基于层次最优传输的日文汉字字符间距离图

Dominic Schuhmacher[†]

数学随机过程研究所，G¨ottingen大学

2023年4月6日

arXiv:2304.02493v1 [stat.ME] 5 Apr 2023

摘要

我们提出了一个通用框架，用于根据汉字之间的差异性为其分配距离。该术语的具体含义可能取决于实际应用。我们唯一的假设是，两个汉字之间的差异性可以通过对部件的嵌套结构进行最优匹配，并对所得惩罚值取加权平均来恰当地表示。对于匹配代价，我们提出了若干模块，这些模块可以自由组合，也可以替换为其他模块，包括配准后部件之间相对不平衡的墨水输运、配准所需变换之间的距离，以及预先指定标签之间的差异。

我们给出一个由此得到的汉字距离函数的具体示例，作为概念验证。基于该函数，我们通过多维缩放生成二维汉字地图，并列出一张表格，其中包含随机选取的 100 个 J¯oj¯o 汉字及其 16 个最近邻。

我们的汉字距离函数可帮助非中日韩背景的日文学习者掌握汉字读写能力。此外，这些函数还可协助汉字词典的编辑者展示其资料，也可用于文本处理和光学字符识别系统，以评估错误发生的可能性。

1 引言

日文通常被认为拥有世界上最复杂的书写系统之一。除了两套各含 48 个基本字符、并可通过附加变音符号进行变形的假名文字外，汉字也是其中不可或缺的一部分；这些汉字源自传统汉字书写体系，并经过改造后用于日文。一般而言，汉字用于名词（包括专有名词）、形容词和动词的前部（大致相当于词干），有时也用于副词；而假名则用于其他词类、语法形式（包括形容词和动词的词尾）、许多外来词，以及没有对应汉字或对目标读者而言使用汉字不太合适的词语。

日本文部科学省目前发布的标准清单规定了 2136 个日常使用汉字（常用汉字）。[1] 该清单旨在为

[†]舒 hmacher@math.uni-goettingen.de

[1]参见 https://www.bunka.go.jp/kokugo_nihongo/sisaku/joho/joho/kijun/naikaku/汉字 /

日常生活中的书面日语（包括报纸和广播）提供指引。它是官方文件规定使用的字符集，也是日本教育体系中初中及以下阶段汉字教学的依据。司法部发布的另一份清单则规定，另有 863 个汉字可用于日本户籍中的姓名（人名用汉字）。[2]

实际日常使用的汉字集合与上述列表有很大差异。一方面，有一部分汉字较少遇到（其中包括一些列入 J¯oy¯o 表的汉字，因为它们用于《日本国宪法》）。另一方面，常用汉字的数量更多，其中包括科学、技术、艺术及其他专业领域的词语，这些汉字未列入 J¯oy¯o 表，因此许多日本人认识的汉字数量超过 3000 个。日本官方汉字能力检定（Nihon Kanji N¯oryoku Kentei，简称 Kanken）的最高两个级别，分别要求准确掌握约 3000 和 6000 个汉字。综合性汉字词典的收录量通常从 5000 条到远超 10000 条不等。免费的在线 KANJIDIC 项目[3] 由 Jim Breen 发起，并由电子词典研究与开发小组（EDRDG）维护，提供有关 13,108 个汉字的信息；这些汉字收录于三项主要日本工业标准的文本处理字符集中（这些字符集是 Unicode 中所有 CJK 统一表意文字的子集）。规模最大的汉字词典《大汉和辞典》（DaiKanWa Jiten）最初由诸桥辙次编纂，于 1955 年至 1966 年间分 13 卷首次出版，收录约 50,000 个汉字（其 2018 年电子版收录 51,110 个）。

对于日文的外国学习者[4] 而言，要大致理解中级水平文本所需掌握的字形数量（估计约1000个）可能显得令人气馁。然而，间隔重复系统（SRS）的成功案例，结合技术进步以及互联网服务的广泛普及，如今已使这一过程对雄心勃勃的日文学生来说变得相当轻松。从主要汉字学习网站Wanikani[5] 抓取的用户统计数据显示，共有10,143名用户至少达到30级（大约掌握1000个汉字），而2167名用户处于60级（在此级别他们会遇到Wanikani提供的2074个汉字中的最后一个）。[6]

间隔重复系统的全部潜力尚未释放。例如，许多系统目前使用仅根据项目级别调整的固定时间重复间隔。人工智能的最新进展允许根据用户能力、所学项目的复杂性以及项目回忆的详细历史来优化重复时间，从而使学习过程更加高效愉悦。

间隔重复系统的一个缺点是其固有的顺序性。新项目按照特定顺序呈现，这使得相邻汉字之间的相似性和相互关系显而易见（或至少很容易指出）。但是，随着系统中的汉字数量增加，要将新呈现的汉字稳妥地置于已知汉字的整体脉络中，也就是（有意识或无意识地）追踪相似性，会变得越来越困难。

---

[2]参见 https://www.moj.go.jp/MINJI/minji86.html[3] http://www.edrdg.org/wiki/index.php/KANJIDIC_项目[4]本文中，我们将“外语学习者”一词用于指称掌握了与汉字无关的一种书写系统，且可能还掌握了其他书写系统的功能性知识的人。这一说法隐含了一个大致的最低年龄界限，但我们并无意对年龄或族裔施加任何其他限制。为求简洁，文中使用女性代词，但这并不意味着特指任何性别。[5]https://www.wanikani.com/[6]数据由 Wanikani 用户 saraqael 于 2022年 收集，参见 https://community.wanikani.com/t/ new-combined-user-statistics/58741。

以及与此前汉字的差异，以便能够区分它们。

举一个具体例子，来看 Wanikani 中以下四个汉字：粋（第 34 级）、酔（第 35 级）、枠（第 39 级）和 砕（第 47 级）。把它们并排展示时，很容易注意到它们的相似性（右侧部分由表示 9 的「九」和表示 10 的「十」构成）。学习者可以迅速复习这些字，并分辨它们不同的含义和读音，这种效果很可能会持久。然而，当学习者在间隔重复系统中遇到第三个，尤其是第四个汉字时（通常分别是在初次见到前两个字约两个月和四个月后），他们不太可能还记得先前发现的所有联系；因此，在间隔重复系统的后续阶段或日常阅读中再次遇到这四个汉字中的任何一个时，都很容易发生混淆。还要注意，这些汉字左侧部分所表示的具体事物（米代表稻米，酉代表酒，木代表树木或木材，石代表石头）可能比抽象的数字更容易留下深刻印象。但偶尔，学习者看到其中一个汉字时，脑海中还是会浮现 9 和 10；于是，这四个汉字在含义以及（大多截然不同的）读音上的界限都会变得模糊。

在这种情况及其他情况下，如果学习者能够快速查找相似的汉字，了解哪些其他字符及其隐藏关联会妨碍她回忆读音和含义，将会很有帮助。时不时地搜索与任意已学汉字相似的汉字，也是个好主意。这两种策略都能为已知汉字这座摇摇欲坠的建筑增添支柱和横梁，使它更快、更高地向天空耸立。还有一个令人欣喜的副作用：学习者也会因此接触到新的汉字，只需将其不同部分与相似的已学汉字进行对比，就能几乎不费脑力地把它们添入这座建筑中。

本文提出一个数学框架，根据汉字之间的差异度来定义它们之间的距离。最后这个术语有意保持宽泛和不具体。预测某人会在多大程度上认为两个对象相似或不同，是一个复杂的问题，认知心理学、神经科学和计算机科学中的模式识别等多个领域都对此有所探讨。这种判断可能取决于多种粗略和细致的个体因素，包括对这些对象的既有了解、总体教育程度、社会经济背景、性别和族裔。即使对同一个人而言，这种判断也可能因其心理和情绪状态、过往联想、当前情境以及呈现方式而有很大差异。[7] 充分细致地探讨这些方面超出了本文作者的能力，本文也不会尝试这样做。相反，本文旨在提供一套抽象的构件，用于构建汉字距离映射；讨论可与之配合使用的各种模块；最后给出一个具体但多少有些临时性的距离映射，作为概念验证。

拥有合适的汉字距离，其用途远不止帮助外国学习者为自己的汉字大厦搭建桁架。面向非日本人的教材和汉字词典作者，也能借此获得编排和组织所呈现材料的支持；参见 Kandr´aˇc (2021)。文本处理和光学字符识别系统则可根据其他汉字在较近距离内是否符合当前语境，评估潜在错误发生的可能性；参见 Nagata (1998) 和阪本 (Sakamoto) et al. (2019)。

我们需要澄清一些术语。给定一个（通常很大的）汉字集合 $\mathcal{X}$，如果任意对称函数 d: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ 满足以下识别条件，我们就称其为距离映射

[7] 汉字学习者都知道，字体类型稍有变化，就可能让人能否认出汉字截然不同。

性质：对于任意 $\mathfrak{K}^{(1)}$, $\mathfrak{K}^{(2)} \in \mathcal{X}$

$$d(\mathfrak{K}^{(1)}, \mathfrak{K}^{(2)}) = 0 \quad \text{if and only if} \quad \mathfrak{K}^{(1)} = \mathfrak{K}^{(2)}. \tag{1.1}$$

如果三角不等式也成立，即

$$d(\mathfrak{K}^{(1)}, \mathfrak{K}^{(2)}) \leq d(\mathfrak{K}^{(1)}, \mathfrak{K}^{(3)}) + d(\mathfrak{K}^{(3)}, \mathfrak{K}^{(2)}) \quad \text{for all } \mathfrak{K}^{(1)}, \mathfrak{K}^{(2)}, \mathfrak{K}^{(3)} \in \mathcal{X}, \tag{1.2}$$

则映射 d 称为**度量**。在这两种情况下，我们都将数值 $d(\mathfrak{K}^{(1)}, \mathfrak{K}^{(2)})$ 简称为距离。本文中的**相似性**一词大多是宽泛意义上的用法。不过，由于我们的距离映射通常以 1 为上界，读者完全可以将 $1 - d(\mathfrak{K}^{(1)}, \mathfrak{K}^{(2)})$ 视为 $\mathfrak{K}^{(1)}$ 与 $\mathfrak{K}^{(2)}$ 之间的相似性，并将 $1 - d$ 视为相似性映射。

1.1 汉字距离或相似性的其他来源

各种汉字学习工具都充分认识到，查找和参考相似汉字的必要性。我们再来看看 Wanikani，它会为许多汉字列出相似汉字。从用户的角度来看，这些相似性通常非常准确，因为它们大概是由人工挑选的。不过，缺点是往往会明显遗漏一些汉字，而且倾向于只收录涉及某些部件的相似汉字。继续以上文讨论的四个汉字为例，截至 2023 年 2 月 23 日，Wanikani 列出的相似汉字如下：

