摘　要

矩阵分解是一种在人工智能领域中广泛应用的数学技术，特别是在推荐系统、自然语言处理和图像处理等领域。本文将主要讨论矩阵分解在人工智能中的应用，并讨论矩阵分解在人工智能不同领域的实际意义。

关键词：矩阵分解，人工智能，SVD，PCA

目　录

第 1 章　引　　言

随着人工智能和深度学习研究的不断发展，矩阵理论在其中的应用越来越广泛。矩阵是深度学习数学基础的重要组成部分，对于掌握和理解深度学习算法具有至关重要的作用。这篇论文主要讨论矩阵分解在人工智能中的应用，并讨论矩阵分解在人工智能不同领域的实际意义。

1.1　矩阵分解

矩阵分解是将一个矩阵拆解成多个子矩阵的过程，通常是通过将原始矩阵表示为其他形式的矩阵相乘的形式来实现。这种拆解可以有多种形式，常见的包括奇异值分解（SVD）、QR 分解、LU 分解等。

最常见的矩阵分解之一是奇异值分解（SVD），它将一个矩阵分解成三个矩阵的乘积:

$$A = U \Sigma V^T$$

其中，A 是原始矩阵，U 和 V 是正交矩阵，Σ 是对角矩阵，对角线上的元素称为奇异值。这种分解可以用于降低矩阵的维度、压缩信息、解决最小二乘问题等。

另一个常见的矩阵分解是 QR 分解，将一个矩阵分解成一个正交矩阵和一个上三角矩阵的乘积:

$$A = QR$$

在 QR 分解中，Q 是正交矩阵，R 是上三角矩阵。这种分解常用于求解线性方程组和最小二乘问题。

LU 分解将一个矩阵分解成一个下三角矩阵和一个上三角矩阵的乘积:

$$A = LU$$

其中，L 是下三角矩阵，U 是上三角矩阵。LU 分解常用于求解线性方程组，特别是在多次求解相同系数矩阵但不同常向量的线性方程组时效率更高。

矩阵分解技术在数学、工程、统计学和计算机科学等领域中都有广泛的应用，是很多算法和模型的基础。

1.2　人工智能

人工智能（Artificial Intelligence，AI）是一门计算机科学的分支，旨在使计算机系统能够模仿人类智能的各种方面。这包括理解自然语言、学习、推理、规

划、感知、移动和操作对象等多方面的技能。

人工智能的核心目标是使计算机系统能够执行通常需要人类智能的任务，而不需要人类的直接干预。这涉及到对复杂问题的处理、模式识别、自动化决策等方面。人工智能的方法和技术广泛而多样，包括但不限于以下几种：

1. 机器学习：机器学习是人工智能的一个重要分支，其目标是让计算机系统能够从数据中学习并改进性能，而不需要明确地进行编程。机器学习方法包括监督学习、无监督学习、强化学习等。

2. 深度学习：深度学习是机器学习的一种特殊形式，它使用人工神经网络来模拟人脑的神经元结构，从而实现对数据的高级抽象和表征学习。

3. 知识表示与推理：这些技术致力于将人类知识表示在计算机中，并利用逻辑、推理和推断来处理复杂的问题。常见的方法包括专家系统、规则引擎、图形推理等。

4. 自然语言处理：自然语言处理（Natural Language Processing，NLP）旨在使计算机能够理解、解释和生成自然语言的技术。这包括文本分析、语义理解、机器翻译等。

5. 计算机视觉：计算机视觉涉及使计算机系统能够理解和分析图像或视频内容的技术。这包括目标检测、图像分类、人脸识别等任务。

6. 感知与机器人：这些技术使计算机系统能够感知周围环境，并进行交互。这包括传感器技术、机器人控制、自主导航等。

人工智能的应用非常广泛，包括但不限于自动驾驶、医疗诊断、智能客服、金融风控、电子商务推荐系统等。随着技术的不断发展和进步，人工智能将在更多领域中发挥重要作用，并对人类社会产生深远的影响。

