

復旦大學

本科毕业论文



论文题目：基于深度半监督的遥感图像分类研究

院 系：计算机科学技术学院

专 业：计算机科学与技术

姓 名：赵阳旻 学 号：14307130067

指导教师：池明旻 职 称：副教授

日 期：2018 年 6 月 12 日

目录

第一章 绪论	6
1.1 研究背景	6
1.2 研究目的	6
1.3 全文结构	6
第二章 高光谱成像技术	8
2.1 成像原理简介	8
2.2 分辨率的权衡	10
第三章 浅层的半监督算法	11
3.1 半监督学习的有效性	11
3.2 TSVM 模型	12
3.2.1 二分类的 SVM 模型	12
3.2.2 多分类的 TSVM 模型	13
第四章 半监督深度学习	16
4.1 流形嵌入	16
4.1.1 分析学观点	16
4.1.2 LapSVM	16
4.2 基于流形嵌入的半监督深度学习	18
4.2.1 output 模型	19
4.2.2 internal 与 auxiliary 模型	20
第五章 应用与实现	21
5.1 有关空间的正则化	21
5.2 基于 tensorflow 的实现	23
5.2.1 神经网络结构	23
5.2.2 目标函数	25
5.2.3 实现技巧	26
5.3 不同数据集上的测试	29

目录	3
5.3.1 数据集介绍	29
5.3.2 实验设置	31
5.3.3 实验结果	32
5.3.4 实验结论	36
第六章 总结与展望	37
附录 A 主要符号	38
附录 B 主要定义	40
附录 C 主要定理	42
附录 D 推导变换	43

摘要

高光谱遥感传感器使人们能够通过遥感图像对地表进行深度研究，包括资源的分类、探测、地质与化学的构成分析等等。因此，高光谱图像在农业、环境、城市规划、矿业、国防等诸多领域显得日益重要。不仅如此，在民用方面，随着移动互联网科技的发展，其与遥感技术关系也亦密切。诸如谷歌地图、高德地图、百度地图等地图应用，均离不开遥感技术的发展。

机器学习技术拥有强大的预测能力，如今已经成为遥感图像分析的重要手段。然而遥感图像基于其本身的特点，标记的代价很高。为此，基于社交媒体图片帮助遥感图像进行标记的方法被提出，也已经成为了新的研究热点。即便如此，研究一系列半监督学习算法仍然是非常关键的。

本文立足于高光谱图像与半监督学习算法，首先介绍高光谱图像的成像原理，并分析其由此而得的重要特点。随后基于浅层的机器学习模型介绍半监督算法，尤其是支持向量机，由此回顾各式半监督学习算法在遥感图像分类上的应用成果，并引出对构造正则化项的探讨。接下来在深度学习模型上运用正则化项，并结合高光谱图像的特点进行扩展。使得能够在标记不足的情况下尽可能提高模型的分类准确率。

关键词：遥感、高光谱、图像分类、半监督学习、机器学习、深度学习、正则化项

Abstract

Hyperspectral sensors enable researchers to study the surface of earth by remote sensing images, including the classification and detection of resources, the chemical and geologic analysis and so on. Hence, hyperspectral images have been increasingly important in the fields of agriculture, environment, urban planning, mining and defense. Furthermore, in the aspect of commercial product, with the development of mobile internet technology, the connection between remote sensing and internet also becomes tighter. The applications like Google Map, AMAP, Baidu Map cannot work without remote sensing technology. Machine learning algorithms have become the critical methods in the analysis of remote sensing image due to their strong abilities of prediction. However, it is rather expensive to label all pixels of the image. For the sake of this problem, a novel method of labelling with the help of social media images has been proposed and become a hot topic. Even so, it is still necessary to study some semi-supervised algorithms in the task of classification of remote sensing images.

This paper concentrates on hyperspectral images and semi-supervised algorithms. Firstly, the mechanism of hyperspectral images is introduced. The important feature of the image would be therefore fetched from the analysis. Then the semi-supervised algorithms based on shallow machine learning models especially the Support Vector Machine would be learned. So that the results of various semi-supervised algorithms in the task of classification of hyperspectral images would be reviewed. And the discussion of constructing regularization is proposed from the analysis above. Afterwards, the regularization technique is applied to the deep learning model and expanded according to the feature of hyperspectral images. Finally, the accuracy of the classification can be improved without sufficient labels.

Index terms: Remote sensing; Hyperspectral images; Image classification; Semi-supervised learning; Machine learning; Deep learning; Regularization

第一章 绪论

1.1 研究背景

随着卫星遥感技术的快速发展与传感器研发水平的提高，遥感图像被广泛运用在天气预报、国土测绘、资源探测等多个领域。也因如此，遥感图像的复杂性在不断提高：从红外光谱图，到高光谱、高空间、高时间分辨率的图像，遥感图像种类与信息也越来越多。但是，对于遥感图像中地表覆盖物的分类标注长期停留在基础阶段：传统上，为了获得图像地表的准确分类，常常需要专业人员实地勘定，代价高昂。因此，如何以较低的成本，充分利用现有的标记数据，从而准确地标记大量遥感数据，已经成为了一个亟待解决的问题。这正是本文所要探讨的方法。

1.2 研究目的

本课题拟采用基于深度半监督学习的方法，实现对于遥感图像中未标记像素的分类与预测。遥感图像的标注成本很高，因此准确标注的像素点较少，这适用于半监督学习的应用场景。同时，随着深度学习技术的快速发展，相应的深度半监督学习技术也得到了一定提高。相比于传统的浅层半监督学习算法，深度半监督算法在易用性、泛用性等特点上有所提高，可以结合遥感高光谱图像的固有特点，实现分类任务，并得到适用于遥感图像分类的深度半监督模型。从而，能够充分利用标记数据，实现对遥感图像的高效分类，并分析此类模型的有效性。

1.3 全文结构

论文第(二)章将首先介绍高光谱成像的原理与其图像的特点。(2.1)节将简单介绍其成像机制，并由此介绍高光谱图像具有“三维数据体”的结构特点。(2.2)节讨论遥感图像中空间分辨率与光谱分辨率之间存在的权衡关系，这是我们在(五)一章中实现模型改进的重要基础。

第(三)章着重于浅层的半监督学习算法。(3.1)一节以多元 Gaussian 分布为基础，分析了半监督学习的有效性，这是本文讨论半监督学习问题的根基。(3.2)一节立足于 SVM 算法，给出了二元有监督的 SVM 简单推导，并将其扩展至多元半监督的 TSVM

模型。在模型的扩展中，运用了修改正则化项的方法，这是我们（五）一章中实现模型改进的重要思路。

(四) 一章介绍了半监督深度学习模型。以前人对神经网络添加有关流形的正则化项的工作为全章脉络，(4.1) 一节重点讨论了数据的流形性质。(4.2) 一节以 (4.1) 中提出的模型为基础，运用 (3.2) 的思路，将神经网络的目标函数扩展为含有基于流形的正则化项的目标函数，从而初步建立了一个宽泛的半监督深度学习模型。

第 (五) 章的主要内容是实验论证。(5.1) 一节以 (2.2) 为基础，沿用 (3.2) 的思路，提出在 (4.2) 的模型基础上再增加有关空间正则化项的想法。(5.2) 一节介绍了算法实现中所运用的深度学习结构，细化了 (5.1) 中提出的模型，并给出了一些实现方面的技巧。(5.3) 一节介绍了三种数据集，给出了一些实验设置，并在三个数据集上运行了半监督深度学习模型，并且与有监督学习算法及其他半监督学习算法进行了对比，从而得出本文所提出的模型具有一定有效性的结论。

(六) 一章总结了模型的特点，并指出了模型的不足之处，意在希望能得到进一步的提高。

第二章 高光谱成像技术

通常认为遥感 (remote sensing) 技术起初于 20 世纪 60 年代在美国兴起。它的科学内涵可以总结为远程观测，其外延却在不断演变与扩大。因此我们很难给按照单一的标准给遥感图像分类，只能从传感器类型、电磁波波段、空间尺度等多个角度来进行。即便如此，我们也可以大致概括现今不同种类的遥感图像的主要特点，其中包括：

- 高光谱分辨率 (Hyperspectral Remote Sensing)
- 高空间分辨率 (Hyperspace Remote Sensing)
- 高时间分辨率 (Hyper-temporal Remote Sensing)

本文主要讨论高光谱图像，因此本章也主要介绍高光谱图像的成像原理以及图像特征。这将为我们在 (五) 章中基于 (四) 章所描述的算法的改进奠定基础。

2.1 成像原理简介

电磁波在真空中总是以光速 c 传播。假定其电场或磁场的振动周期为 T ，波长为 λ ，则频率 v 与波长的关系为：

$$c = \frac{\lambda}{T} = \lambda \cdot v$$

如果按照电磁波频率 v 划分，即得若干波段，形成了电磁波波谱 (Electromagnetic Spectrum)。常见的电磁波波谱如图 (2.1) 所示。

当太阳光的电磁辐射到达地表后，地表物体会吸收、透射、反射太阳辐射，即：

$$E_{\text{太阳辐射}} = E_{\text{反射}} + E_{\text{吸收}} + E_{\text{透射}}$$



图 2.1: 电磁波波谱

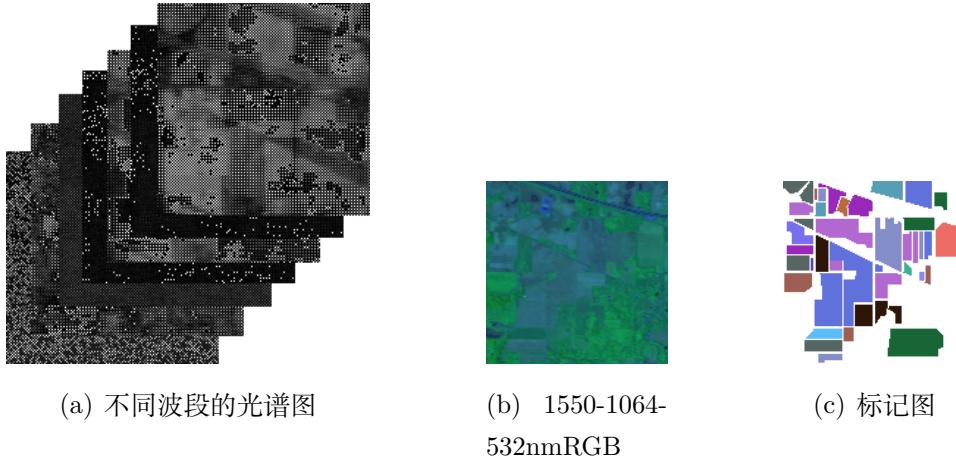


图 2.2: Indian Pines 高光谱图像的数据结构

高光谱的遥感传感器主要依赖于地表物体所反射的不同波段的电磁波进行成像。不同物体对不同波段的电磁波的吸收 $\frac{E_{\text{吸收}}}{E_{\text{太阳辐射}}}$ 与反射 $\frac{E_{\text{反射}}}{E_{\text{太阳辐射}}}$ 能力有差异，因此在连续的光谱波段上呈现出了不同的特征。例如植被场景饱含叶绿素，叶绿素对红、蓝光的吸收能力强，对绿光的反射能力强。因此视觉上植被呈现出绿色，而在反射光谱上表现出对应的特征。水场景则不同，水对蓝、绿的反射能力较强，而对其他频率的电磁波吸收能力强。因此大海、湖泊多呈现蓝绿色，在反射光谱上也自有其特征。但是需要注意的是，其他场景（岩石、公路、建筑等）的反射光谱没有明显规律可循，缺乏直观特征。

值得一提的是，遥感图像能通过研究宏观场景的反射光谱进行分类，而在微观领域，也可以通过分析原子与分子的吸收光谱从而对其物理、化学结构进行判断。可见这是一种具有普适性的方法。

高光谱分辨率的遥感器将场景中反射的电磁波分为数百个狭窄而连续的波段带，即 $[v_i, v_{i+1}]_{i=1, \dots}$ 。一般高光谱图像涵盖了可见光、红外等多个波段。于是，高光谱图像中每个像素都包含了数百个波段的信息（而非如 RGB 图像只含三通道的信息）。

以美国航空局 (National Aeronautics and Space Administration, NASA) 于 1998 年总结的成像光谱仪 AVIRIS(Airborne Visible Infrared Imaging Spectrometer) 为例 [3]。AVIRIS 自 1987 年开始服务。其光谱分辨率为 10nm，它是当时第一台能够以 10nm 间隔测量 400 至 2500nm 范围内太阳反射光谱的图像传感器，其谱段实际数量为 224。这个范围相当于紫外线与可见光频段至微波频段。而它的空间分辨率则是 20m，测量范围可长达 800km。

美国 Purdue 大学的学者曾用 AVIRIS 于 1992 年 6 月 12 日在印第安纳州 (Indiana) 拍摄了一组农场的高光谱数据 [6]，如图 (2.2) 所示，其中 RGB 图像是取 1550nm、1064nm、532nm 三个波段为 R、G、B 值合成的图像。Indian Pines 图像的长、宽均为 145 像素，有 200 个波段。照片包含了大量的农场、森林、公路、房舍等不同场景，它们被分为 16 个不同类别以及无标记像素。

可以看到，高光谱图像整体可以视作三维数据体。高光谱图像在空间上展示了位置等地理信息，在光谱上展示了该像素点的物理、化学性质（通过光学反射来表征）。

2.2 分辨率的权衡

RGB 三通道图像通常只能反应光亮强度，但高光谱图像能够通过各个波段的数值反应更全面的信息。我们可以想象，最理想的图像应该同时具有高空间分辨率（例如我国达到 0.8m 的高分 2 号卫星 (GF-2) 这一类亚米级别的卫星）与高光谱分辨率的特征，这样遥感图像像素能够在标记单纯的条件下具有丰富的特征。可惜的是，这是很难实现的，其原因在于高空间与高光谱分辨率之间存在一种权衡关系 (trade-off)。

如图 (2.3) 所示，遥感图像的空间分辨率可以由瞬时视场 (Instantaneous Field of View, IFOV) 给出。IFOV 是遥感传感器可视的圆锥形场域，其在地表的瞬时投影为地表 IFOV，决定了空间分辨率。显然，IFOV 可以由其张角 (IFOV 角) 与海拔高度所决定：IFOV 角越小，海拔越低，IFOV 越小，空间分辨率越高 [16]。