精粹: 框架 醉酒: 分
配 酸 醋 烈酒 斟酒
框架: 精粹 粉碎: —

在较低层级中，相似的汉字似乎更为完整，因为列表正随着时间推移逐渐增长。

据作者所知，唯一提供相似汉字搜索功能的在线汉字词典名为 Niai[8]。虽然源代码和实际评分可在 GitHub[9], 上获取，但具体评分方法并未公开。从 GitHub 上的致谢部分来看，该方法很可能基于使用 KRADFILE/RADKFILE[10] 中公开分解数据进行部件匹配；该数据由 Jim Breen 发起，并由 EDRDG 维护。因此，这种方法可能只是下文所述构建方法的一种特例，只不过目标是相似性而非距离。其结果似乎相当可靠，且完全自动化。不过，该方法似乎不容易识别形状相似但并不完全匹配的部件，而且较好的匹配结果中往往混杂着几项不明确的相似项。对于上面的四个汉字，截至 2023 年 2 月 23 日，我们得到以下结果（结果仅限于 J⁻oy⁻o 汉字）：

精粹: 醉酒 框架 粉碎 降 真
醉酒: 精粹 框架 粉碎 傘 烧
框架: 枯萎 染 精粹 醉酒 植
粉碎: 概括 活 砂 坚硬 碑

至少在英文文献中，似乎鲜有研究从方法论角度探讨汉字距离或相似性。

[8] https://niai.mrahhal.net/similar [9] https://github.com/mrahhal [10] http://www.edrdg.org/krad/kradinf.html

粋	断 0.1818	枠 0.2000	料 0.2000	粉 0.3000	析 0.3000	新 0.3077	数 0.3846	砕 0.4000	卒 0.4000	科 0.4000
酔	酢 0.2500	酌 0.2727	配 0.2727	酷 0.2857	酬 0.3077	酪 0.3077	卑 0.3636	枠 0.3636	車 0.3636	昇 0.3636
枠	析 0.1250	粋 0.2000	株 0.3000	柳 0.3333	砕 0.3333	断 0.3636	酔 0.3636	杉 0.3750	卒 0.3750	松 0.3750
砕	砲 0.3000	研 0.3333	枠 0.3333	許 0.3636	群 0.3846	粋 0.4000	破 0.4000	訴 0.4167	碑 0.4286	故 0.4444

表1：与这四个汉字最接近的十个汉字及其笔画编辑距离 s.

一个值得注意的例外是 Yencken and Baldwin（2006, 2008）。在前一篇论文中，研究者开展并分析了相似性实验，共有来自20种语言背景的179名参与者，包括中文和日文母语者、中文和日文二语学习者，以及此前没有汉字使用经验的参与者（非CJK组）。研究者提出了相对简单的相似性计算方法：一种使用渲染图像的 ℓ_1-距离，另一种则是基于 KRADFILE/RADKFILE 的部首词袋余弦相似性。除非CJK组参与者外，部首词袋方法与实验中评分者给出的平均或中位数相似性判断总体上吻合良好（以秩相关衡量）。

在 Yencken and Baldwin（2008）中，作者介绍了更复杂的方法。其中包括一种“部首袋加字形”距离映射：如果汉字的 SKIP编码（Halpern，2022）中的首位数字不一致，就将他们此前的余弦相似性设为 1。SKIP编码用于对四种基本汉字结构进行分类。这种距离映射在与人工相似性判断的总体秩相关方面表现最佳。作者还利用 Apel and Quint（2004年）的层级数据，将汉字分解为笔画序列和笔画树；如今，这些数据由 kanjiVG 网站维护。[11]在相关性实验中，所得的笔画编辑度量和树编辑度量表现并不突出，但在后续的两项实验中，它们识别高度相似汉字对的效果远胜于其他度量。

其中一位作者的网站上提供了一个文件，包含基于笔画编辑度量得出的成对相似性数据。[12] 提取上述四个汉字各自最相似的十个汉字，并将相似性转换回笔画编辑距离，即得表1。

1.2 贡献

我们提出了一个用于为汉字之间赋予距离的一般框架（第 2 节）。该框架作出了一个相当简约的假设：相异性可以通过以下方式得到充分表达——以最优方式比较由笔画集合构成的嵌套结构，并对比较所得的惩罚值进行加权平均；我们将这些笔画集合称为部件。

惩罚值根据各种新旧构建要素计算得出，例如比较部件的标签、相对位置、尺寸或畸变，以及将一个部件转换为另一个部件时，墨迹需要 transport 多远、需要添加或擦除多少墨迹。这些构建要素可以根据需要进行添加和替换

[11]http://kanjivg.tagaini.net[12]
https://lars.yencken.org/datasets/kanji-confusion

粹	枠 0.0596	砕 0.1210	粒 0.1315	粉 0.1323	料 0.1326	粘 0.1345	枯 0.1397	粧 0.1397	粗 0.1431	辞 0.1437
酔	酢 0.0594	酎 0.1028	酌 0.1056	配 0.1083	酷 0.1187	鮮 0.1203	酬 0.1234	酪 0.1235	酸 0.1268	酵 0.1333
枠	粋 0.0596	枯 0.0875	枝 0.1067	砕 0.1109	染 0.1137	乗 0.1312	祥 0.1328	析 0.1400	朽 0.1400	杉 0.1407
砕	枠 0.1109	粋 0.1210	辞 0.1238	砂 0.1483	酔 0.1544	砲 0.1549	研 0.1551	計 0.1552	枯 0.1572	群 0.1583

表 2：四个汉字最接近的十个汉字及其汉字距离。

具体取决于我们的需求，例如所需的精确度和复杂程度，或可用的计算时间。

我们在第 3.2 节中提出了一系列有用的构建模块，并在第4节中做出初步参数选择，以获得作为概念验证的具体距离映射。作为预览，我们在表 2 中给出了上述四个汉字的十个最近邻及相应距离（尺度从 0 到 0.25）。与表 1 相比，该表中相似性不那么明显的汉字更少，尤其是当我们查看表格右半部分（最近邻编号 6–10）时。

然而，本文的重点并不在于具体的距离映射，该映射仍有相当大的优化空间，并且理想情况下应针对特定任务进行调整。相反，重点在于将整体构造呈现为部件的多层次匹配，讨论比较部件的重要目标可能是什么，并展示可用于此比较的若干模块。

我们的方法根植于一种现代结构方法，可称为层次最优传输。在宏观层面，我们通过匹配部件来求解最优传输问题（由于部件结构具有层次性，这一过程本身也具有层次性）；而在确定部件之间的代价时，我们也使用最优传输来计算部件渲染图像之间的（相对非平衡）Wasserstein 距离。该度量解决了 Yencken and Baldwin（2006）中提到的（相对） ℓ_1 度量的根本问题。它并非对所有位置的像素差绝对值取平均，而是计算将一幅图像转换为另一幅图像时墨迹必须移动的平均距离。作为更详细说明的预览，读者可参阅图 3 右侧面板。独立于左侧面板来看，这可以视为汉字日（日、太阳）和臼（石臼）的像素图像差异图。尽管这两个汉字已经缩放并对齐，但线条几乎没有任何位置能精确重合（右上方和左下方的少许部分除外；在那里，白色的正线与黑色的负线相互抵消）。因此， ℓ_1 距离相对较大，而墨迹输运的平均距离（箭头所示）则相当小。

当然，计算 Wasserstein 距离的成本远高于计算 ℓ_1-距离。不过，如今在普通笔记本电脑上，用不到 0.1 秒就能完成图 3 中的 transport 计算，完全可行（例如使用 R 软件包 transport 中的函数 `unbalanced`，参见 Schuhmacher 等人，2023）。事实证明，使用较低的分辨率（例如图 3 中的 32×32 而非 64×64）就足够了。此外，如果时间确实非常紧迫，我们可以采用更快的近似计算方法；参见 Chizat 等人（2018）。

作为一个 ast贡献，我们提到新的R包kanjistat（Schuhmacher，2023 ），

它提供了一系列用于处理汉字的计算和统计工具。除此之外，它可以导入和转换本文中提到的几个数据集，并计算所描述的各种部件距离以及整体汉字距离。该软件包的部分内容仍在开发中，整个软件包将在未来不断改进。

2 嵌套部件结构之间的最优传输

本节重点介绍汉字距离映射的抽象定义，将较有争议的设计和参数选择留待后续章节讨论。

从全局来看，我们认为汉字由（有时会相互重叠的）部件构成，而部件由笔画组成。笔画 是乘积空间中的一个元素，其中包含我们希望在汉字距离中表示的所有所需信息。若选择 $U = [0,1]^2$ 作为绘制汉字的画布，通常至少包含一条路径 $f\colon\ [a,b] \to U$，或一个路径连通子集 $F \subset U$；两者都表示某种预先指定字体类型中的笔画。其他信息还可以包括笔画编号 s、笔画类型分类 τ，甚至更多信息，从而可以用例如一个元组 $(f,\ s,\ \tau)$ 来表示该笔画。部件 是有限个笔画构成的集合。[13]

随后，我们将一个汉字建模为一个有限序列 $\mathfrak{K} = (\mathfrak{K}_l)_{0\le l\le L+1}$，其中部件的分解粒度逐步细化 $\mathfrak{K}_l = \{\mathfrak{C}_{l,i} \,|\, 1 \le i \le m_l\}$，从第 $l = 0$ 层所有笔画都属于同一个部件开始，到第 $l = L+1$ 层每一笔画都属于一个独立部件结束。记作 $[m] := \{1,\dots,m\}$，更精确地说，我们要求：

(1) 对每个 $l \in [L+1]$ 和每个 $i \in [m_l]$，都存在一个 j 属于 $[m_{l-1}]$，使得 $\mathfrak{C}_{l,\,i} \subset \mathfrak{C}_{l-1,\,j}$（各层之间的嵌套性质）。(2) 对每个 $l \in [L+1]$ 和 $i,\ i' \in [m_l]$，若满足 $i \neq i'$，则有 $\mathfrak{C}_{l,\,i} \setminus \mathfrak{C}_{l,\,i'} \neq \emptyset$（同一层上的部件彼此不嵌套）。(3) $\mathfrak{K}_0 = \{\mathfrak{C}_{0,1}\} = \big\{\{S_1,\dots,S_n\}\big\}$。(4) $\mathfrak{K}_{L+1} = \{\mathfrak{C}_{L+1,1},\dots\mathfrak{C}_{L+1,n}\} = \big\{\{S_1\},\dots,\{S_n\}\big\}$。

性质 (1) 和 (4) 蕴含覆盖性质，即 $\bigcup_{i=1}^{m_l} \mathfrak{C}_{l,\,i} = \{S_1,\dots,S_n\}$ 对于每个 l 都成立。虽然这些并集通常互不相交，但我们在同一层级上的唯一要求是 (2)。我们将 $\mathfrak{C}_{0,\,1}$ 称为根，并将任意 $\mathfrak{C}_{L+1,\,i}$ 称为叶节点，尽管包含关系不一定会形成树结构。

索引 L 表示我们研究汉字距离时关注的最后一个层级。我们通常不关注将其分解为第 $L+1$ 层的各个叶节点，因为这一层并未包含比第 L 层更多的信息（甚至不比第 0 层更多）。不过，如果我们仍想纳入这些叶节点，始终可以将这一分解的副本作为倒数第二层添加。此外，我们可以省略任意第 $l \in [L]$ 层，以得到更简单的分解层级结构。这两种做法都不会破坏上述性质。

我们将剩余索引的集合 $K = \{(l,\ i) \,|\, 1 \le l \le L,\ 1 \le i \le m_l\}$ 称为汉字的本质索引集。脉络是指任意集合 $V = \{(l,\ i_l) \mid 0 \le l \le L\} \subset K$

