Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 830

Sebastian Wuttke

Aktives Lernen mit Segmentierung und Clusterbildung zur bildbasierten Klassifikation der Landbedeckung

München 2019

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 | ISBN 978-3-7696-5242-0

Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 830

Aktives Lernen mit Segmentierung und Clusterbildung zur bildbasierten Klassifikation der Landbedeckung

Von der Ingenieurfakultät Bau Geo Umwelt

der Technischen Universität München

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Vorgelegt von

Dipl.-Inform. Sebastian Wuttke

Geboren am 15.01.1984 in Eisenach

München 2019

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5242-0

Adresse der DGK:



Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759

E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender: Prof. Dr. techn. Roland Pail

Referent: Prof. Dr.-Ing. Uwe Stilla

Korreferenten: Prof. Dr.-Ing. Christian Heipke (Uni Hannover)

Tag der mündlichen Prüfung: 07.12.2018

© 2019 Bayerische Akademie der Wissenschaften, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

ISSN 0065-5325 ISBN 978-3-7696-5242-0

Kurzfassung

Die Klassifikation von Landbedeckungsarten ist eine bedeutende Grundlage, um informierte politische, wirtschaftliche und gesellschaftliche Entscheidungen treffen zu können. Die hierzu erforderlichen Informationen können aus Daten der Fernerkundung gewonnen werden. Die Menge dieser Daten steigt aufgrund technologischer Weiterentwicklungen stetig an. Eine vollständige, manuelle Auswertung ist alleine aufgrund der schieren Datenmenge nicht mehr möglich. Seit einiger Zeit spielt maschinelles Lernen daher eine immer wichtiger werdende Rolle. Hierbei kommen am häufigsten überwachte Lernverfahren zum Einsatz, welche viele Trainingsbeispiele benötigen. Obwohl eine Unmenge an Stichproben zur Verfügung steht, ist das Beschaffen der zugehörigen *Ground-Truth*-Klasseninformationen sehr ressourcenaufwendig.
Das Ziel dieser Arbeit ist es, den Ressourceneinsatz während der Trainingsphase zu verringern, indem die Anzahl der benötigten *Ground-Truth*-Informationen reduziert wird. Dies geschieht durch die Identifikation jener Stichproben, welche besonders hilfreich für den Lernerfolg sind und durch das Ignorieren redundanter Informationen.
Das vorgestellte Verfahren besteht aus drei Schritten: (i) Segmentierung, (ii) Clusterbildung und (iii) aktives Lernen. Der Segmentierungsschritt baut auf der Glattheitsannahme auf und nutzt den SLIC-Algorithmus (*simple linear iterative clustering*), um die Merkmalsvektoren des Eingangsbildes in Repräsentantenvektoren zu überführen. Der Clusterbildungsschritt basiert auf der Clusterannahme und nutzt den *bisecting* k-Means-Algorithmus, um die Repräsentantenvektoren in einem Binärbaum zu organisieren. Der dritte Schritt stutzt diesen Baum in einem iterativen Prozess mit Hilfe der *Active-queries*-Methode, einem aktiven Lernverfahren. Das dabei entstehende Pruning ist in jeder Iteration optimal bezüglich eines definierten Klassifikationsfehlers. Daher ist es zu jedem Zeitpunkt möglich, den Trainingsvorgang abzubrechen und das aktuelle Zwischenergebnis in eine Klassifikationskarte des Eingangsbildes zu überführen.
Das Verfahren wurde mit drei verschiedenen Datensätzen aus dem Gebiet der Fernerkundung getestet. Sie zeigen ländliche und urbane Teile der deutschen Städte Abenberg, Potsdam und Vaihingen. Die acht durchgeführten Experimente untersuchen die verschiedenen Parameter der Methode und führen einen Vergleich mit Methoden auf dem aktuellen Stand der Forschung durch. Zur statistischen Auswertung wurde jedes Experiment zehnmal wiederholt und der Wilcoxon-Vorzeichen-Rang-Test angewendet. Dieser Test stellt fest, ob die Unterschiede zwischen den untersuchten Methoden statistisch signifikant sind. Die Ergebnisse zeigen, dass der Segmentierungsschritt den größten Einfluss auf die erreichte Klassifikationsgüte hat, gefolgt vom Schritt des aktiven Lernens. Die vorgestellte Methode erreicht im Vergleich zum passiven Lernen eine Reduktion der Trainingskosten um 95% im Durchschnitt über alle drei Datensätze.

Abstract

The classification of land cover types is an important basis for making informed political, economic, and social decisions. The information required for this purpose can be obtained from remote sensing data. The amount of this data is steadily increasing due to continuous technological development. A complete, manual evaluation is no longer possible due to the sheer volume of data. For some time now, machine learning has played an increasingly important role. The most frequently used machine learning methods are supervised, which require many training examples. Although a plethora of samples are available, obtaining the associated ground truth class labels is very resource intensive.
The goal of this work is to reduce the amount of resources used during the training phase by minimizing the number of ground truth information needed. This is done by identifying those samples that are particularly helpful for learning and by ignoring redundant information.
The presented method consists of three steps: (i) segmentation, (ii) clustering, and (iii) active learning. The segmentation step builds on the smoothness assumption and uses the simple linear iterative clustering (SLIC) algorithm to transform the feature vectors of the input image into representative vectors. The clustering step is based on the cluster assumption and uses the bisecting k-means algorithm to organize the representative vectors in a cluster hierarchy. The third step uses the active-queries method, an active learning method, to prune the cluster hierarchy in an iterative process. The resulting binary tree is optimal in each iteration with respect to a defined classification error. Therefore, it is possible at any time to cancel the training process and to transform the current intermediate result into a classification map of the input image.
The procedure was tested with three different remote sensing data sets. They show rural and urban parts of the German cities Abenberg, Potsdam and Vaihingen. The eight experiments carried out examine the various parameters of the method and compare it with current state-of-the-art methods. For statistical evaluation, each experiment was repeated ten times and the Wilcoxon signed-rank test applied. This statistical test determines if the differences between the methods studied are statistically significant. The results show that the segmentation step has the greatest impact on the achieved classification quality, followed by the active learning step. The presented method achieves a 95% reduction in training costs averaged over all three data sets compared to passive learning.

Inhaltsverzeichnis

Kurzfassung **iii**

Abstract **iv**

Inhaltsverzeichnis **v**

Abbildungsverzeichnis **ix**

Tabellenverzeichnis **xi**

Abkürzungsverzeichnis **xiii**

Notation **xv**

1 Einleitung **1**
1.1 Motivation . . . 1
1.2 Problemstellung und Ziel der Arbeit . . . 3
1.3 Aufbau der Arbeit . . . 3

2 Stand der Wissenschaft **5**
2.1 Landbedeckungsklassifikation . . . 5
2.2 Clusterbildung . . . 6
2.2.1 k-Means Algorithmus . . . 6
2.2.2 Hierarchische Clusterbildung . . . 7
2.2.3 *Bisecting* k-Means . . . 8
2.3 Segmentierung . . . 9
2.3.1 SLIC-Algorithmus . . . 10
2.4 Aktives Lernen . . . 11
2.4.1 Probabilistisches aktives Lernen . . . 12
2.5 Historische Entwicklung und Abgrenzung . . . 14

3 Grundlagen **17**
3.1 Aktives Lernen . . . 17
3.1.1 Einführendes Beispiel . . . 17
3.1.2 Allgemeines Schema . . . 20
3.1.3 Orakel . . . 21
3.1.4 Selektionsstrategie . . . 22
3.1.5 Stoppkriterium . . . 25
3.1.6 Szenarien . . . 26
3.2 Landbedeckungsklassifikation . . . 28
3.2.1 Distanzmaß . . . 28
3.2.2 Häufig verwendete Distanzmaße . . . 29
3.3 Clusterbildung . . . 30

3.4 Segmentierung 32

4 Segmentierung, Clusterhierarchie und aktives Lernen 33
4.1 Überblick 33
4.2 Segmentierung 35
4.2.1 Lokale Merkmale 35
4.2.2 Anpassung des SLIC-Algorithmus 35
4.2.3 Repräsentantenvektoren 36
4.3 Clusterhierarchie 37
4.3.1 Globale Merkmale 37
4.3.2 Erstellen der Hierarchie 38
4.3.3 Anpassung von *bisecting* k-Means 38
4.3.4 Zusammenspiel mit Segmentierungsschritt 38
4.4 Aktives Lernen 40
4.4.1 Optimales Pruning bestimmen 40
4.4.2 Optimale Stichprobe bestimmen 44
4.4.3 Orakelanfrage und Aktualisierung 45
4.4.4 Berücksichtigung der lokalen Dichte 46
4.5 Komplexitätsbetrachtung 48
4.5.1 Segmentierung 48
4.5.2 Clusterbildung 49
4.5.3 Aktives Lernen 49
4.5.4 Gesamtkomplexität 51

5 Experimente und Daten 53
5.1 Versuchsaufbau 53
5.1.1 Bewertungsmethode 54
5.1.2 Budget 54
5.1.3 Qualität 54
5.1.4 Statistischer Test 55
5.2 Experimente 56
5.2.1 Einfluss der Segmentierung 56
5.2.2 Einfluss des Segmentierungsparameters k 56
5.2.3 Einfluss des Clusterparameters B 57
5.2.4 Einfluss des aktiven Lernens 57
5.2.5 Einfluss des lokalen Dichte-Parameters σ 57
5.2.6 Einfluss der lokalen Dichte 58
5.2.7 Vergleich auf verschiedenen Datensätzen 58
5.2.8 Vergleich verschiedener Methoden 58
5.2.9 Zusammenfassung 59
5.3 Datensätze 59
5.3.1 Abenberg 59
5.3.2 Potsdam 59
5.3.3 Vaihingen 61
5.3.4 Zusammenfassung 63

6 Ergebnisse und Diskussion 65
6.1 Einfluss der Segmentierung 65
6.2 Einfluss des Segmentierungsparameters k 66
6.3 Einfluss des Clusterparameters B 67

6.4 Einfluss des aktiven Lernens 68
6.5 Einfluss des lokalen Dichte-Parameters σ 69
6.6 Einfluss der lokalen Dichte 71
6.7 Vergleich auf verschiedenen Datensätzen 72
6.8 Vergleich verschiedener Methoden 75
6.9 Allgemeine Diskussion 79

7 Zusammenfassung und Ausblick **83**
7.1 Zusammenfassung 83
7.2 Ausblick 84

Eigene Veröffentlichungen **87**

Literaturverzeichnis **89**

Lebenslauf **101**

Abbildungsverzeichnis

3.1 Einführendes Beispiel der Schwellenwertsuche auf dem Intervall [0,1]. 18
3.2 Schematischer Unterschied zwischen passivem und aktivem Lernen. 21
3.3 Verschiedene Szenarien des aktiven Lernens. 27
3.4 Clusterings mit verschiedenen Eigenschaften. 31

4.1 Überblick des Ablaufs der vorgestellten Methode. 34
4.2 Gegenüberstellung von Merkmals- und Repräsentantenvektoren. 36
4.3 Merkmalsraum und Clusterhierarchie nach einer bzw. drei binären Teilungen. 39
4.4 Gegenüberstellung von Unter- und Übersegmentierung. 40
4.5 Vergleich zweier Prunings. 41
4.6 Beispielhafte Verteilung der Klassenlabel für einen Knoten. 43
4.7 Visualisierung einer Iteration. 46
4.8 Veranschaulichung der Berechnung des Label-Häufigkeitsvektors. 48

5.1 Gesamtübersicht Datensatz Abenberg. 60
5.2 Gesamtübersicht Datensatz Potsdam. 61
5.3 Gesamtübersicht Datensatz Vaihingen. 62

6.1 Lernkurven Experiment 1. 65
6.2 Lernkurven Experiment 2. 67
6.3 Lernkurven Experiment 3. 68
6.4 Lernkurven Experiment 4. 69
6.5 Lernkurven Experiment 5. 70
6.6 Einflussbereich der Repräsentantenvektoren. 71
6.7 Lernkurven Experiment 6. 72
6.8 Lernkurven Experiment 7. 73
6.9 Detailansicht der Fehlklassifikationen im Vaihingen-Datensatz. 74
6.10 Lernkurven Experiment 8. 75
6.11 Lernkurven der vorgestellten Methoden auf den Datensätzen Abenberg und Potsdam. 77
6.12 Klassifikationsergebnisse der vorgestellten Methoden auf dem Vaihingen-Datensatz. . . 78
6.13 Vergleich verschiedener Distanzmaße des SLIC-Algorithmus. 79
6.14 Ergebnisse der einzelnen Methodenschritte. 80
6.15 Klassifikationsergebnisse für den Vaihingen-Datensatz während des Trainings. 81
6.16 Klassifikationsergebnisse für den Potsdam-Datensatz während des Trainings. 82

Tabellenverzeichnis

2.1 Übersicht zu verwandten Arbeiten. 15

5.1 Übersicht zu allen Experimenten und Parametern. 59
5.2 Charakteristiken der drei verwendeten Datensätze. 63

6.1 Verbesserungen durch die vorgestellte Methode für drei Datensätze. 73
6.2 Statistische Unterschiede der untersuchten Methoden. 76
6.3 Vergleich der untersuchten aktiven Methoden mit passivem Lernen. 76

Abkürzungsverzeichnis

AUC	area under the curve ... 55
BRDF	bidirectional reflectance distribution function ... 2
CIELAB	CIE-L*a*b*-Farbraum ... 10
CIE	Commission Internationale de l'Éclairage ... 10
CNN	convolutional neural net ... 1
CPU	central processing unit ... 1
CRF	conditional random field ... 1
DGPF	Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V. ... 61
DSM	digital surface model ... 60
GPU	graphics processing unit ... 1
GSD	ground sampling distance ... 63
ISPRS	International Society for Photogrammetry and Remote Sensing ... 59
LiDAR	light detection and ranging ... 61
mcPAL	multi-class PAL ... 13
mcPAL*	multi-class PAL mit Segmentierung ... 75
NDVI	Normalized Difference Vegetation Index ... 5
OPAL	optimiertes probabilistisches aktives Lernen ... 13
PAL	probabilistisches aktives Lernen ... 12
PCA	principle component analysis ... 9
RBF	radiale Basisfunktion ... 6
RGB	rot grün blau ... 10
SA	spectral angle ... 30
SAM	spectral angle mapper ... 87
SCHAL	Segmentierung, Clusterhierarchie, aktives Lernen ... 33
SCHPAL	Segmentierung, Clusterhierarchie, probabilistisches aktives Lernen ... 47
SLIC	simple linear iterative clustering ... 9
SVM	Support Vektor Maschine ... 5
TOP	true ortho photo ... 60
UAV	unmanned aerial vehicle ... 2

Notation

Symbol	Beschreibung
a, α, β	Skalarer Wert
$\mathbf{x}, \mathbf{q}$	Mehrdimensionaler Vektor z. B. ein Merkmalsvektor, Koordinaten eines Pixels
$\mathbf{x}^{(1)}, \mathbf{q}^{(x)}$	Einzelne Komponente eines Vektors, z. B. X-Koordinate eines Pixels
$\mathbf{D}, \mathbf{perf}(\mathbf{x}, y)$	Funktion, falls relevant mit Parameterangabe
$\mathbb{N}, \mathbb{R}$	Die Menge der natürlichen beziehungsweise reellen Zahlen
$\mathcal{U}, \mathcal{L}$	Mathematische Menge von Elementen, z. B. Menge von Klassenlabeln $\Omega = \{1, 2, \ldots, k\}, k \in \mathbb{N}$
$(\mathbf{x}, y)$	Mathematisches Tupel, z. B. zusammengehöriges Paar aus Stichprobe und Klassenlabel $\mathcal{L} = \{(\mathbf{x}, y)\}, \mathbf{x} \in \mathcal{U}, y \in \Omega$
$\mathbf{q}_i, y_i$	Element (Vektor oder Skalar) einer indexierten Menge, z. B. Pixel $\mathbf{q}$ aus einem Bild: $\mathcal{I} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\}, \mathbf{q}_i \in \mathcal{I}, 1 \leq i \leq k$ oder Klassenlabel y aus der Menge aller Klassenlabel: $y \in \Omega$

1 Einleitung

1.1 Motivation

Die Vereinten Nationen definieren Landbedeckung als „die wahrgenommene (bio-) physikalische Bedeckung der Erdoberfläche“* [Di Gregorio & Jansen, 2000]. Im Kontrast hierzu ist Landnutzung definiert als „die Arrangements, Aktivitäten und Beiträge von Menschen zum Erstellen, Verändern oder Erhalten bestimmter Landbedeckungstypen“† [Di Gregorio & Jansen, 2000]. Obwohl Landnutzung und Landbedeckung im umgangssprachlichen Gebrauch synonym verwendet werden, ist es wichtig, sie zu unterscheiden. Landbedeckung gibt lediglich die Materialart an, bezieht sich also auf rein physikalische Eigenschaften. Landnutzung auf der anderen Seite bezeichnet die Verwendung des betrachteten Gebietes, welche trotz des selben Materials unterschiedlich sein kann. Ein Beispiel für eine Landbedeckung ist Asphalt, wohingegen die Landnutzung sowohl Straße, Radweg oder Parkplatz sein kann.

Die Klassifikation von Landnutzungs- und Landbedeckungsarten ist eine bedeutende Grundlage, um informierte politische, wirtschaftliche und gesellschaftliche Entscheidungen treffen zu können [Anderson, 1976]. Beispiele hierfür sind die Vorhersage von Auswirkungen der Landbedeckung auf Luftqualität [Akbari et al., 2003], die Analyse der Folgen schnell fortschreitender Urbanisierung [Mohan et al., 2011] und das Kartografieren von Risiken, Auswirkungen und Gefährdungen im Kontext der globalen Erwärmung [Stephenne et al., 2017].

Methoden des maschinellen Lernens automatisieren diese Klassifikationsaufgaben zunehmend. Eine Verbesserung der Klassifikationsleistung ist auf drei verschiedene Arten möglich: höhere Geschwindigkeit, höhere Genauigkeit und mehr Robustheit. Die Geschwindigkeit kann zum Beispiel durch Anpassung der Algorithmen an *graphics processing units* (GPUs) anstatt *central processing units* (CPUs) geschehen [Ngo et al., 2012]. Große Verbesserungen der Genauigkeit konnten in letzter Zeit durch Fortschritte in der Verwendung von *convolutional neural nets* (CNNs) [Marmanis et al., 2016; Volpi & Tuia, 2017] erzielt werden. Eine robustere Klassifikation kann beispielsweise durch den Einsatz von *conditional random fields* (CRFs) und dem Integrieren kontextbasierter Zusammenhänge zwischen Landbedeckung und Landnutzung erzielt werden [Albert et al., 2017].

Unabhängig hiervon wächst die Menge der zu klassifizierenden Daten in der Fernerkundung auf mehreren Ebenen beständig an. Auf der Hardware-Ebene steigt die räumliche und spektrale Auflösung der Sensoren, was zu mehr Daten pro aufgenommenem Bild führt. Auf der Ebene hochentwickelter Anwendungen wird Fernerkundung in einem weltweiten Maßstab eingesetzt [Taubenböck et al., 2012; Kuffer et al., 2016] wie zum Beispiel mittels der Sentinel- [Schreier et al., 2008] und TerraSAR-X-Satelliten [Werninghaus & Buckreuss, 2010]. Im Vergleich zu älteren Sensoren ist die Größe der erfassten Flächen drastisch angestiegen. Auf der Ebene weniger anspruchsvoller Anwendungen erlauben sinkende Sensorpreise immer mehr Nutzern den Zugang zu Fernerkundungstechnologien. Dies steigert den Bedarf für Plattformen, die gleichzeitig mit mehreren Sensortypen arbeiten können [Schilling et al., 2013; Haraké et al., 2016]. Der Einsatz solcher Plattformen vergrößert die Datenmenge, die pro Fläche aufgenommen wird. Die

*„the observed (bio)physical cover on the earth's surface“

†„the arrangements, activities and inputs by people to produce, change or maintain a certain land cover type“

fortschreitende Entwicklung reduziert das Gewicht handelsüblicher Sensoren und erlaubt somit den Einsatz von unbemannten Fluggeräten (*unmanned aerial vehicle*, UAV) für breitgefächerte Anwendungsszenarien, wie beispielsweise landwirtschaftliche Überwachungsaufgaben [Colomina & Molina, 2014; Hird et al., 2017]. Dies führt zu einem insgesamt häufigeren Einsatz von Fernerkundungstechnologien. Ein weiteres Beispiel für das steigende Datenaufkommen sind modernste Erdbeobachtungssatelliten. Diese erreichen das Petabyte-Level und übersteigen die Kapazität der zur Erde übertragbaren Datenmenge. Hier sind Alternativen erforderlich, wie zum Beispiel das Hochladen des Algorithmus auf den Satelliten, so dass nur noch das Endprodukt zur Erde übertragen werden muss [Wagner et al., 2014]. Die aufgenommenen Rohdaten stehen somit gar nicht mehr zur Verfügung und können daher auch nicht manuell ausgewertet werden.

Jedoch selbst für auf der Erde aufgenommene Daten ist eine vollständige manuelle Auswertung nicht mehr möglich. Daher ist es erforderlich, maschinelle Auswerteverfahren einzusetzen. Die beiden Hauptansätze hierfür sind unüberwachte und überwachte Klassifikation. Bei ersterer werden Cluster von einander getrennt, ohne dass die Anzahl zu erstellender Klassen vorab bekannt ist. Es ist schwierig, hierfür gute Heuristiken zu finden [Lee & Crawford, 2004, 2005]. Dem gegenüber ist bei überwachten Methoden die Anzahl der zu trennenden Klassen zwar bekannt, jedoch müssen den Daten auch noch Klassenlabel passend zugeordnet werden. Es gibt zwei Arten, diese Zuordnung zwischen Daten und Klassenlabeln zu erstellen.

Die erste Möglichkeit ist, synthetische Daten zu bekannten Klassenlabeln zu erstellen. Um möglichst realistische Bilddaten zu simulieren, müssen jedoch viele Parameter bekannt sein wie zum Beispiel (i) Beleuchtung, (ii) Oberflächeneigenschaften der Objekte, beispielsweise die bidirectional reflectance distribution function (BRDF) [Nicodemus, 1965], (iii) Absorptions-, Transmissions- und Streuungseigenschaften der Atmosphäre [Rahman et al., 1993], (iv) Sensorspezifikationen sowie (v) Bodenfeuchte [Rüdiger et al., 2009]. Diese vielen Unbekannten führen zu sehr komplexen und nur sehr aufwendig zu berechnenden Modellen.

Die zweite Möglichkeit ist, für bekannte Daten die dazu korrespondierenden Klassenlabel zu erstellen. Für kleine und mittlere Gebiete kann dies *in situ* geschehen. Das heißt, es wird vor Ort bestimmt, welche Landbedeckungsart vorherrscht und die zugehörigen Daten werden mit dem entsprechenden Klassenlabel versehen. Diese Art der Klassenlabel wird auch als *Ground Truth* bezeichnet. Für größere Gebiete ist diese Art der Label-Bestimmung zu aufwendig und somit ungeeignet. Eine Alternative ist die Bildinterpretation. Hierbei wird die Bedeckungsart nicht vor Ort bestimmt, sondern durch einen Experten anhand des aufgenommenen Bildmaterials festgestellt. Diese Art der Klassenlabel wird auch als *Sensed Truth* bezeichnet [Klausmann et al., 1999]. Weitere Alternativen sind das Ableiten des Klassenlabels von bestehenden Informationsquellen wie zum Beispiel Karten [Kaiser et al., 2017]. Jedoch sind Karten eine Visualisierung und daher stets mit Generalisierungsfehlern, im Vergleich zur *Ground Truth*, behaftet [Hake, 2002]. Das automatische Wiederverwenden von Klassenlabeln oder Klassifikationssystemen ist Thema des Transfer-Lernens [Pan & Yang, 2010; Demir et al., 2012] und der Domänenadaption [Patel et al., 2015; Tuia et al., 2016]. Ein Nachteil dieser Ansätze ist jedoch, dass sie das Problem der Beschaffung von Klassenlabeln nicht lösen, sondern lediglich zu einem anderen Datensatz verlagern. Es entstehen also nach wie vor hohe Trainingskosten für das Beschaffen der *Ground Truth*.

Daher bleibt ein Problem bestehen: ungelabelte Daten sind oftmals in großer Fülle vorhanden, die dazugehörigen Klassenlabel müssen jedoch erst beschafft werden. Dies verursacht Aufwand und benötigt den Einsatz von Ressourcen wie beispielsweise Geld oder menschliche Annotationszeit [Bailly et al., 2017]. Das übliche Vorgehen hierbei ist, einen Teil der ungelabelten Daten auszuwählen und nur für diese die Label-Informationen zu beschaffen. Die Auswahl erfolgt dabei jedoch zufällig und nur einmalig zu Beginn des Trainings [Settles, 2012]. Im Gegensatz dazu wird beim aktiven Lernen die Auswahl der Trainingsbeispiele aktiv gesteuert. Dieser Ansatz spielt in der vorliegenden Arbeit eine hervorgehobene Rolle.

1.2 Problemstellung und Ziel der Arbeit

Das in dieser Arbeit betrachtete Problem besteht darin, dass der Trainingsaufwand für überwachte Landbedeckungsklassifikation sehr hoch ist. Der Grund hierfür sind die hohen Kosten, die mit der Beschaffung der Label-Informationen einhergehen. Um diese Kosten zu reduzieren, nutzen bestehende Arbeiten den Ansatz des aktiven Lernens [Dasgupta & Hsu, 2008; Tuia et al., 2012; Muñoz-Marí et al., 2012]. Ziel der vorliegenden Arbeit ist es, die bestehenden Ansätze zu vereinen, anzupassen und zu erweitern, so dass die Effizienz der Landbedeckungsklassifikation steigt. Die Steigerung kann dabei auf zwei Arten geschehen: (i) Verbesserung der Gesamtklassifikationsgenauigkeit unter Verwendung des selben Trainingsaufwandes oder (ii) Reduzierung der benötigten Trainingsbeispiele bei gleichbleibender Gesamtklassifikationsgenauigkeit. Um dies zu erreichen, wird beim aktiven Lernen ein Nützlichkeitsmaß definiert und eine spezielle Selektionsstrategie eingesetzt. Dieses Vorgehen fokussiert den Annotationsaufwand auf die hilfreichsten Stichproben und reduziert so die Anzahl der benötigten Trainingsbeispiele und den damit verbundenen Ressourceneinsatz.

Die vorliegende Arbeit widmet sich den folgenden Forschungsfragen:

- Führt die Kombination von Segmentierung, Clusterbildung und aktivem Lernen zu effizienterem Training?
- Welchen Beitrag leisten die einzelnen Schritte der Methode?
- Wie können Informationen über die lokale Dichte integriert werden und unterstützen sie den Lernprozess?
- Wie unterscheiden sich die Ergebnisse der vorgestellten Methode im Vergleich zu den Methoden nach dem Stand der Forschung?

Hierzu wird eine aus drei Schritten bestehende Methode vorgestellt: (i) Segmentierung, (ii) Clusterbildung und (iii) aktives Lernen. Der erste Schritt nutzt die Glattheitsannahme, um die Redundanz der Eingangsdaten zu reduzieren. Der zweite Schritt nutzt die Clusterannahme, um die in den Daten vorhandenen Strukturen in eine Clusterhierarchie zu überführen. Der dritte Schritt wendet ein aktives Lernverfahren auf diese Hierarchie an. Des Weiteren untersucht diese Arbeit, ob durch die Integration lokaler Dichteinformationen die Lerneffizienz steigt. Die vorgestellte Methode wird in einem Einzelfaktor-Versuchsaufbau [Montgomery, 2013] untersucht, mit aktuellen Methoden aus dem Stand der Forschung verglichen und anhand von drei verschiedenen multispektralen Datensätzen demonstriert.

1.3 Aufbau der Arbeit

Dieses Kapitel gab eine einleitende Motivation zur Problemstellung und nannte die in dieser Arbeit untersuchten Forschungsfragen. Das zweite Kapitel stellt den Stand der Wissenschaft vor und geht dabei insbesondere auf Veröffentlichungen ein, die in engem Zusammenhang mit der hier vorgestellten Methode stehen. Kapitel drei gibt eine Einführung in die Grundlagen des aktiven Lernens sowie einen Überblick über Segmentierung und Clusterbildung. Das vierte Kapitel bildet den Hauptteil dieser Arbeit und stellt die dreistufige Methode detailliert vor. Die durchgeführten Experimente und verwendeten Datensätze werden in Kapitel fünf vorgestellt. Kapitel sechs präsentiert die Ergebnisse und diskutiert sie. Den Abschluss der Arbeit bildet Kapitel sieben mit einer Zusammenfassung und einem Ausblick auf mögliche Folgearbeiten.

2 Stand der Wissenschaft

Wie bereits in Kapitel 1 erläutert, ist die Landbedeckungsklassifikation ein sehr wichtiger Teilbereich der Fernerkundung. Dieses Kapitel nennt aktuelle Forschungsansätze und stellt Lösungen vor, die methodisch mit der vorliegenden Arbeit zusammenhängen. Zur verwendeten Notation sei auf den Eingangs dargestellten Überblick hingewiesen.

2.1 Landbedeckungsklassifikation

Es gibt eine Vielzahl an unterschiedlichen Ansätzen, um Landbedeckungen zu klassifizieren [Weng, 2012]. Hierzu zählen Ansätze, bei denen Merkmalsextraktoren manuell definiert werden. Beispiele hierfür sind der Normalized Difference Vegetation Index (NDVI) [Rouse et al., 1974] und andere Indizes (Hydrocarbonindex [Kühn et al., 2004], Wasserindex [Gao, 1995]), die direkt auf physikalischen Eigenschaften der zu identifizierenden Klassen basieren. Ebenso wurden objektbasierte Ansätze mit manuellen Regeln kombiniert, um beispielsweise Schatten und deren erzeugende Objekte zu segmentieren [Zhou & Wang, 2008; Zhou et al., 2009].

Eine große Rolle spielen auch pixelbasierte Ansätze [Myint et al., 2011] sowie die damit verwandten subpixelbasierten Methoden [Foody & COX, 1994; Tran et al., 2014; Lu et al., 2017]. Statistische Lernverfahren hingegen, wie zum Beispiel Maximum-Likelihood-Schätzer, können auch ohne manuell erzeugte Merkmalsextraktoren verwendet werden [Dean & Smith, 2003; Wuttke et al., 2014]. Um auch nicht-lineare Effekte bearbeiten zu können, wurden verschiedene Erweiterungen vorgestellt [Okujeni et al., 2014]. Hierzu zählen *Random Forests* [Breiman, 2001; Rodríguez-Galiano et al., 2012], Support Vektor Maschinen (SVMs) [Cortes & Vapnik, 1995; Schölkopf et al., 2000; Huang et al., 2002; Wuttke et al., 2016] und k-Nächster-Nachbar-Methoden [Gjertsen, 2007; Blanzieri & Melgani, 2008; Ma et al., 2010; Wuttke et al., 2012]. Eine weitere, sehr wichtige Kategorie der parameterfreien statistischen Lernverfahren sind künstliche neuronale Netze [Kavzoglu & Mather, 2003; Zhou & Wang, 2008]. Werden diese so konstruiert, dass sie spezielle Eingangsschichten haben, sind sie in der Lage selbstständig Merkmalsextraktoren zu erlernen und man spricht von *Deep Learning* [LeCun et al., 2015; Zhu et al., 2017]. Hier gewinnen vor allem CNNs an Bedeutung [Makantasis et al., 2015; Hu et al., 2015; Kussul et al., 2017]. Die große Anzahl zusätzlicher Schichten (insbesondere die vollständig verbundenen) bedeutet jedoch, dass die Anzahl der zu optimierenden Parameter nun im Bereich von hunderten von Millionen liegt [LeCun et al., 2015]. Damit einhergehend steigt der Bedarf für Trainingsdaten, die mit den wahren Klassenlabeln versehen sind. Solche Trainingsdaten in ausreichender Menge zur Verfügung zu stellen, ist somit ein großes Problem.

Es gibt verschiedene Ansätze, die versuchen, dieses Problem zu umgehen. Die wichtigsten sind (i) teilüberwachtes Lernen [Zhu, 2008; Muñoz-Marí et al., 2012; Huo et al., 2015], (ii) Domänenadaption beziehungsweise Transferlernen [Bruzzone & Marconcini, 2010; Pan & Yang, 2010; Durbha et al., 2011; Paul et al., 2016; Tuia et al., 2016] und (iii) aktives Lernen [Settles, 2009; Tuia et al., 2011]. Der erste Ansatz ist, wie der Name vermuten lässt, zwischen überwachtem und unüberwachtem Lernen angesiedelt. Er nutzt eine kleine Menge bekannter Klassenlabel, um daraus die Klassenlabel bisher unbekannter Stichproben zu extrapolieren und vergrößert so die

verfügbare Trainingsmenge. Der zweite Ansatz ist für Problemstellungen geeignet, bei denen Trainingsdaten in ausreichender Menge zwar in der Ausgangsdomäne vorhanden sind, jedoch nicht in der Lösungsdomäne. Es soll nun eine geeignete Transformation gelernt werden, so dass die bekannten Klassenlabel aus der Ursprungsdomäne auf Stichproben aus der Zieldomäne übertragen werden können. Gelingt dies, stehen in der Zieldomäne ausreichend viele Trainingsbeispiele zur Verfügung, so dass auch hier ein Lernverfahren trainiert werden kann. Der dritte Ansatz, die benötigte Trainingsmenge zu reduzieren, ist für diese Arbeit von besonderem Interesse. Aktives Lernen erreicht diese Reduktion durch eine spezialisierte Selektionsstrategie. Ebenso relevant für die vorliegende Arbeit sind Ansätze, die auf Segmentierung sowie Clusterhierarchien basieren und auf der Pixel- und Superpixel-Ebene eingesetzt werden. Die folgenden Unterkapiteln erläutern entsprechende Arbeiten detaillierter.