光谱分辨率则和传感器本身的性质有关。光谱分辨率由波段的宽度所决定：所分波段宽度越窄，光谱分辨率越高，理想情况当然是获得连续光谱。然而现实中传感器由于噪音、设备等问题，是存在探测阈值的。

若图像中某一像素的功率阈值为 P_ϵ ，场景中频率为 v 的电磁波在单位面积 S 上反射的功率恒定为常数 $C = \frac{P_v}{S}$ ，则该频率的波段若要被传感器感知，则应满足： $\frac{P_v}{S} \cdot S_{\text{地表 IFOV}} = C \cdot S_{\text{地表 IFOV}} \geq P_\epsilon$ 。以湖泊为例，水对非蓝、绿波段的可见光反射较弱，若这些波段要被传感器探测，则必须要求增加地表 IFOV。如此一来，则空间分辨率不得不降低。

高空间图像对地表的空间结构表达细致，高光谱的光谱信息丰富，但二者之间的矛盾长期存在于遥感成像技术中。为了解决这个问题，学者们提出了利用图像融合 (Image Fusion) 的方法，Pohl 与 Genderen 在 1998 年对当时的遥感图像融合方法做了回顾 [13]。他们将遥感图像融合分为：像素级 (Pixel Level)、特征级 (Feature Level)、决策级 (Decision Level) 三个层次。Pohl 与 Genderen 回顾了诸多方法，包括 IHS(Intensity-Hue-Saturation) 变换等。遥感图像融合不在本文探讨之列，故相关方法略去不述。

回到高光谱图像上。至此，我们已经大致介绍了高光谱分辨率的遥感图像的数据结构特征，以及空间-光谱分辨率之间的权衡。后者将成为我们在 (五) 章中将现有的半监督深度学习模型 [20] 应用在遥感图像分类任务上的重要基础。



图 2.3: 空间分辨率

第三章 浅层的半监督算法

一般认为半监督学习的研究始于 Purdue 大学的 Shahshahani 与 Landgrebe 的工作 [34]。这几位学者在他们的工作中论证了无标记样本对于提高训练准确率的帮助。同时，这份工作就是在高光谱的遥感数据集上完成的。本章将根据历史的发展，分别介绍生成式方法 (Generative Methods)(3.1) 以及转导支持向量机 (Transductive Support Vector Machine)(3.2)。这几种方法在遥感数据集上都得到了验证，证明半监督学习算法对于遥感图像分类具有良好的效果。

3.1 半监督学习的有效性

Shahshahani 与 Landgrebe 的工作主要集中在 Gaussian 最大似然 (Gaussian Maximum-Likelihood, GML) 分类器上。进一步的成果在 Tadjudin 与 Landgrebe 的工作中展现。他们在 2000 年提出用 EM(Expectation Maximization) 算法对 Gaussian 混合模型 (Gaussian Mixture Model, GMM) 进行优化，并取得了良好的成果 [35]。

在本节，我们借助 GMM 来论证半监督学习的有效性。

多元 Gaussian 分布的定义如 (B.1) 所示。显然，密度函数 $p(\mathbf{x})$ 完全由参数 $\boldsymbol{\mu}$ 与 $\boldsymbol{\Sigma}$ 决定，故记 $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 。在此基础上，有 Gaussian 混合模型 (GMM) 的定义 (B.2)。

给定样本 \mathbf{x} ，其真实类别为 $y \in \mathcal{Y} = \{1, 2, \dots, k\}$ 。若 \mathbf{x} 由 GMM 生成，且每一个类别 $i \in \mathcal{Y}$ 对应一个混合成分 $p(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ ，则样本 \mathbf{x} 的概率密度由式 (B.2) 给出。

令分类器为 f ，则预测 \mathbf{x} 的类别为 $f(\mathbf{x}) \in \mathcal{Y}$ 。定义指示性的参数 $\Phi = i$ 表示 \mathbf{x} 属于混合成分 $p(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ 。则有后验概率：

$$\begin{aligned} p_M(y = j|\mathbf{x}) &= \sum_{i=1}^k p_M(y = j, \Phi = i|\mathbf{x}) \\ &= \sum_{i=1}^k p_M(y = j|\Phi = i, \mathbf{x}) \cdot p_M(\Phi = i|\mathbf{x}) \end{aligned} \tag{3.1}$$

其中 $p_M(y = j|\Phi = i, \mathbf{x})$ 与标记 $y = j$ 相关，为样本 \mathbf{x} 由第 i 个成分生成而类别为

j 的概率。则有：

$$p_M(y = j|\Phi = i, \mathbf{x}) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \quad i, j \in \{1, 2, \dots, k\} \quad (3.2)$$

$p_M(\Phi = i|\mathbf{x})$ 是标记无关项。且由 Bayes 定理 (C.1) 可知， \mathbf{x} 由第 i 个混合成分生成 (也即 $\Phi = i$)，其后验概率为：

$$p_M(\Phi = i|\mathbf{x}) = \frac{\alpha_i \cdot p_M(\mathbf{x}|\Phi = i)}{p_M(\mathbf{x})} = \frac{\alpha_i \cdot p(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i)}{\sum_{i=1}^k \alpha_i \cdot p(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i)} \quad (3.3)$$

对于分类器 f ，考虑最大化后验概率：

$$\begin{aligned} f(\mathbf{x}) &= \arg \max_{j \in \mathcal{Y}} p_M(y = j|\mathbf{x}) \\ &= \arg \max_{j \in \mathcal{Y}} \sum_{i=1}^k p_M(y = j|\Phi = i, \mathbf{x}) \cdot p_M(\Phi = i|\mathbf{x}) \end{aligned} \quad (3.4)$$

由此可见，如果 f 接收了大量无标记样本，式 (3.4) 的估计或许会更加准确。至于如何求解式 (3.4)，也即估计模型参数 $\Theta = \{(\alpha_i, \boldsymbol{\mu}_i, \Sigma_i)\}_{i \in \{1, 2, \dots, k\}}$ ，则由 EM 算法 [15] 给出。

这个方法被 Tadjudin 与 Landgrebe 运用在高光谱遥感图像的分类中，并取得了良好的效果 [35]。

3.2 TSVM 模型

有标记的数据集常常不大，因此存在着过拟合、欠拟合等风险，于是分类效果变得很差。转导支持向量机 (TSVM) 最初是为了解决这一类问题而被提出的。在 1999 年，Joachims 解决了 TSVM 的二次优化问题 [21]。同时，理论与实验均证明了模型的有效性。最初，TSVM 模型是针对二分类问题的学习方法。Bruzzone 等在 2006 年将它扩展为了多分类的模型 [11]。

3.2.1 二分类的 SVM 模型

二分类 SVM 模型在 1995 年被 Cortes 与 Vapnik 提出，用于解决文本分类问题 [11]。模型希望通过超平面：

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (3.5)$$

将样本空间 $\mathcal{X}_l \subset \mathbb{R}^n$ 依据类别标记 $\mathcal{Y} = \{-1, +1\}$ 分隔开 (其中令 $l = |\mathcal{X}_l|$, 为标记样本数量)。也即, 一个良好的超平面应该满足:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \Leftrightarrow \begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq +1, & y_i = +1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1, & y_i = -1 \end{cases} \quad \forall i \in \{1, 2, \dots, l\} \quad (3.6)$$

依据几何直觉, 超平面应该尽量处于两类样本的中间位置。则距离超平面最近的样本点应该使式 (3.6) 取等号。这一对异类样本被称为支持向量 (support vector), 它们到超平面的 Euclidean 距离为:

$$\frac{|\mathbf{w}^T \mathbf{x} + b|}{\sqrt{\mathbf{w}^T \mathbf{w}}} = \frac{1}{\sqrt{\mathbf{w}^T \mathbf{w}}} \quad (3.7)$$

而它们相互之间的距离 $\frac{2}{\sqrt{\mathbf{w}^T \mathbf{w}}}$ 被称为间隔 (margin)。一个好的超平面为了平衡划分, 并且增强模型的鲁棒性, 应该取最大间隔:

$$\max_{\mathbf{w}, b} \frac{2}{\sqrt{\mathbf{w}^T \mathbf{w}}} \Leftrightarrow \min_{\mathbf{w}, b} \frac{\mathbf{w}^T \mathbf{w}}{2} \quad \text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall i \in \{1, 2, \dots, l\} \quad (3.8)$$

考虑到模型的泛化性, 应该允许一些样本不完全满足约束条件。为此, 引入松弛变量 (slack variables) $\xi_i \geq 0$ 。则优化问题变为:

$$\begin{aligned} & \min_{\mathbf{w}, b} \left\{ \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^l \xi_i \right\} \\ & \text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \quad \xi_i \geq 0, \quad \forall i \in \{1, 2, \dots, l\} \end{aligned} \quad (3.9)$$

也就是软间隔支持向量机 (Soft-margin SVM)。

通常对式 (3.9) 采用 Lagrange 乘子法求其对偶问题得解。其变换推导过程在附录 (D.1) 中。从式 (D.5) 中解出 \mathbf{w}, b, ξ 的方法有很多, 其中一个著名的代表是 SMO (Sequential Minimal Optimization) 算法 [30]。

3.2.2 多分类的 TSVM 模型

在二分类 SVM 的基础上, Joachims 在 1999 年推导出了半监督的 SVM 算法 [21]。

和 (3.1) 一样, 我们令: $D_l = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l)\}$ 表示标记样本集, $D_u = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \dots, \mathbf{x}_{l+u}\}$, 表示无标记样本, 其中 $l + u = |\mathcal{X}| = m$ 。仅凭借 D_l , 通过 (3.2.1) 节, 我们可以训练得到一个有监督的 SVM 模型: f^* 。根据 f^* , 可以对 D_u 进行预测, 从而得到它的“伪标记” (pseudo-label): $\mathbf{y}^* = (f^*(\mathbf{x}_{l+1}), f^*(\mathbf{x}_{l+2}), \dots, f^*(\mathbf{x}_{l+u}))$ 。

得到伪标记 \mathbf{y}^* 后, 如式 (3.9), 添加 D_u 与 \mathbf{y}^* , 则可以得到:

$$\begin{aligned} & \min_{\mathbf{w}, b} \left\{ \frac{\mathbf{w}^T \mathbf{w}}{2} + C_l \sum_{i=1}^l \xi_i + C_u \sum_{i=l+1}^{l+u} \xi_i \right\} \\ \text{s.t. } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, l \\ & y_i^*(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = l+1, l+2, \dots, l+u \\ & \xi_i \geq 0, \quad \forall i \in \{1, 2, \dots, l+u\} \end{aligned} \quad (3.10)$$

其中 C_l 与 C_u 是一组常数参数。式 (3.10) 引入 \mathbf{y}^* 后, 可以将原本的无标记样本 D_u 视作有标记样本, 其标记即是伪标记。训练好式 (3.10), 即得到 TSVM 模型: f 。

直接求解式 (3.10) 显然是不合理的。因为 f^* 对于 D_u 的预测 (也即 \mathbf{y}^*) 很可能是不准确的。为了解决这个问题, 初始时必须设置 $\frac{C_u}{C_l} \rightarrow 0$ 以降低 \mathbf{y}^* 对整体优化的影响。此后, 需要不断迭代, 使伪标记 \mathbf{y}^* 尽可能合理, 在此过程中使 $\frac{C_u}{C_l} = r$ 。其中 r 是常数, 可以设定为 1, 或是其他平衡性的数值。

至于如何判断使 \mathbf{y}^* 尽可能合理, Joachims 在 [21] 中提出了一个策略: 对于无标记的样本 \mathbf{x}_i 与 \mathbf{x}_j , 若它们异类: $y_i^* \cdot y_j^* = -1$, 且松弛系数满足 $\xi_i^* + \xi_j^* > 2$, 则认为是分类可能是错误的, 交换 y_i^* 与 y_j^* : $y_i^* = -y_i^*, y_j^* = -y_j^*$ 。个中理由十分简单, 由式 (3.10), 应有: $\xi_i^* + \xi_j^* \geq 2 - (y_i^* \mathbf{w}^T \mathbf{x}_i + y_j^* \mathbf{w}^T \mathbf{x}_j + b(y_i^* + y_j^*))$ 。不失一般性, 令 $y_i^* = +1, y_j^* = -1$, 则可以得到:

$$\xi_i^* + \xi_j^* \geq 2 - \mathbf{w}^T (\mathbf{x}_i - \mathbf{x}_j) \quad (3.11)$$

如图 (3.1) 所示, 注意到 \mathbf{w} 是超平面的法向量, 方向由负类指向正类; $\mathbf{x}_i - \mathbf{x}_j$ 是由负类指向正类的向量, 与 \mathbf{w} 夹角小于 $\frac{\pi}{2}$ 。因此, $\mathbf{w}^T (\mathbf{x}_i - \mathbf{x}_j) > 0$ 。考虑到不等式 (3.11) 取等的情况, 应修改 $\mathbf{x}_i - \mathbf{x}_j$ 的方向, 也即交换 y_i, y_j 。

在迭代中, t 时的无标记系数为 $C_{u,t}$, 伪标记为 \mathbf{y}_t^* 。通过上述的策略改进伪标记至于 \mathbf{y}_{t+1}^* , 再增加 C_u (例如成倍增加): $C_{u,t+1} = 2C_{u,t}$, 直至于满足条件 $\frac{C_u}{C_l} = r$ 即可。

至于多分类的情况, 则更为复杂, 但大体上和有监督的情形类似。多分类的 SVM 很早就被提出了。Jason Weston 在 1998 年的一篇综述中总结了两种策略 [19]:

- **One-against-All**

OAA 策略将构造 k 个 SVM 分类器 $\{f_i | i = 1, \dots, k\}$, 第 i 个分类器的超平面将 i 类与其他 $k-1$ 个类 $\mathcal{Y} - \{i\}$ 分离开。