[13]为简化记法，我们不在部件的定义中明确添加更多信息，例如标签或该部件在汉字中的相对位置。从形式上说，这些信息可以纳入笔画层级。

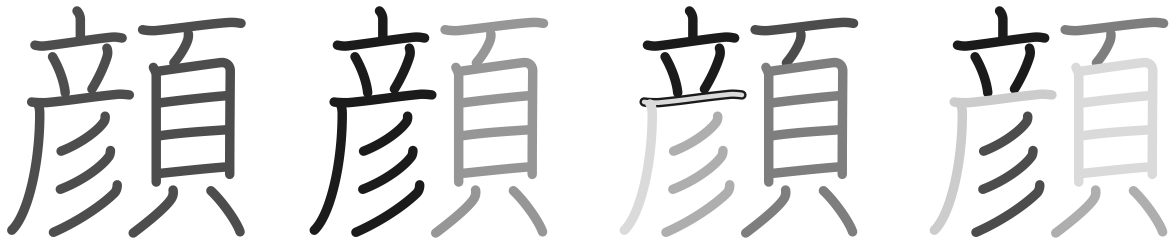


图 1：汉字 颜在层级 0、1、2、3 的分解分别得到 1、2、5、6 个部件。每个层级中，各部件的灰度随机重新分配。注意层级 2 中部件 立和 厂的重叠。图中未展示层级 $4 = L + 1$；该层级仅将汉字分解为 18 个独立笔画。

其部件构成了从根到叶子前一层的极大包含链，即$\mathfrak{C}_{0,\ 1} \supset \mathfrak{C}_{1,\ i_1} \supset \ldots \supset \mathfrak{C}_{L,\ i_L}$。

图 1 展示了基于 kanjiVG 文件的汉字 颜的分解层级。与本文中的所有示例一样，可以使用 R 包 kanjistat 中的标准命令重现该图。该图说明分解后的部件不一定互不重叠：在层级 2 中，立的底部笔画与 厂的顶部笔画共用；到层级 3 时，立的上部单独构成一个部件，这一重叠随之消解。假设我们按从上到下、再从左到右的顺序为部件编号，则脉络的示例如下：集合 {(0，1)，(1，2)，(2，5)，(3，5)} 对应部件颜、页、贝、目；集合{(0，1)，(1，1)，(2，2)，(3，2)} 对应部件颜、彦、厂、厂。

我们提出的汉字距离映射通过在所有层级尽可能匹配两个汉字的部件，同时避免产生嵌套来定义。距离则由各部件距离的加权和给出，并根据汉字中未覆盖部分的权重加入惩罚项。

我们首先引入一种适当的非负权重结构 $w = (w_{l,\ i})_{(l,i)\in K}$，满足 $\sum_{i=1}^{m_l} w_{l,\ i} = 1$，适用于任意 l，并描述各层级内不同部件的相对重要性。通常，$w_{l,\ i}$ 是部件$\mathfrak{C}_{l,\ i}$某种尺寸属性的递增函数，例如绘制该部件所需墨迹的相对量。由于本文使用的是 kanjiVG 中恒定宽度的线条信息，因此部件 $\mathfrak{C}_{l,\ i}$ 的墨迹量可通过对笔画长度 $|S|$ 求和来充分表示，其中 $S \in \mathfrak{C}_{l,\ i}$。

我们还为每个高于 $l = 1$ 的层级引入一个因子（$1 - \varepsilon$），表示沿部件层级向下时权重逐渐“流失”的过程。这会在一定程度上促使匹配较大的部件，而非其各个独立部分。尤其是，它会对未按适当方式排列、因而无法整体匹配的子部件匹配施加惩罚项。因此，我们的权重为

$$w_{l,i} = \frac{\sum_{S\in\mathfrak{C}_{l,i}} |S|}{\sum_{S\in\mathfrak{C}_{0,1}} |S|} (1-\varepsilon)^{\max\{0,\, l-1\}}. \tag{2.1}$$

接下来，需要一个对称函数 $\mu\colon [0,1]\times[0,1] \to [0,1]$，用于确定将具有给定权重 w 和 w'的两个部件匹配后得到的单一贡献权重 $\mu(w,\ w')$。我们要求，对于任意有限权重向量 (v_i)、$(v_{i'})$，当 $\sum_{i=1}^{m} v_i = \sum_{i'=1}^{n} v'_i = 1$时，$\mu(w,\ w') \geq \min\{w, w'\}$ 且 $\sum_{i=1}^{m}\sum_{i'=1}^{m'} \mu(v_i,\ v'_{i'}) \leq 1$，这样便可将 $\mu(w,\ w') = \frac{w+w'}{2}$ 作为最大选择。几何平均数或调和平均数是不错的

折中方案，因为相比最小值，它们会给个体权重不同的匹配项赋予更高权重；但只要其中一个权重趋于零，它们本身也会趋于零（算术平均数则不同）。

最后，我们将 ϱ 记作一个取值于 $[0,1]$、定义在部件之间的距离映射。适当的构造将在第 3 节讨论。除其他内容外，我们的 $\varrho(\mathfrak{C},\ \mathfrak{C}')$ 通常会结合以下两者：标准化字形图 $\mathfrak{C}$，$\mathfrak{C}'$之间的最小笔画搬运成本，以及为使这些字形图尽可能相似所需的平移、缩放和畸变惩罚值。

现在可以如下定义汉字距离映射 d。设 $\mathfrak{K}^{(1)}$，$\mathfrak{K}^{(2)}$ 为两个汉字，其嵌套分解分别为 $\big(\{\mathfrak{C}^{(1)}_{l,i} \mid 1 \le i \le m_l\}\big)_{0\le l\le L+1}$ 和 $\big(\{\mathfrak{C}^{(2)}_{l',i'} \mid 1 \le i' \le m'_{l'}\}\big)_{0\le l'\le L'+1}$。用 K_1 和 K_2 表示相应的本质索引集，用 $\mathcal{V}_1$ 和 $\mathcal{V}_2$ 表示 $\mathfrak{K}^{(1)}$ 和 $\mathfrak{K}^{(2)}$的脉络集。此外，设 $w^{(1)} = (w^{(1)}_{l\,i}\,,\,)_{(l,i)\in K_1}$ 和 $w^{(2)} = (w^{(2)}_{l'\,i'}\,,\,)_{(l',i')\in K_2}$ 为根据（2.1）定义的两个权重结构。令 $a \in (0,1]$，并定义

$$
\begin{aligned}
d(\mathfrak{K}^{(1)},\mathfrak{K}^{(2)}) = \min_{(e_{(l,i),\,(l',i')})} \Bigg(& \sum_{(l,i)\in K_1} \sum_{(l',i')\in K_2} e_{(l,i),\,(l',i')}\, \mu(w^{(1)}_{l,i}, w^{(2)}_{l',i'})\, \varrho\big(\mathfrak{C}^{(1)}_{l,i}, \mathfrak{C}^{(2)}_{l',i'}\big) \\
& + a\Bigg(1 - \sum_{(l,i)\in K_1} \sum_{(l',i')\in K_2} e_{(l,i),\,(l',i')}\, \mu(w^{(1)}_{l,i}, w^{(2)}_{l',i'})\Bigg)\Bigg) \in [0,a],
\end{aligned}
\tag{2.2}
$$

其中最小值是在满足条件的 $(e_{(l,i),\,(l',i')}) \in \{0,1\}^{K_1\times K_2}$ 上取得的

$$
\begin{aligned}
\sum_{(l,i)\in V_1} \sum_{(l',i')\in K_2} e_{(l,i),(l',i')} &\le 1 \quad \text{for all } V_1 \in \mathcal{V}_1, \\
\sum_{(l,i)\in K_1} \sum_{(l',i')\in V_2} e_{(l,i),(l',i')} &\le 1 \quad \text{for all } V_2 \in \mathcal{V}_2.
\end{aligned}
$$

因此，$e_{(l,i),(l',i')} = 1$ 表示部件 $\mathfrak{C}^{(1)}_{l\,i}$ 和 $\mathfrak{C}^{(2)}_{l'\,,i'}$ 相匹配，并产生加和贡献 $\varrho\big(\mathfrak{C}^{(1)}_{l\,i}\,,\,,\mathfrak{C}^{(2)}_{l'\,,i'}\big)$，其权重为 $\mu(w^{(1)}_{l\,i}\,,\,,w^{(2)}_{l'\,i'},)$。考虑所有匹配后，若总贡献的权重不足 1，则差额计为 a。侧约束意味着每条脉络至多只能匹配一个部件（且仅匹配一次）。特别是，如果 $e_{(1,0),(l',i')} = 1$ 对于某个 $(l',i') \in K_2$成立，即顶层的整个第一个汉字已匹配（例如，徵和惩可能适用这种情况），那么对于所有 $(l,i) \in K_1 \setminus \{(1,0)\}$ 和所有 $(l'',i'') \in K_2$，都必须有 $e_{(l,i),(l'',i'')} = 0$ 成立；这意味着我们不能将徵的任何子部件与惩的心部件匹配（也不能与其他任何部件匹配）。

图2展示了汉字 颜和须的一个稍复杂示例。对于第4节中使用的更具体的距离，部件页和彡按图中所示进行匹配，前者的代价很小，后者的代价则稍高。

优化问题 (2.2) 是一个带有线性侧约束的二元线性规划，可以用标准软件求解。由于这是一个规模较小的问题，变量通常少于 100–200 个，因此不会显著增加计算时间。

从实践角度来看，或许可以根据某种标准，进一步精简本质索引集（进而精简脉络），即省略某些包含链末端的若干索引。例如，可以忽略笔画少于三画的部件。这需要对上述定义稍作调整。我们可以采用更一般的本质索引集，也可以将该标准纳入距离映射 ϱ，例如，如果要省略 $\mathfrak{C},\mathfrak{C}'$ 中的某一个，则令 $\varrho(\mathfrak{C},\mathfrak{C}') = a$（或更大）。

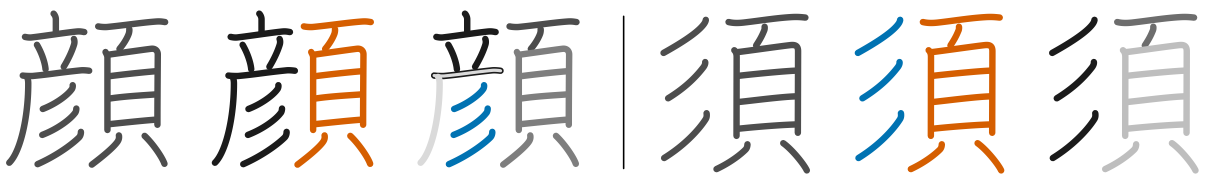


图 2：汉字 颜和须在层级 0、1、2 下的分解及其部件匹配。橙色和蓝色表示部件页和彡相匹配，即 $e_{(1,2),\ (2,2)} = 1$ 和 $e_{(2,3),\ (2,1)} = 1$。左侧汉字仍可为立和厂（或其子部件）建立匹配，但右侧汉字中已没有可供匹配的内容。