第 2 章　矩阵分解在人工智能中的应用

2.1　矩阵分解的作用

矩阵分解，是指将一个矩阵表示成多个矩阵连乘的形式。矩阵分解主要有三个作用。值得注意的是，不同的矩阵分解方法可能会同时有以下多种作用，例如 PCA 既可以降维也可以去除噪声点，防止过拟合等。从时间上来说，1999 年前主流的矩阵分解方法有 PCA，SVD 和 LSI 等，主要用于降维，聚类分析和数据预处理；1999 年到 2006 年间，常用的矩阵分解方法有 PLSA，NMF 和 LDA 等，主要用于低纬度特征学习和聚类分析；2006 年后的矩阵分解有 MF，PMF 和 SVD++ 等，主要用于特征学习，推荐系统和大数据分析等。

2.1.1　降维与压缩

首先，矩阵分解后的每个小矩阵能够更加容易的求逆，这里以 LDU 分解为例。

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{c}{a} & 1 \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & \frac{ad-bc}{a} \end{bmatrix} \begin{bmatrix} 1 & \frac{b}{a} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{c}{a} & 1 \end{bmatrix} \begin{bmatrix} a & b \\ 0 & \frac{ad-bc}{a} \end{bmatrix}, \quad a \neq 0 \tag{2.1}$$

对于上述矩阵，我们利用 LDU 分解将其分解为上三角，对角矩阵和下三角矩阵。

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -\frac{b}{a} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{a} & 0 \\ 0 & \frac{a}{ad-bc} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{c}{a} & 1 \end{bmatrix} = \begin{bmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}, \quad a \neq 0 \tag{2.2}$$

可以看出，分解后得到的三个小矩阵的逆都很容易得到。

说到矩阵数据降维方法，就不得不提到 PCA（主成分分析），PCA 首先将 mxn 的矩阵 X 的每一行进行零均值化（减去这一行的均值），求出协方差矩阵 C，然后求出 C 的特征值及其对应的特征向量，将特征向量按照特征值的大小从上到下按行排列成矩阵，取前 K 行作为矩阵 P，Y=PX 即降维到 k 维的数据。其中 C：

$$C = \frac{1}{m} X X^T \tag{2.3}$$

PCA 能很好的降维及避免多变量间的相关性，有助于减轻维度灾难和提高计算效率。

提到 PCA，那就要提到 SVD（奇异值分解），下式 U 和 V 都是正交矩阵，Σ

是对角矩阵，其中的对角元素 σ 成为矩阵 A 的奇异值。

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \tag{2.4}$$

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T = V\Sigma^2 V^T \tag{2.5}$$

$$\sigma_i = \sqrt{\lambda_i} \tag{2.6}$$

由上面两个式子，我们可以得到，A 的奇异值 σ 是 $A^T A$ 的特征值 λ 的平方根，所以说奇异值的分解关键在于对 $A^T A$ 进行特征值分解。而 PCA 关键在于对 C 进行特征值分解，所以说两者解决的问题都很相似，都是对一个实对称矩阵进行特征值分解，并且两者可以轻易等价。

此外我们考虑，既然两者等价，PCA 问题转化为 SVD 问题求解有什么好处? 一般而言，X 的维度很高，方阵的特征值分解计算效率不高，而 SVD 除了特征值分解这种求解方式外，还有更高效更准确的迭代求解法，避免了 $A^T A$ 的计算。此外，PCA 其实只与 SVD 的右奇异值（V）的压缩效果相同，如果只取 V 的前 k 行作为变换矩阵乘以 X，起到压缩行即降维的效果，如果只取 U 的前 k 行作为变换矩阵乘以 X，起到压缩列即去除冗余样本的效果，而由上述 PCA 可知，它是取特征向量的前 k 行作为变换矩阵，所以得到一个降维的效果。