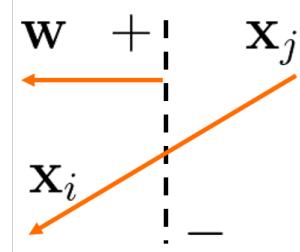


图 3.1: 对于伪标记合理的判断

- **One-against-One**

OAO 策略则构造 $\binom{k}{2} = \frac{k(k-1)}{2}$ 个分类器 $\{f_{i,j} | i \neq j; i, j = 1, \dots, k\}$, 它们的超平面则遍历地区分 $i, j \in \mathcal{Y}$ 。

这个方法被延续到多分类 TSVM 的构造中。我们既可能采用 OAA 策略, 构造 k 个二分类的 TSVM 分类器, 并最终对 k 个分类器的结果进行投票, 选出预测结果; 也可能采用 OAO 策略, 构造 $\binom{k}{2}$ 个分类器。不过, 正如 Bruzzone 在 2006 年的相关工作中指出的, 我们实际上只能采用 OAA 策略构造模型 [11]。

对于一个输入 $\mathbf{x} \in \mathcal{X}$, 假定其真实标记为 $y \in \mathcal{Y}$ 。如果采用 OAO 策略, 则每个 TSVM 给出的结果 $f_{i,j}(\mathbf{x})$ 判断了 $y = i$ 或 $y = j$ 。这对于有监督模型是没问题的, 但对于无标记样本而言, 每个 TSVM 不应仅仅给出 i, j 的判断, 还应该判断一切剩余类别 $\mathcal{Y} - \{i\}$ 的可能性。否则无标记样本会被 TSVM 判定为无法分类。如此一来, 我们也只能够采用 OAA 策略进行构造了:

$$y = \arg \max_{i \in \mathcal{Y}} \{f_i(\mathbf{x})\} \quad (3.12)$$

Bruzzone 的工作验证了多分类 TSVM 在遥感图像分类中的有效性 [11]。

第四章 半监督深度学习

在基于神经网络的深度学习方法兴起以后，其半监督学习方面也有所进展。Weston Jason 在 2012 年的工作考察了传统的浅层学习方法，总结了其嵌入 (embedding) 的原理，并将其运用在神经网络的结构中，取得了良好的效果 [20]。本章将以 Jason 的工作为主要依托，阐明其工作原理，并在此基础上展示模型在遥感图像分类任务上的改动。为此，本章将首先介绍流形嵌入 (manifold embedding) 原理 (4.1)，再结合 Jason 的工作，讨论流形嵌入如何与深度学习结合在一起 (4.2)。

4.1 流形嵌入

4.1.1 分析学观点

从数学分析的角度看，流形 (manifold) 与同胚 (homeomorphism) 的概念紧密相连。在本节主要简单地介绍一下微分同胚、微分流形，这方便接下来的概念理解。

在俄罗斯著名数学家 Zorich 的著作《数学分析》中，微分同胚的概念紧随着隐函数定理 [41]。作为微分学的主要成就之一，隐函数定理内容非常丰富，其推论包括了反函数定理。而微分同胚的概念即为反函数定理服务而被引入。微分同胚的定义见 (B.3)，在此基础上，可以定义微分流形 (B.4)。

例如二维曲线 $\{(x, y) | x^2 + y^2 = 1; x, y \in \mathbb{R}\}$ 是一维流形，同胚于 \mathbb{R}^1 : $x = \sin \phi$, $y = \cos \phi$ ，参数 ϕ 构成了直线空间 \mathbb{R}^1 。而此曲线的每一个局部都可以近似为 Euclidean 空间 \mathbb{R}^1 。

由定义 (B.3)、(B.4) 及此例可见，两个同胚的流形在邻域 (局部) 中具有相同的结构，它们只是相差一个可微的坐标变换，但是在表达方式上却存在着繁简之分。这带来的启发是：如果样本集合 \mathcal{X} 在一个 n^* 维流形 $\mathcal{M} \subset \mathbb{R}^n$ 上 (也即 \mathcal{M} 嵌入 (embed) 在 \mathbb{R}^n 中)，则在低维的 \mathbb{R}^{n^*} 空间中寻找对应的集合 \mathcal{Z} ，使其中的点 $\mathbf{z}_i \in \mathcal{Z}$ 能够表示样本集合中的点 $\mathbf{x}_i \in \mathcal{X}$ ，也即 \mathbf{x}_i 与 \mathbf{z}_i 的邻域相同胚。

4.1.2 LapSVM

流形嵌入在统计学习方面的应用，常常集中在降维与度量等方面。不过，实际上它也和半监督学习关系密切。本节介绍 Laplacian 支持向量机 (Laplacian SVM, LapSVM)，

它将为我们在下一节理解深度学习框架奠定非常重要的基础。

LapSVM 最初由 Belkin 于 2006 年提出 [8]。有别于 TSVM，LapSVM 虽然也是半监督学习模型，不过它是基于流形嵌入的。到此为止，我们已经展现了一条完整的模型演变路线：最初，学者们借助 GMM 模型论证了半监督学习的有效性；接着，二分类的 SVM 作为一个强力工具被提出，并且发展出了 TSVM，以及结合了流形嵌入的 LapSVM。不仅如此，在深度学习兴起以来，基于流形嵌入的神经网络结构也被提出。

在 (3.2.2) 一节中，我们给出了 TSVM 的优化过程，式 (3.10) 是基于参数 \mathbf{w} 与 b 的。实际上优化过程有更一般的形式：

$$f^* = \arg \min_{f \in \mathcal{H}_\kappa} \left\{ \Omega(\Theta) + C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) \right\} \quad (4.1)$$

其中 \mathcal{H}_κ 是用核方法表示 SVM 的再生核希尔伯特空间 (B.6)； $loss(\cdot)$ 是损失函数，描述 f 与标记样本 \mathcal{X}_l 的契合程度； $\Omega(\Theta)$ 是正则化项 (regularization)，描述了 f 的性质。如果我们用 hinge 损失： $loss_H(f(\mathbf{x}), y) = \max\{0, 1 - y \cdot f(\mathbf{x})\}$ 来描述式 (3.10)，则有：

$$\min_{\mathbf{w}, b} \left\{ \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^l loss_H(1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \right\} \quad (4.2)$$

正则化的数学内涵非常丰富，它的历史可以追溯至苏联科学家 Tikhonov 在 1963 研究不适定问题 (ill-posed problem) 所做的工作 [36]。其后，正则化贯穿了机器学习算法，SVM 也可以通过式 (4.1) 看作是正则化的一个例子。

在式 (3.10) 中， $\Omega(\Theta) = \frac{\mathbf{w}^T \mathbf{w}}{2}$ 被称作 L^2 正则化项，它描述了支持向量间隔的大小。之所以以它为优化对象，是由我们在 \mathbb{R}^n 中的几何直觉使然。Belkin 在 LapSVM 的工作中即着手于这一点，他采用了不同的正则化方式 (这一点有别于 TSVM)，将之扩展为：

$$\Omega(\Theta) = \frac{\mathbf{w}^T \mathbf{w}}{2} + \gamma \|f\|_I^2 \quad (4.3)$$

其中 $\frac{\mathbf{w}^T \mathbf{w}}{2}$ 本质上反应了数据 \mathcal{X} 、模型 f 与环境 \mathbb{R}^n 三者的相互作用。如果样本集合 \mathcal{X} 在某一 n^* 维流形 \mathcal{M} 上， $\|f\|_I^2$ 可以是沿着 \mathcal{M} 惩罚 f 的惩罚项。也即是说， $\|f\|_I^2$ 是基于流形环境 \mathcal{M} 的正则化项。 γ 为正则化项的系数。

我们对比一下 TSVM 模型 (式 (3.10))，可以看到：TSVM 将无标记样本 D_u 放在了损失函数中，而 LapSVM 则依据 \mathcal{X} 整体的流形性质，将 D_u 作为正则化项。这是两类模型主要的不同之处。

Laplacian Eigenmaps

至于如何构造 $\|f\|_I^2$ ，则依赖于 Laplace-Beltrami 算子。这里已经涉及到了微分几何中 Riemannian 流形的概念，为了方便理解，我们只给出它在 Euclidean 空间中的定义 (B.5)，并只考虑它的离散近似，也即是 Laplacian 矩阵与 Laplacian Eigenmaps。

Belkin 的工作具有连贯性，他在 2003 年即为了数据降维研究了 Laplacian Eigenmaps[7]，并最终将之运用在 LapSVM 中。我们在附录中给出了 Laplace-Beltrami 算子的相关命题，这将有助于我们理解 Laplacian Eigenmaps 的几何性质，并最终理解 LapSVM 的正则化，也即 $\|f\|_I^2$ 的构造。

命题 (C.4) 中的不等式 (C.4) 给出了结论：相邻点函数值之差 $|f(\mathbf{s}) - f(\mathbf{t})|$ 可由梯度 $\|\nabla f(\mathbf{s})\|$ 估计。因此，在数据降维时，如果想要保证相邻点经过 f 降维以后也相邻，也即 f 具有保相邻性，则可以直接优化流形 \mathcal{M} 上的积分：

$$f^* = \arg \min_f \left\{ \int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2 \right\} \quad (4.4)$$

求解式 (4.4) 需要求梯度的散度，因此 Laplace-Beltrami 算子作用于斯。虽然 Riemannian 流形上的情况比较复杂，但我们可以求相对简单的离散情形。

回顾式 (4.4)，我们之所以设计这样的优化，是因为希望 f 具有保相邻性，也即原本相邻的点经过映射依然相邻。为此，我们先为样本空间 $\mathcal{X} \subset \mathbb{R}^n$ 构造邻接矩阵 \mathbf{W} 。定义样本点 \mathbf{x}_i 的邻域为： $U(\mathbf{x}_i) = \{\mathbf{x}_j \in \mathcal{X}, \|\mathbf{x}_j - \mathbf{x}_i\| \leq \epsilon \in \mathbb{R}^+\}$

$$W_{ij} = \begin{cases} 1, & \mathbf{x}_j \in U(\mathbf{x}_i) \\ 0, & \mathbf{x}_j \notin U(\mathbf{x}_i) \end{cases} \quad (4.5)$$

设待求映射为 $f(\mathbf{x}_i) = \mathbf{z}_i \in \mathbb{R}^{n^*}, i = 1, \dots, m$ 。如此，以保相邻性为目的的优化目标非常清晰：

$$\min_{\mathbf{z}} \left\{ \sum_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|^2 W_{ij} \right\} \quad (4.6)$$

式 (4.6) 实质上就是式 (4.4) 的离散形式：邻接矩阵 \mathbf{W} 上求和对应流形 \mathcal{M} 上积分，差分 $\|\mathbf{z}_i - \mathbf{z}_j\|^2$ 对应于梯度 $\|\nabla f(\mathbf{x})\|^2$ 。求解式 (4.6) 的过程在推导 (D.2) 中。

式 (4.4) 与 (4.6) 的动机非常明确：相邻的点降维后也相邻。因此，LapSVM 的流形正则化项 $\|f\|_I^2$ 也依据这个原则构造：

$$\|f\|_I^2 = \sum_{i,j=1}^m (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} \quad (4.7)$$

这样一来，我们终于构造了 LapSVM 模型的正则化项：

$$\Omega(\Theta) = \frac{\mathbf{w}^T \mathbf{w}}{2} + \gamma \sum_{i,j=1}^m (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} \quad (4.8)$$

4.2 基于流形嵌入的半监督深度学习

如果我们要将无监督数据运用在深度神经网络模型中，则可以参考 SVM 的扩展过程。TSVM 将通过“伪标记”，将无标记样本融入损失函数 $= loss(f(\mathbf{x}), y)$ 中，其表现

为一系列对应的松弛变量 $\{\xi_i\}_{i=l+1,\dots,u}$ 。LapSVM 则依据数据整体的几何性质，将无标记样本融入正则化项 $\Omega(\Theta)$ 。显然，后者的构造方式更具有普遍性，容易被运用在其他模型中。Jason 正是利用了这一点，他在 2012 年即基于此构造了半监督的深度学习模型 [20]。

我们可以看到，贯穿 (4.1.2) 一节的假设是保相邻性：原本相邻的样本在低维空间也相邻。这与大部分半监督学习的假设不谋而合：同一结构（聚类、流形）中的样本有可能具有相同的标记。从这个观点看，无标记样本虽然无益于标记本身，但有助于补全数据的分布。因此，在正则化项方面做文章是合理的。

如给出 $\Omega(\Theta)$ 中流形正则化的一般形式，可以概括为：

$$\sum_{i,j=1}^m R(g(\mathbf{x}_i), g(\mathbf{x}_j), W_{ij}) \quad (4.9)$$

和式 (4.6) 一样，我们用邻接矩阵上的求和表示流形上的积分。 $R(\cdot)$ 表示某种正则化函数， g 的是有关流形的映射，它未必和模型 f 相同。

4.2.1 output 模型

我们可以根据 Jason 的工作构建半监督的深度学习模型。不失一般性，在这里我们直接用全连接层表示各层。

一个简单的 output 模型则如图 (4.1) 所示。其中 \mathbf{x} 指输入； $S_i(\cdot)$ 是第 i 层的激活函数； \mathbf{w}_i ， \mathbf{b}_i 是对应的网络参数； $h_i(\cdot)$ 指第 i 层的隐藏层结果。

图 (4.1) 中的网络只有三层全连接，但足以展现模型的关键之处。输出 $f(\cdot)$ 是通过对最后的隐藏层 $h_3(\cdot)$ 激活 $S_O(\cdot)$ 得到的。流形嵌入 (embedding)，也即降维，则通过映射 $g : \mathbf{x} \mapsto \mathbf{w}_3^T h_2(\mathbf{x}) + \mathbf{b}_3 \in \mathbb{R}^k$ 实现。

根据式 (4.1) 所提到的一般性目标函数的构造原则，我们可以得到 output 模型的目标函数：

$$\begin{aligned} & C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \Omega(\Theta) \\ & = C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \gamma \sum_{i,j=1}^m R(g(\mathbf{x}_i), g(\mathbf{x}_j), W_{ij}) \end{aligned} \quad (4.10)$$