注 1。 若无更具体的假设， d 并非度量。它始终是距离映射，即具有对称性和同一性，因为 ϱ 具有这些性质。[14] 然而，三角不等式还需要更多条件。我们的目标是量化（不）相似性，而人们普遍认为，人类的相似性判断和识别任务通常具有非度量性质，参见例如 Tversky（1977）、Laub 等人（2006）、Scheirer 等人（2014）或 Nachshon 等人（2022），因此三角不等式未必是我们所追求的。另一方面，在展示汉字空间景观（见第4节）或进一步对汉字空间进行数学研究时，能够使用一个度量或许会很有意思。假设 ϱ 是度量，且 μ 取最小值，我们猜想 (2.2) 定义了一个度量。如果我们固定 $l = L$ 和$l' = L$，不在层级之间进行匹配，则可以证明 所得结果是一个度量。如果我们允许根据拆分权重对部件进行多重匹配，而不是按照最小权重进行一对一匹配，结论依然成立。这样构造出的整体方法与 Yurochkin 等人（2019）的方法相似。

3 部件比较

上一节通过将问题分解为针对ϱ的多层级部件比较，为测量汉字之间的距离提供了整体策略，但仍需构建合适的函数 ϱ。在很大程度上，这些部件并不比实际汉字更简单。不过，我们现在可以完全专注于部件的全局特征——整体外观、形状、面积和位置——并相信第 2 节中针对 d 的多层级构造能够处理基于其子结构的任何相似性或距离。

3.1 一般考虑

正如引言中所述，作者无法给出一个明确的答案，即单个人或更大群体的人认为两个汉字或两个组成部分有多么不同，以及其他方面，因为缺乏良好的数据基础和任何自己的

[14]请注意，在识别性质的语境中，部件或汉字是否相等，取决于我们在对汉字建模时纳入了哪些信息（严格来说，是笔画层级的信息）。如果不够谨慎，这种相等关系可能弱于语言学意义上的两个汉字相等。例如，如果只纳入笔画类型而不纳入笔画位置，那么就会有部 = 陪、招 =拐和呆 =杏。前两组可以通过为部件添加标签来区分，但最后一组需要位置的信息。本文后续部分将始终采用足以从语言学意义上识别部件（及汉字）的信息。

心理语言学专业知识。尽管如此，我还是想通过若干方面和例子，让读者意识到这个问题是多么多方面的，并理清我们下面定义的函数 ϱ defined below 考虑了哪些方面、没有考虑哪些方面（但也许应该考虑，而且肯定可以考虑）。

首先，考虑四个部件：牛（牛）、午（中午）、干（干燥）和千（千）。读者或许会认同，这一序列中任意两个相邻元素看起来至少都颇为相似。我们暂且不考虑结构视角（它们都至少有一条竖笔和两条横笔），因为这属于上一节讨论的内容；下文将通过汉字之间墨水输运的最小总距离（相对的非平衡距离），加上必须增添或擦除的墨迹量，来描述整体相似性。对于这些例子，这种方法似乎相当有效。尤其是，与使用 kanjiVG 数据或 KRADFILE 中提供的部件标签相比，这种方法灵活得多。依据这两种数据，我们只能判断两个部件是否完全相同。即便这一判断也有些问题，因为有些部件根本没有标签，或者至少标签并不完全准确（因为相应部件在 Unicode 中没有精确的对应项）。但更大的局限在于，这种方法无法识别彼此非常相近的部件，例如传统部首 阝（Unicode 码位 2ECF，在汉字左侧，是阜的变体）和 阝（Unicode 码位 2ED6，在汉字右侧，是邑的变体）；又如上文序列牛、午、干、千中任意两个相邻元素。

使用最优墨迹传输会引出一个问题：应该使用哪种字体。虽然在某些日文字体（哥特体）中，干和千的顶笔看起来可能非常相似（笔画粗细均匀且没有装饰），但在明朝体中，两者看起来更不相同（笔画粗细不均且带有装饰），例如本文所用的字体；而在一些描绘真实笔触的字体中，两者可能看起来差异很大。究竟是使用单一的“代表性”字体更好，还是在多种字体下对总墨水输运距离取平均值，目前并不明确，这可能取决于汉字距离的具体目标群体和目标。为求简便，我们使用 kanjiVG 数据提供的简单笔画路径信息。由于这些信息源自教科书字体，就纯粹的路径信息而言，它颇具代表性。不过，它显然无法代表各种明朝体和书法字体中的可变笔画宽度信息。

另一个需要考虑的重要问题是，是否应将学习汉字时习得的其他知识也纳入考量。即使在哥特体中，部件干和千的顶笔看起来相当相似，正在学习手写汉字的人（以及许多中高级学习者）也能轻易区分它们，因为他们知道这实际上是两种截然不同的笔画：干中的从左向右的横画横画 ，以及千中的从右上向左下的左払い左払い 。

可以说，更关键的知识涉及部件的含义和读音。例如，考虑去和先，无论是作为完整汉字部件，还是作为法和洗的一部分。这些部件在纯粹的现象学层面上可能已经看起来有些相似，但通过知道它们分别代表密切相关的时间概念过去 和之前，感知到的相似性（以及混淆它们的风险）会显著增加。

所有这些附加项（字体、变化的笔画粗细、笔画类型、读音和含义）都不 包含在我们下面的函数 ϱ 中，尽管它们可以包含，会遇到一些个别但总体上相当类似的问题，就像我们下面面临的那些问题。

除最优墨迹传输之外，我们还纳入了位置、尺寸和形状差异的简单描述符。请看以下十个汉字，它们都包含

一个山部件：崔、密、嵐、岬、岳、崎、惩、仙、岸、両。你会如何根据相似性对它们分组？如果你识别出了所有汉字中的山部件（根据你以往的经验，这可能需要一些时间），你大概会得出三个基本分组：$G_1 = \{$岬，崎$\}$、$G_2 = \{$密，岳$\}$，以及$G_3 = \{$崔、嵐，岸$\}$。也许你不知道该如何处理剩下的汉字，于是把它们各自分成一组；或者，你把仙归入 G_1，把惩归入 G_3，也许还很宽泛地把両归入G_2。关键在于，建立这三个基本分组时，你主要依据的是山部件的位置，或许在较小程度上也依据其尺寸和长宽比；这些特征在每组内部大致完全一致。你可能也会像我一样发现，G_1 与另外两组的差异，比那两组彼此之间的差异更大，这是因为山部件的长宽比截然不同。不过，这种印象很大程度上取决于你在多大程度上把山部件看作图形，而不是仅仅把它看作表示山的部件。

我们注意到，懲中的山部件与 G_3中的山部件位置相似，长宽比也大致相同。尽管如此，我们仍觉得懲与 G_3中的汉字相当不同，这部分是因为山部件的尺寸不同，部分也是因为该部件在汉字中的暴露度 不同。这是我们目前在下文的 ϱ 中尚未纳入的另一个方面。在 G_3中，山部件非常突出；而在懲中，山部件则紧密地夹在另外三个部件之间。

除了包含许多显著部件的汉字外，由于单个（部分）包围部件的存在，部件暴露度不足的情况也很常见。比如，比较両、風、国和園中被包围的部件，以及岳/仙、蛍、玉和遠中相应（暴露度更高）的部件。我们计算出的距离由墨水输运以及针对位置、尺寸和形状差异的惩罚值构成；由于忽略了暴露度差异，这一距离可能远小于人们感知到的不相似程度。不过，没有数据很难判断这一点，而且这种现象可能在很大程度上因人而异。

比部件的暴露度更一般的概念，或许是它与相邻部件之间的相互作用，而这同样尚未纳入我们当前的距离映射。若所有部件完全匹配，且汉字的整体结构也相同，例如「部」与「陪」或「招」与「拐」，所产生的相似感可能会比合理的加性位置惩罚所能体现的更强。这也可能适用于整体结构不同的情况，例如「变」与「赦」。

3.2 基于最优传输的部件比较

梳理完目前尚未考虑的诸多方面后，我们现在定义剩余的量，即标准化部件之间（相对非平衡的）最小墨水输运量，以及针对平移、缩放和畸变的惩罚值。

对于部件 $\mathfrak{C}$ 和 $\mathfrak{C}'$之间的非平衡最小墨水输运，我们根据 kanjiVG 笔画信息绘制它们，并对齐其位置、尺寸和畸变。为此，我们考虑这些部件的边界框，先将其置于画布 $U = [0,1]^2$ 的中心，再重新缩放，使其 x轴和 y轴的最大值略低于 1。[15] 由此得到两幅（可能较粗糙的）像素图像 $C = (c_{rs})_{(r,s)\in G}$,

[15]为简便起见，我们目前使用边界框，但这方面仍有很大的改进空间。我们可以不采用严格的边界框，而是允许对离群值进行一定程度的裁剪，例如用墨迹分布沿 x轴和 y轴的第5和第95百分位数替代最小值和最大值。这样可以改善图3中的匹配效果，例如，与黑色部件顶部相比，它会去除白色部件顶部更多的像素。此外，还有更复杂的图像

$C' = (c'_{rs})_{(r,s)\in G}$，其中 $G = \{\frac{1}{2N}, \frac{3}{2N}, \cdots, \frac{2N-1}{2N}\}$ 是像素中心的 $N \times N$ 网格。记 $\delta_{(r,s),(r',s')}$ 为 (r,s)，$(r',s') \in G$ 之间的欧几里得距离。然后，我们计算部件之间的非平衡 Wasserstein 度量（参见 Heinemann 等人，2023；M¨uller 等人，2023 及其中的参考文献），其形式为

$$d_{\mathrm{UBW}}(\mathfrak{C},\mathfrak{C}') = \left(\min_{\Pi}\left(\sum_{r,s,r',s'} \delta^p_{(r,s),(r',s')}\,\pi_{(r,s),(r',s')} + \big(\|C\|_1 + \|C'\|_1 - 2\|\Pi\|_1\big)\frac{b^p}{2}\right)\right)^{1/p}$$
$$\in \big[0, b\max\big\{\|C\|_1, \|C'\|_1\big\}\big], \tag{3.1}$$

其中，最小值是在所有满足以下条件的 $\Pi = (\pi_{(r,s),(r',s')}) \in \mathbb{R}_+^{G\times G}$ 中取得的

$$\sum_{r',s'} \pi_{(r,s),(r',s')} \le c_{r,s} \quad \text{for all } (r,s) \in G,$$
$$\sum_{r,s} \pi_{(r,s),(r',s')} \le c'_{r',s'} \quad \text{for all } (r',s') \in G,$$

此外

$$\|C\|_1 = \sum_{r,s} c_{r,s},\ \|C'\|_1 = \sum_{r',s'} c'_{r',s'},\ \|\Pi\|_1 = \sum_{r,s,r',s'} \pi_{(r,s),(r',s')}.$$

换句话说，$d_{\mathrm{UBW}}(\mathfrak{C},\mathfrak{C}')$ 表示将 $\mathfrak{C}$ 转换为 $\mathfrak{C}'$ 时的代价，以最小总墨水输运量衡量，并允许以每单位 $b/2^{1/p}$ 的代价添加或删除墨迹。[16] 我们始终选择 $p = 1$ 和 $b = 0.4$（即 0.4 是我们假定墨迹仍有关联时的最大距离，对应于添加墨迹的代价为 0.2，删除墨迹的代价也为 0.2）。图 3 展示了潟和陽的右上部件示例。