值得注意的是 SVD 的时间复杂度是 $O(n^3)$ 或者是 $O(nm^2)$, 所以 SVD 的速度是很慢的。在 SVD 分解矩阵 Σ 中，是按照奇异值从大到小顺序排列，在很多情况下前 10% 甚至 1% 的奇异值的和就占了全部的奇异值之和的 99% 以上。也就是说, 剩下的 90% 甚至 99% 的奇异值几乎没有什么作用。因此, 我们可以用前面 r 个大的奇异值来近似描述矩阵，所以在 SVD 中 r 的取值很重要，就是在计算精度和时间空间之间做取舍。这里介绍 SVD++ 是指在 SVD 的基础上对于 baseline Predictors 引入隐式反馈。概括来说，就是从评分 = 兴趣 + 偏见转化为，显示兴趣 + 偏见 + 隐式兴趣。

LSI（Latent Semantic Indexing), 或者可以称为 LSA（Latent Semantic Analysis），是基于 SVD 的一种应用。在文本主题模型中，经过一次 SVD，就可以得到文档和主题的相关度，词和词义的相关度以及词义和主题的相关度。LSI 的缺点其实就是 SVD，SVD 计算非常耗时，尤其是处理文本时，词和文本数都是非常大的，对于这样的高维矩阵做 SVD 很难。解决办法就是使用在主题模型中使用 NMF，可以提高矩阵分解的速度。NMF（非负矩阵分解，一种无监督学习方法），就是指将一个非负的大矩阵分为两个非负的小矩阵。对于 PCA 和 NMF 的基分析可以知道，PCA 的基是指向四面八方的，相互正交着。NMF 的原数据首先就是只分布在非负子空间里面的，然后它的基则在这个非负子空间靠近边缘的区域，像一组长短不一、间隔不一的伞骨。那如何对它求解呢，首先定义为下式有界优化

问题：

$$\min_{W,H} \quad f(W,H) \equiv \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}(V_{ij}-(WH)_{ij})^2 \text{subject to} \quad W_{ia} \geqslant 0, H_{bj} \geqslant 0, \forall i,a,b,j. \tag{2.7}$$

这里提供一个标准有界优化问题解法：

$$W^{k+1} = \max(0, W^k - \alpha_k \nabla_W f(W^k, H^k)), H^{k+1} = \max(0, H^k - \alpha_k \nabla_H f(W^k, H^k)), \tag{2.8}$$

我们从中可以总结出来 NMF 的优点为：1、得到的基是 parts-based 的，2、整个优化问题是非凸的，但是通过上述式子先固定其中一个待估计矩阵，子问题是凸的，能够使用梯度下降方法和保证算法的收敛性。NMF 解决了 SVD 的计算复杂问题（SVD 特征向量两两正交，计算慢），且求得的所有矩阵都为正值，更好理解。两者的不同在于 SVD 根据奇异值的大小确定保留哪些隐含维度，NMF 直接根据指定的隐含维度分解矩阵。

此外，LSI 得到的不是一个概率模型，缺乏统计基础，结果难以直观的解释。（解决：pLSI(也叫 pLSA) 和隐含狄利克雷分布 (LDA) 这类基于概率分布的主题模型来替代基于矩阵分解的主题模型。）由于本文是介绍矩阵分解在人工智能中的应用，于是不做过多介绍，我们由上述引入概率分布的思想，进一步介绍 PMF（Probabilistic Matrix Factorization），概率矩阵分解。

大多数存在的协同过滤算法不能处理以下两种情况：1. 不能处理大规模数据 2. 不能处理评分非常少的用户数据。概率矩阵分解模型可以解决大规模、稀疏且不平衡的数据。这篇文章主要介绍概率矩阵模型 PMF。

若用户 U 的特征矩阵满足均值为 0，方差为 σ 的高斯分布，则有下面两式：

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} N(U_i|0,\sigma_U^2 I) \tag{2.9}$$

$$p(V|\sigma_V^2) = \prod_{i=1}^{N} N(V_i|0,\sigma_V^2 I) \tag{2.10}$$

我们也可以得到评分矩阵 R 的条件概率如下：

$$P(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[N(R_{ij}|U_i^T V_j,\sigma^2)]^{I_{ij}} \tag{2.11}$$