2.2 Clusterbildung

Ein zweistufiger Ansatz zur Clusterbildung unter Verwendung spektraler und räumlicher Informationen wurde in [Marcal & Castro, 2005] präsentiert. Marcal und Castro setzen eine zuvor durchgeführte unüberwachte Klassifikation voraus. Das Ergebnis hiervon ist eine geringe Anzahl von Klassen („few tens of classes“), welche anschließend durch ihr agglomeratives hierarchisches Clusterverfahren verarbeitet werden. Das hierfür benötigte Clusterdistanzmaß definieren sie als eine Linearkombination aus vier Indizes: (i) spektrale Ähnlichkeit, (ii) räumliche Begrenzung, (iii) räumliche Kompaktheit und (iv) Klassengröße. Für den ersten Index verwenden sie die Mahalanobis-Distanz [Mahalanobis, 1936]. Der zweite Index ist das Verhältnis der zur jeweiligen Klasse gehörenden Pixelanzahl und der Anzahl gemeinsamer Nachbarpixel. Der dritte Index ist der Anteil von Nachbarschaftspixeln zur Gesamtanzahl an Pixeln der selben Klasse. Hiermit soll vermieden werden, dass bereits sehr kompakte Klassen vereint werden. Der vierte Index ist die normierte relative Größe der zu vergleichenden Cluster, gemessen anhand der zugehörigen Pixel. Ihre vorgestellte Methode berechnet nun paarweise die Distanz und vereint die beiden am nächsten zueinander liegenden Cluster. Dies wird solang wiederholt, bis nur noch ein Cluster übrig bleibt. In der vorliegenden Arbeit wird ebenfalls eine Kombination aus spektralen und räumlichen Komponenten verwendet. Anstatt agglomerativer Clusterbildung („*bottom up*“) wird jedoch unterteilende Clusterbildung („*top down*“) eingesetzt, wie im Abschnitt 4.3 detaillierter erläutert.

Ein anderer Ansatz für ein Clusterverfahren ist das Anpassen des Kerns einer Support Vektor Maschine, so dass die Clusterannahme (siehe Abschnitt 3.3) ausgenutzt werden kann. Huo et al. [2015] nutzen hierfür eine Linearkombination aus drei Kernen: (i) radiale Basisfunktion (RBF), (ii) Kodierung der Clusterbeziehungen und (iii) räumliche Konsistenz. Der erste Kern ist der allgemein übliche RBF-Kernel [Vert et al., 2004]. Der zweite Kern basiert auf den Ergebnissen eines Ensembles von Clusterverfahren. Je häufiger zwei Elemente im selben Cluster landen, desto ähnlicher sind sie sich. Für den dritten Kern wird zunächst eine unüberwachte hierarchische Segmentierung durchgeführt. Die Ähnlichkeit zweier Segmente entspricht der Aggregationsentfernung auf dem Level der Hierarchie, auf welchem die beiden Elemente vereint wurden. Die drei Kerne werden gewichtet zusammengefasst, so dass die Summe der drei Gewichte 1 ist. Diese Arbeit ist ein gutes Beispiel dafür, wie die Clusterannahme für das Auswerten von Fernerkundungsbildern genutzt werden kann.

2.2.1 *k*-Means Algorithmus

Eines der bekanntesten und einfachsten Clusterverfahren ist der k-Means Algorithmus. Der Begriff „k-Means“ wurde erstmals von MacQueen [1967] verwendet. Die in diesem Abschnitt vorgestellte Variante basiert jedoch auf der Arbeit von Lloyd [1982]. Diese Variante läuft nach einem ein-

fachen Schema ab: es beginnt mit einem Initialisierungsschritt gefolgt von sich abwechselnden Zuweisungs- und Aktualisierungsschritten, bis ein Stoppkriterium erfüllt ist. Die drei einzelnen Schritte sind im Folgenden beschrieben.

❑ **Initialisierungsschritt**

Der namensgebende Parameter k des Algorithmus gibt an, in wie viele Cluster die Daten unterteilt werden sollen. Zur Initialisierung werden zufällig k Werte aus den Daten bestimmt und als initiale Clusterzentren festgelegt: $\mathbf{c}_i$ mit $i = \{1, \ldots, k\}$.

❑ **Zuweisungsschritt**

In diesem Schritt wird jedes Element $\mathbf{x}$ dem Clusterzentrum zugewiesen, zu dem es die kürzeste Distanz $\mathbf{D}$ hat. Trifft dies für ein $\mathbf{x}$ auf mehrere Zentren zu, wird es trotzdem nur einem zugeordnet, da es sich um ein striktes Clustering handelt. Abschnitt 3.2.1 geht genauer auf häufig verwendete Distanzmaße ein.

$$\mathcal{C}_i = \{\mathbf{x} : \mathbf{D}(\mathbf{x}, \mathbf{c}_i) \leq \mathbf{D}(\mathbf{x}, \mathbf{c}_j) \ \ \forall j = 1, \ldots, k\} \tag{2.1}$$

❑ **Aktualisierungsschritt**

Nach der Zuweisung aller Elemente werden die Positionen der Clusterzentren aktualisiert. Hierfür wird der Mittelwert aller zu einem Cluster gehörender Elemente bestimmt und als neues Zentrum festgelegt:

$$\mathbf{c}_i' = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x} \tag{2.2}$$

Die letzten beiden Schritte werden wiederholt, bis sich die Zuordnungen nicht mehr ändern. Der Algorithmus konvergiert stets auf ein lokales Optimum. Es kann jedoch nicht garantiert werden, dass dies auch das globale Optimum ist. Als Abhilfe wird der Algorithmus daher häufig mehrmals mit unterschiedlicher Initialisierung ausgeführt und anschließend das beste Ergebnis verwendet.

Unter gewissen Annahmen besitzt die hier vorgestellte Variante eine lineare Laufzeit [Hartigan & Wong, 1979]. Diese Annahmen sind, dass die Dimension der Daten und die Anzahl der Cluster konstant sind und dass nur eine geringe Anzahl von Iterationen bis zur Konvergenz benötigt wird. Har-Peled & Sadri [2005] sowie Arthur & Vassilvitskii [2006] zeigen, dass diese Annahmen in der Praxis häufig erfüllt sind.

2.2.2 Hierarchische Clusterbildung

Ein Vergleich verschiedener Clusterverfahren befindet sich in [Senthilnath et al., 2012]. Dort stellen Senthilnath et al. außerdem einen zweistufigen Ansatz vor. Für das Clusterverfahren der ersten Stufe vergleichen sie drei Varianten. Die erste Variante nutzt *mean shift clustering* [Fukunaga & Hostetler, 1975]. Hierbei werden die Trainingsbeispiele als Wahrscheinlichkeitsdichtefunktion interpretiert, so dass die Regionen größter Dichte den lokalen Funktionsmaxima entsprechen. Durch Gradientenaufstieg können diese Maxima gefunden werden und bilden die Zentren der gesuchten Cluster. Dies wird solang wiederholt, bis die Anzahl der gefunden Cluster dem Bayeschen Informationskriterium [Schwarz, 1978] entspricht. Die zweite Variante nutzt *niche particle swarm optimization* [Brits et al., 2002]. Hierbei werden die Trainingsbeispiele als Vogelschwarm interpretiert. Jeder einzelne Vogel merkt sich seine bisher beste Position und besitzt eine Geschwindigkeit, welche in jeder Iteration angepasst wird. Die Anpassung ist eine Mischung aus

der vorherigen Geschwindigkeit und abhängig von der aktuellen Entfernung zur persönlich besten Position. Unterschreitet der Wert der aktuellen Position eine feste Schwelle, wird der Vogel einem neuen Teilschwarm zugewiesen. Insgesamt werden mehrere Teilschwärme gebildet, so dass am Ende des iterativen Prozesses mehrere lokale Maxima identifiziert werden können. Die dritte Variante nutzt *glow worm swarm optimization* [Krishnanand & Ghose, 2005]. Hierbei werden die Trainingsbeispiele als Leuchtkäfer interpretiert. Jedes Individuum leuchtet entsprechend der Güte seiner aktuellen Position und einem sich über die Zeit abschwächenden Wert, welcher abhängig von den zuletzt besuchten Positionen ist. Jeder Leuchtkäfer hat eine räumlich beschränkte Wahrnehmung und bewegt sich in jeder Iteration auf einen von ihm wahrgenommenen Punkt zu, der heller leuchtet als er selbst. Mit steigender Anzahl an Iterationen wird der Wahrnehmungsradius für alle Leuchtkäfer reduziert, bis schließlich alle Bewegungen stoppen. Die konvergierten Positionen sind die gesuchten Clusterzentren. Die Untersuchungen von Senthilnath et al. zeigten, dass *glow worm swarm optimization* die genauesten und robustesten Ergebnisse liefert. Nachdem der erste Schritt die Clusterzentren identifizierte, werden diese im zweiten Schritt als Eingabe für ein k-Means Clustering verwendet. Nun wird jede Stichprobe dem ihr am nächsten liegenden Zentrum zugewiesen. Jedem dieser so entstandenen Cluster wird mittels eines Maximumentscheids über die in diesem Cluster bekannten Klassenlabel ein Gesamtlabel zugewiesen. In der Arbeit von Senthilnath et al. steht der Begriff „hierarchisch“ für das Verwenden von zwei Stufen, nicht für das Erstellen einer Clusterhierarchie wie es in der vorliegenden Arbeit der Fall ist. Der Ansatz des Mehrheitsentscheids, um das erstellte Clustering in das Label-Bild zu überführen, wird jedoch übernommen.

Die ersten Arbeiten, die Clusterbildung und Segmentierung kombinieren, stammen von Lee & Crawford [2004, 2005]. In [Lee & Crawford, 2004] stellen sie einen zweistufigen Prozess vor, der auf Hyperspektraldaten arbeiten, also Bildern mit deutlich mehr als drei Kanälen. Die erste Stufe arbeitet lokal und führt ein räumliches *Region-growing*-Verfahren durch. Hierbei werden die einzelnen Pixel zu Ketten zusammengefasst, so dass die enthaltenen Pixel möglichst ähnlich zueinander sind. Ihr verwendetes Abstandsmaß basiert auf der Intra-Cluster-Varianz. Sie setzen dafür jedoch die Annahme voraus, dass keine Korrelation zwischen den Hyperspektralbändern besteht. Als Stoppkriterium für ihr *Region-growing*-Verfahren „*mutual clostest neighbor*“ verwenden sie das Schwarzsche Informationskriterium [Schwarz, 1978]. Die zweite Stufe ist ein globales (räumlich nicht eingeschränktes) hierarchisches Clusterverfahren. Sie setzten dabei ein agglomeratives Verfahren ein und nutzen die Mahalanobis-Distanz als Abstandsmaß.

In [Lee & Crawford, 2005] adaptieren sie den zweistufigen Prozess. In der ersten Stufe verwenden sie nun *region merging* mit dem Bayeschen Informationskriterium welches auch in *Markov-Random-Fields* [Kindermann & Snell, 1980] verwendet wird. Die Einschränkung auf räumliche Lokalität bleibt weiterhin bestehen. Die zweite Stufe arbeitet global und überführt die Regionen mit einem kontextfreien Ähnlichkeitsmaß in eine Clusterhierarchie. Diesen Schritt optimierten sie durch den Einsatz eines eigenen *Multiwindow*-Verfahrens, welches das Bild in nicht überlappende Bereiche einteilt. Diese Unterteilung bedingt jedoch eine Sonderbehandlung der Regionen, die über die Bereichsgrenzen hinausgehen. Der Ansatz die „lokale Segmentierung“ und die „globale hierarchische Clusterbildung“ zu kombinieren wird in der vorliegenden Arbeit aufgegriffen und mit aktivem Lernen als dritten Schritt erweitert.

2.2.3 *Bisecting* k-Means

Der *bisecting* k-Means Algorithmus ist eine Spezialisierung des zuvor beschriebenen k-Means Algorithmus (Abschnitt 2.2.1). Namensgebend ist die Festlegung der Clusteranzahl auf $k = 2$. Das Ergebnis dieser Variante ist kein flaches sondern ein hierarchisches Clustering. Zu Beginn befinden sich alle Elemente in einem Cluster, der Wurzel. Anschließend werden drei Schritte wiederholt bis der Algorithmus terminiert.

Die drei sich wiederholenden Schritte sind:

1. Wähle den Cluster, der am meisten Elemente enthält. Enthält dieser lediglich ein Element, terminiere den Algorithmus.
2. Führe für die Elemente des ausgewählten Clusters den k-Means Algorithmus mit $k = 2$ aus.
3. Ersetze den im ersten Schritt ausgewählten Cluster mit den beiden im zweiten Schritt gefundenen Clustern.

Die Laufzeit des *bisecting* k-Means Algorithmus ist wie in der herkömmlichen Variante linear. Die *bisecting* Variante erzeugt Cluster, die relativ homogene Größen haben. Dieses Ergebnis ist besser als bei der herkömmlichen Variante, welche deutlich variierende Clustergrößen erzeugt [Steinbach et al., 2000].

2.3 Segmentierung

Die in der Einleitung genannten Cluster- und Glattheitsannahmen spielen auch in der Arbeit von Hasanzadeh & Kasaei [2010] eine wichtige Rolle. Sie bezeichnen sie dort als spektrale und räumliche Redundanzen. Des Weiteren identifizieren sie als drittes die Intraklassenredundanz. Diese existiert in sehr dichten Clustern, spielt in der vorliegenden Arbeit jedoch nur eine untergeordnete Rolle. Ihr Ansatz zur multispektralen Segmentierung nutzt einen zweistufigen, pixelbasierten Prozess. Zunächst nutzen sie die Hauptkomponentenanalyse (*principle component analysis*, PCA [Jain, 1989]) zur Dimensionsreduktion gefolgt von einer Wasserscheidentransformation [Beucher & Lantuejoul, 1979]. Die entstandenen Regionen werden im zweiten Schritt mit einer gewichteten *fuzzy C-means* Methode zu Clustern gruppiert. Hierbei muss jedoch die Anzahl der Cluster als Eingabeparameter vorab bekannt sein. Abschließend nutzen sie eine angepasste *membership-connectedness*-Segmentierung [Hasanzadeh & Kasaei, 2008]. Diese geht von lokal gewählten Saatpunkten aus und erstellt zusammenhängende Objekte. Nun weist ein Maximum-Klassifikator jedes Pixel dem mit ihm am stärksten verbundenen Objekt zu. Sie argumentieren, dass dies zu einer klareren Segmentierung führt („*crisp segmentation map*“).

Ein weiterer pixelbasierter Ansatz stammt von Bruzzone & Carlin [2006]. Basierend auf einer Mehrschwellensegmentierung erstellen sie eine Hierarchie nach dem *Bottom-up*-Prinzip. Aus dieser Hierarchie extrahieren sie kontextbasiert eine große Anzahl spektraler, räumlicher und relationaler Merkmale. Da die Menge der Trainingsbeispiel im Vergleich mit der Anzahl der Merkmale bei ihnen gering ist, nutzen sie einen SVM-Klassifikator. Eine von ihnen genannte Alternative ist die Verwendung von Methoden zur Merkmalsreduktion. Diese nutzen sie jedoch nicht, da hierfür meist eine Gauß-Verteilung angenommen werden muss. Die vorliegende Arbeit wählt einen anderen Ansatz und verwendet ein *top-down*-basiertes Clusterverfahren zum Erstellen der Hierarchie. Zudem ist das hier angewendete Clusterverfahren kontextfrei.

Mit pixelbasierten Ansätzen verwandt sind superpixelbasierte Ansätze, welche zu den erfolgreichsten Segmentierungsverfahren zählen [Achanta et al., 2012]. Achanta et al. geben einen guten Überblick zu bestehenden Methoden. Sie führen einen empirischen Vergleich zwischen fünf verschiedenen Superpixel-Algorithmen durch, welche sie in die beiden Gruppen graphenbasiert und gradientenabstiegsbasiert unterteilen. Sie kommen zu dem Ergebnis, dass keine der untersuchten Methoden alle drei von ihnen aufgestellten Kriterien erfüllt: (i) Einhaltung der Bildgrenzen, (ii) schnell zu berechnen und (iii) Verbesserung der Qualität von Folgeprozessen. Als Lösungsvorschlag stellen sie eine iterative Segmentierungsmethode vor: *simple linear iterative clustering* (SLIC). Diese Methode zeigt gute Ergebnisse, die Implementierung ist frei verfügbar und einfach an eigene Vorgaben anzupassen. Sie dient daher als Grundlage des Segmentierungsschrittes der in der vorliegenden Arbeit vorgestellten Methode. Die Grundidee ist, sowohl spektrale als auch räumliche

Informationen zu verwenden. Sie definieren dazu ein gemeinsames Distanzmaß. Darauf basierend führen sie ein k-Means Clustering durch. Eine Besonderheit dabei ist, dass sie den Suchraum drastisch einschränken und so nur die Distanzen innerhalb der lokalen Nachbarschaft berechnet werden müssen. Eine detaillierte Beschreibung der Original-Methode ist im nächsten Abschnitt zu finden. Die in dieser Arbeit durchgeführten Anpassungen sind in Abschnitt 4.2 beschrieben.

2.3.1 SLIC-Algorithmus

Der SLIC-Algorithmus wurde von Achanta et al. [2012] vorgestellt und ist ein superpixelbasiertes Verfahren. Solche Verfahren lösen die rigide zweidimensionale Struktur eines Bildes auf und gruppieren die Pixel in sogenannte Superpixel. Das Ziel ist, im Bild vorhandene Redundanzen zu reduzieren und die Komplexität der weiteren Verarbeitungsschritte zu verringern. Die Basis für den SLIC-Algorithmus ist das bereits zuvor in 2.2.1 beschriebene k-Means Verfahren. Es gibt jedoch drei wichtige Unterschiede:

1. Festgelegte Initialisierung statt zufälliger Auswahl.
2. Drastische Reduktion der Anzahl zu berechnender Distanzen.
3. Einsatz eines spezialisierten Distanzmaßes.

Der Algorithmus ist sowohl für Graustufen als auch für Farbbilder mit drei Kanälen definiert; üblicher Weise rot grün blau (RGB). Für Bilder in RGB-Darstellung findet ein Vorbereitungsschritt statt, der diese in die CIELAB-Darstellung überträgt. In diesem Farbsystem sind Farben so definiert, wie sie bei Standard-Lichtbedingungen von einem Normalbeobachter wahrgenommen werden. Das heißt, insbesondere sind die Farben von der Erzeugungs-, Geräte- und Wiedergabeart unabhängig. Namensgebend ist die französische Internationale Beleuchtungskommission, *Commission Internationale de l'Éclairage* (CIE) und der L*a*b*-Farbraum. Er ist in der EN ISO 11664-4 „Colorimetry – Part 4: CIE 1976 L*a*b* Colour space“ genormt [International Organization for Standardization, 2008]. Jede Farbe ist hierbei durch drei Koordinaten eindeutig definiert:

- L^* Dieser Wert beschreibt die Helligkeit (Luminanz) der Farbe und nimmt Werte von 0 bis 100 ein. Hierbei steht L*=0 für Schwarz und L*=100 für Weiß.
- a^* Dieser Wert beschreibt die Lage der Farbe auf der Grün-Rot-Achse. Der Minimalwert -170 steht hierbei für Grün und der Maximalwert +100 steht für Rot.
- b^* Dieser Wert gibt die Koordinate auf der Gelb-Blau-Achse an. Seine Werte reichen von -100 (Blau) bis +150 (Gelb).

Anschließend findet der Initialisierungsschritt statt. Entgegen der Originalvariante werden die k_S Saatpunkte der Clusterzentren $\mathbf{c}_i$ in einem regelmäßigem Gitter über das Bild verteilt. Die Gitterweite beträgt hierbei $S = \sqrt{N/k_S}$, wobei N die Anzahl der Bildpixel ist. Die zu clusternden Elemente werden durch Konkatenation der Farb- und Koordinatenwerte der entsprechenden Pixel gebildet. Jedes Element ist somit ein fünfdimensionaler Vektor $\mathbf{q} = [\mathbf{q}^{(L*)}, \mathbf{q}^{(a*)}, \mathbf{q}^{(b*)}, \mathbf{q}^{(x)}, \mathbf{q}^{(y)}]^\top$.

Das Berechnen aller Distanzen im Zuweisungsschritt ist mit sehr hohem Aufwand verbunden. Die von Achanta et al. [2012] eingeführte Einschränkung reduziert den Aufwand jedoch drastisch gegenüber anderen Segmentierungsverfahren. Die erwartete Ausdehnung eines Superpixels beträgt $S \times S$. Für jedes Clusterzentrum c_i werden daher nur die Distanzen zu Pixeln berechnet, die in einem Teilbereich des Bildes $\mathcal{I}'_i$ der Größe $2S \times 2S$ um das Zentrum liegen:

$$\mathcal{C}'_i = \left\{\mathbf{q} : \mathbf{D}_{SLIC}(\mathbf{q}, \mathbf{c}_i) \leq \mathbf{D}_{SLIC}(\mathbf{q}, \mathbf{c}_j) \;\; \forall j = 1, \ldots, k_S, \, \mathbf{q} \in \mathcal{I}'_i\right\} \tag{2.3}$$

Die dritte Änderung, die Achanta et al. vornehmen, ist der Einsatz eines spezialisierten Distanzmaßes. Sie nutzen hierfür eine gewichtete Kombination der spektralen und räumlichen Komponenten und definieren das Distanzmaß wie folgt:

$$\mathbf{D}_{SLIC} = \sqrt{\left(\frac{\mathbf{D}_{spektral}}{m}\right)^2 + \left(\frac{\mathbf{D}_{räumlich}}{S}\right)^2} \tag{2.4}$$

Die spektralen und räumlichen Teilmaße basieren dabei auf dem euklidischen Distanzmaß:

$$\mathbf{D}_{spektral}(\mathbf{q}_i, \mathbf{q}_j) = \sqrt{\left(\mathbf{q}_i^{(L^*)} - \mathbf{q}_j^{(L^*)}\right)^2 + \left(\mathbf{q}_i^{(a^*)} - \mathbf{q}_j^{(a^*)}\right)^2 + \left(\mathbf{q}_i^{(b^*)} - \mathbf{q}_j^{(b^*)}\right)^2} \tag{2.5}$$

$$\mathbf{D}_{räumlich}(\mathbf{q}_i, \mathbf{q}_j) = \sqrt{\left(\mathbf{q}_i^{(x)} - \mathbf{q}_j^{(x)}\right)^2 + \left(\mathbf{q}_i^{(y)} - \mathbf{q}_j^{(y)}\right)^2} \tag{2.6}$$

Da die beiden Einzelmaße nicht den selben Wertebereich besitzen, werden sie über die Faktoren S und m normiert. Die Normierung basiert auf den im Suchbereich auftretenden Maximalwerten. Für die räumliche Distanz ist dies $S = \sqrt{N/K}$. Der Maximalwert für die spektrale Distanz ist jedoch a priori nicht bekannt. Achanta et al. nutzen hierfür den in der vorherigen Iteration aufgetretenen globalen Maximalwert, um m festzulegen.

Als Stoppkriterium haben sie das Residuum R untersucht. Dieses ist über die 2-Norm der Verschiebung aller Clusterzentren definiert:

$$R = \sum_{i=1}^{k_S} \sqrt{(\mathbf{c}_i - \mathbf{c}_i')^2} \tag{2.7}$$

Die Schritte werden solang wiederholt bis der Fehler konvergiert, dass heißt die Clusterzentren sich nur noch um sehr geringe Werte verschieben. Sie haben jedoch festgestellt, dass in der Regel zehn Iterationen ausreichen, um zufriedenstellende Ergebnisse zu erreichen.

2.4 Aktives Lernen

Es gibt zahlreiche Ansätze, die Theorie des aktiven Lernens zu erschließen und mathematische Beweise herzuleiten [Hanneke, 2014]. Jedoch müssen dafür teils sehr strenge Annahmen gelten, wie zum Beispiel der „realisierbare Fall“. Hierbei wird davon ausgegangen, dass sich alle Stichproben fehlerfrei linear voneinander trennen lassen und ein Verfahren existiert, das diesen perfekten Klassifikator zuverlässig finden kann [Cohn et al., 1994; Dasgupta, 2005; Balcan & Blum, 2005]. Unter diesen strengen Voraussetzungen lassen sich beweisbare Aussagen über den Erfolg des aktiven Lernens treffen, siehe hierzu auch Abschnitt 3.1.1. Es gibt Arbeiten, die diese Anforderungen abschwächen und versuchen Aussagen, auch unter dem Vorhandensein von Rauschen, zu treffen [Balcan et al., 2006; Kääriäinen, 2006]. Diese Arbeiten blieben jedoch vorwiegend theoretischer Natur und wurden nicht auf Fernerkundungsdaten getestet.

Die vorliegende Arbeit soll auf Daten angewendet werden, die mit Sensoren der Photogrammetrie und Fernerkundung aufgenommen wurden. Solche Daten verletzen die obigen Annahmen, da sie immer mit Rauschen belegt sind. Es gibt jedoch Ansätze und Konzepte, die ohne diese strengen Annahmen auskommen und auch auf Daten gute Ergebnisse liefern, die unter natürlichen Bedingungen aufgenommen wurden (im Vergleich zu synthetischen Daten aus kontrollierten Bedingungen) [Beygelzimer et al., 2010; Wuttke et al., 2015]. Im Folgenden wird auf Methoden eingegangen, die thematisch eng mit diesen Ansätzen verwandt sind.

Eine der ersten Arbeiten, die hierarchische Clusterbildung mit aktivem Lernen verbindet, stammt von Dasgupta & Hsu [2008]. Die Motivation für ihre Arbeit ist der *Sampling Bias* des

aktiven Lernens. Dieser führt dazu, dass das Lernverfahren suboptimale Ergebnisse liefert, falls zu Beginn des Trainings ungünstige Stichproben ausgewählt wurden. Dies ist der Fall, wenn Ausreißer verwendet werden oder ganze Cluster unentdeckt bleiben. Um dies zu verhindern, schlagen sie vor, die Auswahl der Stichproben durch ein clusterbasiertes Vorgehen zu steuern. Voraussetzung ist eine bereits durchgeführte hierarchische Clusterbildung der Daten, so dass eine Baumstruktur vorliegt. Mit Informationen aus dieser Hierarchie kann anschließend die Auswahl der Stichproben so gesteuert werden, dass keine Cluster unentdeckt bleiben und Ausreißer keinen negativen Einfluss haben. Liegen ausreichend viele Informationen über die zugeordneten Klassen vor, kann der Baum gestutzt werden. Anschließend überführt ein Mehrheitsentscheid für jedes Blatt den gestutzten Baum in eine Klassifikation für den gesamten Datensatz.

Dasgupta und Hsu wendeten ihre Methode nur auf Probleme der optischen Zeichenerkennung und natürlichsprachlichen Textverarbeitung an, die Grundlagen lassen sich jedoch auch auf die Landbedeckungsklassifikation übertragen [Tuia et al., 2012]. Tuia et al. erweiterten das Verfahren mit Methoden des aktiven Lernens um eine spezielle Auswahlstrategie. Diese basiert auf der Zuverlässigkeit der Klasseninformationen und der Größe des aktuell betrachteten Clusters. Sie untersuchen dabei sechs verschiedene Strategien, wobei vier hiervon teilweise zufällig vorgehen und somit eher zur Kategorie des passiven Lernens zählen. Sie wenden die erweiterte Methode auf drei verschiedene Fernerkundungsdatensätze an und zeigen, dass aktives Lernen auf allen drei besser ist als passives Lernen.

Muñoz-Marí et al. [2012] ist die Folgearbeit der gleichen drei Autoren. Hierin reduzieren sie die Berechnungskomplexität für das Erstellen der Clusterhierarchie. Anstatt *Ward's minimum variance* [Ward, 1963] setzen sie den *bisecting* k-Means Algorithmus [Kashef & Kamel, 2009] ein. Hierbei wird k-Means mit dem Wert $k = 2$ wiederholt angewendet, bis der gesamte Datensatz unterteilt oder ein Stoppkriterium erfüllt ist. Durch den kleinen Wert von k reduziert sich der Aufwand deutlich und ermöglicht es somit auch größere Datensätze effizient zu verarbeiten. In ihrer Arbeit nutzen sie jedoch keine Segmentierung, so dass der Vorteil der Stabilisierung der Eingangsdaten nicht genutzt wird. Dies wurde vom Autor der vorliegenden Arbeit in [Wuttke et al., 2017] gezeigt. Diese Veröffentlichung stellt gleichzeitig den Vorläufer zu der hier vorgestellten Methode dar.

Eine Alternative für den Umgang mit Ausreißern stellen Zhu et al. [2010] vor. Sie argumentieren, dass der *Selection Bias* des aktiven Lernens vor allem durch die Verwendung der Unsicherheit bzw. Entropie als Auswahlkriterium auftritt („*Uncertainty Sampling*“). Ausreißer zeichnen sich vor allem dadurch aus, dass sie in Regionen des Merkmalsraums geringer Dichte auftreten. Um das Auswählen von Ausreißern zu reduzieren, integrieren Zhu et al. die lokale Dichte in die Selektionsstrategie. Als Maß für die Dichte nutzen sie die mittlere Entfernung der k nächsten Nachbarn zu der in Frage kommenden Stichprobe. Für die finale Selektionsstrategie schlagen sie zwei Varianten vor: (i) Multiplikation von Dichte- und Unsicherheitsmaß und (ii) Sortierung nach Unsicherheit mit anschließender Neusortierung der besten N Kandidaten mit dem Dichtemaß. Sie wenden die Methode auf sechs Datensätze aus dem Bereich der natürlichsprachlichen Textverarbeitung an. Ihr Ergebnis ist, dass beide dichtebasierten Methoden signifikant besser sind als reines *Uncertainty Sampling*. Untereinander ergaben sich jedoch gemischte Ergebnisse. Sie stellen außerdem fest, dass die dichtebasierten Selektionsstrategien schlechte Ergebnisse liefern, wenn schiefe Klassenverteilungen vorliegen. Der Ansatz dichtebasierte Informationen mit aktivem Lernen zu verbinden, erscheint vielversprechend und wird in der vorliegenden Arbeit weiter verfolgt.

2.4.1 Probabilistisches aktives Lernen

Ein sehr junger Ansatz, Dichteinformationen in Methoden des aktiven Lernens zu integrieren, ist probabilistisches aktives Lernen (PAL). Arbeiten hierzu, die methodisch mit der vorliegenden Arbeit verwandt sind, werden im Folgenden Abschnitt vorgestellt.

Die PAL-Rahmenstruktur wurde von Krempl et al. [2014a,b] veröffentlicht und basiert auf der Glattheitsannahme (siehe auch Abschnitt 3.4). Die Schlussfolgerung die sie ziehen ist, dass die Nützlichkeit der Informationen, die durch das Versehen einer Stichprobe mit einem Klassenlabel gewonnen werden, von zwei Faktoren abhängt: (i) der A-posteriori-Wahrscheinlichkeit des Klassenlabels und (ii) der Dichte der Stichproben in der Nachbarschaft. In den ersten Arbeiten ist die PAL-Rahmenstruktur nur für Zwei-Klassen-Probleme definiert (lediglich die Klassen „positiv“ und „negativ“).

Sie modellieren die wahre A-posteriori-Wahrscheinlichkeitsverteilung P der positiven Klasse als eine Beta-verteilte Zufallsvariable. Deren Realisierung p wird als Parameter der Bernoulli-verteilten Zufallsvariable Y verwendet. Die Realisierung dieser bestimmt schließlich das Klassenlabel des Kandidaten. Hieraus ergibt sich, dass die Anzahl der positiven Stichproben in der Nachbarschaft binomialverteilt ist.

$$\begin{aligned} P &\sim \text{Beta}_{n \cdot \hat{p}+1, n \cdot (1-\hat{p})+1} \\ Y &\sim \text{Bernoulli}_{p} \\ (n \cdot \hat{p}) &\sim \text{Binomial}_{n,p} \end{aligned} \tag{2.8}$$

Der Parameter $\hat{p}$ ist hierbei der Anteil positiver Klassenlabel in der Nachbarschaft und n die absolute Anzahl bekannter Klassenlabel. Zusammengefasst ergeben beide Werte die Label-Statistik $ls = (n, \hat{p})$. Die erwartete Nützlichkeit einer Stichprobe (probabilistischer Nutzen, **pgain**) kann nun über den Erwartungswert der beiden Zufallsvariablen bestimmt werden:

$$\mathbf{pgain}(ls) = E_p \left[E_y \left[\mathbf{gain}_p(ls, y) \right] \right] \tag{2.9}$$

Der herkömmliche Nutzen (**gain**) einer Stichprobe ist dabei über die Leistungssteigerung durch Hinzufügen der neuen Stichprobe definiert:

$$\mathbf{gain}_p(ls, y) = \mathbf{perf}_p \left(\frac{n\hat{p} + y}{n+1} \right) - \mathbf{perf}_p(\hat{p}) \tag{2.10}$$

Als Leistungsfunktion **perf** (englisch *performance*) kann zum Beispiel die Klassifikationsgenauigkeit verwendet werden.

Der probabilistische Nutzen **pgain** wird nun noch mit der lokalen Dichte $d_{\mathbf{x}}$ gewichtet, so dass die nützlichste Stichprobe bestimmt werden kann:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{U}}{\text{argmax}} \left(d_{\mathbf{x}} \cdot \mathbf{pgain}(ls_{\mathbf{x}}) \right) \tag{2.11}$$

Eine Erweiterung der PAL-Rahmenstruktur spezialisiert sich auf Fälle, in denen die Kosten einer Fehlklassifikation nicht für alle Klassen identisch ist. Das heißt, ein falsch-positiv Fehler verursacht andere Fehlerkosten als ein falsch-negativ Fehler. In ihrer Arbeit über optimiertes probabilistisches aktives Lernen (OPAL) [Krempl et al., 2015] integrieren Krempl et al. dies, indem sie die Fehlklassifikationskosten mit dem Informationsgewinn vergleichen und eine Fallunterscheidung durchführen. Eine weitere Ergänzung ist die Erweiterung der PAL-Rahmenstruktur auf mehrere gleichzeitige Anfragen. Diese Betrachtungen wurden jedoch nur für binäre Zwei-Klassen-Probleme durchgeführt, so dass sie in dieser Arbeit nicht direkt angewendet werden können.