一种看似可行的方法是直接计算原始部件之间的 $d_{\mathrm{UBW}}(\mathfrak{C},\mathfrak{C}')$，而不是先对它们进行归一化，再对平移、缩放和畸变添加惩罚值。然而，这样做会低估形状相似性的显著影响。例如，部和陪永远不可能具有（相对）较小的整体 d-距离，因为外观相似的部件之间的 d_{UBW}-距离会接近可能的最大值，这是因为它们仅相差约 $0.4 = b$。即使是位于相同位置的差异很大的部件之间的 d_{UBW}-距离，也小于这个值。

下文中我们始终考虑相对 d_{UBW}-距离，即除以 $b\max\{\|X\|_1, \|Y\|_1\}$，这使最大距离达到 1。虽然 d_{UBW} 始终是一个度量，但除以$b\max\{\|X\|_1, \|Y\|_1\}$ 会破坏三角不等式，除非我们只考虑总墨迹量处于某个范围内的汉字（Schuhmacher 和 Wirth，2023）。不过，相对 d_{UBW}-距离映射与我们感知差异的方式更加吻合：在已有许多笔画的部件上增加一笔，比在笔画较少的部件上增加一笔所造成的差异更小，例如比较史和吏，与丨和十。

接下来，我们介绍平移、缩放和畸变惩罚。与上文对部件进行的归一化保持一致，我们以边界框为基础定义这些惩罚值

配准方法，这些方法基于一组变换使部件尽可能相似。[16]在上述公式中将因子 $\frac{b^p}{2}$ 除以2是有争议的。我们这样做是为了确保因子2不会出现在距离的上界中。另一方面，如果我们省略除以2，我们会有更直观的每单位添加/删除墨迹成本 b，并且 d_{UBW}-距离对于实际部件仍然很少超过 $b\max\{\|C\|_1, \|C'\|_1\}$。

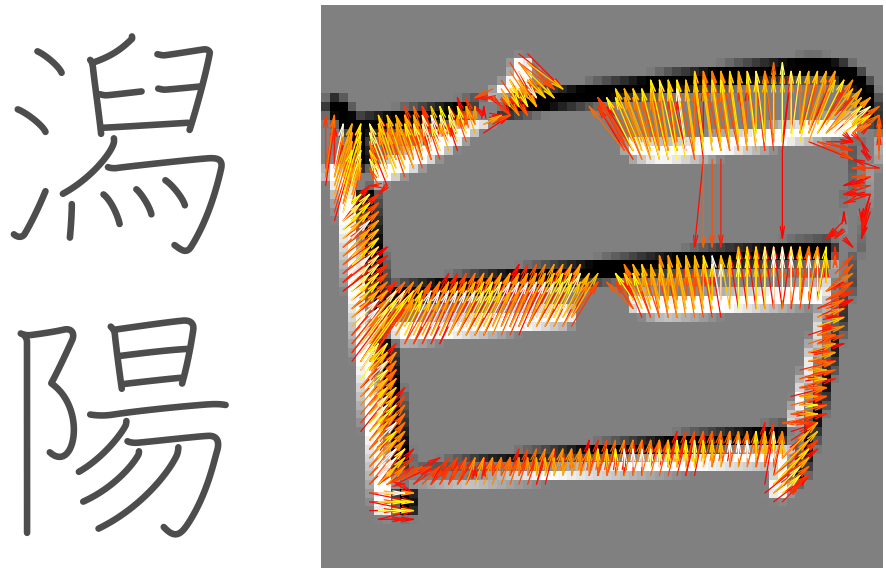


图 3：潟和陽右上部件之间的非平衡最优传输。左侧：根据 kanjiVG 中的笔画信息绘制的汉字。右侧：灰度图像显示将它们的 x-和 y-扩展的最大值缩放至 1 后，部件之间的差异。白色表示高度正值，黑色表示高度负值；灰色背景对应值为零。箭头表示热色（红色至橙色：较少墨迹；黄色至白色：较多墨迹）像素之间的最优传输方案。原地保留的质量未予标示，因为它对传输成本没有贡献。参数为 $p = 1$,$b = 0.4$。我们得到 ^{1} $_{\text{最大值}\{|\mathfrak{C}|,|\mathfrak{C}'|\}} d_{\mathrm{UBW}}(\mathfrak{C},\ \mathfrak{C}') = 0.061272$；平移、对数尺度和对数畸变惩罚值分别为 0.041694、0.363346 和 0.013505。

。部件的边界框。平移惩罚 $\tau(\mathfrak{C},\ \mathfrak{C}')$ 就是两个边界框中心之间的欧几里得距离。我们将边界框各边长度的几何平均数作为尺度，并将这些边长的比值作为畸变。随后，我们将尺度惩罚 $\sigma(\mathfrak{C},\ \mathfrak{C}')$ 定义为尺度比值的绝对对数值，将畸变惩罚 $\chi(\mathfrak{C},\ \mathfrak{C}')$ 定义为两个部件之间畸变比值的绝对对数值。

目前，我们对相对的 d_{UBW}-距离及各项惩罚项的递增函数进行线性组合。更复杂的选择将在第 3 节第 3.3 小节中讨论。使用递增函数似乎至关重要，因为我们的感知似乎完全不会随这些量线性变化：例如，如果部件之间的相对 d_{UBW}-距离非常小，我们很容易将它们感知为相似。如果距离稍大一些，这种相似性可能不会被有意识地察觉。不过，之后记忆变得模糊时，我们仍可能把这些部件混淆（例如图 3 中汉字右侧的部件，我们大概不会在第一眼看到时就把它们混淆）。一旦超过某个距离阈值，我们的大脑恐怕就不会再将这些部件联系起来。我们猜测，这三种惩罚值也有类似的表现，尽管它们与其他效应之间的相互作用可能更复杂。

然后，我们将递增函数 $\psi_{\mathrm{dist}}, \psi_{\mathrm{trans}}, \psi_{\mathrm{scale}}, \psi_{\mathrm{distort}} \colon [0,1] \to [0,1]$ 应用于相应的量，以压低较小的值并放大较大的值。为简洁起见，

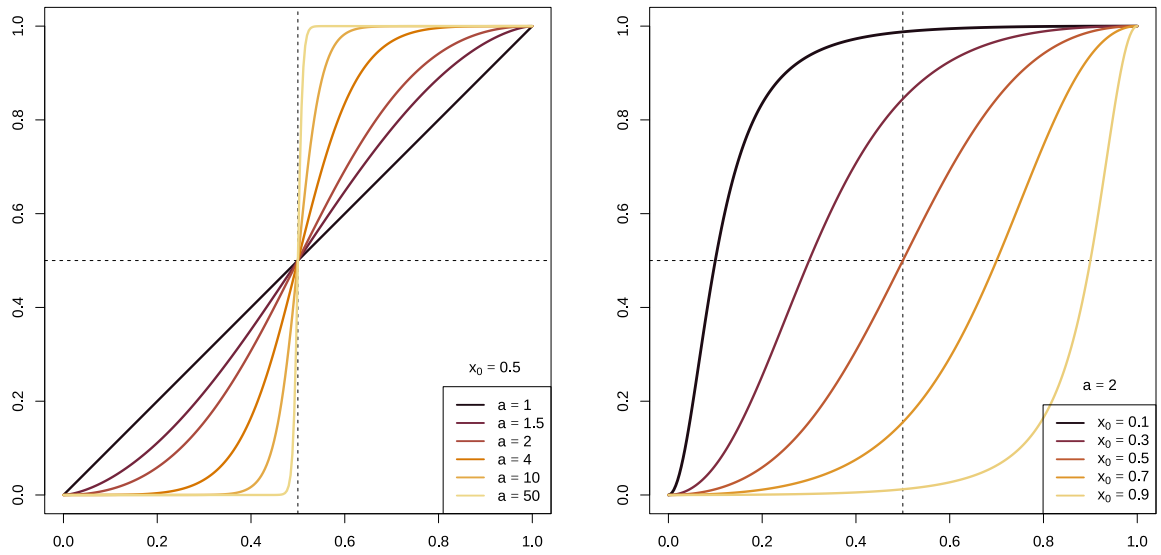


图4：用于变换距离以及变换、缩放和失真惩罚值的函数 ψ。参数 a 决定其形状，使函数在 $[0,1]$ ($a=1$) 上的恒等映射与阶跃函数 $1_{[1/2,1]}$ ($a\to\infty$) 之间插值。参数 x_0 决定点 x，在该点 $\psi(x)=1/2$。

我们选择

$$\psi(x)=\psi^{(\alpha,x_0)}(x)=\frac{1}{1+\left(\frac{x_0}{1-x_0}\frac{1-x}{x}\right)^{\alpha}},\quad x\in[0,1],$$

参数 $\alpha\geq 1$ 和 $x_0\in$ （0,1）分别决定变换的形状和平移，参见图4。该函数通过将标准 logit 函数代入广义逻辑函数并重新参数化平移量得到。[17]

最后，结合上述所有要素，我们得到 ϱ

$$\begin{aligned}\varrho(\mathfrak{C},\mathfrak{C}')=\lambda_0\,\psi_0\Big(\tfrac{1}{b\max\{|\mathfrak{C}|,|\mathfrak{C}'|\}}\,d_{\mathrm{UBW}}(\mathfrak{C},\mathfrak{C}')\Big)\\ +\lambda_1\,\psi_1\big(\tau(\mathfrak{C},\mathfrak{C}')\big)+\lambda_2\,\psi_2\big(\sigma(\mathfrak{C},\mathfrak{C}')\big)+\lambda_3\,\psi_3\big(\chi(\mathfrak{C},\mathfrak{C}')\big),\end{aligned}\tag{3.2}$$

其中 $\lambda_i\geq 0$ 是满足 $\sum_{i=0}^{3}\lambda_i=1$的参数。在缺乏数据且没有针对总体距离的具体目标时，我们在第4节中通过有依据的反复试验来选择各项参数；具体参数取值见公式（4.1）。一旦有了（足够多的）数据，似乎也有必要在线性表达式的右侧加入交互项；另见下一小节。

3.3 根据数据估计函数 ϱ

从长远来看，理想的做法是根据数据，估计 $\varrho(\mathfrak{C},\mathfrak{C}')$ 与 $^1/_{\text{最大}\{|\mathfrak{C}|,|\mathfrak{C}'|\}}\,d_{\mathrm{UBW}}(\mathfrak{C},\mathfrak{C}')$、$\tau(\mathfrak{C},\mathfrak{C}')$、$\sigma(\mathfrak{C},\mathfrak{C}')$ 和 $\chi(\mathfrak{C},\mathfrak{C}')$（或用于对部件进行配准的变换的其他合适惩罚值）之间的函数关系。收集此类数据的测试项目可以参考文献中已有的类似项目；参见 Yencken and Baldwin（2006, 2008）。

[17]标准 logit 函数为 $(0,1)\to\mathbb{R}, u\mapsto\log(\frac{u}{1-u})$，广义逻辑函数 $\mathbb{R}\to(0,1)\ y\mapsto(1+\exp(-\alpha(y-\beta)))^{-1}$ 对应参数 $\alpha>0$ (此处) 和 $\beta\in\mathbb{R}$。

例如（展示两个汉字）：“请在 0 到 1 的连续尺度上评定相似性。”或者（展示三个汉字）：“汉字 A 和汉字 B 中，哪一个与汉字 C 更接近？”不过请注意，这些项目在此首先用于训练，而非测试汉字距离映射，因此可以（也应该）有策略地设计。

