': grey\_forecast.round(2),  '修正后预测值': final\_forecast.round(2),  '残差修正量': np.array(residuals\_correction).round(2)  })  print("\n未来6个月销量预测：")  print(result) |