Dieser Nachteil wurde in der Erweiterungen der PAL-Rahmenstruktur für Mehrklassenprobleme behoben: multi-class PAL (mcPAL) [Kottke et al., 2016]. Kottke et al. ersetzen hierbei ihre vorherige binomiale Modellierung der A-posteriori-Wahrscheinlichkeit durch eine multinomiale Modellierung. Anschließend überführen sie die aufgestellten Gleichungen in eine geschlossene Form. Ihre Methode wenden sie auf sechs verschiedene Datensätze an, von denen jedoch

keiner aus der Fernerkundung stammt. Zudem handelt es sich nur um sehr kleine Datensätze, da die vorgeschlagene Methode trotz der Vereinfachungen eine quadratische Komplexität besitzt (siehe hierzu auch Abschnitt 4.5). Dennoch ist die Mehrklassen-Erweiterung sehr nützlich, da die interessantesten Probleme der Landbedeckungsklassifikation mehr als zwei Klassen enthalten.

2.5 Historische Entwicklung und Abgrenzung

Hier soll ein kurzer historischer Überblick zur Entwicklung des aktiven Lernens anhand ausgewählter Arbeiten gegeben werden. Er beginnt mit den ersten Zügen im *Optimal Experimental Design* 1972, geht über die erste veröffentlichte Nennung 1990, bis zu den jüngsten Entwicklungen um das probabilistische aktive Lernen 2014.

1972 [Fedorov] Optimal Experimental Design

1988 [Angluin] Queries and Concept Learning

1990 [Atlas et al.] Erste Nennung „active learning“

1992 [Seung et al.] Query by Committee

1994 [Lewis & Catlett] Uncertainty Sampling

1994 [Cohn et al.] Expected error reduction

1998 [McCallum & Nigam] Expectation Maximization für Pool-basiertes aktives Lernen

2000 [Schohn & Cohn] Aktives Lernen mit Stützvektormaschinen

2000 [Campbell et al.] Aktives Lernen für Large Margin Klassifikatoren

2000 [Tong & Koller] Bayessches aktives Lernen

2001 [Roy & McCallum] Expected error reduction

2005 [Souvannavong et al.] Partition Sampling, Maximierung des globalen Informationsgewinns

2006 [Balcan et al.] Agnostisches aktives Lernen

2007 [Hanneke] Disagreement Coefficient

2008 [Settles et al.] Expected Model Change

2008 [Settles & Craven] Information Density Framework (Exploration vs. Erschließung)

2012 [Settles] Erstes Buch über Aktives Lernen

2014 [Krempl et al.] Probabilistisches aktives Lernen

Die vorangegangenen Abschnitte präsentierten und charakterisierten Methoden, die mit der vorliegenden Arbeit verwandt sind. Der nennenswerteste Unterschied ist, dass keine der Methoden alle drei Ansätze von Segmentierung, Clusterbildung und aktivem Lernen gleichzeitig kombiniert. Tabelle 2.1 gibt hierzu eine Übersicht der präsentierten Arbeiten. Vorläufer der in dieser Arbeit vorgestellten Methode wurden vom Autor bereits in [Wuttke et al., 2017] und [Wuttke et al., 2018] veröffentlicht.

Tabelle 2.1: Übersicht zu verwandten Arbeiten aus den Bereichen Segmentierung (S), Clusterhierarchien (CH) und aktives Lernen (AL). Hervorgehoben sind Arbeiten des Autors.

Methode	S	CH	AL
[Lee & Crawford, 2004] Multistage hierarchical clustering	☒	☒	☐
[Lee & Crawford, 2005] Bayesian multistage hierarchical clustering	☒	☒	☐
[Marcal & Castro, 2005] Spectral and spatial hierarchical clustering	☐	☒	☐
[Bruzzone & Carlin, 2006] Context-driven feature extraction	☒	☒	☐
[Dasgupta & Hsu, 2008] Hierarchical active learning	☐	☒	☒
[Hasanzadeh & Kasaei, 2010] PCA, watershed, fuzzy c-means	☒	☐	☐
[Senthilnath et al., 2012] Mean shift clustering and k-means	☐	☒	☐
[Tuia et al., 2012; Muñoz-Marí et al., 2012] Active queries	☐	☒	☒
[Krempl et al., 2014b] Probabilistic active learning (PAL)	☐	☐	☒
[Krempl et al., 2015] Optimized probabilistic active learning (OPAL)	☐	☐	☒
[Huo et al., 2015] SVM with RBF-kernel & similarity measures	☐	☒	☐
[Kottke et al., 2016] Multi-class probabilistic active learning (mcPAL)	☐	☐	☒
[Wuttke et al., 2017, 2018] Segmented active queries	☒	☒	☒
Diese Arbeit	☒	☒	☒

3 Grundlagen

Dieses Kapitel stellt die theoretischen Grundlagen des aktiven Lernens vor, die für Kapitel 4 benötigt werden. Es folgen Erläuterungen zur Landbedeckungsklassifikation, Clusterbildung und Segmentierung.

3.1 Aktives Lernen

Ein zentrales Konzept der vorliegenden Arbeit ist das aktive Lernen. Dabei handelt es sich um eine Variante des überwachten maschinellen Lernens. Das heißt, neben den zu unterscheidenden Elementen werden auch Informationen über deren Klassenlabel benötigt. Dabei entsteht der Trainingsaufwand nicht durch das Beschaffen der Elemente – denn diese stehen meist in großer Zahl zur Verfügung – sondern im Beschaffen der Klasseninformationen, der *Ground Truth*. Die Quelle dieser Klasseninformationen ist oftmals ein nur sehr aufwendig zu automatisierender Entdeckungsprozess oder ein nicht vollständig verstandener Zusammenhang. Beispiele hierfür sind das Extrahieren semantischer Informationen aus Texten oder die Bestimmung der Landbedeckung ausschließlich aus dem Spektrum des reflektierten Lichts. Wäre das Bestimmen der Klasseninformationen einfach möglich, bestünde kein Bedarf für den Einsatz von maschinellen Lernverfahren. Da die Details des Entdeckungsprozesses oft unbekannt sind, hat sich in der Literatur zu aktivem Lernen die Bezeichnung „Orakel“ durchgesetzt. Die Herausforderung ist, eine gute Klassifikationsleistung des maschinellen Lernverfahrens zu erreichen und dabei möglichst wenig Kosten beim Beschaffen der Klassenlabel zu verursachen.

Aktives Lernen versucht diese Kosten gering zu halten, indem es jene Stichproben identifiziert, die dem Trainingsprozess besonders stark helfen, falls ihr Klassenlabel beschafft wird. Der nächste Abschnitt gibt zur Einführung ein kurzes Beispiel. Anschließend wird das allgemeine Schema erläutert und einzelne Bestandteile detailliert beschrieben. Es folgt eine Unterteilung in verschiedene Einsatzszenarien.

3.1.1 Einführendes Beispiel

Ein einfaches Beispiel soll das Prinzip des aktiven Lernens verdeutlichen. Es folgt den Erläuterungen aus Hanneke [2014]. Gegeben sei eine Menge von Zahlen $\mathcal{X}$ aus dem Intervall $[0, 1]$ und ein Schwellenwert t ebenfalls aus diesem Intervall. Jede der Zahlen die kleiner oder gleich dem Schwellenwert ist, sei der Klasse -1 zugeordnet und jede Zahl die größer als der Schwellenwert ist, sei der Klasse +1 zugeordnet.

Die Klassenlabel seien in der Menge $\mathcal{Y}$ zusammengefasst.

$$\begin{aligned}
&\mathcal{X} = \{x_1, x_2, \ldots, x_n\}, n \in \mathbb{N}, x_i \in \mathbb{R}, 0 \leq x \leq 1 \\
&\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}, n \in \mathbb{N}, y_i \in \{-1, +1\} \\
&t \in \mathbb{R}, 0 \leq t \leq 1 \\
&y_i = \begin{cases} -1, \text{ falls } x_i \leq t \\ +1, \text{ falls } x_i > t \end{cases}
\end{aligned} \tag{3.1}$$

Es seien nur einige Zahlen aus $\mathcal{X}$ und ihre zugehörigen Klassenlabel aus $\mathcal{Y}$ bekannt, jedoch nicht der wahre Wert von t. Aufgabe eines Klassifikators ist es nun, basierend auf der Eingabe von Tupeln $(x, y), x \in \mathcal{X} \times y \in \mathcal{Y}$ den Wert $\hat{t}$ zu schätzen, so dass dieser um maximal ε vom wahren Wert des Schwellenwertes t abweicht. Abbildung 3.1 stellt dieses Beispiel grafisch dar.

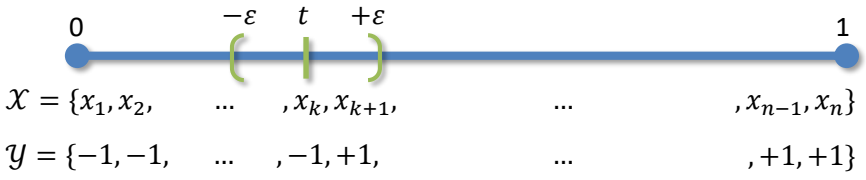


Abbildung 3.1: Einführendes Beispiel der Schwellenwertsuche auf dem Intervall [0,1]. Ziel ist es, den unbekannten Schwellenwert t mit möglichst wenig Trainingsbeispielen zu bestimmen, so dass der Fehler kleiner als ε ist.

In diesem einfachen Beispiel gibt es keine Überlappung zwischen den Klassen. Das heißt, sie lassen sich anhand eines einzigen Schwellenwertes perfekt trennen. Dies wird in der Literatur als der „realisierbare Fall“ bezeichnet. Eine sehr einfache Konstruktion eines Klassifikators ist das Festlegen des Schwellenwertes auf den Mittelwert der größten Zahl mit Klasse -1 und der kleinsten Zahl mit Klasse +1, basierend auf der Menge der bekannten Tupel $\{(x_i, y_i)\}$.

$$\begin{aligned}
x_{max} &= \max_{i=1}^{n} (\{x_i : y_i = -1\} \cup \{0\}) \\
x_{min} &= \min_{i=1}^{n} (\{x_i : y_i = +1\} \cup \{1\}) \\
\hat{t} &= \frac{x_{min} + x_{max}}{2}
\end{aligned} \tag{3.2}$$

Der Erfolg dieses Klassifikators hängt von der Lage der vorhandenen Trainingsbeispiele ab. Im klassischen maschinellen Lernen wird keine besondere Strategie zur Auswahl der Trainingsbeispiele verfolgt. Sie werden „passiv“ bestimmt, das heißt es handelt sich um zufällig gezogene Stichproben. Damit der Fehler gemäß Aufgabenstellung akzeptabel ist, muss gelten: $|\hat{t} - t| \leq \varepsilon$. Eine hinreichende Bedingung für diesen Fall ist, dass jeweils mindestens ein Trainingsbeispiel in den Intervallen $\mathcal{I}_{-1} = [t - \varepsilon, t]$ und $\mathcal{I}_{+1} = (t, t + \varepsilon]$ liegt. Es gilt somit:

$$\exists x_i, x_j \in \mathcal{X} : i \neq j, x_i \in \mathcal{I}_{-1} \wedge x_j \in \mathcal{I}_{+1} \Rightarrow \text{abs}(t - \hat{t}) \leq \varepsilon \tag{3.3}$$

Die Größe der Grundgesamtheit ist 1 und die Größe der Intervalle $\mathcal{I}_{-1}$ und $\mathcal{I}_{+1}$ ist jeweilsε. Somit beträgt die Wahrscheinlichkeit, dass ein zufällig gezogenes Element außerhalb eines der Intervalle liegt $1 - \varepsilon$. Die Wahrscheinlichkeit, dass ein Intervall nach n zufällig aus der Grundgesamtheit gezogenen Elementen leer ist, beträgt $(1 - \varepsilon)^n$. Dieses Ereignis sei bezeichnet als L_i

mit $i \in \{-1, +1\}$. Für die gemeinsame Betrachtung von Ereignissen beider Intervalle kann die Boolesche Ungleichung (siehe Formel 3.4) herangezogen werden. Diese besagt, dass die Summenwahrscheinlichkeit für das Eintreten einer Gruppe von Ereignissen gleich oder kleiner als die Summe der einzelnen Eintrittswahrscheinlichkeiten ist [Hazewinkel, 2002]. Die Wahrscheinlichkeit, dass mindestens eines der Intervalle leer bleibt, kann somit wie folgt nach oben abgeschätzt werden:

$$\begin{aligned} \mathbb{P}\left(\bigcup_i L_i\right) &\leq \sum_i \mathbb{P}(L_i) && (3.4)\\ &= \mathbb{P}(L_{-1}) + \mathbb{P}(L_{+1}) \\ &= (1-\varepsilon)^n + (1-\varepsilon)^n \\ &= 2(1-\varepsilon)^n && (3.5) \end{aligned}$$

Im Umkehrschluss beträgt somit die Wahrscheinlichkeit, dass bei n zufällig gezogenen Elementen keines der beiden Intervalle leer bleibt, mindestens $1 - 2(1-\varepsilon)^n$. Tritt dieses Ereignis ein, befindet sich in beiden Intervallen mindestens ein Element und der Fehler ist akzeptabel.

Die Frage ist nun, wie groß n gewählt werden muss, so dass ein akzeptables Ergebnis mit ausreichend hoher Wahrscheinlichkeit $(1-\delta)$ eintritt. Hierbei hilft die Abschätzung $1 - \varepsilon \leq e^{-\varepsilon}$.

$$\begin{aligned} 1 - 2(1-\varepsilon)^n \geq 1 - 2(e^{-\varepsilon})^n &\overset{!}{\geq} 1 - \delta \\ -2e^{-\varepsilon n} &\geq -\delta \\ 2e^{-\varepsilon n} &\leq \delta \\ \ln 2 + \ln e^{-\varepsilon n} &\leq \ln \delta \\ -\varepsilon n &\leq \ln \delta - \ln 2 \\ \varepsilon n &\geq \ln 2 - \ln \delta \\ n &\geq \frac{1}{\varepsilon} \ln \frac{2}{\delta} && (3.6) \\ \rightsquigarrow n &\in \mathcal{O}\left(\frac{1}{\varepsilon}\right) && (3.7) \end{aligned}$$

Wird die Wahrscheinlichkeit δ, mit der ein akzeptabler Fehler erreicht wird, auf einen bestimmten Wert festgelegt, ist der Term $\ln \frac{2}{\delta}$ konstant und kann für Aufwandsabschätzungen vernachlässigt werden. Das Resultat ist eine Aussage über die benötigte Anzahl von Trainingsbeispielen des hier betrachteten passiven Klassifikators. Es sind $1/\varepsilon$ Stichproben notwendig, damit der Klassifikationsfehler kleiner als ε ist.

Im Vergleich hierzu folgt die Untersuchung eines aktiven Klassifikators. Es werden die Werte U und O eingeführt: die untere und obere Schranke. Diese werden mit $U = 0$ und $O = 1$ initialisiert. Anschließend wird der Wert x^* bestimmt und dessen Klassenlabel y^* beim Orakel angefragt. Basierend auf der Antwort findet entweder eine Aktualisierung der unteren oder der oberen Schranke statt:

$$\begin{aligned} x^* &= \frac{O+U}{2} && (3.8)\\ \text{Falls } y^* = +1 &\rightsquigarrow O := x^* \\ \text{Falls } y^* = -1 &\rightsquigarrow U := x^* \end{aligned}$$

Die Anfrage der Klassenlabel und die Aktualisierung der Schranken wird solang wiederholt, bis ein Stoppkriterium erfüllt ist. Der geschätzte Schwellenwert ist anschließend der Mittelwert aus oberer und unterer Schranke: $\hat{t} = (O+U)/2$. Der Algorithmus bestimmt somit „aktiv“ welche Stichproben angefragt und für das Training verwendet werden.

Nach der Initialisierung ist $\hat{t} = 0,5$ und es gilt $|\hat{t} - t| \leq 0,5$. Nach dem ersten Schritt gilt entweder $O = 0,5$ oder $U = 0,5$ und somit $\hat{t} = 0,25$ oder $\hat{t} = 0,75$. Es folgt $|\hat{t}-t| \leq 0,25$. In jedem weiteren Schritt wird der maximale Restfehler halbiert. Die allgemeine Berechnungsvorschrift des Restfehlers ist somit: $\varepsilon = (1/2)^{n+1}$. Diese Gleichung soll nun so umgestellt werden, dass die Anzahl der benötigten Trainingsbeispiele n in Abhängigkeit des zu unterschreitenden Restfehlers ε angegeben werden kann:

$$
\begin{aligned}
\left(\frac{1}{2}\right)^{n+1} &\leq \varepsilon \\
2^{-(n+1)} &\leq \varepsilon \\
\log_2 2^{-(n+1)} &\leq \log_2 \varepsilon \\
-(n+1) &\leq \log_2 \varepsilon \\
n+1 &\geq -\log_2 \varepsilon \\
n+1 &\geq \log_2 \left(\frac{1}{\varepsilon}\right) \\
n &\geq \log_2 \left(\frac{1}{\varepsilon}\right) - 1 \qquad (3.9) \\
\rightsquigarrow n &\in \mathcal{O}\left(\log \frac{1}{\varepsilon}\right) \qquad (3.10)
\end{aligned}
$$

Der aktive Klassifikator benötigt somit logarithmisch weniger Trainingsbeispiele als der passive Klassifikator. Dies ist eine exponentielle Reduktion des Trainingsaufwandes. Dies ist intuitiv verständlich, da die Strategie des aktiven Klassifikators dem Vorgehen der Binärsuche entspricht und diese den Suchraum in jeder Iteration halbiert.

Dieses einfache Beispiel verdeutlicht die Grundidee des aktiven Lernens anhand eines sehr einfachen Klassifikators. Im Folgenden wird ein allgemeines Schema für aktive Lernverfahren vorgestellt, welches sich auch auf komplexere Beispiele anwenden lässt.

3.1.2 Allgemeines Schema

Die zu klassifizierenden Daten sind d-dimensionale Vektoren und werden als Stichproben bezeichnet $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \mathbf{x} \in \mathbb{R}^d, d, n \in \mathbb{N}$. Ziel ist es, das Klassenlabel $y \in \Omega = \{1...c\}$ für jede Stichprobe vorherzusagen. Ein allgemeines aktives Lernverfahren ist ein Fünf-Tupel: $\mathbf{C}, \mathbf{O}, \mathcal{U}, \mathcal{L}, \mathbf{S}$.

- **Klassifikator C:** Prognostiziert auf den Trainingsdaten $\mathcal{L}$ basierend das Klassenlabel y einer Stichprobe $\mathbf{x}$: $\mathbf{C}_{\mathcal{L}} : \mathcal{X} \to \Omega, \mathbf{x} \mapsto y$
- **Orakel O:** Stellt unter Ressourcenaufwand das wahre Klassenlabel bereit: $\mathbf{O} : \mathcal{X} \to \Omega, \mathbf{x} \mapsto y$
- **Ungelabelte Stichproben $\mathcal{U}$:** Menge aller Stichproben, deren jeweiliges Klassenlabel dem Klassifikator noch unbekannt ist und erst beim Orakel angefragt werden muss: $\mathcal{U} = \{\mathbf{x}_i\}_{i=l}^{u}$.
- **Gelabelte Stichproben $\mathcal{L}$:** Menge von Stichproben, deren jeweiliges Klassenlabel bekannt ist. Solch ein Paar wird auch als Trainingsbeispiel bezeichnet. $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=u+1}^{u+l}$ mit $u \gg l$

- **Selektionsstrategie S:** Eine Funktion, die jeder Stichprobe einen Nützlichkeitswert zuordnet anhand dessen entschieden wird, welche Stichprobe beim Orakel angefragt werden soll: $\mathbf{S} : \mathcal{X} \to \mathbb{R}, \mathbf{x} \mapsto s$

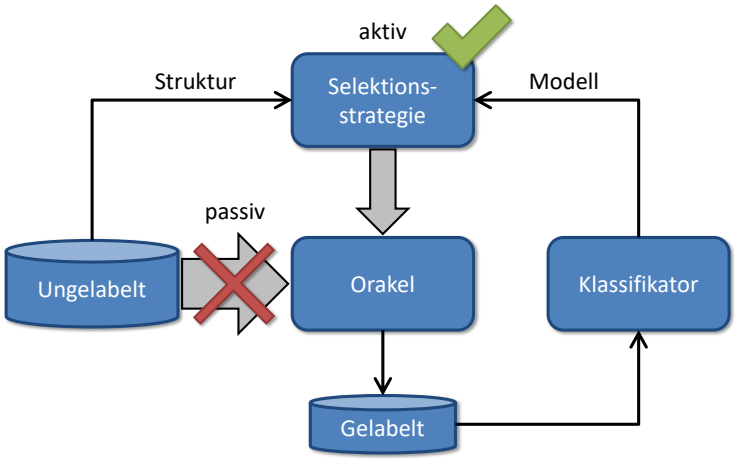


Abbildung 3.2: Schematischer Unterschied zwischen passivem und aktivem Lernen. Im Gegensatz zu herkömmlichen („passiven“) Lernverfahren kommt beim aktiven Lernen eine Selektionsstrategie zum Einsatz. Diese Strategie reduziert die Menge der mit Klassenlabeln zu versehenden Stichproben deutlich. Sie kann dabei auf strukturellen Informationen der Daten selbst oder auf dem aktuellen Modell des Klassifikators beruhen.

Das Zusammenspiel der genannten Bestandteile ist in Abbildung 3.2 dargestellt. In den meisten Anwendungsgebieten überwachter Klassifikation steht eine sehr große Menge ungelabelter Stichproben zur Verfügung. Diese Stichproben werden bei passiven Verfahren direkt durch einen externen Prozess („Orakel“) mit einem Klassenlabel versehen. Der hierdurch entstehende kleinere Pool bildet die Trainingsmenge, welche dem Training des Klassifikators zugrunde liegt. Der Unterschied beim aktiven Lernen ist, dass keine große Menge Klassenlabel beschafft wird, da dies mit einem großen Ressourceneinsatz einhergehen würde. Stattdessen wird eine Bewertung aller Stichproben anhand einer Strategie durchgeführt und lediglich die geeignetsten ausgewählt. Diese Strategie basiert auf strukturellen Informationen wie zum Beispiel identifizierten Clustern in den Daten oder dem aktuellen gelernten Modell des Klassifikators. Durch die bessere Eignung der Trainingsbeispiele wird erreicht, dass eine geringere Menge von Klassenlabeln angefragt werden muss und dennoch die gleiche Klassifikationsgenauigkeit wie bei passiven Verfahren möglich ist. Dieses Vorgehen reduziert somit den Trainingsaufwand bei gleichbleibender Klassifikationsleistung.

Der Kreislauf von Stichprobe selektieren, Orakel anfragen und Klassifikator trainieren wird solang fortgeführt, bis ein Stoppkriterium erfüllt ist. Hierauf geht Abschnitt 3.1.5 genauer ein.

3.1.3 Orakel

Wie bereits erwähnt, bezeichnet „Orakel“ den Prozess, der die wahren Klassenlabel zur Verfügung stellt. Dies ist für unüberwachte Klassifikation unerlässlich. In passiven Lernverfahren wird nicht darauf eingegangen woher diese Informationen stammen oder wie sie beschafft werden. Es ist jedoch in der Regel so, dass dies so aufwendig ist, dass es nicht in ausreichend großer Menge durchgeführt werden kann. Wäre dies der Fall, gäbe es keinen Bedarf das Klassifikationsverfahren zu automatisieren.

In vielen Anwendungen der Fernerkundung werden die Klassenlabel als *Ground Truth* bezeichnet und durch den Menschen bestimmt. Dies kann *in situ* geschehen, durch Bildinterpretation oder durch Adaption anderer Informationsquellen. In jedem Fall bedeutet dies jedoch einen hohen Ressourcenaufwand. Beispiele für diesen Aufwand sind Geld für Reisen zu schwer erreichbaren Gebieten und menschliche Arbeitszeit für manuelle Bildinterpretation. Es ist daher wünschenswert, diesen Ressourceneinsatz zu reduzieren oder zumindest so effizient wie möglich zu nutzen. Genau hierbei soll das aktive Lernen Unterstützung bieten.

Die Art, wie beim Orakel angefragt wird, kann sehr unterschiedlich sein. In [Wuttke et al., 2012] untersuchte der Autor der vorliegenden Arbeit, ob der Trainingsaufwand für sowohl Experten als auch Laien reduziert wird, wenn sie Unterstützung durch aktives Lernen erhalten. Die Unterstützung geschah dabei durch die Anzeige von Regionen in denen der Klassifikator noch sehr unsicher ist. Das Ergebnis zeigte, dass beide Gruppen deutlich weniger Stichproben markieren mussten im Vergleich zur Situation ohne Rückmeldung über den Zustand des Klassifikators. Experten, die aufgrund ihrer Erfahrung den Zustand des Klassifikators gut einschätzen konnten, mussten generell nur wenige Trainingsbeispiele markieren, um zu guten Ergebnissen zu kommen. Dennoch half ihnen die Unterstützung durch aktives Lernen besser geeignete Trainingsbeispiele auszuwählen, so dass die Klassifikationsleistung stieg. Laien benötigten in der Regel deutlich mehr Trainingsbeispiele, konnten durch die aktive Unterstützung die Menge jedoch um bis zu 75% reduzieren.

Der beim Orakel tatsächlich entstehende Aufwand ist nur sehr schwer zu modellieren und von Aufgabe zu Aufgabe unterschiedlich. Daher wird in der vorliegenden Arbeit als Verallgemeinerung die Anzahl an getätigten Anfragen als Approximation für den Aufwand verwendet.

3.1.4 Selektionsstrategie

Der wichtigste Unterschied zwischen herkömmlichem „passiven“ und dem hier vorgestellten „aktiven“ Lernen ist der Einsatz einer speziellen Selektionsstrategie. Beispiele für häufig eingesetzte passive Methoden zur Auswahl der Trainingsbeispiele sind laut Congalton [1991]:

- **Zufälliges Ziehen:** Dies ist die einfachste Methode. Die für das Training zu verwendenden Stichproben werden zufällig aus der Grundgesamtheit gezogen.
- **Stratifizierte Stichprobe:** Hierbei wird die Grundgesamtheit zunächst nach den gewünschten Eigenschaften in mehrere Schichten unterteilt. Aus diesen wird anschließend zufällig gezogen und jeweils mit dem Umfang der Schicht gewichtet.
- **Gitterbasierte Stichprobe:** Eine insbesondere in der Fernerkundung oft eingesetzte Variante ist, ein regelmäßiges Gitter über das zu untersuchende Gebiet zu legen und die Klassenlabel nur für die Gitterpunkte zu bestimmen.
- **Expertenbasiert:** Hierbei wählt ein mit dem zu untersuchenden Gebiet und der angewendeten Methode vertrauter Experte manuell die Stichproben aus, zu denen die Klassenlabel beschafft werden sollen.

Congalton [1991] geht auf die einzelnen Nachteile dieser Auswahlmethoden ein. Er sagt, dass bei zufälligem Ziehen unterrepräsentierte Stichproben vernachlässigt werden und für stratifiziertes Ziehen die wahre Verteilung der Grundgesamtheit sehr genau bekannt sein muss, was in vielen Anwendungen oft nicht der Fall ist. Wird beim gitterbasierten Ziehen das Gitter unabhängig vom zu untersuchenden Gebiet gewählt, entspricht es im Ergebnis dem zufälligem Ziehen. Wird das Gitter hingegen an lokale Gegebenheiten angepasst, kann es zu Wiederholungseffekten kommen [Congalton, 1991; Mu et al., 2015]. Ein Gegenvorschlag sind expertengestützte oder systematische

Auswahlverfahren. Diese können als Zwischenschritt auf der Entwicklung zum aktiven Lernen betrachtet werden, da sie zwar auf die besonderen Gegebenheiten des konkreten Problems eingehen, dies jedoch nur einmal vor Beginn des Trainingsprozesses tun und sich nicht an die fortschreitende Entwicklung des Klassifikators anpassen. Ebenso findet hierbei die Auswahl weiterhin manuell statt. Aktives Lernen setzt diese Entwicklung konsequent fort und bewertet die Nützlichkeit automatisiert in jeder Iteration erneut.

Der Erfolg von aktiven Selektionsstrategien hängt vom Zusammenspiel mit dem Klassifikator ab, wie vom Autor in [Wuttke et al., 2014] gezeigt. Einige Strategien erfordern, dass der Klassifikator nicht nur das Klassenlabel sondern auch dessen A-posteriori-Wahrscheinlichkeit angibt. Andere wiederum stellen keine Anforderungen an den Klassifikator selbst, erfordern jedoch den Einsatz mehrerer Klassifikatoren gleichzeitig. Einige der wichtigsten Strategien werden im Folgenden vorgestellt. Eine weitere mögliche Einteilung der verschiedenen Selektionsstrategien ist in [Tuia et al., 2011] zu finden.

❑ Query by committee [Seung et al., 1992]

Ein Komitee aus mehreren Klassifikatoren wird gebildet. Dabei kann es sich um verschiedene Parametrisierungen eines Klassifikators handeln oder um die selbe Parametrisierung, jedoch unter Verwendung unterschiedlicher Teilmengen der Trainingsdaten. Hierfür können beispielsweise *bagging*-Ansätze [Breiman, 1996] und *boosting*-Ansätze [Freund & Schapire, 1995] verwendet werden, wie bei *query-by-bagging* und *query-by-boosting* geschehen [Abe & Mamitsuka, 1998]. Dabei klassifizieren alle Komiteemitglieder die unbekannten Stichproben. Diejenigen Stichproben, bei denen Einvernehmen im Komitee herrscht, sind für das weitere Lernen uninteressant, da sie bereits sehr zuverlässig klassifiziert werden können. Besonders interessant hingegen sind diejenigen Stichproben, die zu großer Uneinigkeit führen. Werden diese beim Orakel angefragt, ist der Informationsgewinn besonders groß. Die Uneinigkeit des Komitees kann dabei über die Wahl-Entropie [Dagan & Engelson, 1995] bestimmt werden:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmax}} - \sum_{y\in\Omega} \frac{\operatorname{vote}_{\mathcal{K}}(\mathbf{x}, y)}{|\mathcal{K}|} \log \frac{\operatorname{vote}_{\mathcal{K}}(\mathbf{x}, y)}{|\mathcal{K}|} \tag{3.11}$$

Hierbei ist $\mathcal{K}$ das Komitee aus mehreren Klassifikatoren. Die Anzahl an Stimmen des Komitees dafür, dass die Stichprobe $\mathbf{x}$ das Klassenlabel y erhalten sollte, ist $\operatorname{vote}_{\mathcal{K}}(\mathbf{x}, y)$.

❑ Uncertainty sampling [Lewis & Catlett, 1994]

Diese Strategie benötigt einen Klassifikator, dessen Ergebnis bezüglich der Sicherheit oder Unsicherheit bewertet werden kann. Solch ein Sicherheitsmaß kann zur Bestimmung der Nützlichkeit für den weiteren Trainingsprozess verwendet werden. Hierzu wird der Klassifikator mit seinem aktuellen Trainingsstand auf jede noch ungelabelte Stichprobe angewandt und die Klassifikationssicherheit betrachtet. Stichproben, die bereits sehr sicher klassifiziert werden, tragen nur wenig neue Informationen bei und brauchen daher nicht zum Training hinzugezogen werden. Stattdessen werden die Stichproben ausgewählt, deren Klassifikationsergebnis mit der größten Unsicherheit belegt ist.

Im Falle eines Maximum-Likelihood-Schätzers ist dies direkt der *Likelihood*-Wert [Wuttke et al., 2012]:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmax}} - \sum_{y\in\Omega} P_{\mathbf{C}}(y|\mathbf{x}) \log(P_{\mathbf{C}}(y|\mathbf{x})) \tag{3.12}$$

Hierbei ist $P_{\mathbf{C}}(y|\mathbf{x})$ die A-posteriori-Wahrscheinlichkeit, dass der Klassifikator $\mathbf{C}$ für die Stichprobe $\mathbf{x}$ das Klassenlabel y vorhersagt.

Für SVMs kann als Approximation der Unsicherheit die Distanz der klassifizierten Stichprobe von der Entscheidungs-Hyperebene herangezogen werden (wobei ein kleiner Abstand große Unsicherheit bedeutet [Schohn & Cohn, 2000; Tong & Koller, 2000b]):

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmin}} \left(\sum_{i=1}^{m} |\alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle + b| \right) \tag{3.13}$$

Hierbei sind die $\mathbf{x}_i$ die Stützvektoren, und b der Bias der SVM. Die α_i sowie y_i sind die zugehörigen Lagrange-Multiplikatoren und Klassenlabel.