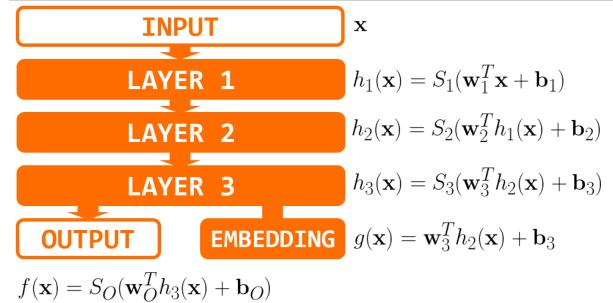


图 4.1: output 模型

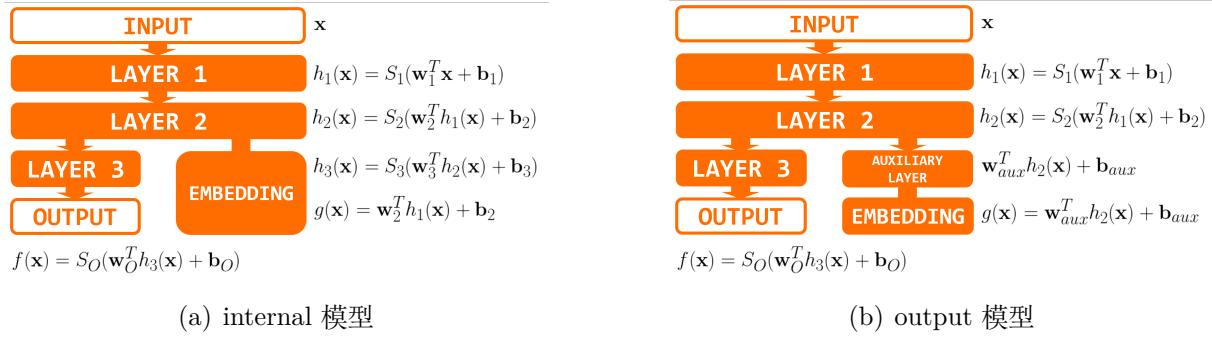


图 4.2: internal 与 auxiliary 模型

可以看到，这个模型和浅层的模型 (LapSVM) 的目标函数是非常相似的。它的目标函数并不复杂，显然可以采用反向传播法 (Back Propagation[14]) 通过梯度下降进行求解。不失一般性，以 \mathbf{w}_3 为例，其推导过程在 (D.3) 中。

4.2.2 internal 与 auxiliary 模型

internal 与 auxiliary 模型如图 (4.1) 所示。它们与 output 模型最主要的区别即在于降维映射的不同。

- internal 模型

internal 模型的映射为 $g : \mathbf{x} \mapsto \mathbf{w}_2^T h_1(\mathbf{x}) + \mathbf{b}_2 \notin \mathbb{R}^k$

- auxiliary 模型

internal 模型的映射为 $g : \mathbf{x} \mapsto \mathbf{w}_{aux}^T h_2(\mathbf{x}) + \mathbf{b}_{aux} \notin \mathbb{R}^k$

从结构上看，internal 模型只截取了前若干层网络作为流形的映射 $g(\cdot)$ 。auxiliary 模型另外增加了一些线性层。这些结构的改变使 f 与 g 的差别更加明显，某些情况下可能有助于提高泛化性能。

不仅如此，这套半监督深度学习的框架非常灵活：纵向上我们可以堆叠不同深度、类型的网络，同时，横向上我们也可以合理地增加正则化项。这为我们接下来解决遥感图像分类问题提供了非常大的改动空间。

Jason 等在多个数据集上检测了模型的有效性，包括：小规模的文本数据 g50c[28]、MNIST 手写数据 [38]、Columbia 大学的目标检测数据 COIL100[27] 等，均取得了不错的效果 [20]。

第五章 应用与实现

根据 (2.2) 一节的讨论，我们已知高光谱图像中存在着光谱与空间分辨率的权衡。我们将根据这一特点，在 (5.1) 一节中按照 (4.2) 一节所展示的思路对模型进行改进，使它适用于高光谱的分类任务。在 (5.2) 一节中，我们将展现该模型是如何基于 tensorflow 实现的，并讨论一些实现的细节。最后，在 (5.3) 一节中，我们会给出不同数据集上的测试结果。

5.1 有关空间的正则化

注意我们在构造流形正则化项时，采用的方法是在邻接矩阵 \mathbf{W} 进行求和。而 \mathbf{W} 则由式 (4.5) 给出。这实际上是光谱空间上的邻域： $U_{\text{光谱}}(\mathbf{x}_i) = \{\mathbf{x}_j, \|\mathbf{x}_i - \mathbf{x}_j\| \leq \epsilon\}$ 。但需要注意的是，遥感图像中的像素除了光谱邻域外，还存在着空间上的邻域。

如图 (5.1) 所示，我们可以直接定义某一像素的周边像素为其空间邻域 $U_{\text{空间}}(\mathbf{x}_i)$ 。对于像素 \mathbf{x}_i 与 \mathbf{x}_j ，我们分情况讨论：

1. $\mathbf{x}_j \notin U_{\text{光谱}}(\mathbf{x}_i)$ 且 $\mathbf{x}_j \notin U_{\text{空间}}(\mathbf{x}_i)$

该情形下 \mathbf{x}_j 很可能与 \mathbf{x}_i 不属于同一个类别；

2. $\mathbf{x}_j \in U_{\text{光谱}}(\mathbf{x}_i)$ 且 $\mathbf{x}_j \notin U_{\text{空间}}(\mathbf{x}_i)$

该情形可能大量存在。比如零散的湖泊、森林、建筑，它们在空间上不相邻，但在光谱上相邻；

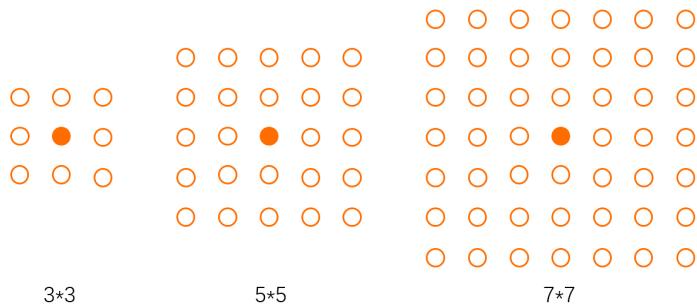


图 5.1: 像素的空间邻域

3. $\mathbf{x}_j \notin U_{\text{光谱}}(\mathbf{x}_i)$ 且 $\mathbf{x}_j \in U_{\text{空间}}(\mathbf{x}_i)$

该情形的数量未必多。显然一张遥感图像在空间上也具有某种“连续性”(否则卷积神经网络(Convolutional Neural Networks, CNN)在池化层选某一代表就失去了意义)，因此空间上相邻的像素有可能是同一类别，它们在光谱上也是相邻的。不过这种情况能够克服仅用光谱邻域的一个缺陷：例如天气情况变化时，同一类别的反射光谱经过云层的吸收后变得不同。但如果它们在空间上相邻，或许还有所补救；

4. $\mathbf{x}_j \in U_{\text{光谱}}(\mathbf{x}_i)$ 且 $\mathbf{x}_j \in U_{\text{空间}}(\mathbf{x}_i)$

这是最理想的情况， \mathbf{x}_j 与 \mathbf{x}_i 很可能属于同一类别。

可以想象，空间正则化项失效的极端情况应如图(5.2)所示，也即标记不存在空间连续性。这种标记混乱的极端情况某种程度上类似于 Dirichlet，处处不连续性。由于高光谱图像的空间分辨率不够高，因此对它的像素采用有关空间的正则化项是合理的。不过必须注意的是：

- 根据上述讨论的第二种情况，我们不宜将空间正则化项的系数设置得过大；
- 考虑到高光谱图像的空间分辨率并不高，因此也不宜将 $U_{\text{空间}}$ 设置得过大。毕竟物理距离越远，属于同一类的可能性越低。
- 在复杂场景，例如都市中，这种方法可能失效。不过考虑到高光谱并不要求高空间，因此高光谱图像本身的应用领域也不在此。

基于上述讨论，我们对式(4.1)进行更激进的扩展：

$$\begin{aligned}
 & C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \Omega(\Theta) \\
 & = C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) \\
 & \quad + \gamma_{\text{光谱}} \sum_{i,j=1}^m R_{\text{光谱}}(g(\mathbf{x}_i), g(\mathbf{x}_j), W_{ij}) + \gamma_{\text{空间}} \sum_{i,j=1}^m R_{\text{空间}}(g(\mathbf{x}_i), g(\mathbf{x}_j), V_{ij})
 \end{aligned} \tag{5.1}$$

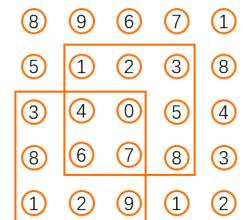


图 5.2: 不连续的空间

其中 V_{ij} 是空间邻域矩阵 \mathbf{V} 中的元素， \mathbf{V} 的定义与式(4.5)相仿。

5.2 基于 tensorflow 的实现

tensorflow 是由谷歌公司 (Google) 于 2016 年公布的机器学习系统 [2]。它通过构建计算图 (computation graph)，方便反向传播的链式求导。tensorflow 能够处理复杂多变的神经网络架构，并且在大规模的情形下运行。keras 是一个高层的深度学习 API，它以 tensorflow 为后端，方便了代码的构建。考虑到我们对正则化项的扩展，其灵活性非常适合我们的任务。

5.2.1 神经网络结构

有一种观点认为神经网络的基础是 Kolmogrov 与 Arnold 于 1957 年发表的一则有关函数逼近的定理 (C.3)，读者可以参考 Funahashi 于 1989 年所发表的文章 [23] 进行更深入的了解。在此基础上，经过数十年的研究，深度学习已经积累了丰富的神经网络结构。

全连接 Fully-Connected

全连接指直接按照定理 (C.3) 所示，对输入向量进行线性变换 (不包括激活) 所得的网络层，它也是我们在 (4.2) 一节中所采用的网络结构：

$$\mathbf{y} = S(\mathbf{w}^T \mathbf{x} + \mathbf{b})$$

这被视作生物学中神经元的一个简单的数学模型，由 McCulloch 与 Pitts 于 1943 年提出 [33]。此模型被称为 MP 神经元模型，并成为了深度学习的滥觞之一。

在 tensorflow 中，全连接网络可以非常容易地被调用，以 python 3 的 tensorflow r1.8 与 keras 为例：

```
tf.contrib.layers.fully_connected
keras.layers.core.Dense
```

卷积网络 Convolutional Networks

卷积神经网络 (Convolutional Neural Networks, CNN) 主要应用在图像处理等领域，它能够处理具有网格结构的数据。CNN 最初由 LeCun 于 1989 年提出 [39]。经过 GPU 与深度学习框架的技术积累，Krizhevsky Alex 于 2012 年在 GPU 上使用 CNN 构建了 AlexNet，并赢得了 ImageNet 比赛 [4]。这一标志性的事件在某种程度上代表了深度学习广泛应用的开端。

CNN 通常包括两个操作：卷积 (convolution) 与池化 (pooling)。卷积操作是数学卷积运算 $\int x(a)w(t-a)da$ 的离散化结果，一般可以用矩阵乘法来表示。池化操作是一种用特征来代表局部的方法，如在 (5.1) 一节中所说的，池化某种程度上利用了图片的语义。

连续性。通常的池化方法包括最大池化 (max pooling)[40] 等。读者可以参考 Goodfellow 与 Bengio 于 2016 年所著的《Deep Learning》[17] 一书，对 CNN 进行更深入的了解。

tensorflow 与 keras 实现 CNN 非常简单：

```
tf.contrib.layers.conv2d
tf.contrib.layers.max_pool2d
keras.layers.convolutional.Conv1D
keras.layers.pooling.MaxPooling1D
```

不过需要注意的是，convolution2d是二维卷积，而像素的光谱是一维数据，因此需要将数据变形为 [1, 224](Indian Pines)。相应的，max_pool2d也需要进行设置。

循环网络 Recurrent Networks

另一种被广泛运用的网络是循环神经网络 (Recurrent Neural Networks, RNN)，它由 Rumelhart 等于 1986 年提出 [14]。RNN 在处理序列数据上具有强大的能力。但是考虑到高光谱图像分类的应用场景，RNN 并不是那么适合，因此不在此赘述。

激活函数 Activation Function

在 Kolmogrov-Arnold 定理 (C.3) 的基础上，有关激活函数的问题也得到了严谨而充分的讨论。Hornik 于 1989 年反思了 Kolmogorov 定理对函数的要求，并由此提出采用具有“挤压”性质的函数 (定义 (B.7)) 的思路 [26]。

Hornik 的研究总结了当时现有的激活函数，并为提出新的激活函数奠定了基础。现今常见的激活函数 (注意，并非皆具有挤压性) 有：

- $\text{sigmoid}(t) = \frac{1}{1+e^{-t}}$
- $\tanh(t) = \frac{2}{1+e^{-2t}} - 1$
- $\text{ReLU}(t) = \max\{0, t\}$

整流线性单元 (Rectified Linear Unit, ReLU) 是现代神经网络常用的激活函数 [24, 37]。

在 tensorflow 与 keras 中，调用激活函数也非常简单：

```
tf.nn.sigmoid
tf.nn.tanh
tf.nn.relu
keras.layers.core.Activation('sigmoid')
keras.layers.core.Activation('tanh')
keras.layers.core.Activation('relu')
```

输出单元

对于多分类的神经网络而言，其输出单元常常用 softmax 来实现：

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j^k e^{x_j}} \quad (5.2)$$

对于 $x_i \rightarrow \pm\infty$ 的上下溢出情况，softmax 的数值稳定在 $[0, 1]$ 区间内，因此很适合用来表示 k 个离散型随机变量的分布。但是 softmax 也存在饱和情况： $\frac{x_1}{x_i} \rightarrow \infty, i = 2, 3, \dots, k$ 时，softmax 数值趋近于 1。

在 tensorflow 与 keras 中，softmax 表示为：

```
tf.nn.softmax
keras.layers.core.Activation('softmax')
```