可能存在两种截然不同的情形。一方面，我们可以通过少量有针对性的问题，从单个学习者或一个规模较小的同质群体中收集少量数据。具体目标可以是先学习参数 $\psi_i, i = 0, 1, 2, 3$，再学习它们的因子 λ_i。例如，ψ_0 的估计可以通过展示若干单个部件和没有强调分解的简单汉字来进行，比如从集合 {牛, 午, 干, 千} 中任选两个，也可以选取差异更明显的汉字，例如 {日, 月, 目}、{山, 川}、{老, 古} 或 {女, 名} 中的汉字对。函数 ψ_1, ψ_2 和 ψ_3 则可以根据更复杂且部件清晰的汉字进行估计：其中一个可识别部件在不同汉字中的位置、尺度和形变各异，而其他部件之间不易建立关联。例如：{案, 查}、{賀, 程} 或 {喋, 割, 鉛}。所有这些示例都是临时选取的；若要提高效率，所展示的汉字集合完全可以根据[1] $\max_{\{|\mathfrak{C}|,|\mathfrak{C}'|\}} d_{\mathrm{UBW}}(\mathfrak{C}, \mathfrak{C}')$、$\tau(\mathfrak{C}, \mathfrak{C}')$、$\sigma(\mathfrak{C}, \mathfrak{C}')$ 和 $\chi(\mathfrak{C}, \mathfrak{C}')$ 的实际取值来设计。

ψ_i 和 λ_i 的估计量最容易通过线性回归得到。对于ψ_i，我们可以先对因变量和自变量都应用logit变换（例如，取值范围为0到1的感知相似度，以及实际的 τ（$\mathfrak{C}$，$\mathfrak{C}'$）值），然后使用变换后的参数 $\tilde{x}_0 = \alpha \log(\frac{1-x_0}{x_0})$。

另一方面，当有大量数据可用，且这些数据来自向一大群异质学习者提出的问题时，可以估计一个比式（3.2）更复杂的函数 ϱ，该函数涉及最优墨迹传输和配准惩罚。这样的函数可以包含更多协变量（例如，来自特定学习者的协变量），并且要么基于灵活的设计而具有许多参数，要么是非参数的。用 x 表示（可能扩展后的）通用协变量向量，用 $\xi = (x_i)_{1\le i\le n}$ 和 $\eta = (y_i)_{1\le i\le n}$ 表示（训练）数据（感知相似度），适用的估计技术包括核平滑方法（参见Hastie et al., 2009），例如（变换后的）Nadaraya-Watson估计量

$$\hat{\varrho}(x; (\xi, \eta)) = \psi\left(\frac{\sum_{i=1}^n y_i\, k\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^n k\left(\frac{x-x_i}{h}\right)}\right),$$

其中，k 是输入空间上的一个适当核函数，$h > 0$ 是带宽参数，[18] 和ψ：$\mathbb{R} \to [0, 1]$ 是逻辑函数。

或者，如果我们不太关注估计量的可解释性特征，也可以使用神经网络，例如采用如下形式的简单前馈网络：

$$\hat{\varrho}(x; (\xi, \eta)) = f_T\big(A_{\hat{\theta}_T(\eta,\xi)}\big(f_{T-1}\big(A_{\hat{\theta}_{T-1}(\eta,\xi)} \cdots f_1(A_{\hat{\theta}_1(\eta,\xi)}(x))\big)\big)\big),$$

其中，A_{θ_t} 表示参数向量为 θ_t 的仿射映射，f_t 是预先指定的（非线性）^激活函数。这里，$\theta_t(\xi, \eta)$表示基于训练数据通过某种梯度下降算法学习得到的参数向量。

4 一个具体的汉字距离映射及初步结果

让我们为（3.2）中尚未确定的参数选取具体取值，以概念验证的方式构造第一个具体距离映射。我们根据目前的讨论，采用一种简单的方法，

[18] 更一般地，将 $k\left(\frac{x-x_i}{h}\right)$ 替换为 $k\big(H^{-1/2}(x - x_i)\big)$，其中 H 是一个正定带宽矩阵。

一些试错，但没有数据。和之前一样，我们在 d_{UBW}中设定 $p = 1$ 和 $b = 0.4$ 。为求简便，我们将 ψ_1、 ψ_2 和 ψ_3 都取为恒等函数。不过，我们采取了两项措施来增强 ψ_0 项对相似性的关注，纳入了这样一种效应：与 d_{UBW}距离项所表明的相比，人们对近似相似的感知更为强烈。一方面，我们直接为 ψ_0选择 $\alpha = 2$ 和 $x_0 = 0.4$ ；另一方面，我们引入 kanjiVG 数据中的部件标签，并允许这些标签覆盖 $d_{\mathrm{UBW}}(\mathfrak{C},\ \mathfrak{C}')$：如果两个部件都有标签且标签匹配，就将其设为零。关于λ_i，显然 λ_0 应占据绝大部分权重，因为总能找到平移、尺度和形变代价都极小的部件匹配；但如果这些部件在最优墨迹传输方面并不相似，距离就应保持较大。我选择了 $\lambda_0 = 0.8$ ，并在查看了一些具体示例、了解各项惩罚值之间的表现后，将剩余的 λ_i 分配给各项变换惩罚。总的来说，我们使用.....而不是 (3.2)

$$\begin{aligned}\varrho(\mathfrak{C}, \mathfrak{C}') = 0.8\, \psi^{(2,\,0.4)}\Big(\tfrac{\mathbb{1}\{\ell(\mathfrak{C})\neq\ell(\mathfrak{C}')\}}{0.2\max\{|\mathfrak{C}|,|\mathfrak{C}'|\}}\, d_{\mathrm{UBW}}(\mathfrak{C}, \mathfrak{C}')\Big) \\ + 0.1\,\tau(\mathfrak{C}, \mathfrak{C}') + 0.05\,\sigma(\mathfrak{C}, \mathfrak{C}') + 0.05\,\chi(\mathfrak{C}, \mathfrak{C}').\end{aligned} \tag{4.1}$$

式 (2.2) 中与汉字距离相关的其他量为 $a = 0.25$,[19] 权重$w_{l\,i}^{(1)}$ ， $w_{l\,i}^{(2)}$ ，如式 (2.1) 所示，第二层之后的递减损失为 0.02，最后还有 $\mu(w,\ w')$ =min(w, w')。

部件及其标签源自 kanjiVG 网站，但经过了进一步处理。kanjiVG 数据由 SVG 文件组成，其中以嵌套的 <g>-元素按层级存储部件。不过，许多此类元素还包含额外信息，从而形成了人为的层级。此外，这些文件遵循笔顺，这意味着逻辑部件有时会被拆分成多个部分。为了获得式 (4.1) 中使用的部件，我们重新合并了这些部分，并移除了一些（对我们的用途而言）不必要的层级；处理时相当谨慎，宁可少删。最终得到的部件属于数据集 kvec，该数据集可在 R 包 kanjistat 中获取。

我们使用（4.1）中的距离映射进行了若干实验。首先，确定了引言中讨论的四个汉字的最近邻（表2）。由于粋和框架十分相似，我们希望进一步考察这两个汉字的邻域。为此，我们分别取粋和框架的八个最近邻；结果发现，这些汉字恰好就是与这两个汉字距离为 < 0.14 的汉字。随后，我们又分别确定了它们的四个最近邻。最终得到42个互不重复的汉字，并计算了它们之间所有的$\binom{42}{2} = 861$ 成对距离。

在图5（左）中，我们可以看到使用 R 软件包 vegan（Oksanen 等人，2022年）通过 Kruskal 非度量多维尺度分析（nMDS）得到的这些汉字的整体映射。距离在几个明显的方面遭到了严重扭曲（事实上，粋和框架应该非常接近，只比析和祈稍远一些；但在映射中，它们也并不十分接近）。这是因为只有两个维度可用，而且 nMDS 首要目标是保持距离比较的次序，而非距离本身的实际大小。尽管如此，这幅映射仍有助于发现这42个汉字之间的相似性。

一个显著特征是结构相同的汉字会形成（有时较为松散且相互重叠的）群组。特别注意上方中间的五个汉字：它们的底部都有树形部件，整体采用上下分割而非左右分割。这种分组值得注意，因为我们并未在距离映射中明确建模这种分割类型。距离映射会自动识别出这一点，因为在这些

[19]在 整体部件比较中的未匹配权重惩罚项。不要与 α混淆 .

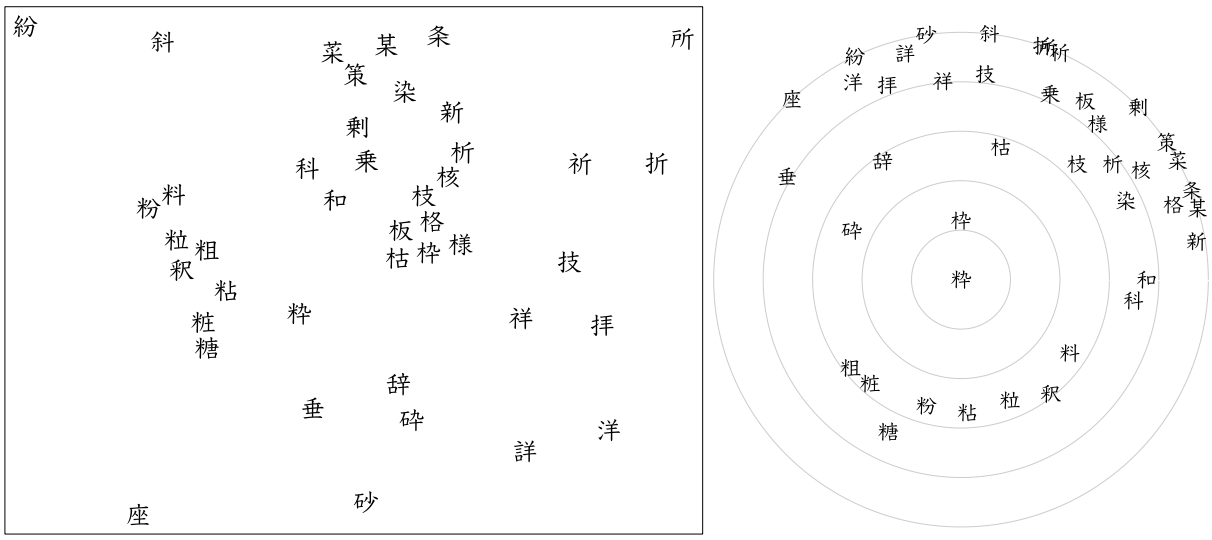


图5：基于粋和枠的邻邻汉字绘制的42个汉字。左图：基于非度量（亦称单调或序数）多维尺度分析绘制的全局地图。请注意，目的是展示全局邻域结构，而非获得良好的簇。右图：基于聚焦（度量）多维尺度分析展示的粋的邻域。圆圈表示与粋的距离每隔0.05的倍数。