❑ Reduzierung des erwarteten Fehlers [Roy & McCallum, 2001]

Diese Strategie basiert auf „was-wäre-wenn"-Szenarien. Hierzu wird der zukünftige Klassifikationsfehler (beispielsweise der 0/1-loss) für den Fall simuliert, dass eine bestimmte Stichprobe in die Trainingsmenge aufgenommen werden würde. Diese Simulation wird für jede ungelabelte Stichprobe und jedes mögliche Klassenlabel durchgeführt. Anschließend wird die Stichprobe ausgewählt, welche den erwarteten Fehler minimiert. Das heißt, unabhängig von der tatsächlichen Antwort des Orakels wird ein Modell mit minimalem Fehler erreicht:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmin}} \sum_{y\in\Omega} P_{\mathbf{C}}(y|\mathbf{x}) \left[\sum_{\mathbf{x}'\in\mathcal{U}} 1 - P_{\mathbf{C}^+}(y'|\mathbf{x}') \right] \tag{3.14}$$

Hierbei ist $\mathbf{C}^+$ der Klassifikator, der mit dem simulierten zusätzlichen Trainingsbeispiel trainiert wurde: $\mathbf{C}^+ = \mathbf{C}_{\mathcal{L}\cup(\mathbf{x},y)}$ und $P_{\mathbf{C}^+}(y'|\mathbf{x}')$ ist die Wahrscheinlichkeit, dass der neue Klassifikator die Stichprobe $\mathbf{x}'$ mit dem Klassenlabel y' versieht. Das sehr häufige Neulernen, welches für diese Strategie notwendig ist, ist jedoch für viele praktische Anwendungen zu aufwendig [Settles, 2009]. Es kann jedoch erfolgreich für unparametrisierte Modelle wie zum Beispiel *Gaussian random fields* eingesetzt werden, da das Neulernen bei diesen kaum Aufwand verursacht [Zhu et al., 2003].

❑ Prototypen basiert [Cebron, 2008]

Diese Strategie nutzt eine Linearkombination der lokalen Dichte und der aktuellen Unsicherheit des trainierten Modells. Auf diese Weise kann auf eine getrennte Explorations- und Erschließungsphase verzichtet werden und stattdessen ein einzelnes Kriterium verwendet werden:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmax}} (1-\alpha)\mathbf{A}(\mathbf{x}) + \alpha\mathbf{B}(\mathbf{x}) \tag{3.15}$$

Hierbei ist $\mathbf{A}$ das Potential, dargestellt durch die gewichtete Summe der Distanzen zu den Nachbarn. Die Klassifikationsunsicherheit $\mathbf{B}$ ist dargestellt durch die Entropie der möglichen Klassenlabel. Der Skalierungsparameter α steuert dabei den Kompromiss zwischen der Exploration neuer Potenziale und der Erschließung bekannter Unsicherheiten.

❑ Probabilistisches aktives Lernen [Krempl et al., 2014b]

Diese Strategie wurde bereits im vorherigen Kapitel in Abschnitt 2.4.1 vorgestellt. Sie ist mit der oben beschriebenen Reduzierung des erwarteten Fehlers verwandt und spielt eine wichtige Rolle in den Untersuchungen dieser Arbeit (siehe Abschnitt 4.4.4). Es wird die Stichprobe mit dem größten gewichteten probabilistischen Nutzen ausgewählt:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmax}} \left(d_{\mathbf{x}} \cdot \mathbf{pgain}(ls_{\mathbf{x}}) \right) \tag{3.16}$$

Unabhängig von der spezifischen Selektionsstrategie kann eine Kostenfunktion verwendet werden, falls die Kosten für das Beschaffen der Klassenlabel nicht für alle Stichproben gleich sind. Im Fall einer *in-situ*-Erhebung könnte beispielsweise die Erhebung einer einzelnen Stichprobe aus einer sehr abgelegenen Region den gleichen Aufwand erfordern wie die Erhebung von fünf Stichproben in leichter erreichbaren Regionen. Daher könnte es sinnvoller sein, fünf mittelmäßig geeignete Stichproben zu erheben, anstatt nur einer einzelnen sehr gut geeigneten Stichprobe. Die Berücksichtigung dieses Kompromisses kann mit spezialisierten Kostenfunktionen erfolgen, wie von Demir et al. [2014] gezeigt.

3.1.5 Stoppkriterium

Das Stoppkriterium erfüllt die Aufgabe, die Iterationen innerhalb des Trainingsprozesses zu stoppen, sobald ein bestimmtes Kriterium erfüllt ist. Hierfür können sowohl intrinsische als auch extrinsische Faktoren herangezogen werden.

❑ Intrinsische Stoppkriterien

Ein offensichtliches Stoppkriterium ist, wenn die Qualität der Klassifikation ein Plateau erreicht [Settles, 2009]. In anderen Worten, sobald die Ableitung der Klassifikationsgüte unter einen Schwellenwert fällt. Ab diesem Punkt ist das weitere Anfragen von Label-Informationen Verschwendung von Ressourcen, da es zu keiner wesentlichen Steigerung der Klassifikationsqualität führt. Ein Problem hierbei ist, auf Basis der wenigen bekannten Trainingsdaten, einen für den gesamten Datensatz repräsentativen Qualitätswert zu bestimmen. Eine oft verwendete Methode ist die Kreuzvalidierung [Witten et al., 2017]. Jedoch ist, insbesondere bei SVMs, das für die Kreuzvalidierung benötigte mehrfache Trainieren des Klassifikators zu aufwendig [Schohn & Cohn, 2000]. Schohn und Cohn schlagen stattdessen vor, das Training zu stoppen, sobald keine ungelabelten Stichproben mehr im Randbereich der aktuellen SVM vorhanden sind. Ein weiteres Problem ist jedoch, dass der Einsatz von Kreuzvalidierung voraussetzt, dass die Trainingsverteilung eine Approximation der Ursprungsverteilung ist. Diese Annahme ist jedoch beim aktiven Lernen absichtlich verletzt, da die Stichproben gezielt so ausgewählt werden, dass in der Ursprungsverteilung vorhandene Redundanzen vermieden werden [Olsson & Tomanek, 2009]. Ein *Hold-out-Set* zu verwenden ist ebenfalls keine geeignete Alternative. Hierfür müssten vorhandene *Ground Truth* Informationen bewusst vorenthalten werden. Dies steht jedoch dem Ziel des aktiven Lernens entgegen, ein möglichst ressourcenschonendes Training durchzuführen.

Eine andere Möglichkeit ist, die Zuverlässigkeit der Klassifikationsergebnisse zu verwenden. Der Konfidenzwert wird entweder direkt vom Klassifikator bestimmt (*Likelihood*-Wert) oder kann von seinem zugrundeliegenden Modell abgeleitet werden. Zhu et al. [2010] schlagen hierfür fünf verschiedene Stoppkriterien vor, die alle auf der Konfidenz des Klassifikators beruhen. Vlachos [2008] nutzt ebenfalls die Konfidenz und stoppt das Training, sobald diese einen deutlichen Abfall zeigt.

Eine weitere Alternative ist, die Prädiktionen für ungelabelte Stichproben als Kriterium zu verwenden. Sobald sich diese Prädiktionen nicht mehr ändern, kann das Training laut Bloodgood & Vijay-Shanker [2009] gestoppt werden, da die Klassifikationsgenauigkeit ihren Konvergenzwert erreicht hat. Olsson & Tomanek [2009] erstellen ein Komitee aus Klassifikatoren und vergleichen deren Abstimmverhalten auf den für am nützlichsten bewerteten Stichproben und einer Validierungsmenge aus ungelabelten Stichproben. Sobald die Häufigkeit der Einstimmigkeiten auf den nützlichsten Stichproben jene Einstimmigkeiten auf der Validierungsmenge übersteigt, beenden sie das Training.

❑ **Extrinsische Stoppkriterien**

Im Gegensatz zu intrinsischen sind extrinsische Stoppkriterien nicht von der Wahl des Klassifikators abhängig. Extrinsische Stoppkriterien sind insofern flexibler und allgemeiner einsetzbar. Laut Settles sind sie daher relevanter für die Praxis [Settles, 2009]. Die vorliegende Arbeit folgt dieser Einschätzung.

Eine Möglichkeit für ein solches Stoppkriterium ist es, die Trainingkosten mit den Fehlerkosten in Relation zu setzen und das Training zu stoppen, sobald erstere die letzteren übersteigen. Fehlerkosten hängen jedoch sehr stark von der konkreten Anwendung ab. Eine andere Möglichkeit ist, insbesondere für zeitkritische Anwendungen, die Ausführungszeit heranzuziehen und das Training nach Ablauf dieser Zeit zu stoppen. Dies hängt jedoch sehr eng mit der eingesetzten Computer-Hardware zusammen und spielt daher in den Untersuchungen dieser Arbeit nur eine untergeordnete Rolle.

Das in dieser Arbeit verwendete Stoppkriterium setzt bei der Anzahl der getätigten Anfragen an. Dies ist ein sehr einfach bestimmbares Maß für den benötigten Aufwand. Es korreliert mit der Ausführungszeit und möglichen Beschaffungskosten für die Klassenlabel. Der größte Nutzen des aktiven Lernens entsteht zudem zu Beginn des Trainings [Krempl et al., 2014b]. Der Vorteil zusätzlicher Trainingsbeispiele wird mit steigender Anzahl immer geringer. Da der Fokus dieser Arbeit auf der Effizienz des Lernvorgangs liegt und nicht auf der maximalen Klassifikationsgenauigkeit, ist es ausreichend diesen anfänglichen Bereich des Trainings zu betrachten. Ist die aktive Lernmethode erfolgreich, erreicht die Klassifikationsgüte ohnehin das Niveau der mit passivem Lernen erreichbaren Klassifikationsgüte, jedoch mit wesentlich geringerem Trainingsaufwand.

3.1.6 Szenarien

Aktives Lernen kann in verschiedenen Szenarien eingesetzt werden. Diese lassen sich auf zwei Arten kategorisieren. Zum einen die Art wie die Selektionsstrategie angewendet wird und zum anderen die Art wie das Orakel befragt wird. Diese sind in Abbildung 3.3 dargestellt. Settles [2009] teilt Selektionsstrategien in drei Kategorien. Als Unterscheidungskriterium nutzt er die Art und Weise, wie die ungelabelten Stichproben zur Verfügung gestellt werden. Die Art der Orakelbefragung wird in der vorliegenden Arbeit in drei weitere Kategorien unterteilt. Diese sechs Kategorien sind im Folgenden erläutert:

- **Membership query synthesis [Angluin, 1988]**: Eine Stichprobe wird künstlich erzeugt (*de novo*), so dass sie in der Region des Merkmalsraums liegt, von der das Lernverfahren aktuell am meisten profitieren würde. Ein Anwendungsbeispiel ist ein Roboter, der eine spezifische Mischung von chemischen Bausteinen erzeugt, um neue Stoffwechselwege in Hefepilzen zu untersuchen. Anstatt zufällige Mischungen auszuprobieren, wird durch eine Selektionsstrategie aktiv bestimmt, welche Mischung als nächstes untersucht werden soll. So werden durch aktives Lernen die Kosten der verwendeten Materialien um dass 100-fache reduziert [King et al., 2004, 2009].

- **Kettenbasiert [Atlas et al., 1990]**: Die unbekannten Stichproben werden einzeln aus der Grundverteilung gezogen und liegen als Kette vor. Für jede Stichprobe muss direkt entschieden werden, ob sie beim Orakel angefragt wird oder nicht. Wird sie nicht angefragt, kann sie nicht zurückgehalten werden und das wahre Klassenlabel bleibt unbekannt. Ein Anwendungsbeispiel ist die Optimierung der Sensorauswahl zur Freund-Feind-Erkennung bei Flugzeugen [Krishnamurthy, 2002]. Hier kann die Entscheidung, welche Sensoren eingesetzt werden sollen, nicht verzögert werden, da das zu klassifizierende Flugzeug aufgrund seiner hohen Geschwindigkeit sonst den Erfassungsbereich der Sensoren verlässt.

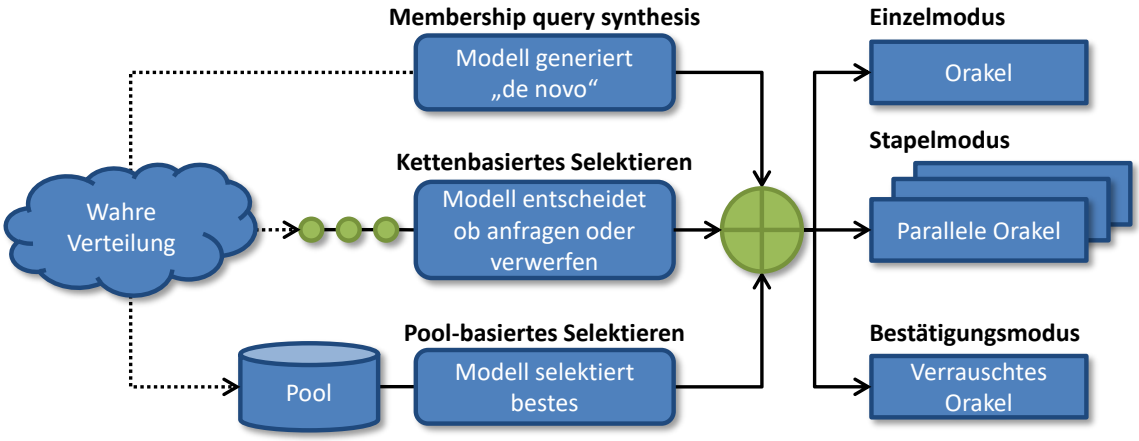


Abbildung 3.3: Verschiedene Szenarien des aktiven Lernens. Unterschieden wird nach der Art wie die unbekannten Stichproben zur Verfügung stehen und nach der Art wie beim Orakel angefragt wird.

- **Pool-basiert [Lewis & Gale, 1994]**: Aus der Grundverteilung steht eine große Menge Stichproben zur Verfügung und die Selektionsstrategie kann daraus frei auswählen. Dies ist eines der am häufigsten anzutreffenden Szenarien. Ein Beispiel hierfür aus der Bildverarbeitung ist das Klassifizieren einer Menge von Bildern nach ihrem Inhalt. Stehen alle Bilder gleichzeitig zur Verfügung, führt eine Selektionsstrategie, die eine diversifizierte Auswahl trifft, zu besseren Ergebnissen als eine zufällige Auswahl [Long & Hua, 2015].

- **Einzelmodus:** Die intuitivste Art der Orakelanfrage ist die Einzelanfrage. Hierbei wählt die Selektionsstrategie eine einzige Stichprobe aus, fragt sie beim Orakel an, erhält das wahre Klassenlabel und integriert sie in das Klassifikatormodell. Dies wiederholt sich bis das Stoppkriterium erfüllt ist. Beispiele sind Anwendungen bei denen ein Experte das Klassenlabel vergibt und anschließend auf die nächste Anfrage wartet.

- **Stapelmodus:** Anstelle einer einzelnen Anfrage werden in diesem Modus mehrere Stichproben gebündelt und auf einmal beim Orakel angefragt. In diesem Fall kann auch bei mehreren Orakeln gleichzeitig angefragt werden, um die Bearbeitung durch Parallelisierung zu beschleunigen. Hierbei ist es wichtig, dass nicht einfach die ersten k Stichproben gewählt werden, welche die Selektionsstrategie ausgibt. Stattdessen sollte ein Diversitätsterm integriert werden, so dass Redundanzen in der Anfragemenge verhindert werden [Xu et al., 2007; Demir et al., 2011].

- **Bestätigungsmodus:** Die bisherigen Kategorien gingen davon aus, dass das Orakel stets das wahre Klassenlabel liefert. Im Gegensatz dazu erlaubt diese Kategorie ein verrauschtes Orakel, bei dem das gelieferte Klassenlabel falsch sein kann. Um diese Unsicherheit zu berücksichtigen, kann es erforderlich sein, eine Stichprobe mehrmals anzufragen. Ein Anwendungsbereich hierfür ist *Crowd-Sourcing*, also das Verwenden von vielen günstigen, aber fehlerbehafteten Orakeln. Ein Einsatzgebiet ist zum Beispiel die maschinelle Übersetzung [Ambati et al., 2010].

Das Szenario, welches in der Bildinterpretation von Fernerkundungsdaten am häufigsten angewendet wird, ist das Pool-basierte Szenario im Einzelmodus. Das gesamte Bild beziehungsweise die Menge aller Pixel ist der Pool von ungelabelten Stichproben und das Klassenlabel für jedes beliebige Pixel kann bei einem Experten angefragt werden. Der Ansatz der vorliegenden Arbeit ist ebenfalls für ein solches Szenario gedacht.

3.2 Landbedeckungsklassifikation

Die Landbedeckungsklassifikation ist ein Teilgebiet der Fernerkundung. Hierin wird das Problem untersucht, aus den aufgenommenen Daten abzuleiten, welcher Bedeckungstyp vorliegt. Hierzu werden in der Literatur zur bildgebenden Fernerkundung vier verschiedene Informationsquellen untersucht:

- **Räumlich:** Die räumliche Entfernung einzelner Pixel zueinander [Gong & Howarth, 1990; Geneletti & Gorte, 2003; Rodríguez-Galiano et al., 2011].
- **Spektral:** Das individuelle Spektrum einzelner Pixel oder deren Relationen [Gross et al., 2013, 2015].
- **Texturell:** Die kleinräumige und regelmäßige Variation der Farbwerte einer Menge von Pixeln [Berberoğlu et al., 2000; Kumar & Dikshit, 2015].
- **Semantisch:** Das gemeinsame Auftreten verschiedener Klassen oder Objekte [Stuckens et al., 2000; Walker & Blaschke, 2008].

Die ersten beiden Informationsquellen werden in der vorliegenden Arbeit unter den Voraussetzungen der Glattheits- und der Clusterannahme ausgewertet. Dies wird in den entsprechenden Abschnitten dieses Kapitels zur Segmentierung und Clusterbildung genauer erläutert. Die letzten beiden Informationsquellen bleiben hier unbetrachtet, da die vorgestellte Methode aus Einzelkomponenten besteht, die nur auf individuellen Pixeln beziehungsweise deren zugehörigen Merkmalsvektoren arbeiten und Textur sowie Semantik nicht im Fokus liegen.

Bilddaten in der Fernerkundung werden häufig mit mehr Kanälen und höherer Auflösung aufgenommen, als den sonst in der Bildverarbeitung üblichen drei Kanälen Rot, Grün und Blau. Werden weniger als 10 Kanäle aufgenommen, wird häufig von Multispektraldaten gesprochen, bei einer größeren Kanalanzahl werden diese als Hyperspektraldaten bezeichnet. In dieser Arbeit wird ein Bild als Menge von Merkmalsvektoren aufgefasst, die den einzelnen Pixeln entsprechen. Jedes Pixel stellt dabei ein durch den Sensor aufgenommenes Stück der Bodenfläche dar. Das Spektrum des von dieser Bodenfläche reflektierten Lichts wurde durch den Sensor in k Kanäle quantisiert und bildet den Merkmalsvektor:

$$\begin{aligned} \mathcal{I} &= \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\} \\ \mathbf{p} &\in \mathbb{R}^k \\ n, k &\in \mathbb{N} \end{aligned} \tag{3.17}$$

Um Aussagen über die Relationen zwischen verschiedenen Pixeln beziehungsweise Merkmalsvektoren treffen zu können, wird ein passendes Distanzmaß benötigt.

3.2.1 Distanzmaß

Ein Distanzmaß ist eine Funktion $\mathbf{D} : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$, die zwei Elementen einer endlichen Menge $\mathbf{p}, \mathbf{q} \in \mathcal{I} = \{1, 2, ..., N\}$ einen Distanzwert zuordnet. Die Funktion muss dabei symmetrisch und positiv semidefinit sein [Hartung & Elpelt, 1984]:

$$\mathbf{D}(\mathbf{p}, \mathbf{q}) = \mathbf{D}(\mathbf{q}, \mathbf{p}) \tag{3.18}$$

$$\mathbf{D}(\mathbf{p}, \mathbf{q}) \geq 0 \tag{3.19}$$

$$\mathbf{D}(\mathbf{p}, \mathbf{q}) = 0 \Leftrightarrow \mathbf{p} = \mathbf{q} \tag{3.20}$$

Die dritte Bedingung besagt, dass wenn zwei Elemente die Distanz 0 haben, müssen beide Elemente identisch sein. Ist nur diese Bedingung nicht erfüllt, handelt es sich um ein Ähnlichkeitsmaß. Erfüllt ein Distanzmaß zusätzlich die Dreiecksungleichung: $\mathbf{D}(\mathbf{p},\mathbf{r}) \leq \mathbf{D}(\mathbf{p},\mathbf{q}) + \mathbf{D}(\mathbf{q},\mathbf{r})$, ist es eine Metrik.

3.2.2 Häufig verwendete Distanzmaße

Es gibt kein Distanzmaß, dass für alle Anwendungen in gleicher Weise gut geeignet ist [Keshava, 2004]. Die Wahl des verwendeten Maßes ist daher von großer Bedeutung für die Klassifikation von Multispektraldaten. In der Literatur häufig verwendete Maße sind: Euklidische Distanz [Campbell & Wynne, 2012], Kosinus-Distanz [Bao & Guo, 2004], Spektraler Winkel [Kruse et al., 1993] und Mahalanobis-Distanz [Mahalanobis, 1936]. Die ersten drei werden in der vorliegenden Arbeit benötigt und sind daher im Folgenden erläutert.

❑ Euklidische Distanz

Die euklidische Distanz ist eines der einfachsten Distanzmaße. Sie basiert auf dem Satz des Pythagoras und kann sehr einfach für höhere Dimensionen, wie zum Beispiel bei Multi- und Hyperspektraldaten, erweitert werden:

$$\mathbf{D}_{euklid}(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n} (\mathbf{p}^{(i)} - \mathbf{q}^{(i)})^2} \tag{3.21}$$

Hierbei ist n die Anzahl der Dimensionen. Dieses Distanzmaß ist jedoch unbeschränkt, dass heißt seine Funktionswerte werden mit zunehmender Kanalanzahl immer größer. Es gibt daher keinen festen Maximalwert. Es existieren skalierte Varianten, die dies verhindern [Keshava, 2004]. Die Auswirkungen dieser Skalierung auf Hyperspektralbilder ist jedoch unklar und sie werden daher selten in der Fernerkundung eingesetzt [Robila, 2005].

❑ Kosinus-Distanz

Eine weitverbreitete Alternative zur Euklidischen Distanz ist die Kosinus-Distanz [Campbell & Wynne, 2012]. Für dieses Maß werden die zu vergleichenden Spektren als Vektoren interpretiert und der Kosinus des von ihnen aufgespannten Winkels berechnet. Da alle Werte eines Spektrums positiv sind, ist der maximale Winkel zwischen zwei Vektoren 90°. Daraus folgt, dass dieses Maß auf das Intervall [0...1] beschränkt ist, wie von Definition 3.19 gefordert. Der Wert ist 1, falls die Vektoren parallel (maximal ähnlich) sind und 0, falls sie orthogonal (maximal verschieden) zueinander sind. Da dies invers zu dem Verhalten der Definition in 3.18 ist, wird der Wert von 1 subtrahiert:

$$\begin{aligned}
\mathbf{D}_{kosinus}(\mathbf{p},\mathbf{q}) &= 1 - cos(\measuredangle(\mathbf{p},\mathbf{q})) \\
&= 1 - \left(\frac{\langle \mathbf{p},\mathbf{q} \rangle}{\|\mathbf{p}\|_2 \, \|\mathbf{q}\|_2} \right) \\
&= 1 - \frac{\sum_{i=1}^{n} \mathbf{p}^{(i)} \mathbf{q}^{(i)}}{\sqrt{\sum_{i=1}^{n} (\mathbf{p}^{(i)})^2} \sqrt{\sum_{i=1}^{n} (\mathbf{q}^{(i)})^2}}
\end{aligned} \tag{3.22}$$

Es ist anzumerken, dass dies trotz der üblichen Bezeichnung kein echtes Distanzmaß ist. Zwei verschiedene Spektren können zwar parallelen Vektoren entsprechen, diese können jedoch unterschiedliche Länge haben. Als Folge ist ihre Kosinus-Distanz zwar 0, es handelt sich jedoch nicht um identische Vektoren. Somit ist die Bedingung der Identität verletzt (3.20) und es handelt sich hierbei um ein Ähnlichkeitsmaß.

❑ Spektraler Winkel

Eine oft verwendete Alternative ist, den tatsächlichen Winkel zwischen den Vektoren zu verwenden. Im englischen wird dieses Maß daher auch als *spectral angle* (SA) bezeichnet [Kruse et al., 1993; Achalakul & Taylor, 2001; Harvey et al., 2002].

$$\begin{aligned}\mathbf{D}_{SA}(\mathbf{p},\mathbf{q}) &= \measuredangle(\mathbf{p},\mathbf{q}) \\ &= \arccos\left(\frac{\langle\mathbf{p},\mathbf{q}\rangle}{\|\mathbf{p}\|_2\,\|\mathbf{q}\|_2}\right) \\ &= \arccos\frac{\sum_{i=1}^{n}\mathbf{p}^{(i)}\mathbf{q}^{(i)}}{\sqrt{\sum_{i=1}^{n}(\mathbf{p}^{(i)})^2}\sqrt{\sum_{i=1}^{n}(\mathbf{q}^{(i)})^2}}\end{aligned} \tag{3.23}$$

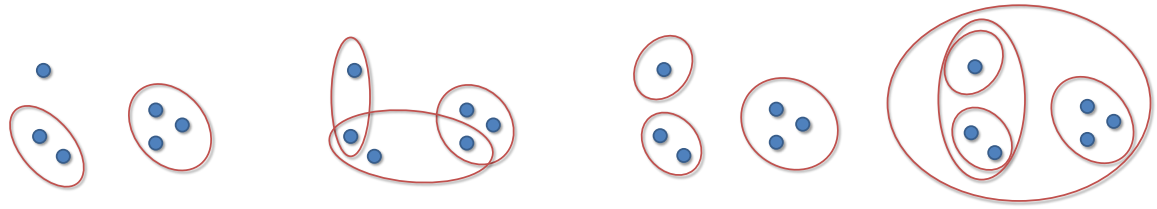
Da die Merkmalsvektoren der Abtastung eines physikalischen Spektrums entsprechen, bestehen sie ausschließlich aus positiven Werten. Der Wertebereich des SA ist daher das Intervall $[0..\pi/2]$. Das Multiplizieren mit $2/\pi$ normiert den Wertebereich auf $[0..1]$. Der SA ist positiv semidefinit, symmetrisch, erfüllt die Dreiecksungleichung und ist invariant gegenüber positiver skalarer Multiplikation. Letzteres entspricht in Hyperspektraldaten einer Änderung der Beleuchtung eines Materials [Keshava, 2004]. Das Verwenden von normierten Vektoren überführt den SA von einem Ähnlichkeitsmaß in ein Distanzmaß. Diese Überführung ist jedoch für die vorliegende Arbeit nicht erforderlich.

3.3 Clusterbildung

Als Clusterbildung wird die Aufgabe beschrieben, Elemente so in bedeutsame Gruppen zu einzuteilen, dass die Unterschiede innerhalb der Gruppen geringer sind als die Unterschiede zwischen verschiedenen Gruppen [Jain & Dubes, 1988; Ester et al., 1999]. Oftmals wird Segmentierung und Clusterbildung synonym gebraucht, letztere ist jedoch allgemeiner anwendbar. Segmentierung wird häufig nur im Bildraum angewandt, während Clusterbildung auch im Merkmalsraum Verwendung findet. Auf die vorliegende Arbeit angewendet bedeutet dies, dass Segmentierung immer zu im Bild zusammenhängenden Pixelmengen führt, sie wird hier daher als „lokal“ bezeichnet. Die Clusterbildung hingegen führt unter Umständen zu nicht zusammenhängenden Pixelmengen. Es wird hier daher auch von „globaler“ Clusterbildung gesprochen.

Existierende Clusterverfahren zu kategorisieren ist schwer, da keine Einigkeit darüber besteht, wie „Cluster“ definiert ist [Estivill-Castro, 2002]. An dieser Stelle soll daher eine Kategorisierung stattfinden, die auf der Art des erstellten Ergebnisses basiert.

Es gibt zwei grundlegende Arten der Zugehörigkeit zu Clustern: hart und weich. In ersterem Fall gehört ein Element entweder zu einem Cluster oder nicht. In letzterem Fall sind auch Abstufungen erlaubt, das heißt Elemente können ebenso nur teilweise zu Clustern gehören. Diese Variante ist auch als *Fuzzy-Clustering* bekannt [Pal et al., 2000; Maulik & Saha, 2010]. Weitere Unterschiede die zur Kategorisierung dienen, sind (siehe auch Abbildung 3.4 für Beispiele):



(a) (b) (c) (d)

Abbildung 3.4: Clusterings mit verschiedenen Eigenschaften. Mit einem Ausreißer (a); vollständig und überlappend (b); vollständig und strikt (c); vollständig und hierarchisch (d).

- **Ausreißer vs. Vollständig:** Muss jedes Element zu mindestens einem Cluster gehören oder kann es auch Elemente geben, die zu keinem Cluster gehören?
- **Überlappend vs. Strikt:** Gehören Elemente zu genau einem Cluster oder können sie auch zu mehreren Clustern gleichzeitig gehören?
- **Hierarchisch vs. Flach:** Können Cluster (und die in ihnen enthaltenen Elemente) in anderen Clustern enthalten sein oder können Cluster nur nebeneinander existieren?

Die in dieser Arbeit vorgestellte Methode erzeugt ein hartes, vollständiges und hierarchisches Clustering. Hierzu wird die Clusterannahme zu Grunde gelegt, welche im nächsten Abschnitt genauer erläutert wird.

Clusterannahme

Die Clusterannahme besagt, dass zwei Elemente, die zu dem selben Cluster gehören, mit hoher Wahrscheinlichkeit auch zur selben Klasse gehören [Chapelle et al., 2010]. Anzumerken ist, dass diese Annahme nicht bedingt, dass eine Klasse aus nur einem Cluster bestehen muss. Dies ist sehr intuitiv, da Landbedeckungstypen in der Regel mehrfach und räumlich getrennt in Bildern auftreten.

Ebenso können Elemente, die zwar im Merkmalsraum zum selben Cluster gehören, im Bildraum weit voneinander entfernt sein. Ein im Merkmalsraum arbeitendes Clusterverfahren operiert somit im Bildraum global. Dies ist ein wichtiger Unterschied zu der als nächstes vorgestellten Segmentierung, da diese im Bildraum nur lokal operiert.

3.4 Segmentierung

Aufgabe der Segmentierung ist es, die Komplexität eines Bildes zu reduzieren, indem ähnliche Pixel zu Regionen zusammengefasst werden. Jede Region sollte dabei redundante Informationen vereinen und so die Folgeschritte der Verarbeitungskette vereinfachen. Sie ist ein nützliches Werkzeug zur Auswertung und Vorbereitung von Fernerkundungsdaten. Ein breiter Überblick über Segmentierungstechniken mit besonderem Augenmerk auf der Fernerkundung ist in [Dey et al., 2010] zu finden. Die Techniken können wie folgt grob unterteilt werden.

- **Bildbasiert:** Techniken, die direkt auf den Pixeln beziehungsweise Subpixeln arbeiten. Hierzu zählen auch Verfahren, die mit Kantendetektion arbeiten [Ali & Clausi, 2001].
- **Modellbasiert:** Techniken, die annehmen, dass die im Bild enthaltenen Objekte einem Modell genügen. Hierzu zählen Techniken wie Schwellenwertentscheider [Pal et al., 2000; Al-Amri et al., 2010], *Markov Random Fields* [Kindermann & Snell, 1980] oder transformative Methoden wie zum Beispiel die Wasserscheidentransformation [Beucher & Lantuejoul, 1979; Carleer et al., 2005].
- **Homogenitätsbasiert:** Diese Techniken werden vor allem bei hoch aufgelösten Bildern eingesetzt. Sie verwenden spektrale und räumliche Distanzmaße [Geneletti & Gorte, 2003; Gross et al., 2013], Form, Textur und Größe von Teilbereichen [Berberoğlu et al., 2000; Kumar & Dikshit, 2015] oder nutzen Kontextwissen [Stuckens et al., 2000].

In der vorliegenden Arbeit spielen vor allem bild- und homogenitätsbasierte Methoden eine Rolle, da der im vorherigen Kapitel erwähnte SLIC-Algorithmus in diese Kategorie fällt. Als Grundlage hierfür dient die Glattheitsannahme. Diese wird im Folgenden detaillierter erläutert.

Glattheitsannahme

Das räumliche Auflösungsvermögen moderner Sensoren wird stetig besser. Somit wird die durch ein Pixel repräsentierte Fläche kleiner. Es können immer feinere Details unterschieden werden. Gleichzeitig bleiben die zu beobachtenden Objekte in ihrer Größe unverändert. Die Folge ist eine steigende Wahrscheinlichkeit, dass benachbarte Pixel zu dem selben Objekt beziehungsweise zu der selben Klasse gehören [Schindler, 2012]. Diese Annahme wird auch als *smoothness assumption* bezeichnet. Chapelle et al. [2010] definieren dies allgemeiner: Sind zwei Elemente x_1, x_2 nahe beieinander, so sollten es ihre zugehörigen Klassenlabel y_1, y_2 ebenso sein.

Schindler [2012] sagt in seiner Arbeit, dass eine konzeptionelle Haupteinschränkung der pixelbasierten statistischen Lernverfahren ist, dass sie benachbarte Pixel als unabhängig voneinander betrachten. Dieses Problem ist stärker in Bildern mit starker spektraler Variation, wie es häufig in urbanen Gebieten der Fall ist. Es ist daher wichtig, auch die Abhängigkeiten benachbarter Pixel zu berücksichtigen. Da es zu aufwendig ist, dies auf globaler Ebene durchzuführen, schlägt er vor, sich an Methoden aus der medizinischen Bildverarbeitung und der Bildrekonstruktion zu orientieren. Diese modellieren Nachbarschaftsbeziehungen nur auf kurze Distanz. Als Beispiele nennt er *graph cuts* [Boykov et al., 2001; Cesa-Bianchi et al., 2010], *message passing* Algorithmen [Wainwright et al., 2005; Kolmogorov, 2006] und *semiglobal matching* [Hirschmuller, 2008]. Die Glattheitsannahme spielt ebenfalls im probabilistischem aktiven Lernen eine Rolle, wie bereits im Abschnitt 2.4.1 erläutert wurde.

