我们来看看 LambdaRank 是如何通过 NDCG 指标定义梯度的。首先，对于 RankNet 的梯度，我们有如下推导：

$$\frac{\partial L}{\partial w_k} = \sum_{(i,j)\in P} \frac{\partial L_{ij}}{\partial w_k} = \sum_{(i,j)\in P} \frac{\partial L_{ij}}{\partial s_i}\frac{\partial s_i}{\partial w_k} + \frac{\partial L_{ij}}{\partial s_j}\frac{\partial s_j}{\partial w_k}$$

进一步的，可以观察到对模型 s 的求导有下面的对称性：

$$\frac{\partial L_{ij}}{\partial s_i} = \frac{\partial\{\frac{1}{2}\left(1 - S_{ij}\right)\left(s_i - s_j\right) + \log\{1 + \exp(-\left(s_i - s_j\right))\}\}}{\partial s_i} = \frac{1}{2}\left(1 - S_{ij}\right) - \frac{1}{1 + \exp\left(s_i - s_j\right)}$$
$$= -\frac{\partial L_{ij}}{\partial s_j}$$

因此，LambdaRank 有如下关于文档x_i和x_j的 Lambda 定义：

$$\lambda_{ij} \stackrel{\text{def}}{=} \frac{\partial L_{ij}}{\partial s_i} = -\frac{\partial L_{ij}}{\partial s_j}$$

我们首先考虑有序对(i, j)，因而有$S_{ij} = 1$，于是上述公式可以进一步简化：

$$\lambda_{ij} \stackrel{\text{def}}{=} \frac{1}{1 + \exp\left(s_i - s_j\right)}$$

至此可以得到每个文档x_i的 Lambda 值为：

$$\lambda_i = \sum_{(i,j)\in I} \lambda_{ij} - \sum_{(j,i)\in I} \lambda_{ij}$$

公式中I是文档对(x_i，x_j)的集合，其中文档x_i排在x_j前面，即更相关。

为了加强排序中顺序前后的重要性，LambdaRank 进一步在 Lambda 中引入评价指标 Z（如 NDCG），把交换两个文档的位置引起的评价指标的变化$|\Delta Z_{ij}|$作为其中一个因子：

$$\lambda_{ij} \stackrel{\text{def}}{=} \frac{1}{1 + \exp\left(s_i - s_j\right)} |\Delta Z_{ij}|$$

可以这么理解 Lambda，Lambda 量化了一个待排序的文档在下一次迭代时应该调整的方向和强度。

可以看出，LambdaRank 不是通过显示定义损失函数再求梯度的方式对排序问题进行求解，而是分析排序问题需要的梯度的物理意义，直接定义梯度，可以反向推导出 LambdaRank 的损失函数为：

$$L_{ij} = \log\{\, 1 + \exp(s_i - s_j)\}\, |\Delta Z_{ij}|$$