5.2.2 目标函数

接下来，我们对式 (5.1) 进行更具体的设计。

监督项损失

一些有关神经网络数值计算的研究表明，对于以 softmax 为输出的网络，某些特定的损失函数（如我们在式 (4.10) 中所使用的平方误差）会不起作用 [10]。在 (5.2.1) 一节中，我们说明了 softmax 输出的饱和情况，为了克服这一点，损失函数应该对其进行补偿。为此情况，tensorflow 专门设计了一类交叉熵 (cross entropy)，能够直接以 One-Hot 编码的标记计算多分类的损失。其他损失，例如 (4.1.2) 一节中所提到的 hinge 损失，也可以使用。

```
tf.losses.softmax_cross_entropy
tf.losses.hinge_loss
```

光谱正则化

光谱上，我们可以按照 (4.1.2) 节的讨论，直接使用 Laplacian Eigenmaps 进行构造：

$$R_{\text{光谱}}(g(\mathbf{x}_i), g(\mathbf{x}_j), W_{ij}) = \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|^2 W_{ij} \quad (5.3)$$

```
def Laplacian_Eigenmaps(gi, gj, wij):
    dist = tf.reduce_sum(tf.square(gi - gj), 1)
    w_ij = tf.to_float(wij)
    loss = tf.multiply(w_ij, dist)
    return tf.reduce_mean(loss)
```

空间正则化

对于高光谱图像的像素空间分布，我们也可以要求它具有某种保相邻的性质，如此，也可以使用 Laplacian Eigenmaps，只是邻接矩阵不同：

$$R_{\text{空间}}(g(\mathbf{x}_i), g(\mathbf{x}_j), V_{ij}) = \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|^2 V_{ij} \quad (5.4)$$

其他可用的正则化方式还包括：

- **MDS**

多维尺度变换 (Multidimensional Scaling, MDS) 由 Kruskal 于 1964 年提出 [5]。与 Laplacian Eigenmaps 一样，MDS 试图保证映射不改变相邻性：

$$R(g(\mathbf{x}_i), g(\mathbf{x}_j), V_{ij}) = (\|g(\mathbf{x}_i) - g(\mathbf{x}_j)\| - V_{ij})^2 \quad (5.5)$$

- **Siamese Networks**

Siamese Networks 是 Bromley 于 1994 年提出的神经网络 [18]。在 2006 年，Hadsell 等为它设计了目标函数 [32]：

$$R(g(\mathbf{x}_i), g(\mathbf{x}_j), V_{ij}) = \begin{cases} \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\| & V_{ij} = 1 \\ \max\{0, m - \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|\}^2 & V_{ij} = 0 \end{cases} \quad (5.6)$$

5.2.3 实现技巧

预算计算

考虑到邻接矩阵 \mathbf{W} 与 \mathbf{V} 在优化中不存在更新，因此其计算可以视作数据的预算计算，这样可以大大节省计算时间。在实验中，我们为光谱项设计了两种邻接矩阵，为空间项设计了一种邻接矩阵。

空间项的邻接矩阵可以直接由图 (5.1) 所得。根据我们在 (5.1) 一节中的讨论，将邻域大小设定为 3×3 。

光谱的邻接矩阵可以设计为“最近邻 (Nearest Neighbor)” 的方式。不同于式 (4.5) 用一个阈值 ϵ ，我们直接选择前 10 个距离最接近的样本构成其邻域。在图像的长、宽、波段数都比较大时，这种方法的时间复杂度很高，因此在模型优化时计算 \mathbf{W} 非常不明智。

光谱的邻接矩阵也可以根据“聚类 (Clustering)” 的方式来设计。 k 平均 (k -Means) 是一种经典的聚类算法，在历史上被各界学者不断重新设计 [22]。它能够将数据 \mathcal{X} 分为给定数量的“簇 (cluster)”。在某种程度上，簇也可以视为一种类别，它反应了数据内部的一些特征。因此，可以设计 $U_{\text{光谱}}(\mathbf{x})$ 为该样本所在的簇，从而实现一种新的邻接矩阵。注意，在这里聚类的数量与标记数量 k 未必相同 (实际上很可能聚类的簇数要大于 k)。

sklearn(scikit-learn)[29] 是一个成熟的 python 机器学习库，它提供了一系列的聚类算法接口，能够高效地实现我们的目的：

```
sklearn.cluster.KMeans
```

預計算完成以后，将结果保存在本地文件，待训练时直接读取，则可以大幅提升训练速度。

归一化

回顾 (5.2.1) 中所讨论的激活函数。由于很多激活函数具有“挤压”性质，因此在输入 $t \rightarrow \infty$ 过大时，激活函数会饱和 $S(t) \rightarrow C$ 。例如有 $\lim_{t \rightarrow \infty} \text{sigmoid}(t) \rightarrow 1$ ，而其后果即是神经网络可能对任意输入都输出相同的结果。

为了避免这种情况，应该对输入数据进行归一化 (normalization) 处理。常用的归一化方法有：

1. min-max 归一化: $t \mapsto \frac{t - \min}{\max - \min}$ ，这种方法将数据归一化到区间 $[0, 1]$ 。
2. Z-Score 归一化: $t \mapsto \frac{t - \mu}{\sigma}$ ，其中 μ 为样本均值， σ 为样本方差。

随机策略

回顾式 (5.1)，可以看到我们对整个数据集进行了求和： $\sum_{i=1}^l$ 与 $\sum_{i=1}^m$ 。在具体实现中，由于计算瓶颈已经通过預計算得到解决，因此可以用“随机 (stochastic)” 的方式进行，如算法 (1) 所示。

关键代码

前馈网络 在实现模型时，我们用 python 的 list 保存 keras 的网络层。这样做是为了方便管理和扩展不同的网络，方便 internal、auxiliary 模型的实现。以 Indian Pines 数据集为例：

Listing 5.1: internal 模型的 embedding 过程

```
indian_g = [
    Reshape((200, 1)),
    Conv1D(kernel_size=2, filters=128),
    Activation('sigmoid'),
    MaxPooling1D(pool_size=2),
    Conv1D(kernel_size=2, filters=64),
    Activation('sigmoid'),
    MaxPooling1D(pool_size=2),
    Flatten()
]
```

Result: 训练好的模型 f

输入标记数据 $D_l = \{(\mathbf{x}_i, y_i)\}_{i=1,\dots,l}$, 无标记数据 $D_u = \{\mathbf{x}_j\}_{j=l+1,\dots,m}$, 构造神经网络 f 与嵌入函数 g ;

while 未满足训练终止条件 **do**

- 随机取一组标记样本 $(\mathbf{x}_a, y_a) \in D_l$;
- 对损失 $loss(f(\mathbf{x}_a), y_a)$ 进行一次梯度下降优化;
- 随机取一对光谱相邻的样本 $\mathbf{x}_b, \mathbf{x}_c \in U_{\text{光谱}}(\mathbf{x}_b)$;
- 对正则项 $\lambda_{\text{光谱}} R(g(\mathbf{x}_b), g(\mathbf{x}_c), W_{ij} = 1)$ 进行一次梯度下降优化;
- 随机取一个无标记样本 $\mathbf{x}_d \in D_u$;
- 对正则项 $\lambda_{\text{光谱}} R(g(\mathbf{x}_b), g(\mathbf{x}_d), W_{ij} = 0)$ 进行一次梯度下降优化;
- 随机取一对空间相邻的样本 $\mathbf{x}_e, \mathbf{x}_f \in U_{\text{空间}}(\mathbf{x}_e)$;
- 对正则项 $\lambda_{\text{空间}} R(g(\mathbf{x}_e), g(\mathbf{x}_f), V_{ij} = 1)$ 进行一次梯度下降优化;
- 随机取一个无标记样本 $\mathbf{x}_g \in D_u$;
- 对正则项 $\lambda_{\text{空间}} R(g(\mathbf{x}_e), g(\mathbf{x}_g), V_{ij} = 0)$ 进行一次梯度下降优化;

end

Algorithm 1: 半监督神经网络

Listing 5.2: internal 模型的输出

```
indian_f = indian_g + [
    Dense(128, activation='sigmoid'),
    Dense(32, activation='sigmoid'),
    Dense(16, activation='softmax')
]
```

如此, 前馈网络即可以通过一个循环实现:

Listing 5.3: 前馈网络的外层封装

```
def forward(name, x):
    assert name in {
        'indian_f', 'indian_g',
    }, 'ERROR: wrong neural network name'
    layerlst = {
        'indian_f': indian_f,
        'indian_g': indian_g,
    }
    y = x
    for layer in layerlst[name]:
        y = layer(y)
    return y
```

目标函数 完成forward函数后, 可以简单地调用各个网络实现算法 (1):

Listing 5.4: 目标函数

```

xa = tf.placeholder(tf.float32, input_dim)
ya = tf.placeholder(tf.float32, output_dim)
xb = tf.placeholder(tf.float32, input_dim)
xc = tf.placeholder(tf.float32, input_dim)
xd = tf.placeholder(tf.float32, input_dim)
xe = tf.placeholder(tf.float32, input_dim)
xf = tf.placeholder(tf.float32, input_dim)
xg = tf.placeholder(tf.float32, input_dim)

fa = forward(dconf['indian_f'], xa)
gb = forward(dconf['indian_g'], xb)
gc = forward(dconf['indian_g'], xc)
gd = forward(dconf['indian_g'], xd)
ge = forward(dconf['indian_g'], xe)
gf = forward(dconf['indian_g'], xf)
gg = forward(dconf['indian_g'], xg)

loss_ = loss_func('softmax')(fa, ya)
    + gamma_1 * (reg_func('Laplacian')(gb, gc, 1.0)
                  + reg_func('Laplacian')(gb, gd, 0.0))
    + gamma_2 * (reg_func('Laplacian')(ge, gf, 1.0)
                  + reg_func('Laplacian')(ge, gg, 0.0))

```

loss_ 即为梯度下降法的优化对象。其中loss_func与reg_func均为外层封装函数，用于调用损失与正则化的计算。

5.3 不同数据集上的测试

5.3.1 数据集介绍

Indian Pines Indian Pines 数据集已在 (2.1) 中介绍，在此不赘述。

Pavia University Pavia University 数据集是由意大利 Pavia 大学的 Gamba 教授用 ROSIS 遥感器所拍摄的其大学的高光谱数据集。

ROSI(SReflective Optics System Imaging Spectrometer) 传感器 [25] 的瞬时视场角约为 8° ，光谱范围为 $[430, 860]\text{nm}$ ，光谱带数为 115，光谱分辨率为 4.0nm 。Pavia University 数据集由 Gamba 于 2003 年拍摄 [31]。其图像大小为 610×340 像素，空间分辨高达 1.3m ，有 103 个波段，是高光谱高空间分辨率 (HSSR) 的遥感图像。以 830nm 的电磁波强度为 R 值、 630nm 为 G 值、 430nm 为 B 值，则 RGB 图像如图 (5.3(b))。Pavia University 数据集有 9 个类别以及未标记，包括：沥青 (asphalt)、草坪、沙砾、树木、着色金属板、土壤、柏油 (bitumen)、砖石、阴影，如图 (5.3(a)) 所示。

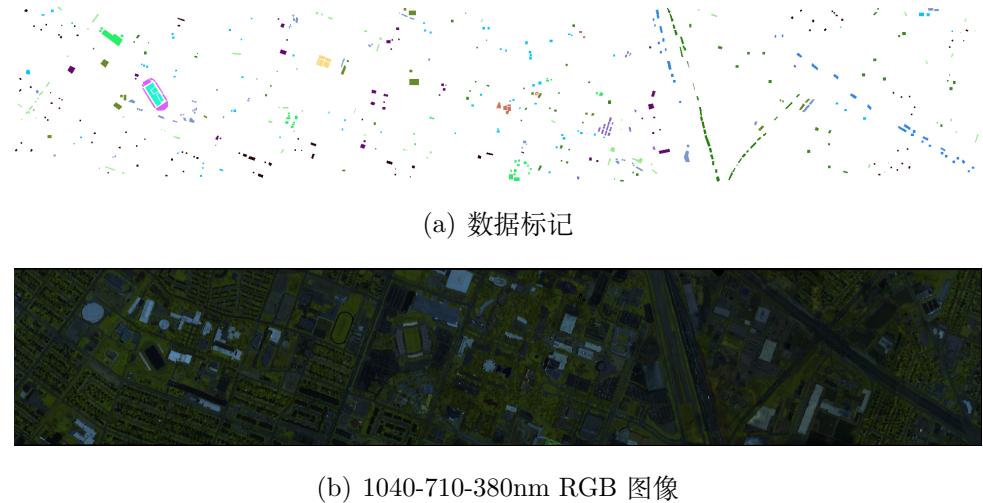


图 5.4: Houston 数据集

Houston Houston 高光谱遥感数据集是由 Houston 大学的 NCALM(National Center for Airborne Laser Mapping) 于 2012 年 6 月所拍摄, 为 2013 IEEE GRSS Data Fusion Contest 比赛数据¹, 其内容是 Houston 大学与周边的环境。

数据集的图像大小为 349×1905 。其光谱范围为 $[380, 1050]\text{nm}$, 波段数为 144, 像素被分为 15 类, 空间分辨率为 2.5m , 截取 $R=1040\text{nm}$, $G=710\text{nm}$, $B=380\text{nm}$, 则可得到如图 (5.4(b))

所示的 RGB 图像, 其数据包括 15 种类别以及无标记样本: 青草、枯草、混合草、树、土壤、水、民宅、商业建筑、马路、公路、铁路、停车场 1、停车场 2、网球场、跑道, 如图 (5.4(a)) 所示。