Je höher die räumliche Auflösung ist, desto eher trifft somit die Glattheitsannahme zu. In der vorliegenden Arbeit dient sie als Entscheidungsgrundlage für die Verwendung eines Segmentierungsverfahrens. Die Entscheidung hierfür fiel, wie im vorherigen Kapitel erläutert, auf den SLIC-Algorithmus.

4 Segmentierung, Clusterhierarchie und aktives Lernen

Das in dieser Arbeit vorgestellte Trainingsverfahren besteht aus drei Schritten: (i) Segmentierung, (ii) Clusterhierarchie und (iii) aktives Lernen. Zur einfacheren Schreibweise wird aus den drei Schritten das Akronym SCHAL (Segmentierung, Clusterhierarchie, aktives Lernen) gebildet und die Methode als die SCHAL-Methode bezeichnet. Dieses Kapitel beginnt mit einem Überblick, um das Zusammenspiel der drei Schritte aufzuzeigen. Anschließend wird jeder Schritt detailliert erläutert.

4.1 Überblick

Die ersten beiden Schritte, das Segmentieren und die Clusterbildung, dienen der Vorbereitung. Sie finden unüberwacht statt, so dass hierfür noch keine Klassenlabel benötigt werden. Im letzten Schritt, dem aktiven Lernen, findet der überwachte Lernvorgang statt. Dieser Schritt wird wiederholt bis ein Stoppkriterium erfüllt ist.

Die Definition eines guten Stoppkriteriums ist kein triviales Problem, wie in Abschnitt 3.1.5 erläutert. Hierfür gibt es mehrere Möglichkeiten:

- **Klassifikationsgenauigkeit:** Das Training wird beendet sobald eine vorgegebene Klassifikationsgenauigkeit erreicht ist. Das wiederholte Bestimmen dieser verursacht jedoch zusätzlichen Aufwand und erfordert eine von der Trainingsmenge unabhängige Testmenge von Klassenlabeln.

- **Zeit:** Das Training wird nach Ablauf einer festen Zeitdauer beendet. Dies wird beeinflusst durch die Berechnungszeit des Lernalgorithmus und die Antwortzeit des Orakels. Während der Methodenentwicklung spielt die Berechnungszeit eine untergeordnete Rolle. Dieses Kriterium ist daher eher für die spätere Endanwendung geeignet.

- **Anfragebudget:** Sobald ein vorgegebenes Budget für Orakelanfragen aufgebraucht wurde, wird das Training beendet. Dieses Kriterium ist das in der Literatur zu aktivem Lernen am häufigsten verwendete.

In dieser Arbeit wurde die dritte Möglichkeit gewählt, da sie am besten zur Aufgabenstellung passt. Jede Orakelanfrage verursacht in der vorgesehenen Anwendung die gleichen Kosten. Daher wird das verbrauchte Anfragebudget durch die Anzahl der verwendeten Trainingsbeispiele gemessen.

Der Gesamtablauf der vorgestellten Methode ist in Abbildung 4.1 visualisiert und in Algorithmus 1 als Pseudocode formuliert. Zeile 9 ist nicht Teil der eigentlichen Methode. Dort wird lediglich das Ergebnis in die geforderte Form überführt. Die Aufrufe für die ersten beiden Schritte

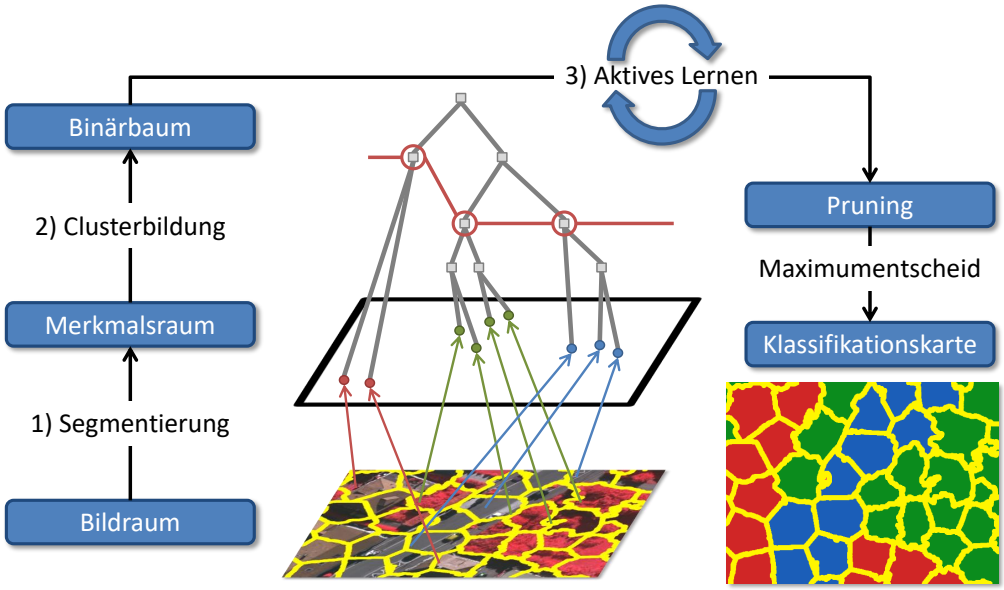


Abbildung 4.1: Überblick des Ablaufs der SCHAL-Methode. Schritt 1 segmentiert das Eingabebild und überführt die Pixel in Repräsentantenvektoren im Merkmalsraum. Diese werden in Schritt zwei in eine binäre Clusterhierarchie eingeordnet. Schritt 3 optimiert auf diesem Binärbaum ein Pruning mittels aktivem Lernen. Das Pruning induziert schließlich über einen Maximumentscheid die Belegung mit Klassenlabeln für das Ergebnisbild.

des Trainingsverfahrens finden in Zeilen 1 und 2 statt. Der dritte Schritt ist auf die Zeilen 4 bis 7 aufgetrennt. Die einzelnen Teilschritte werden in Abschnitt 4.4 beschrieben.

Eingabe:
U: Ungelabelte Daten
Ausgabe:
R: Resultat – Klassenlabel für jedes Pixel
1: S ← doSegmentation(U)
2: H ← doClustering(S)
3: **while not** stopCriterion **do**
4: P ← findOptimalPruning(H)
5: $\mathbf{x}^*$ ← findOptimalSample(H, P)
6: y^* ← queryOracle($\mathbf{x}^*$)
7: H ← updateHierarchy(H, $\mathbf{x}^*$,y^*)
8: **end while**
9: R ← distributeLabels(P)

Algorithmus 1: Trainingsablauf der SCHAL-Methode. Die ersten beiden Methodenschritte finden in Zeile 1 beziehungsweise 2 statt. Der dritte Schritt wiederholt die Teilschritte in Zeilen 4 - 7 bis ein Stoppkriterium (Zeile 3) erfüllt ist.

4.2 Segmentierung

Der Segmentierungsschritt der vorgestellten Methode erhält als Eingabe ein Luftbild. Jedes Pixel dieses Bildes entspricht einem Merkmalsvektor. Dieser Vektor entsteht durch die Quantisierung des Spektrums des aufgenommenen Lichts im Sensor. Die Pixel werden zunächst anhand von lokalen Merkmalen mit einer angepassten Variante des SLIC-Algorithmus segmentiert. Die Merkmalsvektoren werden mit Hilfe der erstellten Segmente anschließend in eine Menge von Repräsentantenvektoren überführt.

4.2.1 Lokale Merkmale

Die Bezeichnung *lokal* bezieht sich in dieser Arbeit auf die Nachbarschaft im Bildraum. Mit „Merkmal“ sind hier sowohl räumliche, als auch spektrale Informationen gemeint. Die Einschränkung auf die lokale Nachbarschaft basiert auf der Glattheitsannahme (siehe Abschnitt 3.4). Um die lokalen Merkmale für die spätere Klassifikation aufzubereiten, wird daher ein Segmentierungsverfahren benötigt, das sowohl die spektralen als auch die räumlichen Komponenten berücksichtigt. Hierfür wurde der SLIC-Algorithmus gewählt. Neben den in Abschnitt 2.3.1 erläuterten Gründen spielte bei der Auswahl eine große Rolle, dass das Distanzmaß des Algorithmus sehr einfach anzupassen ist.

4.2.2 Anpassung des SLIC-Algorithmus

In der Originalversion wandelt der SLIC-Algorithmus die im RGB-Farbraum dargestellten Bilder in den CIELAB-Farbraum um und berechnet darin die euklidische Distanz (Gleichung 2.5). Dieser Farbraum basiert auf der Wahrnehmung des menschlichen Auges und ist demzufolge auf den sichtbaren Teil des elektromagnetischen Spektrums ausgerichtet. Die zur Landbedeckungsklassifikation verwendeten Bilder werden jedoch oftmals nicht im RGB-Farbraum aufgenommen (siehe Abschnitt 3.2). Eine Adaption des SLIC-Algorithmus ist daher erforderlich.

Da die Handhabung der spektralen Merkmale angepasst werden soll, muss die Änderung im spektralen Anteil $\mathbf{D}_{spektral}$ (Formel 2.5) des originalen Distanzmaßes $\mathbf{D}_{SLIC}$ (Formel 2.4) durchgeführt werden. Gesucht ist ein Distanzmaß für Spektren, dass unabhängig von der Anzahl der verwendeten Kanäle ist, so dass es gleichermaßen für Multi- und Hyperspektraldaten verwendet werden kann.

Eine naheliegende Wahl ist, die CIELAB-Umwandlung auszulassen und direkt die euklidische Distanz der Spektren zu berechnen. Dies führt jedoch zu den im Abschnitt über spektrale Distanzmaße (3.2.1) beschriebenen Problemen. Vor allem der unbeschränkte Wertebereich bereitet Probleme, die spektralen und räumlichen Komponenten zu vereinen. Stattdessen wird der *spectral angle* (SA), wie ebenfalls im Abschnitt über spektrale Distanzmaße (3.2.1) beschrieben, verwendet. Dieser ist über das Skalarprodukt $\langle ., . \rangle$ und die 2-Norm $\|.\|_2$ definiert:

$$\mathbf{D}_{SA}(\mathbf{p}, \mathbf{q}) = \arccos\left(\frac{\langle \mathbf{p}, \mathbf{q} \rangle}{\|\mathbf{p}\|_2 \|\mathbf{q}\|_2}\right) \qquad \text{(wiederholt 3.23)}$$

Im Gegensatz zur euklidischen Distanz sind die Werte des SA begrenzt und können daher einfacher in den SLIC-Algorithmus integriert werden. Das neue Distanzmaß ist somit definiert als:

$$\mathbf{D}_{SLIC-SA} = \sqrt{{\mathbf{D}_{SA}}^2 + \left(\frac{\mathbf{D}_{spatial}}{S}\right)^2 m^2} \qquad (4.1)$$

Durch diese Anpassung lässt sich der Segmentierungsschritt der Methode auf Multispektralbilder unabhängig von ihrer Kanalanzahl anwenden. Die so erstellten Segmente werden im nächsten Schritt in Repräsentantenvektoren überführt.

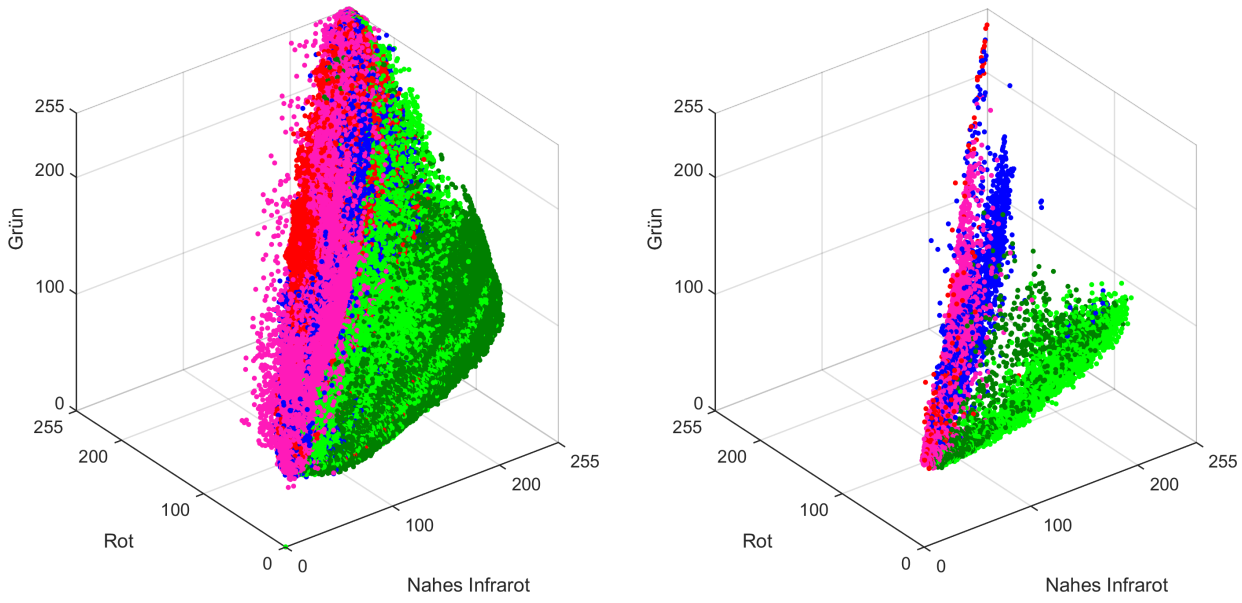


Abbildung 4.2: Bild (a) zeigt die vollständigen Daten von Region 7 des Vaihingen-Datensatzes. In Bild (b) sind nur die berechneten Repräsentantenvektoren dargestellt. Deutlich zu erkennen, ist die starke Ausdünnung, wobei die Relationen der Klassen zueinander erhalten bleibt. Insbesondere Merkmalsvektoren, die im Sättigungsbereich des Sensors liegen, wurden entfernt. Der Sättigungsbereich befindet sich jeweils im oberen, mittleren Teil der Bilder bei den Koordinaten (255, 255, 255).

4.2.3 Repräsentantenvektoren

Die in den erstellten Segmenten organisierten Merkmalsvektoren müssen nun so repräsentiert werden, dass sie durch die nachfolgenden Schritte verarbeitet werden können. In dieser Arbeit wurde die Repräsentation durch ein einzelnen Vektor gewählt – dem Repräsentantenvektor. Diese Entscheidung basiert auf den Aussagen der Glattheitsannahme (siehe Abschnitt 3.4). Daher können nahe beieinander liegende Pixel zusammengefasst werden, ohne dass wesentliche, für die spätere Klassifikation benötigte, Informationen verloren gehen. Vor dem Zusammenfassen repräsentiert ein Merkmalsvektor das Spektrum eines bestimmten Bereiches des aufgenommenen Gebietes. Nach dem Zusammenfassen entspricht ein Repräsentantenvektor dem Gebiet aller zum Segment gehörenden Pixel. Das heißt, der Zusammenhang zwischen Pixel, Repräsentantenvektor und aufgenommener Bodenfläche bleibt erhalten.

Der Vorteil dieser Darstellungsform ist, dass Algorithmen, die direkt auf Pixeln beziehungsweise Merkmalsvektoren arbeiten, unverändert übernommen werden können. Dem gegenüber steht eine Darstellung durch Texturindizes, Formbeschreibungen oder Kontextinformationen, welche jedoch die Algorithmenauswahl stark einschränkt oder umfangreiche Anpassungen notwendig macht.

Ein weiterer Vorteil ist eine drastische Datenreduktion. Hat ein Luftbild vor der Segmentierung mehrere Millionen Pixel, ist bei typischer Parameterwahl anschließend mit etwa 10.000 Repräsentantenvektoren zu rechnen. Dies entspricht einer Reduktion um mehr als zwei Größenordnungen. Diese Reduktion erhöht die Verarbeitungsgeschwindigkeit der nachfolgenden Schritte enorm. Abbildung 4.2 zeigt eine solche Reduktion anhand des Vaihingen-Datensatzes.

Es ist darauf zu achten, wie die Reduktion durchgeführt wird. Zwei naheliegende Lösungen sind (i) ein zufälliges Pixel oder (ii) das im Mittelpunkt des Segmentes liegende Pixel zu wählen. Dies kann jedoch in ungünstigen Fällen zu Fehlern führen, da es sich um einen Ausreißer handeln

kann. Beispiele für solche Ausreißer in urbanen Gebieten sind Dachfenster, kahle Stellen auf einer Wiese oder Straßenmarkierungen. Es gibt jedoch auch Ausreißer, die unabhängig vom aufgenommenen Gebiet auftreten können: Pixel im Sättigungsbereich des Sensors. Diese enthalten kaum verwertbare Informationen und stören das Ergebnis, wenn ihre Merkmalsvektoren mit anderen Merkmalsvektoren kombiniert werden.

Die Auswahl eines Ausreißers kann verhindert werden, indem der bandweise Mittelwert aller N, zum Segment gehörenden Merkmalsvektoren, verwendet wird:

$$\mathbf{r}_{mittel} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{r}^{(i)} \tag{4.2}$$

Vorhandene Ausreißer können jedoch auch hier zu einem verzerrten Ergebnis führen, da der Mittelwert anfällig für Ausreißer ist. In dieser Arbeit wird daher der kanalweise Median verwendet. Hier dargestellt für N Spektren mit jeweils K Kanälen:

$$\mathbf{r}_{median} = \begin{pmatrix} \text{median}\left(\mathbf{r}_1^{(1)}, \ldots, \mathbf{r}_1^{(N)}\right) \\ \vdots \\ \text{median}\left(\mathbf{r}_K^{(1)}, \ldots, \mathbf{r}_K^{(N)}\right) \end{pmatrix} \tag{4.3}$$

Dies führt zum einen dazu, dass Ausreißer entfernt werden und zum anderen, dass im Segment vorhandenes Rauschen herausgemittelt wird. Diese Vorverarbeitung liefert sehr robuste Daten für die Clusterbildung im nächsten Schritt.

4.3 Clusterhierarchie

Der Clusterbildungsschritt der vorgestellten Methode erhält als Eingabe die Menge von Repräsentantenvektoren $\mathcal{R}$. Diese werden als globale Merkmale interpretiert und mit einer angepassten Variante des in Abschnitt 2.2.1 erklärten *bisecting* k-Means Algorithmus in eine Clusterhierarchie eingeordnet.

4.3.1 Globale Merkmale

Die Merkmalsvektoren werden in diesem Schritt als global bezeichnet, da sie im Gegensatz zu den lokalen Merkmalen (Abschnitt 4.2.1) ohne räumliche Zuordnung und ohne Kontext verwendet werden. Ziel dieses Schrittes ist es, die einzelnen Merkmalsvektoren nach ihren Materialien zu gruppieren. Basierend auf der Clusterannahme (siehe Abschnitt 3.3) kann davon ausgegangen werden, dass Merkmalsvektoren, die vom gleichen Material stammen, zueinander ähnlich sind. Es bildet sich somit eine natürliche Hierarchie: Merkmalsvektoren von Wiesenflächen bilden eine von Baumflächen unterscheidbare Gruppe, welche zueinander dennoch ähnlicher sind als zur Gruppe der Straßenflächen (siehe Abschnitt 3.2). Da Materialien wiederholt und über die gesamte Szene verteilt auftreten können, findet – im Gegensatz zum vorherigen Segmentierungsschritt – keine Einschränkung auf die lokale Nachbarschaft statt. Dies ist durch die Verwendung von Repräsentantenvektoren gegeben, da diese keinen Ortsbezug mehr besitzen. Gesucht ist nun ein Clusterverfahren, dass eine in den Daten vorhandene Hierarchie aufdecken kann.

4.3.2 Erstellen der Hierarchie

Eine Hierarchie kann *bottom up* oder *top down* erstellt werden. *Bottom up* bedeutet hierbei, dass zunächst jedes Element eine Gruppe für sich bildet und anschließend die zueinander ähnlichsten Gruppen zusammengefasst werden. Dies wird wiederholt, bis nur noch eine Gruppe übrig ist. Diese Gruppe bildet die Wurzel des erstellten Baumes. *Top down* bedeutet, dass zunächst alle Elemente in einer gemeinsamen Gruppe sind. Diese Gruppe wird in Untergruppen geteilt, so dass diese sich möglichst unähnlich sind. Dies wird wiederholt bis jede Gruppe nur noch ein Element enthält. Die so aufgebaute Hierarchie ist in beiden Fällen ein Baum bestehend aus Knoten und Kanten. Jedes Blatt des Baumes enthält genau ein Element. Da es sich um ein hierarchisches Clusterverfahren handelt, enthalten alle darüber liegenden Knoten die Elemente des durch sie induzierten Unterbaumes. Die Wurzel des Baumes enthält somit alle vorhandenen Elemente.

Ein Nachteil von *bottom up* ist, dass alle Einzelgruppen paarweise miteinander verglichen werden müssen. Dies ist sehr aufwendig. Im Gegensatz hierzu werden bei der *Top-down*-Herangehensweise zunächst die deutlichsten Gruppen von einander getrennt. Dies ist unter Einsatz eines passendes Algorithmus effizient möglich. Solch ein *top-down*-basiertes hierarchisches Clusterverfahren ist der in Abschnitt 2.2.3 vorgestellte *bisecting* k-Means Algorithmus. Dieser muss jedoch noch für die Anwendung in dieser Arbeit angepasst werden.

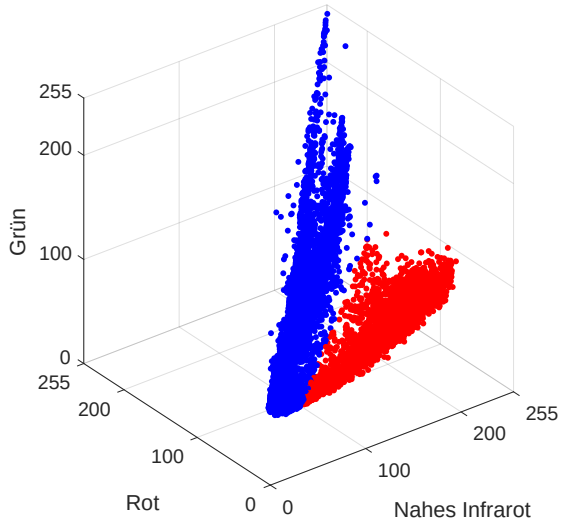
4.3.3 Anpassung von *bisecting* k-Means

Die Wahl des Ähnlichkeitsmaßes ist von großer Bedeutung für die Klassifikation von Multispektraldaten (siehe Abschnitt 3.2). Da die Repräsentantenvektoren aus dem ersten Schritt der vorgestellten Methode auf einer Segmentierung basieren, die mit dem SA (Gleichung 2.5) erstellt wurde, wird auch in diesem Schritt der SA als Ähnlichkeitsmaß verwendet. Es wird auf den räumlichen Anteil verzichtet, da die Repräsentantenvektoren hier, wie im vorherigen Abschnitt beschrieben, globale Merkmale darstellen. Die in diesem Schritt erstellte Clusterhierarchie ist ein Binärbaum, da der *bisecting* k-Means Algorithmus die Menge in jedem Schritt in zwei Teile trennt. Abbildung 4.3 zeigt den aufgeteilten Merkmalsraum mit dem zugehörigen Binärbaum für die erste und die dritte Iteration.

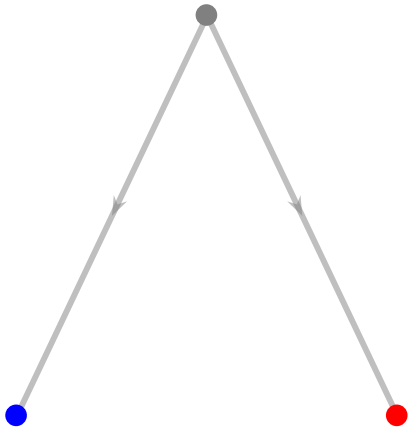
4.3.4 Zusammenspiel mit Segmentierungsschritt

Abhängig von der Parameterwahl findet im ersten Schritt eine Über- oder Untersegmentierung statt. Abbildung 4.4 zeigt jeweils ein Beispiel. Es ist zu erkennen, dass bei Untersegmentierung Bereiche zusammengefasst werden, die nicht zusammengehören. Durch das Überführen in Repräsentantenvektoren ist diese Zusammenfassung unumkehrbar. Die Folge ist, dass einzelne Merkmalsvektoren, die zuvor zu einem Repräsentantenvektor zusammengefasst wurden, für den restlichen Verlauf der Methode nicht mehr getrennt werden können.

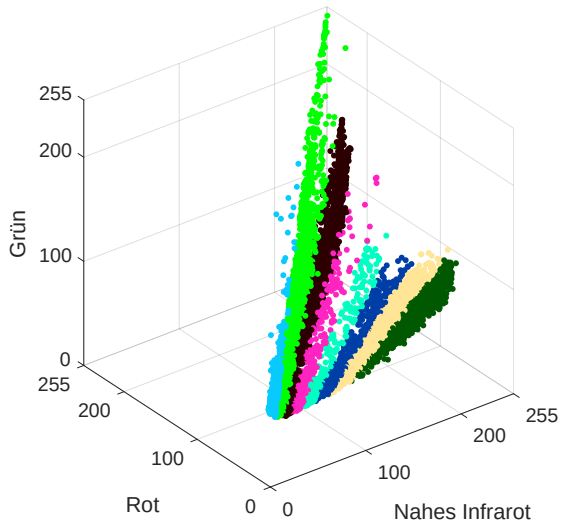
Tritt hingegen eine Übersegmentierung auf, wie auf der rechten Seite von Abbildung 4.4 gezeigt, wurden homogene Flächen in mehrere Segmente geteilt, obwohl hier eine Zusammenfassung sinnvoll ist. Dieses Zusammenfassen kann jedoch nachträglich im zweiten Schritt geschehen und stellt somit keinen Nachteil dar. Daher ist bei der Wahl des Segmentierungsparameters k ein zur Übersegmentierung führender Wert zu bevorzugen.



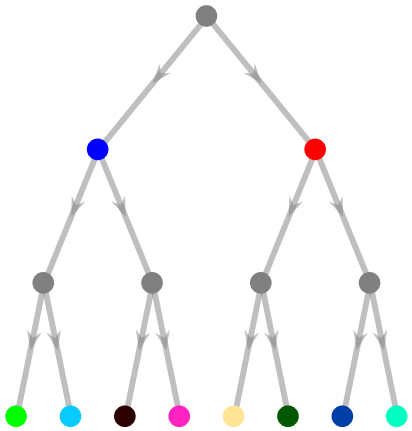
(a)



(b)

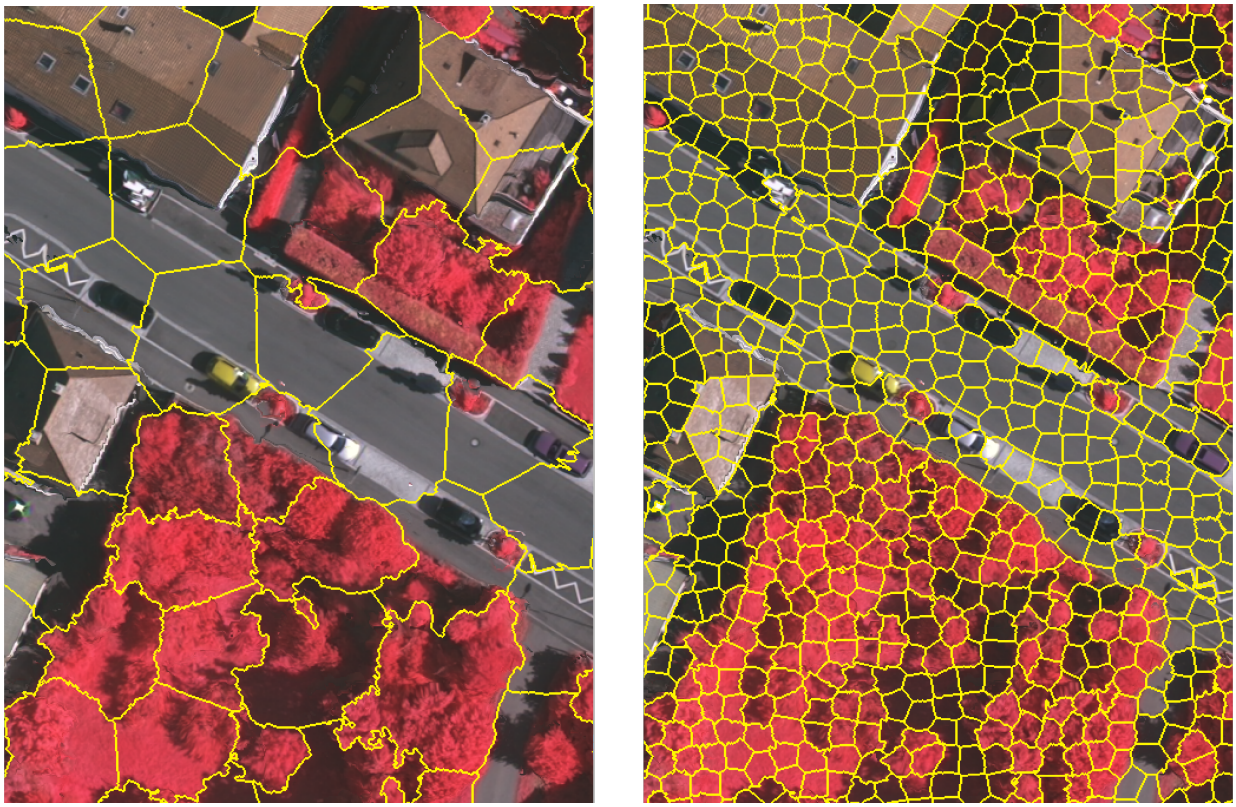


(c)



(d)

Abbildung 4.3: Abbildung (a) zeigt den Merkmalsraum und Abbildung (b) die Clusterhierarchie nach jeweils einer binären Teilung. Abbildungen (c) und (d) zeigen dies nach jeweils drei binären Teilungen. Aufgrund der Verwendung des SA als Distanzmaß, ist deutlich zu erkennen, dass die einzelnen Kegel immer stärker bezüglich ihres spektralen Winkels separiert werden [Wuttke et al., 2018].



(a) (b)

Abbildung 4.4: Bild (a) ist ein Beispiel einer Untersegmentierung ($k = 500$). Die Segmente links oben vereinen Dach und Straße. Rechts oben wird Dach und Vegetation vereinigt. Dies kann durch die Clusterbildung im zweiten Schritt nicht rückgängig gemacht werden und ist daher sehr schlecht. Bild (b) zeigt ein Beispiel von Übersegmentierung ($k = 10.000$). Selbst homogene Flächen (Dach, Straße, Vegetation) wurden mehrfach unterteilt. Dies kann jedoch im darauf folgenden Schritt der Clusterbildung korrigiert werden und hat daher keine negativen Auswirkungen – außer der erhöhten Berechnungszeit.

4.4 Aktives Lernen

Der dritte Schritt der vorgestellten Methode stellt den eigentlichen überwachten, aktiven Lernprozess dar. Die bisherigen Schritte dienten lediglich der Datenaufbereitung und fanden unüberwacht statt. Der im zweiten Schritt erstellte Binärbaum soll nun mit Klassenlabeln versehen werden. Dazu erhalten bestimmte Knoten das zu ihnen am besten passendste Klassenlabel. Diese Klassenlabel werden nach Trainingsende auf alle Repräsentantenvektoren, die im jeweiligen Knoten enthalten sind, angewendet und anschließend auf alle einzelnen Merkmalsvektoren des entsprechenden Segments verteilt. Das Ergebnis ist ein vollständiges Klassifikationsbild. Um die optimale Verteilung der Klassenlabel auf die Knoten zu finden, wiederholt dieser Schritt drei Teilschritte bis ein Stoppkriterium erfüllt ist. Diese Teilschritte sind in den folgenden Abschnitten detailliert erklärt.

4.4.1 Optimales Pruning bestimmen

Das Ergebnis des zweiten Schrittes ist ein Binärbaum $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, bestehend aus der Menge von Knoten $\mathcal{V}$ und Kanten $\mathcal{E}$. Der Baum unterteilt die Repräsentantenvektoren zunächst sehr detailliert, da im ersten Schritt eine Übersegmentierung erstellt wurde. Wird nun jedem Blatt das

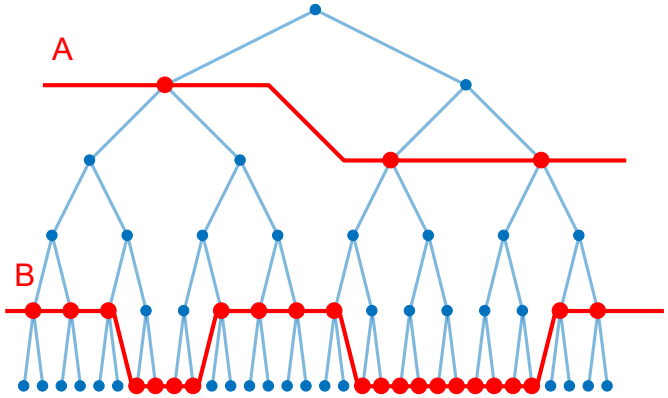


Abbildung 4.5: Das Ergebnis des Pruning-Vorgangs ist eine Menge von Knoten, so dass die durch jeden Knoten induzierten Teilbäume disjunkt sind und die Vereinigung aller Teilbäume alle Blätter des Originalbaumes enthält. Pruning $\mathcal{A}$ ist sehr nahe der Wurzel und resultiert in einer Untersegmentierung, da nur sehr wenige Blätter erhalten bleiben. Pruning $\mathcal{B}$ ist sehr tief im Baum und resultiert in einer Übersegmentierung, da sehr viele Blätter erhalten bleiben [Wuttke et al., 2018].

der *Ground Truth* entsprechende Klassenlabel zugeordnet, führt dies zu einer perfekten Klassifikation, da jedes Blatt nur genau ein Repräsentantenvektor enthält und dieser mit der *Ground Truth* versehen wurde. Dies ist jedoch im Sinne des aktiven Lernens nicht optimal. Da keinerlei Zusammenfassung stattfand, muss die *Ground Truth* für jede einzelne Stichprobe beschafft werden. Dieser Aufwand ist in der Praxis zu groß. In stärker generalisierten Ebenen der Hierarchie sind Repräsentantenvektoren zusammengefasst, so dass sie sich Klassenlabel teilen und weniger *Ground-Truth*-Anfragen benötigt werden, um alle Knoten auf dieser Ebene mit einem Klassenlabel zu versehen. Hierbei kommt es jedoch in der Regel zu Klassifikationsfehlern. Der Extremfall ist das Zusammenfassen in der Wurzel des Baumes, so dass alle Repräsentantenvektoren das selbe Klassenlabel erhalten. Gesucht ist nun eine Abwägung zwischen Klassifikationsfehler und Beschaffungsaufwand der benötigten *Ground Truth*. Die Abwägung kann gesteuert werden, indem der Binärbaum gestutzt wird. Dabei werden die unteren Teile des Baumes abgeschnitten. Das Ergebnis ist ein reduzierter Baum, ein sogenanntes *Pruning* (englisch: *to prune*, stutzen).