树形部件（以及可能匹配的其他部件）之间，平移、缩放和畸变惩罚值几乎为零。

图5右侧更有针对性地展示了单个汉字的邻域。该图基于聚焦（度量）多维尺度分析绘制，参见 Urpa 和 Anders（2019）。这意味着图中显示了中心汉字粋到其他每个汉字的精确距离。后续每个汉字在与该距离对应的圆上的位置，通过最小化它与已放置汉字之间所有距离的应力（即平方偏差之和）来确定。R 软件包 focusMDS 提供了该图的动态版本，用户可以通过点击逐一浏览各个汉字。

正如预期，由于一些相似汉字之间的距离非常小（例如粋与枠之间），这42个汉字之间并不总是满足三角不等式。然而，在$\binom{42}{3} = 11,480$组三个汉字中，只有57组违反了三角不等式。大多数违反情况都涉及非常小的距离（< 0.08），或仅有轻微违反。因此，对于少数距离最近的汉字，focusedMDS 的动态版本有时会产生一些奇怪的效果，但总体而言，这仍是一种非常有用的表示方式。

当然，也存在几种更复杂的方法，可以根据相似性或距离绘制地图；例如，参见 Amid 和 Warmuth（2022）中比较的方法。

最后，我们从2136个常用汉字中均匀随机抽取100个汉字，并在附录中列出它们最近的16个常用邻近汉字。浏览表格后可以发现，相似汉字及其从左到右的排列顺序（包括表示距离区间的颜色）总体上看似合理。如果目标是生成一组相似汉字，供学习者挑选自己感兴趣或正在寻找的汉字，那么总体而言，这些结果似乎非常有用。

另一方面，有时也会有少数汉字排在相当靠前的位置，但它们可能不会被（许多）学习者认为彼此接近。例如，悔（从上往下第十二行）中，母排在第六位，距离为0.123339，这是因为它与右下角部分十分匹配。另一个可能更糟的例子是残（从下往上第三行），其中三排在第五位，距离为

为 0.1352396。经检查，这个距离如此之小，部分是因为三与右侧三个横笔画的匹配所需的笔画搬运成本为零（因为 kanjiVG 标签相符）；但主要是因为，在 kanjiVG 文件中，残的第 5 至第 10 笔画（即其整个右侧部分）不知何故被归入部件三，因此这一匹配获得了不恰当地高达 0.6055 的权重。

5 结论

本文介绍了一种通用的汉字距离映射框架，通过在所有层级上以无嵌套的方式匹配部件的层级结构来实现。其底层的部件距离图通常由两部分构成：已配准部件之间非平衡最优墨迹传输的相对代价，以及适当的配准代价。我们还讨论了许多可能影响部件代价、但目前尚未纳入我们的首个汉字距离（3.2）的其他方面。

这个首个汉字距离是一个原型，已显示出令人期待的结果。不过，艰巨的工作才刚刚开始。要建立权威的通用汉字距离映射（或面向特定受众的、更有针对性的距离图），还需要完成以下步骤。

1. 必须进一步减少部件数据集中的错误和误解。根据我的经验，kanjiVG 数据集是目前公开可用的最佳起点。不过，由于该数据集同时追求许多目标，其结构相当复杂。我们目前用于将 kanjiVG 文件转换为适当部件结构的代码，尚未完整处理所有特殊情况；其中一些情况甚至连人工解读这些文件也颇为困难。同样，我们用于识别部件的标签取自 kanjiVG 中的 `element` 属性，既不完整也不唯一。最好能解决这一问题，例如创建扩展部件标签，将所有相关笔画的 `type` 属性也纳入其中，或者尝试将这些数据与其他来源的分解信息结合起来，例如 KRADFILE。

2. 应采用更复杂的配准流程（例如参见脚注 15），并使用一种惩罚项，要么整体评估所需的变换，要么能够考虑平移、缩放和畸变效应之间的相互作用。

3. 应设计并开展实验，以获取适合训练和评估的数据。理想的目标群体可能是低年级的日本学学生：他们因当前学习而对汉字有所接触，但没有中文文字背景。我们希望排除从未接触过汉字的受试者，因为在相似性差异的感知方面，这类群体可能过于异质。此外，我们也可能排除汉字知识水平较高的受试者，因为对这类群体而言，很难控制先前知识对相似性判断的影响。

4. 整个过程应由认知心理学、心理语言学或（理想情况下）日文汉字／汉字学习理论方面的专家监督。

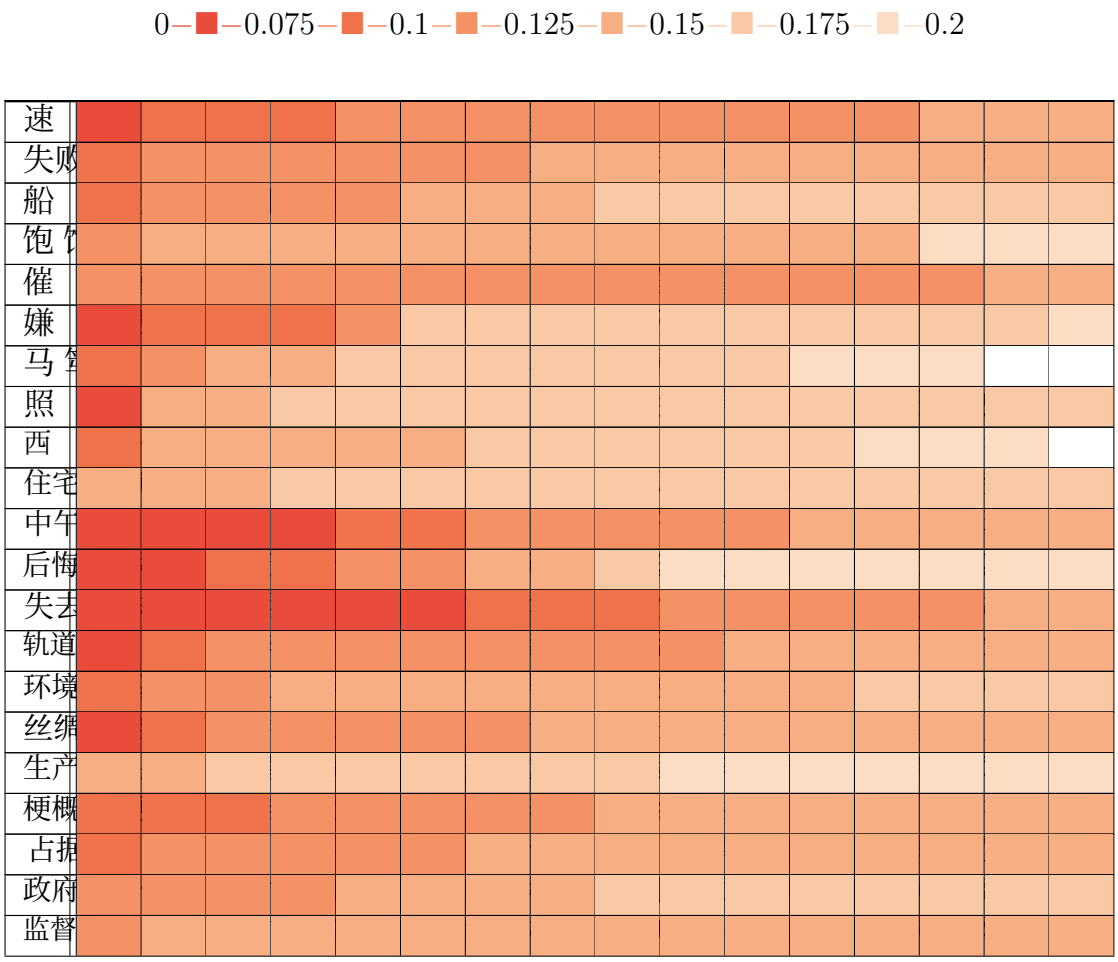
关于上一点提到的汉字，我们注意到，本文所述距离图的许多方面可直接应用于（简体和繁体）中文汉字。唯一的主要障碍是，需要其他数据集来获取分解、标签等。

最后需要提醒的是，应注意不要过度设计汉字距离。本文提供了一系列工具和思路，可以相当自由地组合使用。不过，应根据具体情况，结合专业知识、统计数据分析和常识，审慎地进行选择。