三个数据集的相关信息如表 (5.1) 所示。

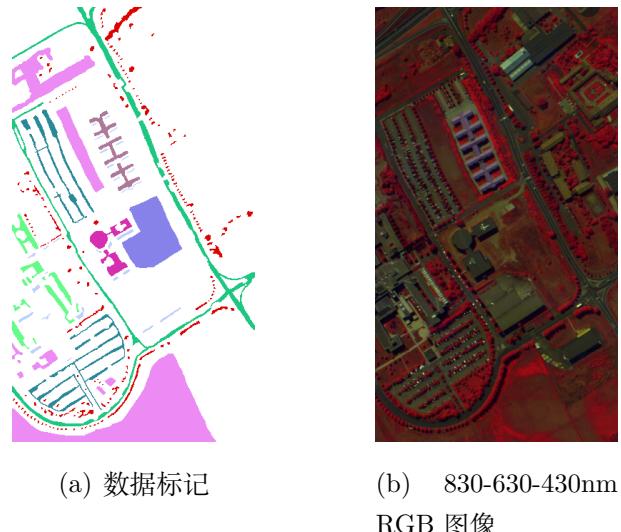


图 5.3: Pavia University 数据集

¹http://hyperspectral.ee.uh.edu/?page_id=459

表 5.1: 数据集信息

数据集	图像宽	图像高	光谱数	类别数	有标记数	无标记数
Indian Pines	145	145	200	16	10249(48.75%)	10776(51.25%)
Pavia University	340	610	103	9	42776(20.62%)	164624(79.38%)
Houston	1905	349	144	15	15029(02.06%)	649816(97.74%)

表 5.2: Indian Pines 数据集的类别平衡

类别	1	2	3	4	5	6	7	8
比例 (%)	00.45	13.93	08.10	02.31	04.71	07.12	00.27	04.66
类别	9	10	11	12	13	14	15	16
比例 (%)	00.19	09.48	23.95	05.78	02.00	12.34	03.77	00.91

表 5.3: Pavia University 数据集的类别平衡

类别	1	2	3	4	5	6	7	8	9
比例 (%)	15.50	43.60	04.91	07.16	03.14	11.76	03.11	08.61	02.21

表 5.4: Houston 数据集的类别平衡

类别	1	2	3	4	5	6	7	8
比例 (%)	08.32	08.34	04.64	08.28	08.26	03.16	08.43	08.27
类别	9	10	11	12	13	14	15	
比例 (%)	08.33	08.16	08.21	08.20	03.12	02.84	04.39	

5.3.2 实验设置

数据集划分

数据集分割实际上包括了：训练集与测试集的分割、有标记样本与无标记样本的分割，这两个部分。实际上，由于测试集必须是有标记样本（否则无法计算精度），因此只要分割有标记数据即可，而全部无标记样本都参与模型训练。以 20% 有标记样本作为训练集、余下的 80% 有标记样本作为测试集是一种合理的划分。这个分割比例可以上浮。需要注意的是，不管有标记样本的划分如何，必须保证每一种类别都以恰当的比例参与了训练。表 (5.2; 5.3; 5.4) 展示了三个数据集的类别比例。以 20% – 80% 划分为例，数据集的分布大致如表 (5.5)。

性能度量

为了验证模型的有效性，需要度量它的性能。对于多分类问题，准确率 (accuracy) 是一种常用的性能度量。实验按照上一小节对数据集的划分规则，每次随机划分数据集，在训练集上训练模型并在测试集上计算准确率，如此重复 10 次，计算准确率的平均值，作为最后的准确率。

表 5.5: 20% – 80% 划分

数据集	训练集		测试集 80% 有标记样本
	无标记样本	20% 有标记样本	
Indian Pines	10776	2050	8199
Pavia University	164624	8555	34221
Houston	649816	2832	12197

表 5.6: 有监督实验 (准确率%)

数据集	LDA	QDA	k-NN
Indian Pines	76.29	38.94	66.82
Pavia University	84.77	86.13	86.26
Houston	74.80	62.88	75.62
数据集	LinearSVC(OAA)	LinearSVC(OAO)	Random Forest
Indian Pines	63.44	69.05	75.12
Pavia University	74.06	83.82	89.15
Houston	60.72	78.33	75.96
数据集	Gaussian NB	Bernoulli NB	Multinomial NB
Indian Pines	50.57	22.97	49.39
Pavia University	67.05	43.61	58.03
Houston	61.70	08.67	64.73
数据集	MLP1	MLP2	CNN
Indian Pines	53.09	62.24	47.70
Pavia University	80.98	78.13	72.91
Houston	71.78	65.36	81.74

5.3.3 实验结果

有监督实验

我们首先对有标记样本进行一些实验。除了无标记样本不参与训练外，实验设置基本与上一节相同，结果如表 (5.6) 所示。

LDA 与 QDA 常见的判别分析 (Discriminant Analysis) 包括线性判别分析 (Linear DA, LDA) 与二次判别分析 (Quadratic DA, QDA)。在 sklearn 中的 API 为：

```
sklearn.discriminant_analysis.LinearDiscriminantAnalysis
sklearn.discriminant_analysis.LinearDiscriminantAnalysis
```

LDA 全图的预测结果如图 (5.5)。注意，为突出结果起见，本节所有图像的标记颜



图 5.5: Houston 数据集, LDA

图 5.6: Houston 数据集, k -NN

色与上一节数据集介绍所用的颜色并不相同。

k -NN k 最近邻 (k -Nearest Neighbors) 是一种常见的监督学习算法, 一定程度上反映了样本的特征分布。在 sklearn 中, 其调用方式如下:

```
sklearn.neighbors.KNeighborsClassifier
```

实验中设置 $k = 20$ 。全图的预测结果如图 (5.6)。

SVM SVM 的实验采用 sklearn 的 sklearn.svm 模块完成。通过调用多分类的宏

```
sklearn.multiclass.OneVsRestClassifier  
sklearn.multiclass.OneVsOneClassifier
```

用户可以实现 (3.2.2) 一节中所描述的 OAO 与 OAA 策略。实验中设置 sklearn.svm.LinearSVC 的随机状态参数 random_state 为 0。全图的预测结果如图 (5.7)。

Naive Bayes 朴素贝叶斯 (Naive Bayes) 的实验采用 sklearn.naive_bayes 模块完成。该模块包括三种不同分布的 NB 模型: Gaussian 贝叶斯模型, 假设后验概率分布服从 Gaussian 分布; Bernoulli 贝叶斯模型, 只考虑某类别样本是否出现; 多项式贝叶斯模型, 考虑某类别样本出现的次数。



图 5.7: Houston 数据集, SVM



图 5.8: Houston 数据集, GaussianNB



图 5.9: Houston 数据集, Random Forest

```
sklearn.naive_bayes.GaussianNB
sklearn.naive_bayes.BernoulliNB
sklearn.naive_bayes.MultinomialNB
```

全图的预测结果如图 (5.8)。

Random Forest 随机森林 (Random Forest) 与决策树 (Decision Tree) 算法密切相关, 是一种性能优秀的集成学习 (Ensemble Learning) 算法。sklearn 也提供了它的实现:

```
sklearn.ensemble.RandomForestClassifier
```

在设置`max_depth=32`的情况下, 随机森林表现出相当优秀的性能。全图的预测结果如图 (5.9)。

神经网络 sklearn 提供了简单的全连接神经网络的 API:

```
sklearn.neural_network.MLPClassifier
```

在设置隐藏层大小为 `MLP1[256, 128, 128, 64]` 与 `MLP2[128, 64, 64]`, 学习率为 10^{-5} , 优化方式为 '`lbfgs`' (一种常用的 Hessian 矩阵优化方法) 的条件下, 其结果如表 (5.6)。

但是如果要实现卷积、池化等复杂的网络结构, 还是需要依靠 tensorflow 与 keras 等工具。实验中对比了两个不同深度的 CNN, 其结构没有任何差别, 皆由卷积、池化、全连接层组成。

半监督实验

与其他半监督算法的对比 实验所用的半监督深度学习模型为 output 模型, 设置 $\gamma_{\text{光谱}} = 0.9$, $\gamma_{\text{空间}} = 0.1$ 。以 Houston 数据集为例, 其所用的网络结构为:

表 5.7: 半监督实验 (准确率%)

数据集	EmbedCNN	LabelSpreading	LadderNetwork
Indian Pines	82.02	63.17	53.85
Pavia University	90.13	77.81	62.70
Houston	86.65	35.86	49.35

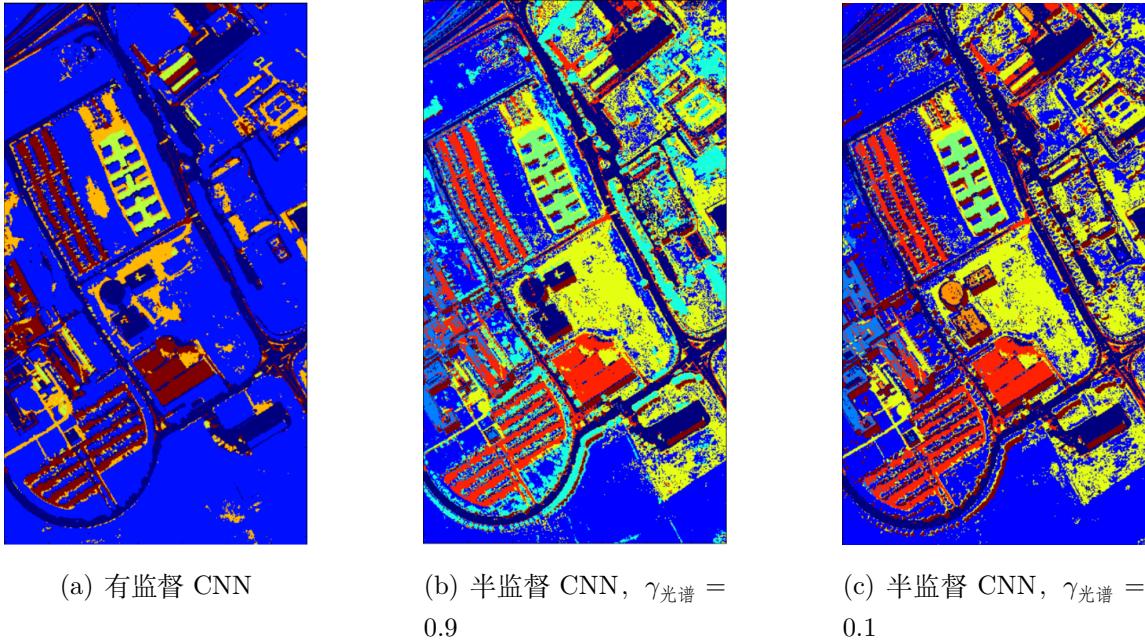


图 5.10: output 模型的相关结果图

```

Conv1D(kernel_size=2, filters=64),
Activation('tanh'),
MaxPooling1D(pool_size=2),
Conv1D(kernel_size=2, filters=32),
Activation('tanh'),
MaxPooling1D(pool_size=2),
Flatten(),
Dense(128, activation='tanh'),
Dense(15, activation='softmax')

```

与之对比的半监督学习模型为: sklearn 所提供的 Label Spreading[9]:

```
sklearn.semi_supervised.LabelSpreading
```

与 Ladder Network[1]²。其对比结果如表 (5.7)。

将神经网络的相关结果绘制成图, 对标记染色, 则得图 (5.10(a); 5.10(b); 5.10(c))

²<https://github.com/CuriousAI/ladder>

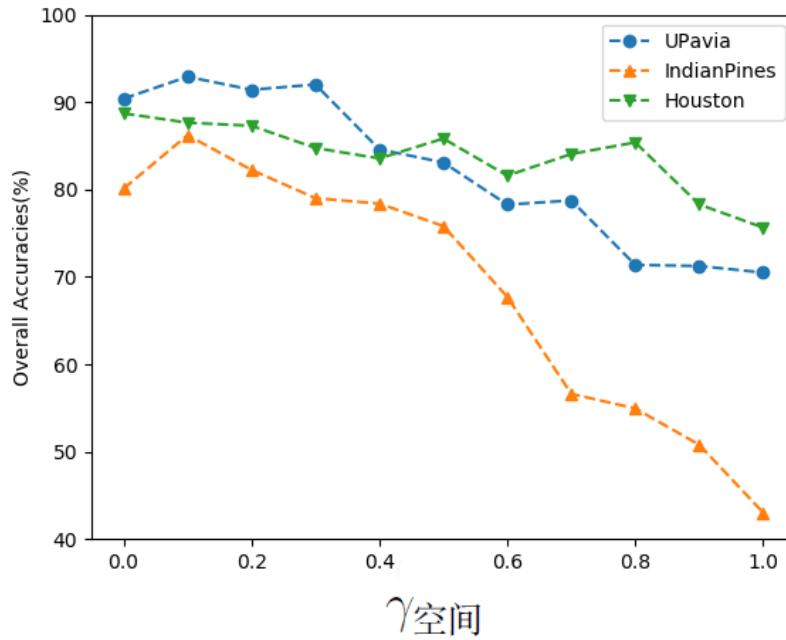


图 5.11: $\gamma_{\text{光谱}}$ 与 $\gamma_{\text{空间}}$ 的对比。 $\gamma_{\text{光谱}} + \gamma_{\text{空间}} = 1.0$ 。

$\gamma_{\text{光谱}}$ 与 $\gamma_{\text{空间}}$ 的对比 不妨令 $\gamma_{\text{光谱}} + \gamma_{\text{空间}} = 1.0$, 变化系数, 则得图 (5.11)。

5.3.4 实验结论

有监督的实验以随机森林的平均性能最好, 在三个数据集上都取得了较高的准确率。但是, 本实验所采用的所有模型都没有良好的泛化性。图 (5.5; 5.6; 5.7; 5.8; 5.9) 中均存在大量白色。这在本节中表示大量原本的无标记样本被分为第 1 类。实际上这并不合理。图 (5.10(a)) 也展现出了相同的问题。这初步说明即使有监督学习能够达到较高的准确率, 但其实是经不住考验的。

另一方面, 半监督学习确实展现了它的有效性。在模型对比的实验中, 半监督深度学习展现了最优的性能, 其他两种模型的性能则不那么优秀, 这可能与参数设置有关。但半监督深度学习模型确实在三个数据集上均取得了较好的结果, 说明它的平均性能很好。

此外, 图 (5.10(b); 5.10(c)) 则与有监督的结果图呈现出不同的状态。在半监督的结果图中, 无标记样本分类结果的多样程度要高一些。这暗示了模型的泛化性能可能要比之前有监督学习要好。

不仅如此, 图 (5.11) 对比了光谱正则化项与空间正则化项, 三个数据集都对光谱正则化项有一定的偏好。这在一定程度上印证了我们在 (5.1) 所作的推测。

第六章 总结与展望

本文所述的模型与 (三; 四) 两章中前人提出的模型都密切相关，甚至可以说是其推论。从实验结果看，此模型比较适用于高光谱图像的半监督分类任务，可以认为基本达到了预期目标。