❑ Definition eines Prunings

Jeder Knoten $v \in \mathcal{V}$ des Baumes T induziert einen Unterbaum $T'(v)$ dessen Wurzel v ist. Ein Pruning ist eine Teilmenge der Knoten des Baumes, so dass alle induzierten Unterbäume disjunkt sind und die Vereinigung der in den Pruning-Knoten enthaltenen Repräsentantenvektoren gleich der Gesamtmenge $\mathcal{R}$ aller Repräsentantenvektoren des Baumes ist (vergleiche Abbildung 4.5). Formal ist ein Pruning $\mathcal{P}$ somit eine Knotenmenge, die folgende Bedingungen erfüllt:

$$\mathcal{P} \subseteq \mathcal{V} \tag{4.4}$$

$$\bigcup_{i=1}^{|\mathcal{P}|} \{r \in \mathcal{P}^{(i)}\} = \mathcal{R} \tag{4.5}$$

$$\forall v, w \in \mathcal{P}, v \neq w : T'(v) \cap T'(w) = \varnothing \tag{4.6}$$

Es existieren mehrere verschiedene Prunings. Wird jeder Knoten eines Prunings mit Hilfe der *Ground Truth* mit einem Klassenlabel versehen, induziert dies ein vollständiges Labeling für alle Repräsentantenvektoren. Prunings weiter „oben“ im Baum benötigen weniger *Ground-Truth*-Anfragen, haben jedoch einen größeren Klassifikationsfehler. Bei Prunings weiter „unten“

im Baum ist dies umgekehrt. Sie haben geringere Klassifikationsfehler, benötigen jedoch mehr *Ground-Truth*-Anfragen. Der Klassifikationsfehler soll im Folgenden approximiert werden.

❑ Klassifikationsfehler

Mit jedem Knoten v ist eine sogenannte Knotenstatistik verbunden. Diese enthält die Anzahl der im Knoten enthaltenen Repräsentantenvektoren (n_v) und die Anzahl ($l_{v,w}$) derer von denen das Klassenlabel als w bekannt. Der relative Anteil an mit w gelabelten Repräsentantenvektoren sei $p_{v,w}$. Wird dieses Label auch für alle anderen im Knoten enthaltenen Elemente verwendet, sind im schlechtesten Fall $1 - p_{v,w}$ Elemente falsch gelabelt. Dies dient als Approximation des Klassifikationsfehler ε_v:

$$\begin{aligned} p_{v,w} &= \frac{l_{v,w}}{n_v} \\ \varepsilon_v &= 1 - \max_{w \in \Omega} p_{v,w} \end{aligned} \tag{4.7}$$

❑ Unsicherheitsschranken

Aufgrund dieser Approximation ist der berechnete Fehlerwert mit Unsicherheit belegt. Je weniger Klassenlabel bekannt sind und je mehr Elemente der Knoten enthält, desto unsicherer ist der approximierte Fehler. Er lässt sich jedoch durch folgende obere (OS) und untere Schranke (US) eingrenzen:

$$p_{v,w}^{OS} = \min\left(p_{v,w} + \Delta_{v,w}, 1\right) \tag{4.8}$$

$$p_{v,w}^{US} = \max\left(p_{v,w} - \Delta_{v,w}, 0\right) \tag{4.9}$$

Bei der Definition des Unsicherheitsterms $\Delta_{v,w}$ folgt die vorliegende Arbeit den Veröffentlichungen von Dasgupta & Hsu [2008] sowie Muñoz-Marí et al. [2012]:

$$\Delta_{v,w} = \frac{c_v}{n_v} + \sqrt{\frac{c_v p_{v,w}(1 - p_{v,w})}{n_v}} \tag{4.10}$$

$$c_v = 1 - \frac{l_v}{n_v} \tag{4.11}$$

$$l_v = \sum_{w \in \Omega} l_{v,w} \tag{4.12}$$

Hierbei ist c_v ein Korrekturterm, der proportional zur Anzahl n_v der Elemente im Knoten v und umgekehrt proportional zur Anzahl der bekannten Klassenlabel pro Knoten l_v ist. Dies spiegelt die oben geforderte Eigenschaft wider, dass mehr bekannte Klassenlabel die Sicherheit der Approximation steigern. Basierend auf dieser Approximation können nun die Kosten einer Fehlklassifikation bestimmt werden.

❑ Fehlklassifikationskosten

Für einen Knoten kommen alle Klassenlabel in Frage, für die gilt: $p_{v,w} > 0$. Dies sind typischerweise mehrere. Die Bestimmung der Fehlklassifikationskosten geschieht hier sehr konservativ. Die Entscheidung für ein bestimmtes Klassenlabel w aus allen möglichen für den Knoten v wird *akzeptabel* genannt, wenn w maximal den doppelten Klassifikationsfehler verursacht wie alle anderen

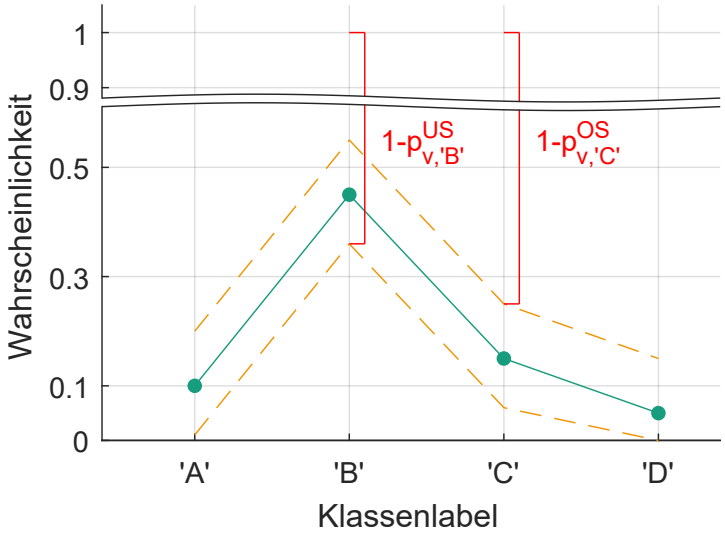


Abbildung 4.6: Beispielhafte Verteilung der Klassenlabel für einen Knoten (Abbildung nach [Wuttke et al., 2018]). Der relative Anteil $p_{v,w}$ ist in grün und die Unsicherheitsschranken $p^{US}_{v,w}$ und $p^{OS}_{v,w}$ sind in orange dargestellt. Klassenlabel B ist *akzeptabel*, da der maximale Klassifikationsfehler für Klassenlabel B $(1 - p^{US}_{v,\text{'B'}} = 0,64)$ kleiner ist als der doppelte minimale Klassifikationsfehler für Klassenlabel C $(2 \cdot (1 - p^{OS}_{v,\text{'C'}}) = 1,5)$.

Klassenlabel w'. Die unteren und oberen Schranken werden bei der Definition einbezogen. Die Kombination (v, w) gilt als akzeptabel, wenn sie folgendes Kriterium erfüllt:

$$\begin{aligned}(v,w)\,\text{akzeptabel} \iff & \quad (1 - p^{US}_{v,w}) < 2 \cdot \min_{w' \neq w} \left(1 - p^{OS}_{v,w'}\right) \\ \iff & \quad p^{US}_{v,w} > 2 \cdot p^{OS}_{v,w'} - 1 \qquad ,\forall w \neq w' \end{aligned} \tag{4.13}$$

Es ist anzumerken, dass mehrere Klassenlabel pro Knoten akzeptabel sein können. Abbildung 4.6 veranschaulicht den Zusammenhang zwischen Unsicherheitsschranken und Klassifikationsfehler.

Mit Hilfe des soeben definierten Akzeptanzkriteriums wird nun der Klassifikationsfehler für die Wahl eines bestimmten Klassenlabels für einen Knoten definiert:

$$\varepsilon_{v,w} = \begin{cases} 1 - p_{v,w} & ,\text{falls } (v,w) \text{ akzeptabel} \\ 1 & ,\text{sonst} \end{cases} \tag{4.14}$$

❑ Optimierung

Wie eingangs beschrieben, besitzt das Pruning, welches nur aus den Blättern des Baumes besteht, die geringsten Fehlklassifikationskosten. Da es jedoch nur unter Zuhilfenahme der gesamten *Ground Truth* erstellt werden kann, ist es im Sinne des aktiven Lernens nicht optimal. Umgekehrt kann das Pruning, welches nur aus der Wurzel besteht, zwar mit einem einzigen Klassenlabel vervollständigt werden, verursacht dabei jedoch sehr hohe Fehlklassifikationskosten. Optimalität ist in diesem Zusammenhang daher ein Kompromiss zwischen Anzahl benötigter Klassenlabel und Klassifikationsfehlern. Da der Baum nur endlich viele Elemente enthält, gibt es auch nur endlich viele Prunings. Mindestens eines von diesen muss den geringsten Klassifikationsfehler besitzen und wird daher als optimal bezeichnet.

Die Suche nach dem optimalen Pruning wird initialisiert mit dem Pruning, das nur aus der Wurzel des Binärbaumes besteht. Für jeden Knoten v des aktuellen Prunings werden die Fehlklassifikationskosten seiner beiden Kindknoten v_l und v_r addiert und mit denen des Knotens selbst verglichen:

$$\varepsilon_{v,w} > \varepsilon_{v_l,w} + \varepsilon_{v_r,w}, \forall w \in \Omega \tag{4.15}$$

Gilt diese Ungleichung für jede Wahl des Klassenlabels w, wird Knoten v im Pruning durch seine beiden Kindknoten ersetzt. Das Ergebnis ist ein Pruning mit geringeren Fehlklassifikationskosten. Dies wird solang wiederholt, bis kein besseres Pruning mehr möglich ist. Um das nun optimale Pruning weiter zu verbessern, sind weitere Label-Informationen erforderlich. Daher gilt, dass in jeder Iteration das Pruning gefunden wird, welches – unter den Voraussetzungen der aktuell verfügbaren Label-Informationen – optimal ist. Da jedes Pruning gleichzeitig eine vollständige Belegung mit Klassenlabeln für das gesamte Bild induziert, kann der Lernprozess nach jeder Iteration abgebrochen werden und liefert das zu diesem Zeitpunkt optimale Ergebnis. Dies ist ein Vorteil gegenüber vielen anderen Lernverfahren, die erst nach ihrer kompletten Abarbeitung eine vollständige Belegung mit Klassenlabeln zur Verfügung stellen.

4.4.2 Optimale Stichprobe bestimmen

Im vorherigen Teilschritt wurde das optimale Pruning unter den Voraussetzungen der aktuell gegebenen Label-Informationen bestimmt. Da an das Orakel nur Anfragen zu einzelnen Repräsentantenvektoren gestellt werden können, muss als nächstes die anzufragende Stichprobe bestimmt werden. Eine Möglichkeit ist die zufällige Auswahl eines bisher ungelabelten Blattes. Dieses Auswahlverfahren erhöht zwar zuverlässig die Menge an vorhandenen Label-Informationen, führt jedoch auf lange Sicht zu mehr Anfrageaufwand als eine aktive Selektionsstrategie.

In diesem Teilschritt wird jener Repräsentantenvektor bestimmt, dessen *Ground-Truth*-Klassenlabel am vorteilhaftesten für den Lernprozess ist. Hierzu wird zunächst der Knoten des Prunings bestimmt, dessen Klassenlabel den größten Mehrwert bietet. Anschließend findet ein Abstieg von diesem Knoten bis zu einem Blatt des Baumes statt. Dieses Blatt enthält den Repräsentantenvektor, dessen Klassenlabel im nächsten Teilschritt beim Orakel angefragt wird. Die folgenden Abschnitte erläutern diese Selektionsstrategie.

❑ Knotenauswahlstrategie

Eine semi-aktive Strategie ist, immer den Knoten zu wählen, der am meisten ungelabelte Stichproben enthält. Selektionsstrategien des aktiven Lernens zeichnen sich jedoch dadurch aus, dass sie Informationen über den aktuellen Zustand des Lernprozesses und dessen Unsicherheiten berücksichtigen. Solch eine Strategie lässt sich hier anwenden, da Informationen zur verbleibenden Unsicherheit durch die Schranken der Klassifikationswahrscheinlichkeit vorliegen.

Für die Strategie spielen hierbei zwei Einflussfaktoren eine Rolle: (i) die Klassifikationsunsicherheit $(1-p^{US}_{v,w})$ und (ii) die Knotengröße (n_v). Für die Klassifikationsunsicherheit gilt, je größer die Unsicherheit ist, desto mehr kann der Lernprozess noch profitieren. Das Abfragen des Klassenlabels für einen Repräsentantenvektor aus einem bereits sehr sicher bestimmten Knoten bringt wenig neue Informationen. Für die Knotengröße gilt, dass der Lernprozess vor allem von Knoten mit vielen enthaltenen Elementen profitiert. Knoten mit wenigen Elementen können zwar schneller mit hoher Sicherheit gelabelt werden, jedoch wirken sie sich auf weniger Repräsentantenvektoren aus. Wird ein Knoten gelabelt, wirkt sich dies auf alle darin enthaltenen Elemente aus. Es profi-

tieren somit umso mehr Merkmalsvektoren von den neuen Label-Informationen, je mehr Elemente der Knoten enthält. Dies wird zusammengefasst in der Auswahlstrategie:

$$v^* = \underset{v \in \mathcal{P}}{\operatorname{argmax}} \left(n_v (1 - p^{US}_{v,w}) \right) \tag{4.16}$$

❑ **Abstiegsstrategie**

Nachdem ein Knoten des aktuellen Prunings ausgewählt wurde, muss ein in diesem Knoten enthaltenes Blatt ausgewählt werden, da beim Orakel nur einzelne Repräsentantenvektoren angefragt werden können. Da es sich um einen Binärbaum handelt, muss wiederholt zwischen dem linken und dem rechten Kind entschieden werden. Erreicht dieser Abstieg ein Blatt, ist der Repräsentantenvektor gefunden, dessen Klassenlabel aktuell den meisten Mehrwert bietet.

Für die Entscheidung, in welchem Kindknoten der Abstieg fortgesetzt werden soll, gibt es wie im Fall der Knotenauswahlstrategie passive und aktive Möglichkeiten. Eine passive Möglichkeit ist die zufällige Auswahl. Aktive Möglichkeiten berücksichtigen die aktuellen Unsicherheiten im Lernprozess. Hier wird die selbe Strategie verwendet wie zur Knotenauswahl, jedoch nur für die beiden Kindkonten v_l und v_r angewendet:

$$v^* = \underset{v \in \{v_l, v_r\}}{\operatorname{argmax}} \left(n_v (1 - p^{US}_{v,w}) \right) \tag{4.17}$$

Die Menge der Knoten, die beim Abstieg betrachtet wurden bilden den *Abstiegspfad* $\mathcal{A} \subset \mathcal{V}$. Nachdem das Blatt mit dem größten Mehrwert bestimmt wurde, steht der Repräsentantenvektor fest, dessen Klassenlabel dem Lernprozess am meisten hilft. Diese Anfrage geschieht im nächsten Teilschritt.

4.4.3 Orakelanfrage und Aktualisierung

Der ausgewählte Repräsentantenvektor wird dem Orakel vorgelegt, welches mit dem zugehörigen Klassenlabel w^* antwortet. Diese neue Label-Information muss nun in die bestehende Hierarchie integriert werden. Alle Werte, die für Entscheidungen während einer Iteration benötigt werden, lassen sich wie folgt zurückführen:

$$\begin{aligned}
\varepsilon_{v,w} &\overset{4.15}{\rightsquigarrow} p_{v,w}\,,\, p^{OS}_{v,w}\,,\, p^{US}_{v,w} \\
p^{OS}_{v,w} &\overset{4.8}{\rightsquigarrow} p_{v,w}\,,\, \Delta_{v,w} \\
p^{US}_{v,w} &\overset{4.9}{\rightsquigarrow} p_{v,w}\,,\, \Delta_{v,w} \\
\Delta_{v,w} &\overset{4.10}{\rightsquigarrow} c_v\,,\, n_v\,,\, p_{v,w} \\
c_v &\overset{4.11}{\rightsquigarrow} l_v\,,\, n_v \\
l_v &\overset{4.12}{\rightsquigarrow} l_{v,w} \\
p_{v,w} &\overset{4.7}{\rightsquigarrow} l_{v,w}\,,\, n_v \\
n_v &\rightsquigarrow \text{konstant} \\
l_{v,w} &\rightsquigarrow \text{variabel}
\end{aligned}$$

Die einzige zu aktualisierende Knotenstatistik ist somit der Wert für $l_{v,w}$, welcher um 1 erhöht werden muss, da ein neues Klassenlabel bekannt ist. Konkret betrifft dies alle Knoten, die das ausgewählte Blatt enthalten. Dies sind alle Knoten auf dem Pfad vom Blatt zur Wurzel (siehe Abbildung 4.7).

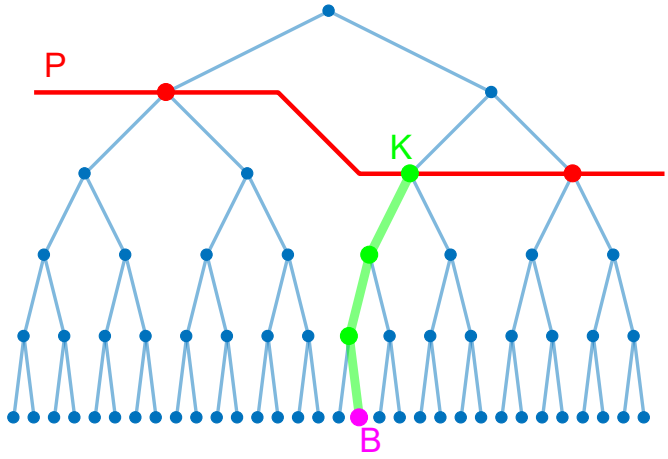


Abbildung 4.7: Visualisierung einer Iteration. Im ersten Teilschritt wird das optimale Pruning $\mathcal{P}$ (rot) bestimmt. Im zweiten Teilschritt findet, vom optimalen Knoten K ausgehend, ein Abstieg (grün) zum optimalen Blatt (B) statt. Das Klassenlabel für den im Blatt enthaltenen Repräsentantenvektor wird beim Orakel angefragt. Es müssen nur die im Abstiegspfad (grün) enthaltenen Knoten aktualisiert werden.

Da das Pruning jedoch nur nach unten wandert und nicht nach oben, haben Knoten oberhalb des Prunings keinen Einfluss mehr. Daher brauchen deren Knotenstatistiken nicht aktualisiert werden. Die Menge der zu aktualisierenden Knoten entspricht genau dem Abstiegspfad $\mathcal{A}$ und die Menge der zu aktualisierenden Knotenstatistiken ist definiert über:

$$\{l_{v,w} \,|\, \forall v \in \mathcal{A} \wedge w = w^*\} \tag{4.18}$$

Hierfür reicht eine einfache Iteration über die Knoten des Abstiegspfads.

4.4.4 Berücksichtigung der lokalen Dichte

Teilschritt 2 zur Bestimmung der Stichprobe mit dem größten Mehrwert berücksichtigt die Unsicherheit und Größe eines Knotens. Größe steht hierbei nur für die absolute Anzahl an enthaltenen Elementen. Dies sagt jedoch noch nichts über die lokale Dichte aus. Daher soll hier eine Alternative für Teilschritt zwei vorgestellt werden, welche die lokale Dichte berücksichtigt.

❑ Lokale Dichte

Die lokale Dichte bezeichnet die Anzahl der Elemente gewichtet mit ihrer Entfernung zueinander. Entfernung steht hierbei für ein Distanzmaß. Die Verwendung der Dichte statt der absoluten Anzahl erlaubt detailliertere Aussagen über einen Knoten. So ist beispielsweise der Mehrwert eines Klassenlabels für einen Knoten mit 10 Elementen, die sehr nahe zueinander sind (sehr ähnlich), höher als für einen Knoten mit 10 Elementen, die eine sehr große Entfernung voneinander haben (sehr unähnlich).

In dieser Arbeit wird die lokale Dichte durch das Summieren der paarweisen Ähnlichkeitswerte der Merkmalsvektoren bestimmt. Die Ähnlichkeitsfunktion ist wiederum eine skalierte Distanzfunktion. Als Ähnlichkeitsmaß wird hierfür, wie in den anderen Schritten der vorgestellten Methode, der SA (Gleichung 3.23) verwendet. Für die Skalierung kommt eine Exponentialfunktion zum Einsatz. Diese Funktion sorgt für eine Normalisierung auf das Intervall [0..1]. Es gilt für die Distanzfunktion $\mathbf{D}_{SA}(\mathbf{p}, \mathbf{p}) = 0$ und nach der Skalierung für die Ähnlichkeitsfunktion: $\mathbf{s}(\mathbf{p}, \mathbf{p}) = 1$.

Über den Parameter σ kann die Stärke der Abschwächung mit zunehmender Entfernung gesteuert werden. Sie ist wie folgt definiert:

$$\mathbf{s}(\mathbf{p}, \mathbf{q}) = e^{-\frac{\mathbf{D}_{SA}(\mathbf{p},\mathbf{q})^2}{\sigma^2}} \tag{4.19}$$

Die lokale Dichte $\mathbf{L}$ an einem bestimmten Ort $\mathbf{p}$ im Merkmalsraum ist nun die Summe der Ähnlichkeit aller Elemente zu diesem Ort:

$$\mathbf{L}(\mathbf{p}) = \sum_{\mathbf{r}\in\mathcal{R}} \mathbf{s}(\mathbf{p}, \mathbf{r}) \tag{4.20}$$

Dies ist unabhängig vom tatsächlichen Klassenlabel der Repräsentantenvektoren. Alle Dichtewerte können somit vorab und unüberwacht berechnet werden.

❑ Integration der PAL-Rahmenstruktur

Die Grundlagen zur Rahmenstruktur des probabilistischen aktiven Lernens wurden bereits in Abschnitt 2.4.1 erläutert. An dieser Stelle soll darauf eingegangen werden, wie die Integration in die SCHAL-Methode geschieht. Als Bezeichnung für diese Variante dient das Akronym SCHPAL (Segmentierung, Clusterhierarchie, probabilistisches aktives Lernen).

Zunächst wird die Label-Häufigkeit h bestimmt. Diese Häufigkeit ist für jede Kombination aus Klassenlabel und Position im Merkmalsraum definiert als die Summe der Ähnlichkeiten aller Repräsentantenvektoren mit dem festgelegten Klassenlabel zur festgelegten Position:

$$h_{\mathbf{p},w} = \sum_{\mathbf{r}\in\mathcal{R},\Omega(\mathbf{r})=w} \mathbf{s}(\mathbf{p}, \mathbf{r}) \tag{4.21}$$

Anschließend wird der probabilistische Nutzen für einen bestimmten Repräsentantenvektor $\mathbf{r}$ bestimmt. Hierzu werden die Label-Häufigkeiten für alle potenziellen Klassenlabel zu einem Vektor $\mathbf{f_r}$ zusammengefasst. Abbildung 4.8 verdeutlicht wie dieser Häufigkeitsvektor zustande kommt. Aus diesem Vektor wird mit Hilfe der PAL-Rahmenstruktur der probabilistische Nutzen errechnet. Dieser Nutzen wird mit dem Dichtevektor gewichtet, um so den Einfluss $\mathbf{E}$ des Repräsentantenvektors zu erhalten:

$$\begin{aligned} \mathbf{E}(\mathbf{r}) &= \mathbf{L}(\mathbf{r}) \cdot \mathrm{perfGain}(\mathbf{f_r}) \\ \mathbf{f_r} &= \left(h_{\mathbf{r},1}\,,\,\ldots\,,\,h_{\mathbf{r},|\Omega|}\right) \end{aligned} \tag{4.22}$$

Anschließend wird der Repräsentantenvektor ausgewählt, der den größten Einfluss auf den weiteren Trainingsverlauf hat. Somit ergibt sich die neue Auswahlstrategie:

$$\mathbf{r}^* = \operatorname*{argmax}_{\mathbf{r}\in\mathcal{R}} \mathbf{E}(\mathbf{r}) \tag{4.23}$$

Anschließend wird der dritte Schritt der SCHAL-Methode fortgeführt wie bisher. Das Klassenlabel für den ausgewählten Repräsentantenvektor wird vom Orakel angefragt, die Knotenstatistiken werden aktualisiert und die nächste Iteration beginnt.

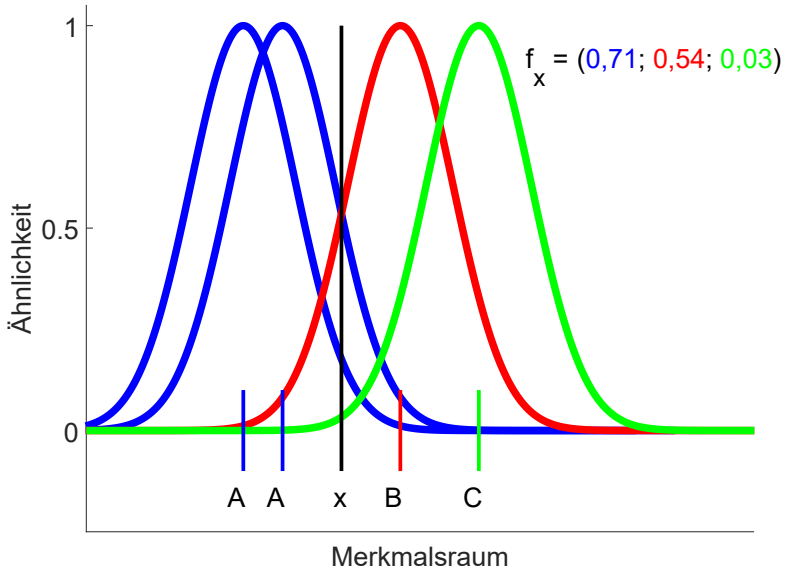


Abbildung 4.8: Beispielhafte Label-Verteilung zur Veranschaulichung der Berechnung des Label-Häufigkeitsvektors $\mathbf{f_x}$. Dargestellt sind vier Elemente aus den Klassen 'A', 'B' und 'C' sowie ihre zugehörige distanzabhängige Ähnlichkeitsfunktion. Um die Label-Häufigkeit an Position $\mathbf{x}$ zu bestimmen, werden die Ähnlichkeiten für jede Klasse separat aufsummiert.

4.5 Komplexitätsbetrachtung

In diesem Abschnitt wird die Komplexität der vorgestellten Methode untersucht. Hierzu wird die $\mathcal{O}$-Notation nach [Knuth, 1976] verwendet, um die verschiedenen Komplexitätsklassen anzugeben.

4.5.1 Segmentierung

Der Segmentierungsschritt besteht aus dem Berechnen der Segmente mit dem SLIC-Algorithmus und der Bestimmung der Repräsentantenvektoren. Der SLIC-Algorithmus hat, wie der zu Grunde liegende k-Means Algorithmus, eine lineare Komplexität (siehe Abschnitt 2.2.1). Die Repräsentantenvektoren werden durch Berechnung des bandweisen Medians bestimmt. Da die Bandanzahl deutlich kleiner ist als die Anzahl der Merkmalsvektoren, kann sie für die Komplexitätsbetrachtung vernachlässigt werden und es dominiert die Komplexität der Medianberechnung. Diese ist mit linearem Aufwand möglich [Blum et al., 1973]. Somit ergibt sich für den ersten Schritt insgesamt eine lineare Komplexität bezüglich der Anzahl N an Pixeln beziehungsweise Merkmalsvektoren des zu klassifizierenden Bildes:

$$\begin{aligned}
\mathbf{f}_{\text{SLIC}} &\in \mathcal{O}(N) \\
\mathbf{f}_{\text{Median}} &\in \mathcal{O}(N) \\
\mathbf{f}_{\text{Segmentierung}} &\in \mathcal{O}(\mathbf{f}_{\text{SLIC}}) + \mathcal{O}(\mathbf{f}_{\text{Median}}) \\
&= \mathcal{O}(N) + \mathcal{O}(N) \\
&= 2 \cdot \mathcal{O}(N) \\
&= \mathcal{O}(N)
\end{aligned} \tag{4.24}$$

4.5.2 Clusterbildung

Der zweite Schritt besteht lediglich aus dem Erstellen der Clusterhierarchie mit dem *bisecting* k-Means Algorithmus. Dieser besitzt lineare Komplexität (siehe Abschnitt 2.2.3). Daher liegt der Clusterbildungsschritt der Methode auch in der linearen Komplexitätsklasse bezüglich der Anzahl $|\mathcal{R}|$ der zu clusternden Repräsentantenvektoren.

$$\mathbf{f}_{\text{Clustering}} \in \mathcal{O}(|\mathcal{R}|) \tag{4.25}$$

4.5.3 Aktives Lernen

Zunächst wurde das optimale Pruning bestimmt, anschließend der darin am besten geeignetste Knoten und schließlich fand ein Abstieg zum am besten geeignetsten Blatt statt. Das Klassenlabel für den im Blatt enthaltenen Repräsentantenvektor wurde beim Orakel angefragt und alle Knoten entlang des Abstiegspfades entsprechend aktualisiert. Da sich in jeder Iteration nur die Knoten entlang des Abstiegspfades ändern, können bei der Implementierung der Methode einige Optimierungen vorgenommen werden.