最后通过贝叶斯公式得到 R，U，V 的联合分布，如下式：

$$\begin{aligned} P(U,V|R,\sigma^2,\sigma_U^2,\sigma_V^2) &= P(R|U,V,\sigma^2)P(U|\sigma_U^2)P(V|\sigma_V^2) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M}[N(R_{ij}|U_i^T V_j,\sigma^2)]^{I_{ij}} \prod_{i=1}^{N} N(U_i|0,\sigma_U^2 I) \prod_{j=1}^{M} N(V_j|0,\sigma_i^2 I) \end{aligned} \tag{2.12}$$

2.1.2　矩阵填充

此外，一个稀疏的矩阵分解为多个稠密小矩阵，也可以理解为从一个很复杂的矩阵信息中，提取一些内在关系的过程，稠密的小矩阵能够更加有效的存储信息。

通过矩阵分解来填充原有的矩阵，这里以协同过滤的 ALS（Alternating least squares）算法为例。ALS 算法，又称交替最小二乘，在推荐系统中用来补全用户评分矩阵。由于用户评分矩阵比较稀疏，将用户评分矩阵进行分解，变成 V 和 U 的乘积（V 和 U 分别表示用户因子向量和物品因子向量），通过求得 V 和 U 两个小的矩阵来补全用户评分矩阵。一般来说，这里的 V 和 U 都是满秩且稠密的。ALS 算法的代价函数如下：

$$f[i] = \arg\min_{w\in\mathbb{R}^d} \sum_{j\in \mathrm{Nbrs}(i)} \left(r_{ij} - w^T f[j]\right)^2 + \lambda ||w||_2^2 \tag{2.13}$$

ALS 算法之所以称之为交替最小二乘法，是因为无论是显示还是隐式代价函数求解，都是凸函数，而且变量耦合在一起，常规的梯度下降算法不能够求解，但是先固定 U 求 V，再固定 V 求 U，如此迭代下去，问题就可以解决了。

2.1.3　有特殊的物理意义

当我们把一个 2x2 矩阵分解为三个因子，一个 2x2 的矩阵可以用来表示平面内的 Affine 变换，也就是先对矩阵进行线性变化，再进行平移变化；也可以理解为对矩阵先做 x 轴上 shearing 变换，再 scaling，再对 y 轴进行 shearing；上下三角矩阵表示对 x 和 y 轴做 shearing 变化；对角矩阵表示对 x，y 轴做 scaling 变换；

所以总的来说，矩阵分解可以用于矩阵填充，降维与压缩，推荐系统，清理异常值与离群点等多种作用。

2.2　不同 AI 领域使用矩阵分解的方法

矩阵分解是一种在人工智能领域中广泛应用的数学技术，特别是在推荐系统、自然语言处理和图像处理等领域。

2.2.1　推荐系统

推荐系统的目标是根据用户的历史行为和偏好向他们推荐相关的商品、内容或服务。常用的方法包括奇异值分解（SVD）、主成分分析（PCA）和潜在因素模型（LFM）等。Netflix Prize 竞赛中采用的矩阵分解方法，以及基于矩阵分解的协同过滤算法。

2.2.2　自然语言处理

在 NLP 任务中，矩阵分解通常用于词嵌入（word embeddings）和文本表示。Word2Vec、GloVe 等词嵌入模型使用了矩阵分解技术。词嵌入在文本分类、命名实体识别、情感分析等任务中广泛应用。

2.2.3　图像处理

图像处理任务中，矩阵分解可用于图像压缩、去噪和特征提取。奇异值分解（SVD）和主成分分析（PCA）常用于图像压缩和特征提取。JPEG 图像压缩算法中使用的离散余弦变换（DCT）可以看作是一种矩阵分解方法的应用。

2.2.4　深度学习

在大规模数据集上进行分析和挖掘时，矩阵分解可用于降低数据维度和提取关键特征。主成分分析（PCA）、因子分析和非负矩阵分解（NMF）等方法常用于降维和特征提取。深度学习模型中的权重矩阵可以看作是对输入数据的特征表示，矩阵分解技术可以用于优化和解释这些权重。因式分解机（Factorization Machines）等模型在深度学习中被用于特征交叉和表示学习。

2.3　总结

矩阵分解在人工智能中应用广泛，是人工智能的不可缺少的数学基础，在实际问题中发挥重要作用，促进了人工智能技术的发展和应用。