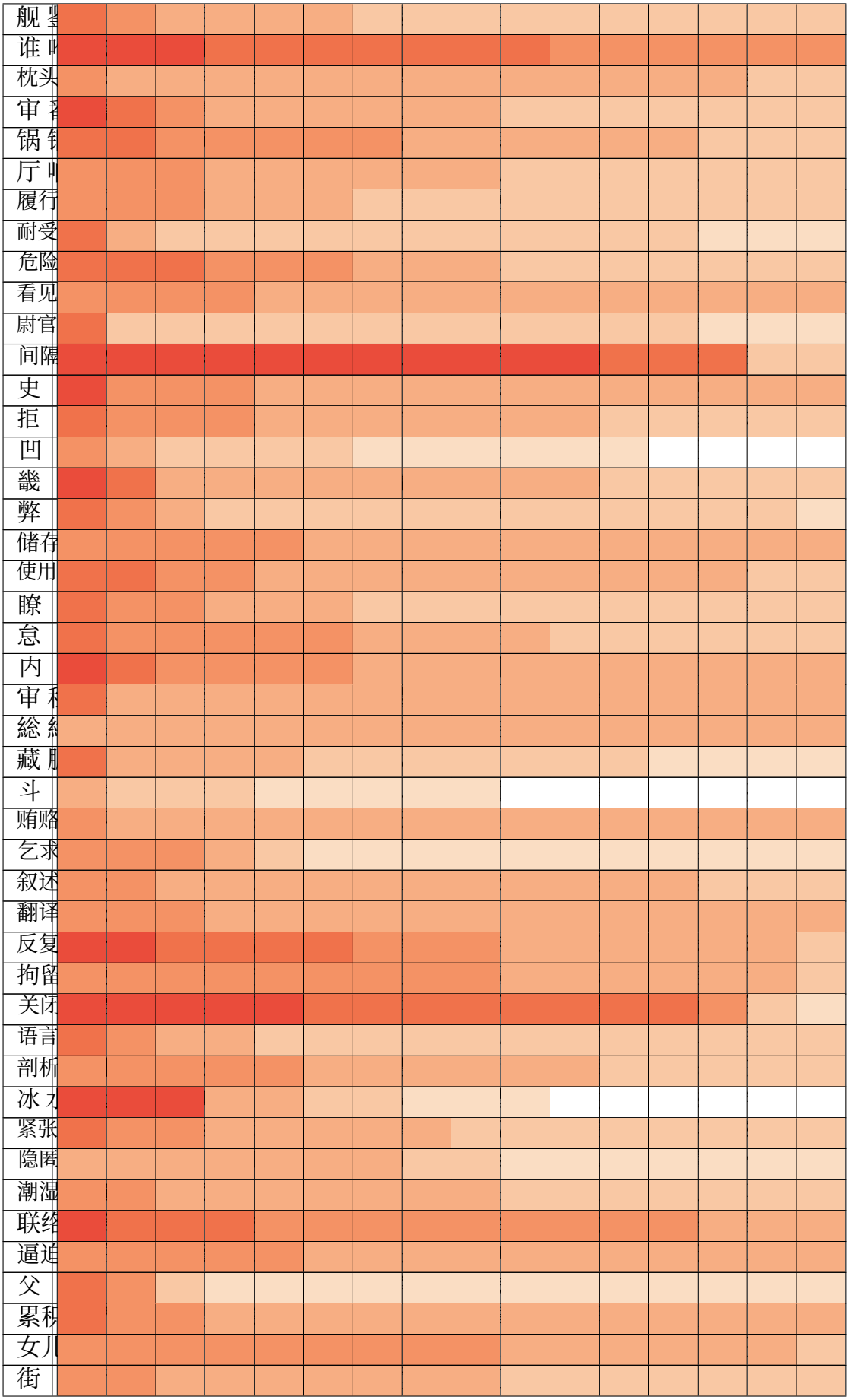
附录：百鬼夜行汉字

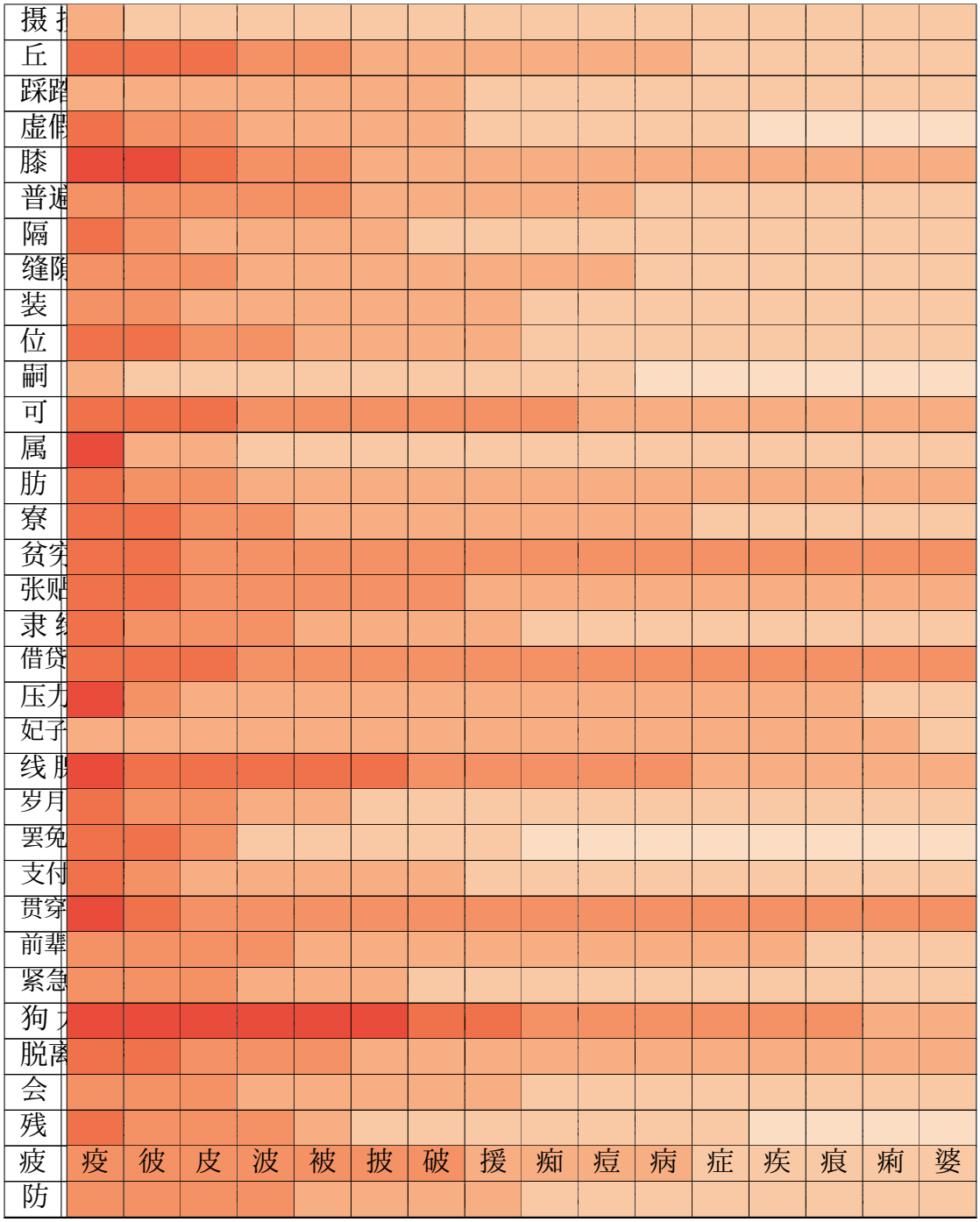
为了让读者评估采用第 4 章中的具体参数设置所得到的汉字距离在多大程度上符合自己的判断，我们计算了从 J⁻oy⁻o 汉字集合中随机抽取的 100 个字符各自的最近邻。为确保客观性和可复现性，我们在启动 R 4.2.2 版本时，直接使用命令 `sample(2136, 100)` 抽取汉字（参照 kanjistat 中 `kbase`、 `kmorph`和 `kvec` 数据集里的索引）。[20]

下表按距离由近到远的顺序，列出了每个汉字的 16 个最近邻。汉字的背景颜色从深红色（距离较小）渐变至白色（距离较大），如下所示：



[20]起始　始种子应在（几乎）所有系统中保持稳定，并在不同　t
版本。T　他前五个索引分别为 360、555、183、1776 和 1220。


舰
谁
枕头
审
锅
厅
履行
耐受
危险
看见
尉官
间隔
史
拒
凹
畿
弊
储存
使用
瞭
怠
内
审
総
藏
斗
贿赂
乞求
叙述
翻译
反复
拘留
关闭
语言
剖析
冰
紧张
隐匿
潮湿
联络
逼迫
父
累积
女儿
街



参考文献

Amid, E. 和Warmuth,M. K.（2022年）。Trimap：使用三元组进行大尺度降维。*ArXiv* 预印本。 https://doi.org/10.48550/arXiv.1910.00204。

Apel, U. 和 Quint, J.（2004年）。构建日本汉字字形词典——基于笔触或笔画组的字符查找，以及以路径数据形式显示汉字。载于增强与使用电子词典研讨会论文集，第36–39页。 https://aclanthology.org/W04-2106。

Chizat, L., Peyr´e, G., Schmitzer, B., 和 Vialard, F.-X.（2018）。非平衡最优传输问题的缩放算法

。计算数学，87(314):2563– 2609. https://doi.org/10.1090/mcom/3303。

Halpern, J.（2022年）。讲谈社汉字学习者词典。Kodansha America，第2版。

Hastie, T., Tibshirani, R. 和 Friedman, J.（2009）。统计学习要素：数据挖掘、推断与预测。Springer，第2版。 https://doi.org/10.1007/978-0-387-84858-7。

Heinemann, F., Klatt, M. 和 Munk, A.（2023）。有限支撑测度的 Kantorovich–Rubinstein 距离与重心：基础与算法。应用数学与优化，87(1):4。 https://doi.org/10.1007/s00245-022-09911-x。

Kandr´aˇc, P.（2021）。基于多准则的汉字排序。硕士论文，布拉迪斯拉发夸美纽斯大学东亚研究系。 https://doi.org/10.13140/RG.2.2.28227.73762。

Laub, J., M¨uller, K.-R., Wichmann, F. A., 和 Macke, J. H. (2006). 诱发人类相似性判断中的度量违例。神经信息处理系统进展，19。 https://papers.nips.cc。

M¨uller, R., Schuhmacher, D., 和 Mateu, J. (2023). 度量空间的方差分析及其在空间数据中的应用。*ArXiv* 预印本。https://doi.org/10.48550/arXiv.2201.08664。

Nachshon, Y., Cohen, H., Ben-Artzi, M., 和 Maril, A. (2022年). 相似性模型：局部区域中的度量。*PsyArXiv* 预印本。 http://doi.org/10.31234/osf.io/fyd4q。

Nagata, M. (1998). 使用字符形状相似性和统计语言模型纠正日文 OCR 错误。在 计算语言学协会第*36*届年会暨第*17*届计算语言学国际会议，第*2*卷，第922–928页。 https://doi.org/10.3115/980691.980721.

Oksanen, J., Simpson, G. L., Blanchet, F. G., Kindt, R., Legendre, P., Minchin, P. R., O'Hara, R., Solymos, P., Stevens, M. H. H., Szoecs, E., Wagner, H., Barbour, M., Bedward, M., Bolker, B., Borcard, D., Carvalho, G., Chirico, M., De Caceres, M., Durand, S., Evangelista, H. B. A., FitzJohn, R., Friendly, M., Furneaux, B., Hannigan, G., Hill, M. O., Lahti, L., McGlinn, D., Ouellette, M.-H., Ribeiro Cunha, E., Smith, T., Stier, A., Ter Braak, C. J., and Weedon, J. (2022年). *vegan*：群落生态学软件包。R 软件包版本 2.6-4, https://CRAN.R-project.org/package=vegan。

R Core Team（2022年）。*R*：统计计算语言与环境。R统计计算基金会，维也纳，奥地利。 https://www.R-project.org。阪本 浩太郎（Sakamoto, K.）、阿部川 明優、岸川 至白、阪本 エリーザ、石下 円香、渋木 英潔和森 辰則（2019）。契約書の OCR 漢字誤り訂正における偏旁冠脚を考慮し た編集距離の検討（关于在合同 OCR 汉字错误纠正中考虑汉字部首的编辑距离研究）。信息处理学会研究报告，2019-NL-242(7):1–7。http://id.nii.ac.jp/1001/00199661/。

Scheirer, W. J., Wilber, M. J., Eckmann, M. 和 Boult, T. E.（2014）。良好的识别是非度量的。模式识别，47(8):2721–2731。 https://doi.org/10.1016/j.patcog.2014.02.018。Schuhmacher, D.（2023）。*kanjistat*：日本汉字字符分析框架。R 软件包版本 0.7.0， https://github.com/dschuhmacher/kanjistat。Schuhmacher, D., B¨ahre, B., Bonneel, N., Gottschlich, C., Hartmann, V., Heinemann, F., Schmitzer, B. 和 Schrieber, J.（2023）。*transport*：最优传输方案与*Wasserstein*距离的计算。R 软件包版本 0.13-2， https://CRAN.R-project.org/package=transport。Schuhmacher, D. 和 Wirth, L.（2023）。属性图的基于分配的度量。编写中。Tversky, A.（1977）。相似性特征。心理学评论，84(4):327–352。 https://doi.org/10.1037/0033-295X.84.4.327。Urpa, L. M. 和 Anders, S.（2019）。聚焦多维缩放：用于探索高维数据的交互式可视化。*BMC*生物信息学，20(1):1–8。https://doi.org/10.1186/s12859-019-2780-y。Yencken, L. 和 Baldwin, T.（2006）。日文汉字正字法邻域建模。载于东方语言计算处理国际会议，第321–332页。Springer。 https://doi.org/10.1007/11940098_33。Yencken, L. 和 Baldwin, T.（2008）。测量和预测正字法关联：日文汉字相似性建模。载于第*22*届计算语言学国际会议论文集（*Coling 2008*），第1041–1048页。https://aclanthology.org/C08-1131。Yurochkin, M., Claici, S., Chien, E., Mirzazadeh, F. 和 Solomon, J. M.（2019）。用于文档表示的层次化最优传输。神经信息处理系统进展，32。 https://proceedings.neurips.cc。