❑ Optimales Pruning bestimmen

Zur Optimierung des Prunings müssen nicht alle Knoten betrachtet werden, sondern nur der in der letzten Iteration ausgewählte Knoten. Dieses Betrachten eines Knotens besitzt eine konstante Komplexität: $\mathcal{O}(1)$. Mit jeder Aufteilung kommen zwei neue Knoten zum Pruning hinzu, die beide potenziell wiederum aufgeteilt werden können, dies entspricht einer exponentiellen Komplexität $\mathcal{O}(2^n)$. Da sich mit jeder Aufteilung die Menge der enthaltenen Repräsentantenvektoren in etwa halbiert, ist die Höhe des Baumes logarithmisch beschränkt bezüglich der Anzahl der Repräsentantenvektoren: $\mathcal{O}(\log_2|\mathcal{R}|)$. Somit ist die resultierende Komplexität linear: $\mathcal{O}(2^{\log_2|\mathcal{R}|}) = \mathcal{O}(|\mathcal{R}|)$. Die Änderung durch ein neues Klassenlabel beschränkt sich jedoch auf einen der beiden Kindknoten, so dass nie beide Knoten gleichzeitig gesplittet werden müssen. Dieses Vorgehen führt daher in der Praxis zu einer weiterhin logarithmischen Komplexität.

$$\mathbf{f}_{\text{Pruning}} \in \mathcal{O}(\log_2|\mathcal{R}|) \tag{4.26}$$

❑ Knotenauswahl

Für die Auswahl des Knotens mit dem größten Mehrwert aus dem aktuellen Pruning gilt: das Pruning hat eine feste Länge und jeder Knoten muss maximal einmal betrachtet werden. Diese Auswahl entspricht der Suche des Maximums auf einer Liste fester länger (siehe Gleichung 4.16). Die Vergrößerung des Prunings in jeder Iteration geschieht nur durch das Aufteilen von Knoten. Hierfür müssen die Kindknoten eine Belegung mit Klassenlabeln besitzen, die das Akzeptanzkriterium (4.13) erfüllt. Da sich in jeder Iteration nur die Werte von Knoten entlang des Abstiegspfades ändern, ändert sich auch maximal eines der Kinder des aufzuteilenden Knotens. Das Anwachsen des Prunings ist somit auf die Länge des Abstiegspfades begrenzt, welche wiederum durch die Höhe des Baumes begrenzt ist. Daher kann die Länge des Prunings in einer Iteration durch $|\mathcal{P}| \leq k \cdot |\mathcal{A}| \leq k \cdot \log_2|\mathcal{R}|$ nach oben abgeschätzt werden. Die Komplexität der Knotenauswahlstrategie ist somit:

$$\begin{aligned}\mathbf{f}_{\text{Knoten}} &\in \mathcal{O}(|\mathcal{P}|) \\ &= \mathcal{O}(k \cdot \log_2|\mathcal{R}|) \\ &= \mathcal{O}(\log_2|\mathcal{R}|)\end{aligned} \tag{4.27}$$

❑ Blattauswahl

Die Auswahl des besten Blattes (Abstiegsstrategie 4.16) entscheidet sich in jedem Schritt für eines der beiden Kinder, bis ein Blatt erreicht ist. Dies führt zu einer logarithmischen Komplexität in Bezug zur Gesamtanzahl der Repräsentantenvektoren:

$$\mathbf{f}_{\text{Abstieg}} \in \mathcal{O}(\log_2 |\mathcal{R}|) \tag{4.28}$$

❑ Aktualisierung

Die Aktualisierung der Knotenstatistiken erfolgt, wie vorher beschrieben, nur auf der Menge des Abstiegspfades und beschränkt sich auf konstante Rechenoperationen. Die Länge des Abstiegspfades wird im Laufe der Iterationen kürzer, da das Pruning im Baum nach unten wandert. Die Maximale Länge ist durch die Höhe des Baumes beschränkt: $|\mathcal{A}| \leq \log_2|\mathcal{R}|$. Aus dieser Beschränkung ergibt sich eine lineare Komplexität bezüglich der Länge des Abstiegspfades und eine logarithmische Komplexität bezüglich der Gesamtmenge an Repräsentantenvektoren:

$$\begin{aligned} \mathbf{f}_{\text{Aktualisierung}} &\in \mathcal{O}(|\mathcal{A}|) \\ &= \mathcal{O}(\log_2 |\mathcal{R}|) \end{aligned} \tag{4.29}$$

Somit ergibt sich für eine Iteration der Methode folgende Komplexität:

$$\begin{aligned} \mathbf{f}_{\text{Iteration}} &\in \mathcal{O}(\mathbf{f}_{\text{Pruning}}) + \mathcal{O}(\mathbf{f}_{\text{Knoten}}) + \mathcal{O}(\mathbf{f}_{\text{Abstieg}}) + \mathcal{O}(\mathbf{f}_{\text{Aktualisierung}}) \\ &= \mathcal{O}(\log_2 |\mathcal{R}|) + \mathcal{O}(\log_2 |\mathcal{R}|) + \mathcal{O}(\log_2 |\mathcal{R}|) + \mathcal{O}(\log_2 |\mathcal{R}|) \\ &= 4 \cdot \mathcal{O}(\log_2 |\mathcal{R}|) \\ &= \mathcal{O}(\log_2 |\mathcal{R}|) \end{aligned} \tag{4.30}$$

❑ Vorberechnung lokaler Dichte

Da alle Repräsentantenvektoren konstant sind, ändern sich ihre Distanzen untereinander nicht. Daher sind auch die Ähnlichkeiten und die lokalen Dichten konstant und können vorberechnet werden. Diese Vorberechnung besitzt jedoch quadratische Komplexität, da die Ähnlichkeit von jedem zu jedem anderen Element benötigt wird.

$$\mathbf{f}_{\text{PAL-Vorberechnung}} \in \mathcal{O}(|\mathcal{R}|^2) \tag{4.31}$$

Zur Berechnung des Label-Häufigkeitsvektors während jeder Iteration genügt es, über die vorberechneten Ähnlichkeiten zu summieren. Dies muss für alle noch ungelabelten Repräsentantenvektoren, die im betrachteten Knoten enthalten sind, durchgeführt werden. Wie zuvor bei der Bestimmung des besten Knotens ergibt sich hier eine logarithmische Komplexität.

$$\mathbf{f}_{\text{Label-Häufigkeit}} \in \mathcal{O}(\log_2 |\mathcal{R}|) \tag{4.32}$$

Die Komplexität für das Optimieren des Prunings bleibt identisch. Das Bestimmen des besten Knotens und Blattes entfällt. Die Komplexität für das Aktualisieren bleibt gleich, da die Aktualisierung weiterhin für die Optimierung des Prunings benötigt wird. Das zusätzliche Speichern des neuen Klassenlabels zur Berechnung des Label-Häufigkeitsvektors kann vernachlässigt werden. Es ergibt sich folgende Komplexität für eine Iteration unter Verwendung der PAL-Rahmenstruktur:

$$\begin{aligned}\mathbf{f}_{\text{PAL-Iteration}} &\in \mathcal{O}(\mathbf{f}_{\text{Pruning}}) + \mathcal{O}(\mathbf{f}_{\text{Label-Häufigkeit}}) + \mathcal{O}(\mathbf{f}_{\text{Aktualisierung}}) \\ &= \mathcal{O}(\log_2 |\mathcal{R}|) + \mathcal{O}(\log_2 |\mathcal{R}|) + \mathcal{O}(\log_2 |\mathcal{R}|) \\ &= 3 \cdot \mathcal{O}(\log_2 |\mathcal{R}|) \\ &= \mathcal{O}(\log_2 |\mathcal{R}|) \end{aligned} \tag{4.33}$$

4.5.4 Gesamtkomplexität

Sei I die Anzahl an Gesamtiterationen, welche durch das Stoppkriterium der vorgestellten Methode festgelegt ist. Diese Anzahl ist deutlich kleiner als die Anzahl der zu klassifizierenden Merkmalsvektoren, da nur ein begrenztes Anfragebudget zur Verfügung steht. Es gilt somit $I \ll |\mathcal{R}|$. Durch den Clusterbildungsschritt wird die Anzahl der zu behandelnden Elemente sehr stark reduziert (typischerweise um den Faktor $k = 400$). Für die Komplexitätsklassen ist dieser Faktor jedoch zu vernachlässigen. Wir setzen daher:

$$\mathcal{O}(|\mathcal{R}|) = \mathcal{O}\left(\frac{1}{k} N\right) = \mathcal{O}(N) \tag{4.34}$$

❑ Ohne lokale Dichte

Die Komplexität ohne die Verwendung der PAL-Rahmenstruktur ermittelt sich wie folgt:

$$\begin{aligned}\mathbf{f}_{\text{SCHAL}} &\in \mathcal{O}(\mathbf{f}_{\text{Segmentierung}}) + \mathcal{O}(\mathbf{f}_{\text{Clustering}}) + I \cdot \mathcal{O}(\mathbf{f}_{\text{Iteration}}) \\ &= \mathcal{O}(N) + \mathcal{O}(|\mathcal{R}|) + I \cdot \mathcal{O}(\log_2 |\mathcal{R}|) \\ &= \mathcal{O}(N) + \mathcal{O}(N) + I \cdot \mathcal{O}(\log_2 N) \\ &= 2 \cdot \mathcal{O}(N) + \mathcal{O}(\log_2 N) \\ &= \mathcal{O}(N) \end{aligned} \tag{4.35}$$

Daher liegt die SCHAL-Methode insgesamt in der linearen Komplexitätsklasse bezüglich der Gesamtanzahl an Merkmalsvektoren beziehungsweise Pixeln. Die Vorteile der Verwendung eines hierarchischen Clusterverfahrens sind jedoch, dass die Iterationen des aktiven Lernens nur logarithmische Komplexität besitzen. Daher sind die Interaktionen mit dem Orakel deutlich effizienter als die Vorberechnungen.

❑ Mit lokaler Dichte

Wird die PAL-Rahmenstruktur verwendet, um die lokale Dichte zu berücksichtigen, ergibt sich folgende Komplexität:

$$\begin{aligned}\mathbf{f}_{\text{SCHPAL}} &\in \mathcal{O}(\mathbf{f}_{\text{Segmentierung}}) + \mathcal{O}(\mathbf{f}_{\text{Clustering}}) + \mathcal{O}(\mathbf{f}_{\text{PAL-Vorberechnung}}) + I \cdot \mathcal{O}(\mathbf{f}_{\text{PAL-Iteration}}) \\ &= \mathcal{O}(N) + \mathcal{O}(|\mathcal{R}|) + \mathcal{O}(|\mathcal{R}|^2) + I \cdot \mathcal{O}(\log_2 |\mathcal{R}|) \\ &= \mathcal{O}(N) + \mathcal{O}(N) + \mathcal{O}(N^2) + I \cdot \mathcal{O}(\log_2 N) \\ &= 2 \cdot \mathcal{O}(N) + \mathcal{O}(N^2) + \mathcal{O}(\log_2 N) \\ &= \mathcal{O}(N^2) \end{aligned} \tag{4.36}$$

Der Aufwand wird von der Vorberechnung der Ähnlichkeitsmatrix dominiert. Daher liegt die gesamte Methode in der quadratischen Komplexitätsklasse. Aufgrund der Größe des Eingabebildes (Millionen Pixel), lässt dich die PAL-Rahmenstruktur nicht direkt auf der Originaldaten

anwenden. Erst die Verwendung der Segmentierung zur Datenreduktion ermöglicht die Anwendung in akzeptabler Bearbeitungszeit. Jedoch ist die Berücksichtigung der lokalen Dichte selbst nach Anwendung der Datenreduktion deutlich aufwendiger als die Variante ohne Berücksichtigung der lokalen Dichte.

5 Experimente und Daten

Dieses Kapitel behandelt die durchgeführten Experimente und die verwendeten Datensätze. Der erste Teil erläutert zunächst den allgemeinen Versuchsaufbau und geht anschließend auf die verwendete Bewertungsmethode ein. Der zweite Teil stellt jedes Experiment einzeln vor. Den Abschluss bildet der dritte Teil, welcher die verwendeten Datensätze und ihre Besonderheiten vorstellt.

5.1 Versuchsaufbau

Ziel der Experimente ist es, den Einfluss und die Auswirkungen der verschiedenen Parameter der vorgestellten Methode zu bestimmen. Hierfür eignet sich der Einzelfaktor-Versuchsaufbau [Montgomery, 2013]. In jedem Experiment werden alle Parameter bis auf einen konstant gehalten. Der einzige variable Parameter nimmt verschiedene Werte an, die Methode wird ausgeführt und die Auswirkungen auf das Ergebnis untersucht.

Dieser Versuchsaufbau kann nur dann zur Optimierung aller Parameter verwendet werden, wenn diese unabhängig voneinander sind. Da dies hier nicht der Fall ist, können Änderungen an einem Parameter zu Verschiebungen des Optimums für andere Parameter führen. Ein Beispiel für solch einen Zusammenhang sind der Segmentierungsparameter k und der Clusterparameter B. Ersterer bestimmt die Anzahl der zu erstellenden Repräsentantenvektoren und letzterer die Anzahl der auf ihnen durchzuführenden Zweiteilungen. Jede Wahl von $B > log_2 k$ hat keine Auswirkungen, da bereits alle Blätter des Baumes nur noch ein Element enthalten und nicht weiter geteilt werden können.

Solch eine globale Optimierung ist jedoch nicht Ziel dieser Arbeit. Das globale Optimum ist ohnehin abhängig von den Eigenschaften des gewählten Datensatzes, wie zum Beispiel der Bodenauflösung und den Arten der zu erkennenden Klassen. Die folgenden Experimente werden daher genutzt, um den Einfluss jedes Parameters einzeln zu untersuchen und so die vorgestellte Methode besser beurteilen zu können.

Die Implementierung der Methode erfolgte in MATLAB. Die Implementation des SLIC-Algorithmus [Achanta et al., 2012] wurde entsprechend der Beschreibung in Abschnitt 4.2.2 angepasst. Der Quellcode für das *Active queries*-Verfahren [Tuia et al., 2012] stand als Referenz zur Verfügung, wurde jedoch vollständig neu implementiert und erweitert (siehe Abschnitt 4.3.3). Aus der mcPAL-Rahmenstruktur [Kottke et al., 2016] wurde lediglich die Funktion zur Berechnung des probabilistischen Nutzens verwendet. Die Integration in die vorgestellte Methode umfasste das Aufbereiten der Daten und die Vorberechnung der Ähnlichkeitsmatrix. Die unveränderten Originalversionen wurden ausschließlich für den Vergleich in Experiment 5.2.8 verwendet. Die eingesetzte Bewertungsmethode nutzt die in MATLAB vorhandenen Statistikfunktionen und wird im Folgenden genauer beschrieben.

5.1.1 Bewertungsmethode

Wie in der Einleitung erwähnt, ist das Hauptziel dieser Arbeit die Effizienzsteigerung unter Nutzung vorhandener Ressourcen und nicht die Verbesserung der maximalen Klassifikationsgenauigkeit. Das heißt, die Klassifikationsqualität allein ist nicht ausreichend. Es muss auch das verwendete Anfragebudget beachtet werden. Ein sehr gutes Maß hierfür ist die Lernkurve. Sie ist eine Abbildung, die einem Budgetwert einen Qualitätswert zuordnet. Es ist eine der am weitesten verbreiteten Evaluierungsmethoden für aktives Lernen [Tuia et al., 2011; Settles, 2012].

Zur Bestimmung der Klassifikationsgenauigkeit eines maschinellen Lernverfahrens wird üblicherweise die vorhandene *Ground Truth* in eine Trainings- und eine Testmenge aufgeteilt. Letztere wird während des Trainings nicht verwendet, sondern dient ausschließlich der Qualitätsbestimmung. Dieses Vorgehen ist bei der SCHAL-Methode nicht möglich, da für das Erstellen der Clusterhierarchie alle zu klassifizierenden Daten bereits während des Trainings bekannt sein müssen. Es kann somit keine Testmenge zurückgehalten werden. Daher wird hier der gleiche Ansatz wie in den Arbeiten von Dasgupta & Hsu [2008], Tuia et al. [2012] und Muñoz-Marí et al. [2012] verwendet: Alle Daten stehen ohne Klasseninformationen zur Verfügung, die *Ground Truth* wird zurückbehalten bis sie beim Orakel angefragt wird und am Ende wird eine Gesamtklassifikationsgenauigkeit bestimmt. Diese Vorgehensweise passt zu dem in der Einleitung vorgestellten Einsatzszenario, da mit dem aufgenommenen Luftbild bereits alle zu klassifizierenden Daten zum Trainingszeitpunkt vorliegen.

Die Auswertung der aktuellen Klassifikationsgenauigkeit kann zu verschiedenen Zeitpunkten stattfinden. In herkömmlichen, passiven Lernverfahren geschieht dies nach Abschluss des Lernprozesses. In aktiven Verfahren geschieht dies bereits zu früheren Zeitpunkten, beispielsweise nachdem 33%, 66% und 100% des Budgets verwendet wurden [Kottke et al., 2016]. Da diese Arbeit und die durchgeführten Experimente unter Laborbedingungen (ausreichend Rechenleistung und Zeit) entwickelt wurden, kann die Auswertung wesentlich öfter stattfinden. Die so erzeugte Lernkurve besitzt deutlich mehr Messpunkte und ist somit genauer und erlaubt detailliertere Einsichten im Vergleich zu Versuchsaufbauten mit weniger Evaluierungszeitpunkten. Welche Arten von Budget und Qualität verwendet wurden, wird in den nächsten Abschnitten erläutert.

5.1.2 Budget

Wie in der Einleitung beschrieben, verursacht das Beschaffen von *Ground Truth* verschiedene Arten von Kosten. Typischerweise treten diese Kosten in Form von Annotationszeit auf. Diese Kosten können variabel in Abhängigkeit von der angefragten Stichprobe oder vom gelieferten Klassenlabel sein. Bei den in dieser Arbeit verwendeten Datensätzen war die *Ground Truth* entweder bereits gegeben oder wurde manuell durch Bildinterpretation erstellt. Daher wird davon ausgegangen, dass alle Anfragen Kosten in der selben Höhe verursachen. Dies bedeutet, dass das Budget direkt als Anzahl der vom Orakel angefragten Stichproben gemessen wird. Das Maximalbudget wurde auf 1.000 Anfragen festgelegt, um das im Praxiseinsatz begrenzte Budget zu simulieren. Die Evaluierung der Qualität fand nach jeder zwanzigsten Anfrage statt. Daraus ergibt sich, dass die Lernkurve durch 50 Stützstellen definiert ist.

5.1.3 Qualität

Zur Bewertung der Klassifikationsgenauigkeit von Multiklassenproblemen, wie sie in dieser Arbeit behandelt werden, wird typischerweise die Konfusionsmatrix verwendet. Sie erlaubt gute Einblicke in die Eigenschaften der untersuchten Methode, da von Verwechslungen bestimmter Klassen Rückschlüsse auf die Funktionsweise möglich sind. Als Qualitätswert für die Lernkurve wird jedoch ein einzelner Wert benötigt. In dieser Arbeit wird hierfür die Gesamtgenauigkeit verwendet. Diese entspricht der Summe der Diagonalelemente der Konfusionsmatrix dividiert durch

die Gesamtanzahl an Stichproben. Anders ausgedrückt, ist es der Anteil an korrekt klassifizierten Stichproben.

Die Aufgabenstellung erfordert ein Klassenlabel für jedes Pixel des Eingabebildes. Da der letzte Schritt der vorgestellten Methode ein Label pro Repräsentantenvektor liefert, müssen diese noch auf die durch sie zusammengefassten Merkmalsvektoren verteilt werden. Dies geschieht, indem jeder Merkmalsvektor eines Segmentes das Label des zugehörigen Repräsentantenvektors erhält.

Die angewendeten Teilschritte der Segmentierung (durch den SLIC-Algorithmus), der Clusterbildung (durch *bisecting* k-Means) und der Knoten-, sowie Blattauswahl sind nicht deterministisch. Daher wird jedes Experiment zehn mal wiederholt und der Mittelwert der errechneten Gesamtgenauigkeit sowie deren Standardabweichung berichtet.

5.1.4 Statistischer Test

Jede Parameterwahl führt zu einer Lernkurve. Daher entstehen in jedem Experiment mehrere Lernkurven. Um zu entscheiden, ob eine Parameterkombination besser ist als eine andere, muss untersucht werden, ob sich die Ergebnisse signifikant unterscheiden. Hierzu müssen die Kurven miteinander verglichen werden. Dies geschieht, indem die Fläche unter der Lernkurve berechnet wird – englisch *area under the curve* (AUC). Der AUC-Wert ist ein Maß für die Güte des gesamten Lernprozesses. Der Wert unterscheidet sich von der vorher beschriebenen Klassifikationsgenauigkeit, da diese ein Maß für lediglich einen Augenblick des Lernvorganges ist.

Durch den AUC-Wert kann eine Parameterkombination nach Ausführung des Trainingsvorgangs als ein einzelner skalarer Wert dargestellt werden. Zwei zu vergleichende Varianten ergeben ein Wertepaar und nach zehnmaliger Wiederholung eine Liste aus zehn gepaarten Werten. Es soll nun eine Aussage getroffen werden, ob sich die beiden Varianten signifikant unterscheiden.

Hierfür bietet sich der Wilcoxon-Vorzeichen-Rang-Test an [Wilcoxon, 1945]. Dieser ist ein parameterfreier statistischer Test, welcher zwei gepaarte Stichproben auf Gleichheit testet. Im Vergleich zu dem häufig eingesetzten Zweistichproben-t-Test (auch bekannt als *Student's t-test* [Student, 1908]) wird jedoch nicht vorausgesetzt, dass den zu vergleichenden Werten eine Normalverteilung zugrunde liegt.

Die Null-Hypothese des Tests besagt, dass beide Messreihen der gleichen Verteilung entstammen und alle beobachteten Differenzen zufälliger Natur sind. Berechnet wird nun die Wahrscheinlichkeit p, dass die Nullhypothese zutrifft. Die Nullhypothese kann abgelehnt werden, wenn $p < 0,05$ gilt. In diesem Fall kann davon ausgegangen werden, dass die Messreihen unterschiedlichen Verteilungen entstammen. Im Fall der durchgeführten Experimente bedeutet dies, dass die beiden untersuchten Parameterkombinationen zu signifikant unterschiedlichen Ergebnissen führen.

5.2 Experimente

Dieser Abschnitt erläutert das Ziel jedes Experiments. Es wird eine Hypothese zu den Auswirkungen des zu untersuchenden Parameters erstellt und beschrieben, wie diese getestet wird.

5.2.1 Einfluss der Segmentierung

Ziel dieses Experiments ist es, zu überprüfen, ob durch den Einsatz von Segmentierung die Lernrate verbessert werden kann. Dies sollte der Fall sein, da durch das Aussortieren von Ausreißern die Effizienz des Lernverfahrens gesteigert wird.

❑ Hypothese

Die Hypothese ist, dass der Segmentierungsschritt das Rauschen in den Daten reduziert, dadurch das Problem vereinfacht wird und sich die Lernrate verbessert. Ist dies der Fall, sollte die Lernkurve höhere Werte früher erreichen als ohne den Segmentierungsschritt. Mit anderen Worten: Erreicht die Variante mit Segmentierungsschritt trotz weniger Anfragen eine höhere Klassifikationsgenauigkeit, hat die Segmentierung einen positiven Einfluss.

❑ Test

Hierzu wird das Lernverfahren in zwei Varianten ausgeführt. In der ersten Variante wird das Lernverfahren wie in Kapitel 4 beschrieben ausgeführt. In der zweiten Variante wird auf die Segmentierung verzichtet und die darauffolgenden Schritte statt auf Repräsentantenvektoren direkt auf den unveränderten Merkmalsvektoren ausgeführt.

Zusätzlich nutzt dieses Experiment die Besonderheit des Vaihingen-Datensatzes, dass Mischpixel in der *Ground Truth* markiert sind (siehe Abschnitt 5.3.3). Hierzu werden die beiden Varianten einmal mit allen Daten ausgeführt und einmal ohne die Mischpixel. Mischpixel zählen als Ausreißer, da sie nicht komplett zu einer Klasse gehören. Es sind somit vier Lernkurven zu vergleichen.

5.2.2 Einfluss des Segmentierungsparameters k

Das vorherige Experiment untersucht den Einfluss der Segmentierung an sich. Nun sollen die Auswirkungen des zugehörigen Parameters k untersucht werden.

❑ Hypothese

Da, wie in Abschnitt 4.3.4 beschrieben, eine Untersegmentierung nicht korrigiert werden kann, ist die Erwartung, dass kleine Werte von k zu schlechterem Lernverhalten führen und große Werte zu besserem.

❑ Test

Die Hypothese wird getestet, indem verschiedene Werte von k untersucht werden. Hierzu wird der während der Entwicklung verwendete Wert von $k = 10.000$ als Basis gewählt. Zum Vergleich wird dieser Wert verdoppelt und halbiert, so dass sich drei zu vergleichende Werte ergeben.

5.2.3 Einfluss des Clusterparameters B

Ziel dieses Experiments ist es, den Einfluss des Clusterparameters B zu untersuchen. Dieser steuert die Anzahl der Zweiteilungen (englisch *bisection*) im Clusterbildungsschritt der Methode.

❑ Hypothese

Kleine Werte für B sollten zu geringerer Klassifikationsgenauigkeit führen, da die Repräsentantenvektoren nicht ausreichend getrennt wurden, so dass Labeling-Fehler unumgänglich sind. Mittlere Werte von B sollten zu Verbesserungen führen. Es gibt jedoch eine Maximalanzahl möglicher Zweiteilungen. Wird B größer gewählt, sollten keine Veränderungen mehr eintreten.

❑ Test

Während der Entwicklung der Methode hat sich gezeigt, dass ein Wert von $B = 5.000$ gute Ergebnisse liefert. Für dieses Experiment werden daher Vergleichswerte gewählt, die deutlich kleiner und deutlich größer sind: $B = \{2.000, 5.000, 8.000\}$.

5.2.4 Einfluss des aktiven Lernens

Dieses Experiment soll den Unterschied zwischen aktivem und passivem Lernen feststellen.

❑ Hypothese

Aktives Lernen verbessert die Effizienz des Lernverfahrens. Diese Verbesserung zeigt sich in einer Lernkurve, die mit geringerem Budget gleiche Qualität erreicht oder mit gleichem Budget höhere Qualität.

❑ Test

Um die Hypothese zu testen, wird im dritten Schritt der Methode das aktive Lernen durch passives Lernen ersetzt. Hierzu wird anstatt den Repräsentantenvektor mit dem größten Mehrwert auszuwählen, ein zufälliger Repräsentantenvektor beim Orakel angefragt.

5.2.5 Einfluss des lokalen Dichte-Parameters σ

Ziel dieses Experimentes ist es, den Einfluss des Skalierungsparameters σ zu untersuchen. Dieser Parameter steuert die Stärke des entfernungsabhängigen Abfalls der Ähnlichkeitsfunktion. Somit steuert er direkt den Einfluss, den die lokale Dichte auf die Auswahl des nächsten Repräsentantenvektors für die Anfrage beim Orakel hat.

❑ Hypothese

Die Hypothese ist, dass es einen optimalen Bereich für die Werte von σ gibt. Zu kleine Werte führen dazu, dass sich Label-Informationen nur auf einen sehr kleinen Bereich auswirken. Es wird erwartet, dass dies zu langsamerem Lernen führt. Zu große Werte von σ führen dazu, dass der Einflussbereich jeder Stichprobe sehr groß ist. Dies führt dazu, dass sich fast alle Stichproben gegenseitig beeinflussen und das Lernergebnis sich immer weiter dem passiven Lernen annähert.

❑ Test

Die Hypothese wird getestet, indem die Lernkurven von Durchläufen mit verschiedenen Werten von σ miteinander verglichen werden. Außerdem werden die daraus resultierenden Ähnlichkeiten der Repräsentantenvektoren miteinander verglichen.

5.2.6 Einfluss der lokalen Dichte

Das vorherige Experiment untersuchte den Einfluss des Skalierungsparameters für die lokale Dichte. Dieses Experiment soll untersuchen, ob die Integration der lokalen Dichte Vorteile gegenüber der Variante ohne lokale Dichte bringt.

❑ Hypothese

Die Integration der lokalen Dichte in den aktiven Lernprozess der vorgestellten Methode steigert die Effizienz des Lernens.

❑ Test

Um die Hypothese zu testen, werden drei Varianten der SCHAL-Methode miteinander verglichen: passiv, aktiv und aktiv mit lokaler Dichte. Es werden die Lernkurven erstellt und untersucht.

5.2.7 Vergleich auf verschiedenen Datensätzen

Dieses Experiment untersucht, wie sich die vorgestellte Methode in der in Kapitel 4 beschriebenen Variante auf verschiedenen Datensätzen verhält.

❑ Hypothese

Die Erwartung ist, dass die Methode unabhängig vom Datensatz effizienteres Lernverhalten zeigt als passives Lernen.

❑ Test

Um die Hypothese zu testen, wird die Methode mit konstanten Parameterwerten auf drei verschiedenen Datensätzen ausgeführt. Zum Vergleich wird die passive Variante mit ebenfalls konstanten Parameterwerten auf den gleichen drei Datensätzen ausgeführt. Die Hypothese, dass die vorgestellte Methode effizienter als die passive Variante trainiert werden kann, wird anhand der resultierenden Lernkurven beurteilt.

5.2.8 Vergleich verschiedener Methoden

In diesem Experiment wird die vorgestellte Methode in der Variante mit und ohne Betrachtung der lokalen Dichte mit verschiedenen Methoden auf dem Stand der Wissenschaft verglichen. Diese sind *active queries* [Tuia et al., 2012], *segmented active queries* [Wuttke et al., 2017] und *multi-class PAL* [Kottke et al., 2016]. Als Basis wird passives Lernen (*random sampling*) herangezogen.

❑ Hypothese

Die Erwartung ist, dass alle aktiven Methoden besser als passives Lernen sind. Des Weiteren wird erwartet, dass die SCHAL- und SCHPAL-Methoden effizienteres Lernverhalten als die anderen Methoden zeigen.

❑ Test

Um die Hypothese zu testen, werden alle Methoden auf dem selben Datensatz ausgeführt. Die SCHAL-Methode wird mit den in den vorangegangenen Experimenten als am besten beurteilten Parametern ausgeführt. Die anderen Methoden werden mit den vorgegeben Standardwerten oder falls nicht vorhanden nach bestem Gewissen optimierten Parametern ausgeführt.

Experiment	k	B	Aktiv	σ	Datensatz
1	$\{\emptyset, 10.000\}$	$\{5.000\}$	{ja}	$\{\emptyset\}$	{Vaihingen}
2	$\{5, 10, 15\} \cdot 10^3$	$\{5.000\}$	{ja}	$\{\emptyset\}$	{Vaihingen}
3	$\{10.000\}$	$\{2, 5, 8\} \cdot 10^3$	{ja}	$\{\emptyset\}$	{Vaihingen}
4	$\{10.000\}$	$\{5.000\}$	{ja, nein}	$\{\emptyset\}$	{Vaihingen}
5	$\{10.000\}$	$\{5.000\}$	{ja}	$\{1, 5, 10\} \cdot 10^{-4}$	{Vaihingen}
6	$\{10.000\}$	$\{5.000\}$	{ja, nein}	$\{5 \cdot 10^{-4}\}$	{Vaihingen}
7	$\{10.000\}$	$\{5.000\}$	{ja, nein}	$\{\emptyset\}$	{Abenberg, Potsdam, Vaihingen}
8	$\{10.000\}$	$\{5.000\}$	{ja}	$\{\emptyset, 5 \cdot 10^{-4}\}$	{Vaihingen}

Tabelle 5.1: Übersicht zu allen Experimenten und den in ihnen verwendeten Parametern. Der Wert $\emptyset$ zeigt an, dass in dieser Variante der entsprechende Teil der Methode (Segmentierung beziehungsweise lokale Dichte) ausgelassen wurde.

5.2.9 Zusammenfassung

Tabelle 5.1 fasst alle acht Experimente zusammen und stellt die jeweiligen Parameterwerte übersichtlich dar.

5.3 Datensätze

Die beschriebenen Experimente wurden auf drei verschiedenen Datensätzen ausgeführt. Alle drei sind luftgestützt aufgenommene optische Fernerkundungsdaten. Bei den aufgenommenen Gebieten handelt es sich um urbane Regionen in Deutschland. Im Folgenden wird jeder Datensatz eingehender beschrieben.

5.3.1 Abenberg

Dieser Datensatz der TU München wurde vom Landesvermessungsamt Bayern aufgenommen. Es ist ein Luftbild der Kleinstadt Abenberg im mittelfränkischen Landkreis Roth. Die Bodenpixelgröße beträgt 20 cm. Für jedes Pixel liegen Intensitätsinformationen für vier Kanäle vor: Infrarot, rot, grün und blau. *Ground Truth* war nicht vorhanden und wurde deshalb selbst erstellt. Hierfür wurde ein 1.000 x 1.000 Pixel großer Bereich ausgewählt in dem Gebäude, Straßen, Wald und offene Erdböden vorhanden sind. Es wurden Label für sieben Klassen erstellt: Gebäude, Baum, niedrige Vegetation, Erde, Schotter, Asphalt, sowie eine Hintergrundklasse für alle nicht zuzuordnenden Pixel. Abbildung 5.1 zeigt das Luftbild und die erstellte *Ground Truth* Karte.

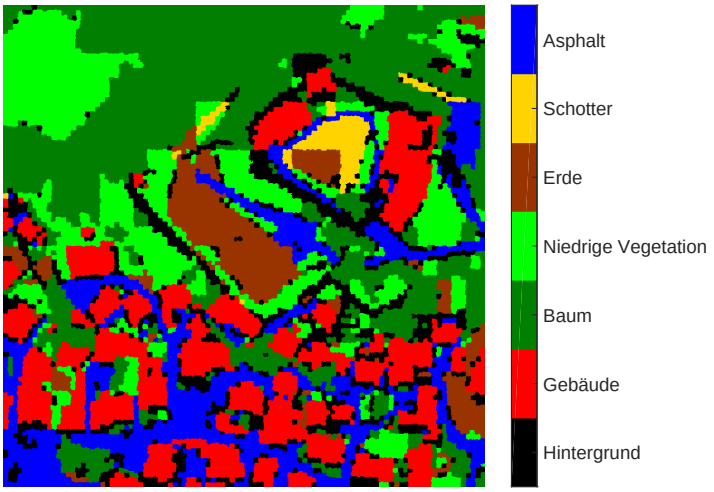
5.3.2 Potsdam

Dieser Datensatz wurde durch die *International Society for Photogrammetry and Remote Sensing* (ISPRS) öffentlich zur Verfügung gestellt* und im Rahmen des „ISPRS 2D Semantic Labeling"-Wettbewerbs aufbereitet. Er zeigt eine typische historische deutsche Stadt mit schmalen Straßen,

*Download unter: `http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html` (letzter Zugriff 10.9.2018)



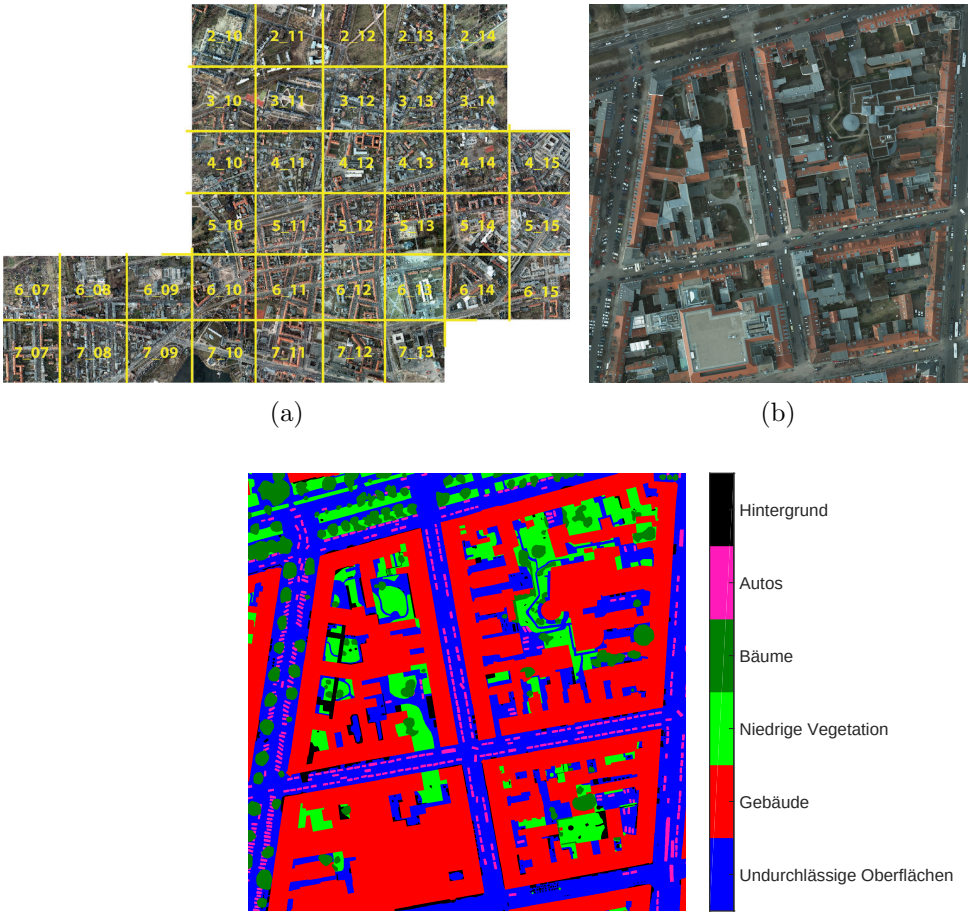
(a) (b)



(c)

Abbildung 5.1: Teilbild (a) zeigt ein Luftbild des gesamten Abenberg-Datensatzes mit Markierung des vergrößerten Ausschnitts (1.000 x 1.000 Pixel), welcher in (b) abgebildet ist. Die Erstellte *Ground Truth* des selben Ausschnitts ist in Teilbild (c) gezeigt. Die Klasse Hintergrund wurde für Pixel vergeben, die nicht eindeutig zugeordnet werden konnten.

großen Gebäudeblöcken und dichter Siedlungsstruktur. Der Datensatz besteht aus sehr hoch aufgelösten *true ortho photos* (TOPs) und einem korrespondierendem digitalen Oberflächenmodell (englisch *digital surface model*, DSM). Er ist in 38 Kacheln unterteilt von denen jede 6.000 x 6.000 Pixel groß ist. *Ground Truth* ist für 24 Kacheln verfügbar. Die verwendeten Klassenlabel sind: Autos, Bäume, niedrige Vegetation, Gebäude, undurchlässige Oberflächen und Hintergrund. Die beiden Klassen Autos und Bäume bezeichnen keine Landbedeckungen sondern Landnutzungstypen. Die Originalbezeichnungen werden jedoch übernommen, um eine bessere Vergleichbarkeit zwischen den Datensätzen herzustellen. Abbildung 5.2 zeigt die Übersicht, eine Detailansicht und die *Ground Truth* für eine Kachel. Für jedes Pixel liegen Intensitätsinformationen für vier Kanäle vor (infrarot, rot, grün und blau) sowie den Höheninformationen des DSM. Diese Arbeit verwendet die spektralen Informationen, jedoch nicht die Höheninformationen.