然而，此模型依然存在诸多不足之处。

泛化性能不足 从结果图 (5.10(b); 5.10(c)) 来看，虽然相比于有监督的模型，其泛化能力得到一定提高，但依然存在大面积连续的无标记样本被分为同一类的问题。这可能与空间正则化项本身有关。或许改变空间邻域 $U_{\text{空间}}$ 的定义方式，或是增加其他的正则化项，会得到更好的结果。

运行代价较高 虽然我们在 (5.2.3) 一节中提出了预算算的技巧，避开了计算时间瓶颈，但算法 (1) 每次只随机地取了一组样本，而没有采用批量 (Batch) 处理的方式。因此一旦数据量增大，神经网络结构变深，则运行实验所占用的计算资源、时间代价都会变得非常高。相关的优化也是一个值得考虑的问题。

应用范围狭窄 本文所述的模型受限于高光谱遥感图像本身，对于其他一般图像的像素级分类借鉴价值不高 (考虑到 RGB 三通道的信息量很难与高光谱数百波段的信息量相比)。此外，训练好的模型与数据本身关系密切，很难迁移到另一张遥感图像上。不仅如此，模型主要考虑的还是高光谱图像，未能与现今遥感图像的高光谱、高空间分辨率的数据融合联系在一起。未来希望能改进模型，在上述方面增加其价值。

展望：全新的标记方法 近来，一些基于社交媒体对城市遥感图像进行标记的方法被提出，并且成为了研究热点 [12]。相关工作进展很快，有关的系统也正在研发。如何将本文所述的模型融入到这样的平台，作为其采集图像、储存数据、分析数据、展示结果的功能一环，也是重要的展望方向。毕竟，如果能通过系统性的方法，从源头上解决无标记数据的问题，则要比半监督算法所带来的提高更为本质。

附录 A 主要符号

默认标记

$\rho(\cdot, \cdot)$

度量

		代数	
\mathbf{x}	样本点		
n	样本维度		
m	样本总数		
l	标记样本数量		
u	无标记样本数量	a	标量
y	样本标记	\mathbf{a}	向量
\mathbf{y}	One-Hot 编码的标记	\mathbf{A}	矩阵
Θ	模型参数	\mathbf{I}	单位矩阵
k	类别总数	\mathbf{O}	零矩阵
C	常数	$diag(\cdots)$	对角矩阵
\mathbf{W}, \mathbf{V}	邻接矩阵	$(\cdot)^T$	矩阵及向量的转置
		$Tr(\cdot)$	矩阵的迹
		$\ \cdot\ _p$	L_p 范数

集合

		映射	
\mathbb{R}	实数集		
\mathbb{N}	自然数集		
\mathbb{Z}	整数集		
\mathbb{R}^n	n 维 Euclidean 空间		
\mathcal{H}	再生核 Hilbert 空间	$f(\cdot)$	模型函数
\mathcal{M}	流形	$g(\cdot)$	流形 embedding 映射
\mathcal{X}	样本空间	$L(\cdot)$	Lagrange 函数
\mathcal{Y}	标记集	$LL(\cdot)$	似然函数
D	数据集	$loss(\cdot)$	损失函数
X	函数定义域	$\Omega(\cdot)$	结构风险
U	邻域	$R(\cdot)$	正则化函数
$C^{(p)}(\cdot; \cdot)$	p 阶光滑函数集	$S(\cdot)$	激活函数
$ \cdot $	有限集的势	$h(\cdot)$	隐藏层

概率

$p(\cdot)$	概率密度函数
$p(\cdot \cdot)$	条件概率密度函数
$P(\cdot)$	概率
$P(\cdot \cdot)$	条件概率
μ	均值
Σ	协方差矩阵

物理

c	真空光速
λ	电磁波波长
v	电磁波频率
T	电磁波周期
E	能量
P	功率

附录 B 主要定义

定义 B.1 (多元高斯分布). 对于 n 维样本空间 \mathcal{X} 中的随机向量 $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, \mathbf{x} 服从 *Gaussian* 分布, 其概率密度函数为:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})} \quad (\text{B.1})$$

其中 $\boldsymbol{\mu}$ 为样本 \mathcal{X} 均值, Σ 为协方差矩阵。[42]

定义 B.2 (高斯混合模型).

$$p_M(\mathbf{x}) = \sum_{i=1}^k \alpha_i \cdot p(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i) \quad (\text{B.2})$$

k 个 *Gaussian* 分布 $p(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i)$ 为混合成分, 与混合系数 (*mixture coefficient*) $\alpha_i \in [0, 1]$ 组成了混合分布。其中有 $\sum_{i=1}^k \alpha_i = 1$ 。[42]

定义 B.3 (微分同胚). 从 \mathbb{R}^m 中的开集 U 到开集 V 上的映射: $f: U \rightarrow V$ 称为 $C^{(p)}$ 类微分同胚 (*diffeomorphism*), 其中 $p = 0, 1, 2, \dots$, 若:

- $f \in C^{(p)}(U; V)$;
- f 为双射;
- $f^{-1} \in C^{(p)}(V; U)$.

其中 $C^{(p)}(U; V)$ 表示由 U 到 V 的一切光滑映射的集合。[41]

定义 B.4 (微分流形). 具有可列拓扑基的 *Hausdorff* 拓扑空间 \mathcal{M} 被称为 n 维微分流形 (*differentiable manifold*), 若 \mathcal{M} 上任何一点都存在一个邻域 U , U 或同胚于整个空间 \mathbb{R}^n , 或同胚于半空间 $H^n = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}_1 \leq 0\}$ 。其中 *Hausdorff* 拓扑空间指任意两个不同点有不相交邻域的拓扑空间。[41]

定义 B.5 (Laplace-Beltrami 算子). 函数 f 定义在 *Euclidean* 空间中 $X \subset \mathbb{R}^n$, 若函数 f 在点 \mathbf{x}_0 上有梯度: $\nabla f = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}_0), \dots, \frac{\partial f}{\partial x_n}(\mathbf{x}_0) \right)$, 且梯度函数 ∇f 有散度: $\nabla \cdot \nabla f = \left(\frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}_0), \dots, \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}_0) \right)$, 则称梯度的散度为 *Laplace-Beltrami* 算子 (*Laplace-Beltrami Operator*): [7]

$$\nabla^2 f = \nabla \cdot \nabla f \quad (\text{B.3})$$

定义 B.6 (再生核 Hilbert 空间). 以实值函数为元素的 Euclidean 空间 $\mathcal{H} = \{f | f : X \mapsto \mathbb{R}\}$ 是再生核 Hilbert 空间 (Reproducing Kernel Hilbert Space, RKHS)[43], 若其具有:

- 完备性: \mathcal{H} 在度量 $\rho(f, g) = \|f - g\|$ 的意义上是完备的, 也即任一基本序列能在 $\|f - g\|$ 意义上收敛;
- 无限维: $\forall n \in \mathbb{N}$, 都可以找到 n 个线性无关的元素;
- 再生性: $\forall f \in \mathcal{H} \forall x \in X$, 存在连续映射 $L_x : f \mapsto f(x)$.

定义 B.7 (挤压函数). 若函数 $\Phi(t) : \mathbb{R} \mapsto [0, 1]$ 满足:

- 单调不降;
- 上下有界。

则称 Φ 是挤压函数 (squashing function)。

附录 C 主要定理

定理 C.1 (Bayes 定理). 设事件 E , 又设事件 F_1, \dots, F_n 互不相容: $\bigcap_{i \neq j} F_i F_j = \emptyset$, 且穷举: $E = \bigcup_{i=1}^n E F_i$ 。若 E 发生, 则 F_j 发生的概率为:

$$P(F_j|E) = \frac{P(EF_j)}{P(E)} = \frac{P(E|F_j)P(F_j)}{\sum_{i=1}^n P(E|F_i)P(F_i)} \quad (\text{C.1})$$

定理 C.2 (Mercer 定理). 若连续函数 $\kappa : [a, b] \times [a, b] \mapsto \mathbb{R}$ 满足:

- 对称: $\kappa(s, t) = \kappa(t, s)$
- 半正定: $\forall f : [a, b] \mapsto \mathbb{R}, \quad \iint f(s)\kappa(s, t)f(t)dsdt \geq 0$

则存在一系列正交的特征函数 $\{\psi_i\}_{i \in \mathbb{Z}^+}$ 及特征值 $\{\lambda_i\}_{i \in \mathbb{Z}^+}$: $\int \kappa(s, t)\psi_i(t) = \lambda_i\psi_i(s)$, 使:

$$\kappa(s, t) = \sum_{i=1}^{\infty} \lambda_i \psi_i(s) \psi_i(t) = \sum_{i=1}^{\infty} (\sqrt{\lambda_i} \psi_i(s)) (\sqrt{\lambda_i} \psi_i(t)) \quad (\text{C.2})$$

定理 C.3 (Kolmogorov-Arnold 定理). 给定 $n \in \mathbb{Z}^+$, 对任意 $[0, 1]^n \subset \mathbb{R}^n$ 上的连续函数 $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, f 可被表示为:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n \Phi_{ij}(x_j) \right) \quad (\text{C.3})$$

其中 g_i 是 $[0, 1]$ 上的连续函数, 取决于 f 。 Φ_{ij} 是 $[0, 1]$ 上的连续单调增函数, 与 f 无关。

定理 C.4 (连续的 Laplacian Eigenmaps). 设 $\mathcal{M} \subset \mathbb{R}^n$ 是一个光滑 n^* 维紧 *Riemannian* 流形, 流形 \mathcal{M} 上的度量记作 $\rho(\cdot, \cdot)$ 。若存在二次可微实值映射 f (也即 *Laplace-Beltrami* 算子可作用): $f : \mathcal{M} \mapsto \mathbb{R}$, 则对于相邻的 $\mathbf{s}, \mathbf{t} \in \mathcal{M}$, 存在不等式 [7]:

$$|f(\mathbf{s}) - f(\mathbf{t})| \leq \rho(\mathbf{s}, \mathbf{t}) \|\nabla f(\mathbf{s})\| + o(\rho(\mathbf{s}, \mathbf{t})) \quad (\text{C.4})$$

附录 D 推导变换

推导 D.1 (软间隔 SVM 的对偶问题). 由 *Lagrange* 乘子法, 变换式 (3.9), 得:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = & \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^l \xi_i \\ & + \sum_{i=1}^l \alpha_i (1 - \xi_i - y_i (\mathbf{w}^T \mathbf{x}_i + b)) - \sum_{i=1}^l \mu_i \xi_i \end{aligned} \quad (\text{D.1})$$

对参数 \mathbf{w} , b , ξ 求极值, 有:

$$\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \mathbf{w} - \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (\text{D.2})$$

$$\frac{\partial L}{\partial b}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \sum_{i=1}^l \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \quad (\text{D.3})$$

$$\frac{\partial L}{\partial \xi_i}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = C - \alpha_i - \mu_i = 0 \Rightarrow C = \alpha_i + \mu_i \quad (\text{D.4})$$

将式 (D.2), (D.3), (D.4) 带入式 (D.1), 则可得到式 (3.9) 的对偶表述:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} & \left\{ \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^l \alpha_i \right\} \\ \text{s.t. } & \sum_{i=1}^l \alpha_i y_i = 0, \\ & \alpha_i \in [0, C], \quad \forall i \in \{1, 2, \dots, l\} \end{aligned} \quad (\text{D.5})$$

推导 D.2 (离散的 Laplacian Eigenmaps). 设待求映射为 $f(\mathbf{x}_i) = \mathbf{z}_i \in \mathbb{R}^{n^*}, i = 1, \dots, m,$

用矩阵 $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_m)^T \in \mathbb{R}^{m \times n^*}$ 来表示。变换式 (4.6):

$$\begin{aligned}
& \sum_{i=1}^m \sum_{j=1}^m \|\mathbf{z}_i - \mathbf{z}_j\|^2 W_{ij} = \sum_{i=1}^m \sum_{j=1}^m (\mathbf{z}_i^T \mathbf{z}_i - 2\mathbf{z}_i^T \mathbf{z}_j + \mathbf{z}_j^T \mathbf{z}_j) W_{ij} \\
&= \sum_{i=1}^m \mathbf{z}_i^T \mathbf{z}_i \left(\sum_{j=1}^m W_{ij} \right) + \sum_{j=1}^m \mathbf{z}_j^T \mathbf{z}_j \left(\sum_{i=1}^m W_{ij} \right) - 2 \sum_{i=1}^m \sum_{j=1}^m \mathbf{z}_i^T \mathbf{z}_j W_{ij} \\
&= 2 \sum_{i=1}^m D_{ii} \mathbf{z}_i^T \mathbf{z}_i - 2 \sum_{i=1}^m \sum_{j=1}^m \mathbf{z}_i^T \mathbf{z}_j W_{ij} \\
&= 2 \sum_{i=1}^m (\mathbf{z}_i \sqrt{D_{ii}})^T (\mathbf{z}_i \sqrt{D_{ii}}) - 2 \sum_{i=1}^m \mathbf{z}_i^T \left(\sum_{j=1}^m \mathbf{z}_j W_{ij} \right) \\
&= 2 \text{Tr}(\mathbf{Z}^T \mathbf{D} \mathbf{Z}) - 2 \text{Tr}(\mathbf{Z}^T \mathbf{W} \mathbf{Z}) = 2 \text{Tr}(\mathbf{Z}^T (\mathbf{D} - \mathbf{W}) \mathbf{Z}) = 2 \text{Tr}(\mathbf{Z}^T \mathbf{L} \mathbf{Z})
\end{aligned} \tag{D.6}$$

其中 $\mathbf{D} = \text{diag}(\sum_{j=1}^m W_{1j}, \sum_{j=1}^m W_{2j}, \dots, \sum_{j=1}^m W_{mj})$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ 被称为 *Laplacian* 矩阵, 对应于 *Laplace-Beltrami* 算子 $\nabla^2(\cdot)$ 。如此, 则式 (4.6) 变换为:

$$\begin{aligned}
& \min \{ \text{Tr}(\mathbf{Z}^T \mathbf{L} \mathbf{Z}) \} \\
s.t. \quad & \mathbf{Z}^T \mathbf{D} \mathbf{Z} = \mathbf{I}
\end{aligned} \tag{D.7}$$

其中条件 $\mathbf{Z}^T \mathbf{D} \mathbf{Z} = \mathbf{I}$ 保证了解的存在性。依然采用 *Lagrange* 乘子法求解:

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{Z}}(\mathbf{Z}) &= \frac{\partial}{\partial \mathbf{Z}} (\text{Tr}(\mathbf{Z}^T \mathbf{L} \mathbf{Z}) + \text{Tr}(\mathbf{A}(\mathbf{Z}^T \mathbf{D} \mathbf{Z} - \mathbf{I}))) \\
&= \mathbf{L} \mathbf{Z} + \mathbf{L}^T \mathbf{Z} + \mathbf{D}^T \mathbf{Z} \mathbf{A}^T + \mathbf{D} \mathbf{Z} \mathbf{A} \\
&= 2 \mathbf{L} \mathbf{Z} + 2 \mathbf{D} \mathbf{Z} \mathbf{A} = \mathbf{O} \\
&\Rightarrow \mathbf{L} \mathbf{Z} = -\mathbf{D} \mathbf{Z} \mathbf{A}
\end{aligned} \tag{D.8}$$

将式 (D.8) 的结果带回式 (D.7), 即得:

$$\text{Tr}(\mathbf{Z}^T \mathbf{L} \mathbf{Z}) = -\text{Tr}(\mathbf{Z}^T \mathbf{D} \mathbf{Z} \mathbf{A}) = -\text{Tr}(\mathbf{I} \mathbf{A}) = -\text{Tr}(\mathbf{A}) = -\sum_i^{n^*} \alpha_i \tag{D.9}$$

是特征值之和。因此只要取特征向量即可。

推导 D.3 (output 模型的梯度下降). 为求梯度, 我们先求向量 $\mathbf{a} \in \mathbb{R}^{n/2}$ 阶范数 $\|\mathbf{a}\|$ 的导数 $\frac{d\|\mathbf{a}\|}{d\mathbf{a}}$:

$$\begin{aligned}
\frac{d\|\mathbf{a}\|}{d\mathbf{a}} &= \frac{d}{d\mathbf{a}} \left(\sum_{i=1}^n a_i^2 \right)^{\frac{1}{2}} = \left[\frac{\partial}{\partial a_1} \left(\sum_{i=1}^n a_i^2 \right)^{\frac{1}{2}}, \dots, \frac{\partial}{\partial a_n} \left(\sum_{i=1}^n a_i^2 \right)^{\frac{1}{2}} \right] \\
&= \left[\frac{\partial}{\partial a_j} \left(\sum_{i=1}^n a_i^2 \right)^{\frac{1}{2}} \right]_{j=1,\dots,n} = \left[\frac{1}{2} \left(\sum_{i=1}^n a_i^2 \right)^{-\frac{1}{2}} \cdot \sum_{i=1}^n \frac{\partial a_i^2}{\partial a_j} \right]_{j=1,\dots,n} = \frac{1}{2\|\mathbf{a}\|} [2a_j]_{j=1,\dots,n} = \frac{\mathbf{a}}{\|\mathbf{a}\|}
\end{aligned}$$

令式 (4.10) 对损失项计算了平方误差 (为求导简单起见。实际上在 (5.2.2) 节中将阐明这不是一个合理的选择), 对正则项采用了 *Laplacian Eigenmaps* 的方式计算 (式 (4.5)):

$$\begin{aligned}
 & C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \Omega(\Theta) \\
 & = C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \gamma \sum_{i,j=1}^m R(g(\mathbf{x}_i), g(\mathbf{x}_j), W_{ij}) \\
 & = C \sum_{i=1}^l \|\mathbf{y}_i - f(\mathbf{x}_i)\|^2 + \gamma \sum_{i,j=1}^m \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|^2 W_{ij}
 \end{aligned} \tag{D.10}$$

对 \mathbf{w}_3 求梯度则有:

$$\begin{aligned}
 & \frac{\partial}{\partial \mathbf{w}_3} \left(C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \Omega(\Theta) \right) \\
 & = \frac{\partial}{\partial \mathbf{w}_3} C \sum_{i=1}^l loss(f(\mathbf{x}_i), y_i) + \frac{\partial}{\partial \mathbf{w}_3} \gamma \sum_{i,j=1}^m R(g(\mathbf{x}_i), g(\mathbf{x}_j), W_{ij}) \\
 & = C \sum_{i=1}^l \frac{\partial loss}{\partial \|\mathbf{y}_i - f(\mathbf{x}_i)\|} \cdot \frac{\partial \|\mathbf{y}_i - f(\mathbf{x}_i)\|}{\partial (\mathbf{y}_i - f(\mathbf{x}_i))} \cdot \frac{\partial (\mathbf{y}_i - f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \cdot \frac{\partial f(\mathbf{x}_i)}{\partial h_3(\mathbf{x}_i)} \cdot \frac{\partial h_3(\mathbf{x}_i)}{\partial \mathbf{w}_3} \\
 & \quad + \gamma \sum_{i,j=1}^m \frac{\partial R}{\partial \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|} \cdot \frac{\partial \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|}{\partial (g(\mathbf{x}_i) - g(\mathbf{x}_j))} \cdot \frac{\partial (g(\mathbf{x}_i) - g(\mathbf{x}_j))}{\partial \mathbf{w}_3} \\
 & = C \sum_{i=1}^l \left(2(f(\mathbf{x}_i) - \mathbf{y}_i) \cdot \frac{\partial S_O(\mathbf{w}_O^T h_3(\mathbf{x}_i) + \mathbf{b}_O)}{\partial (\mathbf{w}_O^T h_3(\mathbf{x}_i) + \mathbf{b}_O)} \cdot \mathbf{w}_O^T \cdot \frac{\partial S_3(\mathbf{w}_3^T h_2(\mathbf{x}_i) + \mathbf{b}_3)}{\partial (\mathbf{w}_3^T h_2(\mathbf{x}_i) + \mathbf{b}_3)} \cdot h_2(\mathbf{x}_i) \right) \\
 & \quad + \gamma \sum_{i,j=1}^m \left(2W_{ij} \cdot \mathbf{w}_3^T (h_2(\mathbf{x}_i) - h_2(\mathbf{x}_j)) \right. \\
 & \quad \left. \cdot \left(\frac{\partial S_3(\mathbf{w}_3^T h_2(\mathbf{x}_i) + \mathbf{b}_3)}{\partial (\mathbf{w}_3^T h_2(\mathbf{x}_i) + \mathbf{b}_3)} \cdot h_2(\mathbf{x}_i) - \frac{\partial S_3(\mathbf{w}_3^T h_2(\mathbf{x}_j) + \mathbf{b}_3)}{\partial (\mathbf{w}_3^T h_2(\mathbf{x}_j) + \mathbf{b}_3)} \cdot h_2(\mathbf{x}_j) \right) \right)
 \end{aligned}$$

如此, 则可用梯度下降法进行优化。

Bibliography

- [1] Rasmus A, Berglund M, and Honkala M et al. “Semi-supervised learning with ladder networks”. In: *Advances in Neural Information Processing Systems* (2015), pp. 3546–3554.
- [2] Abadi et al. “TensorFlow: A System for Large-Scale Machine Learning”. In: *OSDI* 16 (2016), pp. 265–284.
- [3] Green Robert O et al. “Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)”. In: *Remote sensing of environment* 65.3 (1998), pp. 227–248.
- [4] Krizhevsky Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* (2012), pp. 1097–1105.
- [5] Kruskal Joseph B. “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis”. In: *Psychometrika* 29.1 (1964), pp. 1–27.
- [6] Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe. *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. Sept. 2015. doi: doi:/10.4231/R7RX991C. URL: <https://purr.psu.edu/publications/1947/1>.
- [7] Mikhail Belkin and Partha Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation”. In: *Neural computation* (2003), pp. 1373–1396.
- [8] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples”. In: *Journal of machine learning research* (Sept. 2006), pp. 2399–2434.
- [9] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. *Semi-Supervised Learning*. MIT press, 2006.
- [10] Bridle and John S. “Alpha-nets: a recurrent ‘neural’ network architecture with a hidden Markov model interpretation”. In: *Speech Communication* 9.1 (1990), pp. 83–92.

- [11] Lorenzo Bruzzone, Mingmin Chi, and Mattia Marconcini. “A novel transductive SVM for semisupervised classification of remote-sensing images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 44.11 (2006), pp. 3363–3373.
- [12] et al Chi Mingmin. “Big data for remote sensing: Challenges and opportunities”. In: *Proceedings of the IEEE* 104.11 (2016), pp. 2207–2219.
- [13] Pohl Cle and John Van Genderen. “Review article multisensor image fusion in remote sensing: concepts, methods and applications”. In: *International journal of remote sensing* 19.5 (1998), pp. 823–854.
- [14] Rumelhart David, Geoffrey Hinton, and Ronald Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), p. 553.
- [15] Dempster et al. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society* (1977), pp. 1–38.
- [16] Paul Gibson. *Introductory remote sensing principles and concepts*. Routledge, 2013.
- [17] Goodfellow I., Y. Bengio, and Courville A. *Deep learning*. MIT press.
- [18] Bromley J. et al. “Signature verification using a” siamese” time delay neural network”. In: *Advances in Neural Information Processing Systems* (1994), pp. 727–744.
- [19] Weston Jason and Chris Watkins. “Multi-class support vector machines”. In: *Technical Report* 5 (1998).
- [20] Weston Jason et al. “Deep learning via semi-supervised embedding”. In: *Neural Networks: Tricks of the Trade* (2012), pp. 639–655.
- [21] Thorsten Joachims. “Transductive inference for text classification using support vector machines”. In: *ICML* 99 (1999), pp. 200–209.
- [22] Jain Anil K. and Richard C. Dubes. “Algorithms for clustering data”. In: (1988).
- [23] Funahashi Ken-Ichi. “On the approximate realization of continuous mappings by neural networks”. In: *Neural networks* 2.3 (1989), pp. 183–192.
- [24] Jarrett Kevin, Koray Kavukcuoglu, and Yann LeCun. “What is the best multi-stage architecture for object recognition?” In: *Computer Vision, 2009 IEEE 12th International Conference on. IEEE* (2009), pp. 2146–2153.
- [25] B. Kunkel et al. “ROSIS (Reflective Optics System Imaging Spectrometer) - A Candidate Instrument For Polar Platform Missions”. In: *Proc.SPIE* 0868 (1988). DOI: 10.1117/12.943611. URL: <https://doi.org/10.1117/12.943611>.

- [26] Hornik Kurt, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.
- [27] Nene et al. “Columbia object image library (coil-20)”. In: (1996).
- [28] Chapelle Olivier and Alexander Zien. “Semi-Supervised Classification by Low Density Separation”. In: *AISTATS* (2005), pp. 57–64.
- [29] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [30] John Platt. “Sequential minimal optimization: A fast algorithm for training support vector machines”. In: *Technical Report MSR-TR-98-14, Microsoft Research* (1998).
- [31] et al Plaza Antonio. “Recent advances in techniques for hyperspectral image processing”. In: *Remote sensing of environment* 113 (2009), pp. 110–122.
- [32] Hadsell Raia, Sumit Chopra, and Yann LeCun. “Dimensionality reduction by learning an invariant mapping”. In: *2006 IEEE computer society conference* 2 (2006), pp. 1735–1742.
- [33] McCulloch Warren S. and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.
- [34] B. M. Shahshahani and D. A. Landgrebe. “The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon”. In: *IEEE Trans. Geosci. Remote Sens.* 32.5 (Sept. 1994), pp. 1087–1095.
- [35] S. Tadjudin and D. A. Landgrebe. “Robust parameter estimation for mixture model”. In: *IEEE Trans. Geosci. Remote Sens.* 38.1 (Jan. 2000), pp. 439–445.
- [36] Andrei Nikolaevich Tikhonov. “On the solution of ill-posed problems and the method of regularization”. In: *Doklady Akademii Nauk* 151.3 (1963).
- [37] Nair Vinod and Geoffrey E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)* (2010), pp. 807–814.
- [38] LeCun Y et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86 (1998), pp. 2278–2324.
- [39] LeCun Yann. “Generalization and network design strategies”. In: *Connectionism in perspective* (1989), pp. 143–155.
- [40] Zhou YT and R. Chellappa. “Computation of optical flow using a neural network”. In: *IEEE International Conference on Neural Networks* 27 (1988), pp. 71–78.

- [41] Vladimir Antonovich Zorich. 数学分析 (第四版). 高等教育出版社. ISBN: 7040183021.
- [42] 周志华. 机器学习. 1st ed. 清华大学出版社, Jan. 2016. ISBN: 9787302423287.
- [43] 李航. 统计学习方法. 清华大学出版社, 2012. ISBN: 9787302275954.

致谢

本篇论文为我本科阶段的学习成果，其诞生与老师、同学密切相关，是以需要感谢本科阶段一切予我帮助的人们。

首先需要感谢的是我的导师，池老师。在我于实验室的一年多的学习中，池老师给予了我非常大的帮助。最初，我被池老师的理论功底吸引，其后，池老师丰富的实践经验、严谨的治学态度、广博的学术视野都深深地影响了我。池老师是我学术与人生道路上的榜样。

计算机学院的各位老师也为我提供了很大帮助。他们开设了丰富多彩的专业课程，为我打下了本篇论文所需的理论功底，我相信在未来的生涯中犹将受益无穷。

同时，我也要对同学们说声感谢。在三四年学习生活中，计算机学院的同学们与我朝夕相处，相互切磋，极大地提升了我的理论与实践能力。《盘铭》曰：“苟日新，日日新，又日新。”优秀的同学鞭策我不断追赶他们，做新的自己。在此，我要特别感谢实验室的学长们，他们为我答疑解惑，提供了莫大帮助。

最后，我要郑重地感谢我的父母。他们为我能够专心学习付出了很多，我希望能够用学业的进步与人格的成长来回报他们。