(a) (b)

(c)

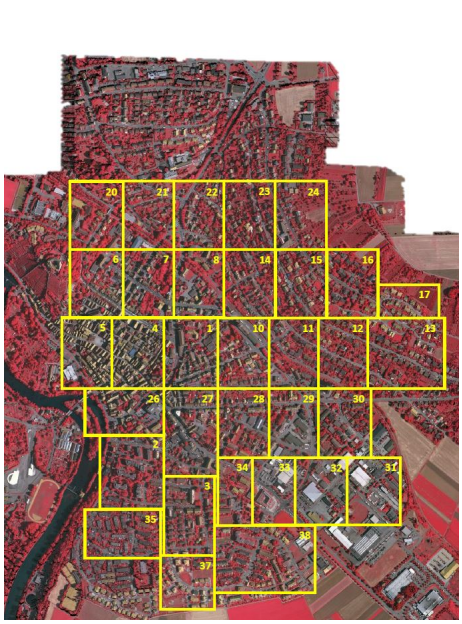
Abbildung 5.2: Teilbild (a) zeigt eine Übersicht über den gesamten Potsdam-Datensatz und die Einteilung der Kacheln. Die in dieser Arbeit verwendete Kachel (5_12) ist in (b) abgebildet und die für sie vorhandene *Ground Truth* ist in (c) dargestellt.

5.3.3 Vaihingen

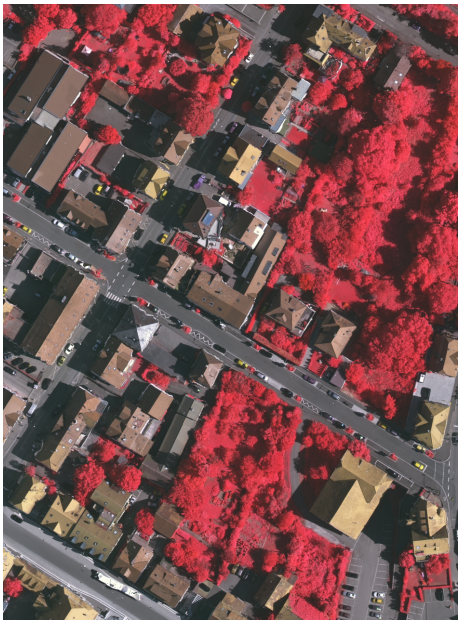
Der Vaihingen-Datensatz wurde für den „ISPRS Benchmark Test für urbane Objektdetektion und -rekonstruktion“ aufbereitet. Er enthält mehrere Luftaufnahmen der Baden-Württembergischen Stadt Vaihingen an der Enz und ist öffentlich verfügbar[†]. Er besteht aus 33 Kacheln von denen jede in etwa 2.000 x 2.000 Pixel groß ist und Intensitätsinformationen für drei Kanäle enthält: Infrarot, rot und grün. Die zur Verfügung stehenden Höheninformationen einer *light detection and ranging* (LiDAR)-Befliegung wurden in der vorliegenden Arbeit jedoch nicht verwendet. Die *Ground Truth* stellt die selben sechs Klassenlabel wie der Potsdam-Datensatz zur Verfügung: Autos, Bäume, niedrige Vegetation, Gebäude, undurchlässige Oberflächen und Hintergrund.

Eine Besonderheit des Vaihingen-Datensatzes ist die Markierung von Mischpixeln. Hierzu wurde um alle Klassensegmente ein drei Pixel breiter Rand gezogen. Mischpixel, enthalten mehr als ein Material, so dass deren Spektren keine reinen Spektren sind. Diese Information wird in Experiment 1 zur Untersuchung des Einflusses der Segmentierung verwendet.

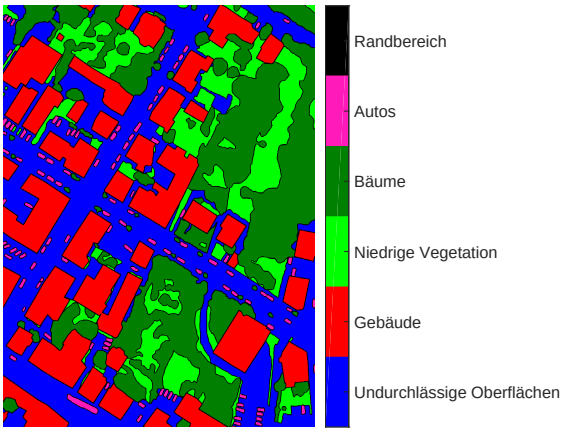
[†]Zur Verfügung gestellt von der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V. (DGPF) [Cramer, 2010]: `www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html`. Download unter: `http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html` (letzter Zugriff 10.9.2018)



(a)



(b)



(c)

Abbildung 5.3: Teilbild (a) zeigt eine Falschfarbenansicht (IR-R-G-Kanäle) des gesamten Vaihingen-Datensatzes und die Einteilung der Kacheln. Teilbild (b) zeigt die in dieser Arbeit verwendete Kachel (7). Die zu dieser Kachel passende *Ground Truth* ist in (c) dargestellt.

5.3.4 Zusammenfassung

Die wichtigsten Eigenschaften wie zum Beispiel die Anzahl der aufgenommenen Kanäle oder der *ground sampling distance* (GSD) sind in Tabelle 5.2 zusammengefasst. Die letzte Spalte zeigt die Histogramme der Klassenverteilung. Die separiert dargestellte Klasse ist die Rückweisungs- oder Hintergrundklasse.

Tabelle 5.2: Charakteristiken der drei verwendeten Datensätze.

Datensatz	# Kanäle	GSD [cm]	# Pixel	# Klassen
Abenberg	4	20	1 [10^6]	6+1 
Potsdam	4	5	36 [10^6]	5+1 
Vaihingen	3	9	4 [10^6]	5+1 

6 Ergebnisse und Diskussion

6.1 Einfluss der Segmentierung

Dieses Experiment untersuchte den Einfluss des Segmentierungsschrittes. In der Variante ohne Segmentierung wurde das Clusterverfahren direkt auf den Merkmalsvektoren des Ursprungsbildes ausgeführt. Durch die fehlende Datenreduktion erhöht sich die zu clusternde Datenmenge im Beispiel des Vaihingen-Datensatzes um einen Faktor von 400.

❑ Ergebnis

Die Auswirkungen auf die Klassifikationsgenauigkeit sind in Abbildung 6.1 dargestellt. Der visuelle Eindruck, dass sich die Leistung der Varianten signifikant unterscheidet, wurde durch den Wilcoxon-Vorzeichen-Rang-Test bestätigt ($p < 0,002$). Das Verwenden des Segmentierungsschrittes steigert die Maximale Klassifikationsgenauigkeit um 18 Prozentpunkte (von 57% auf 75%).

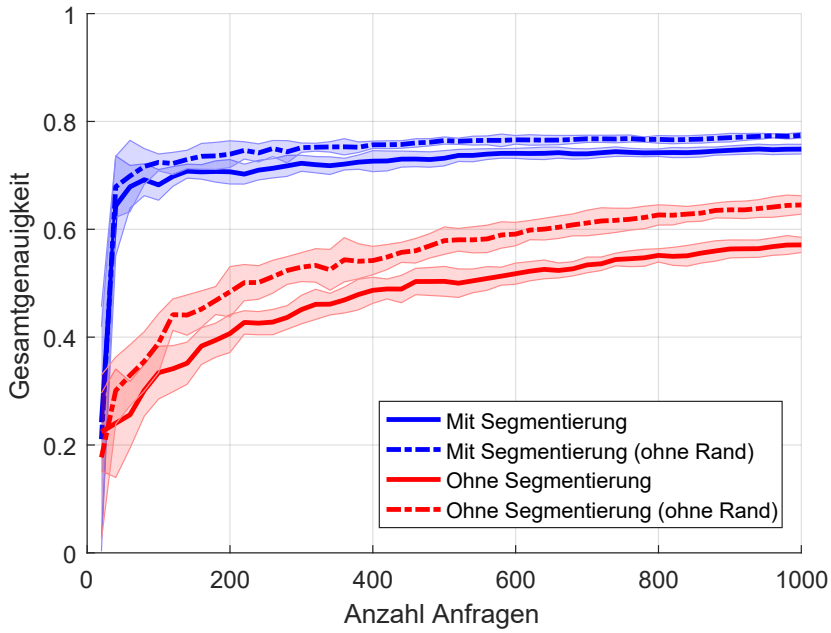


Abbildung 6.1: Ergebnisse des Experiments zur Untersuchung des Einflusses des Segmentierungsschrittes auf dem Vaihingen-Datensatz (Abbildung nach [Wuttke et al., 2018]). Die dargestellten Linien sind die Mittelwerte und die Schattierungen die Standardabweichungen der zehn Wiederholungen. Für die blauen Kurven wurde der Segmentierungsschritt, wie in Abschnitt 4.2 beschrieben, durchgeführt. Für die roten Kurven wurde er übersprungen. Die gestrichelten Linien entstanden durch weglassen der Randpixel vor dem Anwenden des Clusterverfahrens.

Alternativ kann die gleiche Klassifikationsgenauigkeit durch Einsatz des Segmentierungsschrittes mit 90% weniger Anfragen erreicht werden.

Die durchgezogenen Kurven wurden auf dem vollständigen Datensatz erstellt, während die gestrichelten auf der Variante ohne Randpixel erstellt wurden.

❑ Diskussion

Der Einsatz des Segmentierungsschrittes verbessert die Leistung drastisch. Die Leistung sinkt jedoch, wenn auch Mischpixel klassifiziert werden sollen. Bei der Variante ohne Segmentierung verschlechtert sich die Gesamtgenauigkeit um 7,4 Prozentpunkte. Wird hingegen die Segmentierung verwendet tritt nur eine Verschlechterung um um 2,6 Prozentpunkte auf. Die Schlussfolgerung ist, dass durch Verwendung des Segmentierungsschrittes der Umgang mit Mischpixeln verbessert wird. Ein weiteres Indiz hierfür ist die geringere Standardabweichung, wenn die Segmentierung verwendet wird. Da durch die Segmentierung das Rauschen in den Daten herausgemittelt wird, können die im zweiten Schritt erstellten Cluster besser getrennt und somit genauer klassifiziert werden.

Das Fazit ist, dass das Verwenden des Segmentierungsverfahrens einen sehr positiven Einfluss auf die Klassifikationsqualität hat. Der Segmentierungsschritt stellt daher einen sehr wertvollen Beitrag dar.

6.2 Einfluss des Segmentierungsparameters *k*

Dieses Experiment untersuchte, welchen Einfluss die Anzahl der zu erstellenden Segmente im Segmentierungsschritt auf die Lernrate hat.

❑ Ergebnis

Die Ergebnisse in Form der Lernkurven sind in Abbildung 6.2 dargestellt. Es ist eine sehr geringe Standardabweichung für alle drei untersuchten Parameterwerte zu beobachten. Die Lernkurven unterscheiden sich weniger stark als in Experiment 1. Der statistische Test ergab, dass sich die Lernkurven für die Werte 5.000 vs. 15.000 und 10.000 vs. 15.000 signifikant unterscheiden ($p < 0,002$). Der Unterschied zwischen 5.000 und 10.000 ist jedoch nicht groß genug für statistische Signifikanz ($p = 0,13$).

❑ Diskussion

Ein Wert von $k = 5.000$ oder $k = 10.000$ führt bei diesem Datensatz zu geringer beziehungsweise moderater Übersegmentierung. Eine Veranschaulichung des Einflusses von großen und kleinen Werten für k ist auch in Abbildung 4.4 dargestellt. Dies wirkt sich jedoch nicht signifikant auf die Leistung aus. Wird allerdings zu stark übersegmentiert ($k = 15.000$), führt dies zu verminderter Klassifikationsgenauigkeit, da die zu erstellende Hierarchie komplexer ist und somit mehr Anfragen benötigt, um verlässliche Ergebnisse zu erreichen.

Das Fazit ist, das eine moderate Übersegmentierung keine Nachteile für den Klassifikationsvorgang bringt.

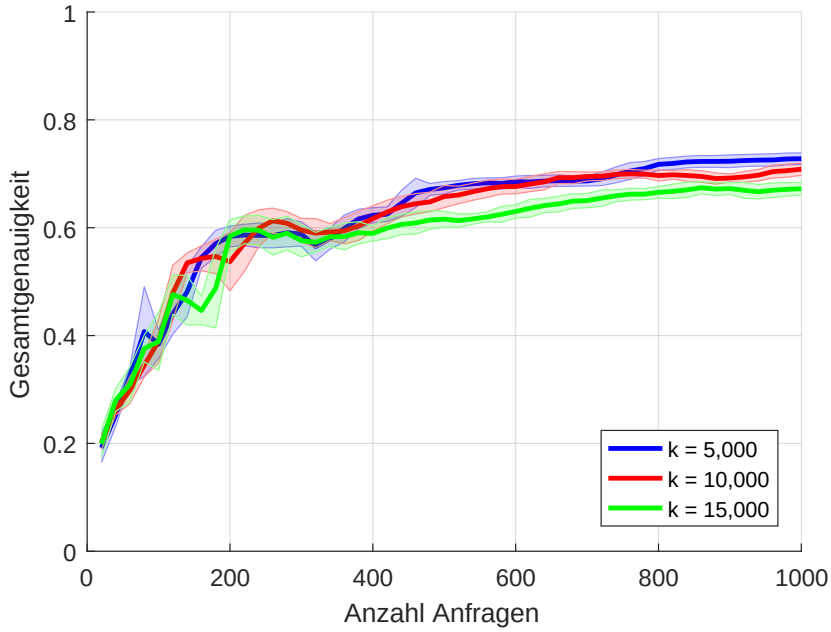


Abbildung 6.2: Vergleich der Lernkurven für verschiedene Werte des Segmentierungsparameters (Abbildung nach [Wuttke et al., 2018]). Die dargestellten Kurven sind die Mittelwerte von zehn Wiederholungen auf dem Vaihingen-Datensatz. Die schattierten Flächen entsprechen den dazugehörigen Standardabweichungen. Kleine Werte führen zu Untersegmentierung und große Werte zu Übersegmentierung. Trotz stark unterschiedlicher Parameterwerte, sind die Auswirkungen gering. Zu starke Übersegmentierung führt jedoch zu schlechteren Leistungen.

6.3 Einfluss des Clusterparameters B

Dieses Experiment untersuchte die Auswirkungen des Clusterparameters B auf die Gesamtgenauigkeit. Der Parameter gibt die maximale Anzahl an durchzuführenden Zweiteilungen während des Clusterbildungsschrittes an.

❑ Ergebnis

Die Lernraten der verschiedenen Parameterwerte sind sehr ähnlich, siehe Abbildung 6.3. Der kleinste Parameterwert ($B = 2.000$) zeigte die schlechteste Leistung. Dies ist auch der einzige Wert, für den sich ein statistisch signifikanter Unterschied feststellen lässt ($p < 0,002$). Der Unterschied zwischen $B = 5.000$ und $B = 8.000$ ist nicht groß genug, um als statistisch signifikant bezeichnet zu werden ($p = 0,13$).

❑ Diskussion

Die Anzahl der Zweiteilungen B während des Clusterbildungsschrittes bestimmt die Tiefe des Binärbaumes. Wird B, im Vergleich zum Segmentierungsparameter k, zu klein gewählt, führt dies dazu, dass sehr viele Segmente in einem Blatt der Clusterhierarchie verbleiben anstatt weiter unterteilt zu werden. Da jedes Blatt nur genau ein Klassenlabel erhält, führt dies zu einer größeren Anzahl von Fehlklassifikationen und somit geringerer Gesamtgenauigkeit (siehe blaue Kurve für $B = 2.000$).

Da ein Baum mit B Zweiteilungen $B + 1$ Blätter besitzt, enthält für die Wahl $B = k/2$ jedes Blatt durchschnittlich zwei Segmente. Diese sind sich sehr ähnlich, da sie sonst bereits in einer

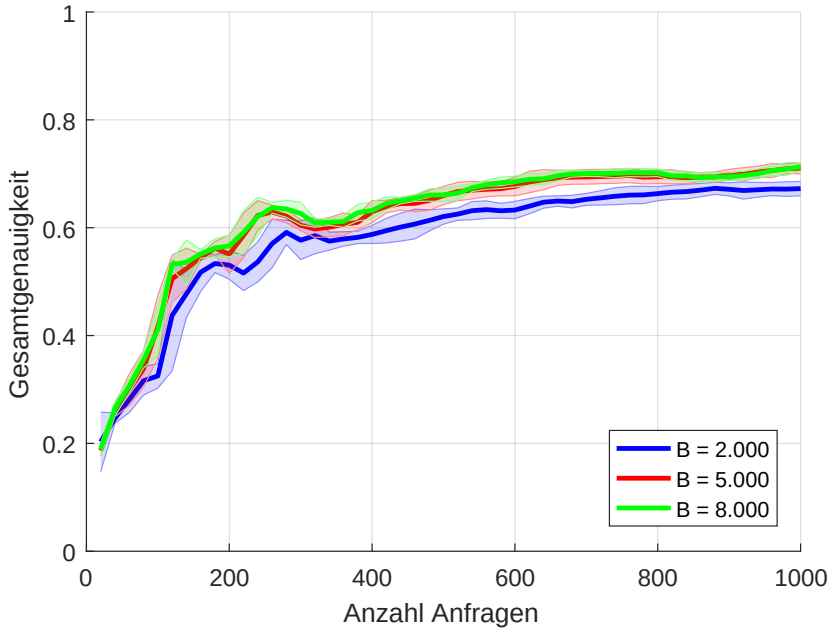


Abbildung 6.3: Die durchgezogenen Linien sind die Mittelwerte und die schattierten Flächen die Standardabweichungen der Gesamtgenauigkeit von zehn Wiederholungen des Experimentes auf dem Datensatz Vaihingen (Abbildung nach [Wuttke et al., 2018]). Der Parameter B hat nur einen geringen Einfluss auf die Lernrate. Wird er jedoch zu klein gewählt (blau), werden die Cluster nicht oft genug aufgeteilt und die maximal erreichbare Genauigkeit sinkt.

früheren Iteration getrennt worden wären. Daher besitzen sie mit hoher Wahrscheinlichkeit das selbe Klassenlabel. Weitere Zweiteilungen bringen demzufolge nur noch sehr wenig bis gar keine Vorteile mehr (siehe rote und grüne Kurven für $B = 5.000$ und $B = 8.000$). Für Parameterwerte $B \geq k$ ändert sich die Hierarchie nicht mehr, da jedes Blatt nur noch genau ein Element enthält und nicht mehr geteilt werden kann. In diesem Fall müssen sehr viele Anfragen gestellt werden, um zu akzeptablen Label-Statistiken (siehe Ungleichung 4.13) zu kommen. Dies führt zu einer verschlechterten Lernrate. Das Fazit ist, dass moderate Werte für B in der Größenordnung $k/2$ zu den besten Lernraten führen.

6.4 Einfluss des aktiven Lernens

Dieses Experiment untersuchte den Einfluss des aktiven Lernens auf die Klassifikationsqualität durch einen Vergleich mit passivem Lernen. In beiden Fällen wurden die ersten zwei Schritte der vorgestellten Methode unverändert angewandt. Lediglich im dritten Schritt wurde die angewendete Selektionsstrategie variiert.

❑ Ergebnis

Das Ergebnis des Vergleichs ist sehr deutlich (siehe Abbildung 6.4). Die maximal erreichte Gesamtgenauigkeit wird durch den Einsatz von aktivem Lernen um 21,7 Prozentpunkte erhöht (von 54,1% auf 75,8%). Des Weiteren ist zu erkennen, dass die Anzahl der benötigten Anfragen, um die gleiche Klassifikationsqualität zu erreichen, um über 90% reduziert wurde (Passiv: 54% mit 1.000 Anfragen; Aktiv: 62% mit 20 Anfragen). Ebenso ist die beobachtete Standardabweichung im aktiven Fall geringer. Der Wilcoxon-Vorzeichen-Rang-Test bestätigt den offensichtlichen Unterschied ($p < 0,002$).

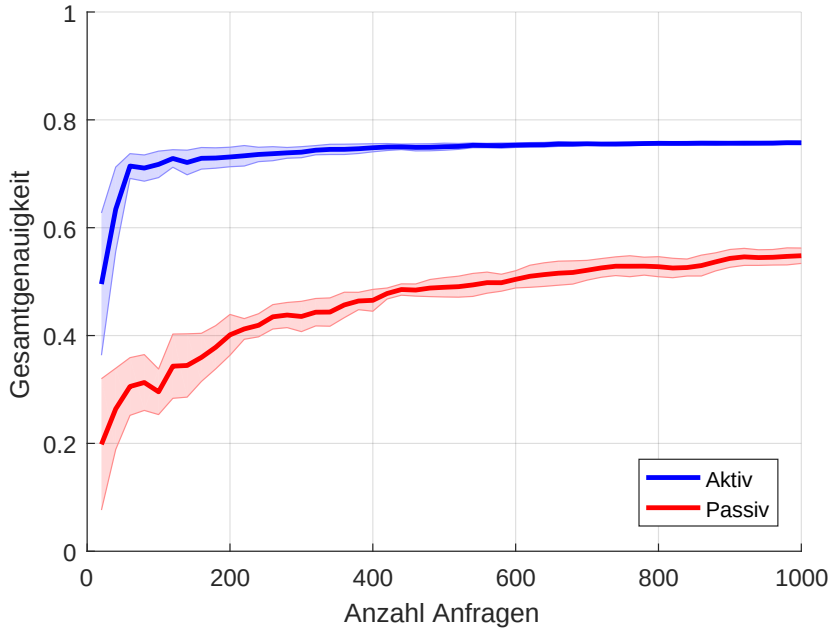


Abbildung 6.4: Vergleich zwischen aktivem und passivem Lernen auf dem Vaihingen-Datensatz. Die durchgezogenen Linien sind die Mittelwerte und die schattierten Flächen die Standardabweichungen der Gesamtgenauigkeit von zehn Wiederholungen des Experimentes. Der Vorteil des aktiven Lernens bezüglich maximaler Gesamtgenauigkeit und Lerngeschwindigkeit ist deutlich ersichtlich.

❑ Diskussion

Die Ergebnisse zeigen sehr eindrucksvoll den Vorteil, der durch aktives Lernen erreicht wird. Es wird sowohl die maximal erreichte Qualität, als auch die Lernrate verbessert. Weiterhin zeigt die kleinere Standardabweichung im aktiven Fall, dass die Ergebnisse wesentlich robuster sind.

Das Fazit ist, wie erwartet, dass aktives Lernen einen deutlichen Vorteil gegenüber passivem Lernen bietet.

6.5 Einfluss des lokalen Dichte-Parameters σ

Dieses Experiment untersuchte welchen Einfluss der Parameter σ der PAL-Rahmenstruktur auf die Gesamtgenauigkeit hat. Hierzu wurden drei verschiedene Werte untersucht.

❑ Ergebnis

Die resultierenden Lernkurven dieses Experimentes sind in Abbildung 6.5 dargestellt. Es ist zu beobachten, dass für $\sigma = 0,05$ die Lernrate zu Beginn sehr gut ist, jedoch die maximal erreichbare Gesamtgenauigkeit für den Wert $\sigma = 0,01$ besser ist. Die Lernkurve für den Parameterwert $\sigma = 0,10$ ist am schlechtesten.

Insgesamt ist zu beobachten, dass das Ergebnis dieses Experiments weniger eindeutig ist als bei anderen Experimenten. Der angewendete statistische Test ergab lediglich zwischen den Werten $\sigma = 0,05$ und $\sigma = 0,15$ einen signifikanten Unterschied ($p < 0,004$). Für die anderen Kombinationen ist der Unterschied statistisch nicht signifikant ($p > 0,05$).

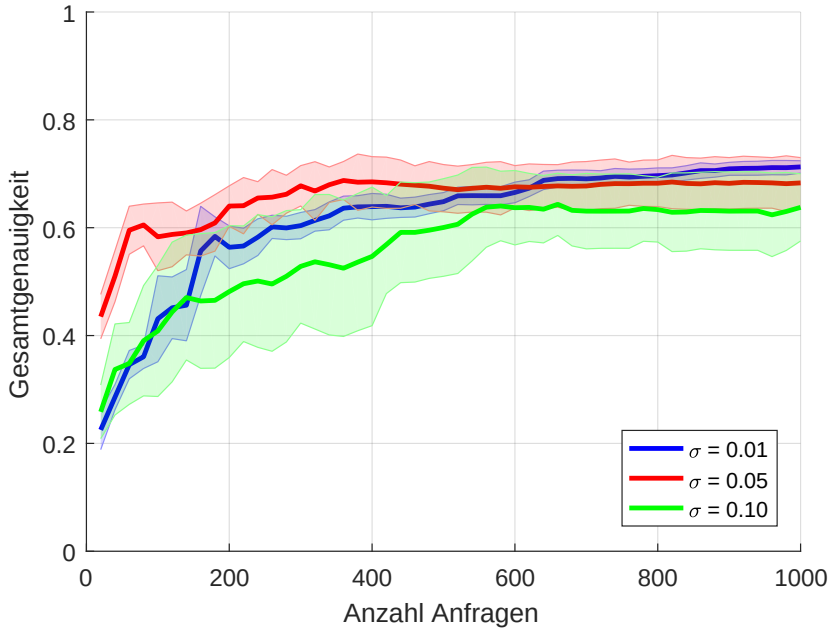


Abbildung 6.5: Der Parameter σ steuert die Größe des Einflussbereichs der einzelnen Repräsentantenvektoren (Abbildung nach [Wuttke et al., 2018] auf dem Datensatz Vaihingen). Kleine Werte bedeuten einen kleinen Einflussbereich während des Trainings. Dies führt zu langsamerer Exploration, weshalb die blaue Kurve zu Beginn schlechter ist als die rote Kurve. Große Werte führen dazu, dass jeder Repräsentantenvektor viele andere beeinflusst. Dies führt zu verringerter Spezialisierung und ist der Grund dafür, dass die rote Kurve am Ende schlechter ist als die blaue Kurve. Im Extremfall beeinflussen sich alle Repräsentantenvektoren gegenseitig, so dass alle gleichberechtigt sind. Das Lernen ähnelt in diesem Fall dem passiven Lernen, da dort die Auswahl der Trainingsbeispiele gleichverteilt stattfindet. Dies zeigt sich darin, dass die grüne Lernkurve am schlechtesten ist.

❑ Diskussion

Der Parameter σ der PAL-Rahmenstruktur skaliert die Überführung des Distanzwertes zwischen zwei Repräsentantenvektoren in einen Ähnlichkeitswert. Dieser wird zur Berechnung des Einflusses einzelner gelabelter Repräsentantenvektoren verwendet. Somit steuert der Parameter direkt die Größe des Einflussbereichs der einzelnen Repräsentantenvektoren. Große Werte führen zu einem großen Einflussbereich und umgekehrt. Das heißt, der Parameter beeinflusst, auf wie viele ungelabelte Repräsentanten sich ein gelabelter Repräsentantenvektor auswirkt.

Um dies zu verdeutlichen, wurden in Abbildung 6.6 für jeden Repräsentantenvektor dessen Ähnlichkeiten zu allen anderen Repräsentantenvektoren sortiert. Als Resultat ist erkennbar, dass größere Werte für σ dazu führen, dass mehr Repräsentantenvektoren einen messbaren Einfluss haben.

Je größer der Einflussbereich ist, desto mehr ähnelt das aktive Lernen dem passiven Lernen. Ein sehr kleiner Einflussbereich fokussiert die Trainingsauswahl auf die einflussreichsten Stichproben, führt jedoch auch zu langsamerem Lernen, da weniger exploriert wird. Andererseits sind die Ergebnisse robuster, was an den kleineren Standardabweichungen erkennbar ist.

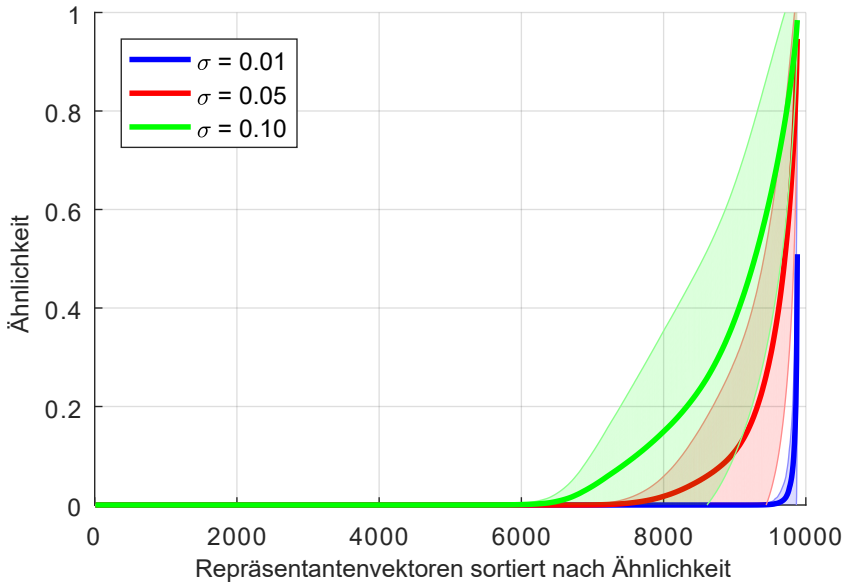


Abbildung 6.6: Durchschnittlicher Einfluss aller Repräsentantenvektoren zueinander (Abbildung nach [Wuttke et al., 2018]). Kleinere Werte für σ führen zu einem kleineren Einfluss. Die durchgezogenen Linien sind der mittlere Einfluss eines Repräsentantenvektors auf alle anderen. Die schattierten Flächen sind die dazugehörigen Standardabweichungen.

6.6 Einfluss der lokalen Dichte

Dieses Experiment untersuchte, ob das Berücksichtigen der lokalen Dichte Vorteile im Trainingsprozess bringt. Hierzu wurden zwei Varianten verglichen: ohne und mit Berücksichtigung der lokalen Dichte. Um einen Vergleichswert zu erhalten, ist in Abbildung 6.7 zusätzlich die Lernkurve des passiven Lernverfahrens dargestellt.

❑ Ergebnis

Die erste Beobachtung ist, dass beide aktiven Varianten signifikant besser sind als passives Lernen (mit Dichte: $p = 0,027$, ohne Dichte: $p = 0,002$). Die zweite Beobachtung ist, dass beide aktiven Varianten deutlich geringere Standardabweichungen haben als passives Lernen. Weiterhin lässt sich beobachten, dass im Vergleich zwischen beiden aktiven Varianten, die Variante ohne Berücksichtigung der lokalen Dichte signifikant bessere Leistungen zeigt ($p = 0,002$), sowohl in der maximalen Gesamtgenauigkeit, als auch in der Lernrate.

❑ Diskussion

Die Schlussfolgerung aus den Ergebnissen ist, dass die Verwendung der mcPAL-Rahmenstruktur zur Berücksichtigung der lokalen Dichte in dieser Anwendung nicht geeignet ist. Dieses Ergebnis ist überraschend, da das mcPAL-Verfahren sehr gute Leistungen auf sechs verschiedenen Datensätzen des UCI-Depots* gezeigt hat [Kottke et al., 2016]. Es muss daher davon ausgegangen werden, dass das Verfahren implizite Annahmen über die Verteilung der Daten trifft, die auf die hier verwendeten Datensätze nicht zutreffen. Es sind daher weitere Untersuchungen notwendig, welche Annahmen erfüllt sein müssen, damit das mcPAL-Verfahren erfolgreich angewendet werden kann.

*UC Irvine Machine Learning Repository [Dheeru & Karra Taniskidou, 2017]. Online verfügbar: `http://archive.ics.uci.edu/ml` (letzter Aufruf: 10.9.2018)

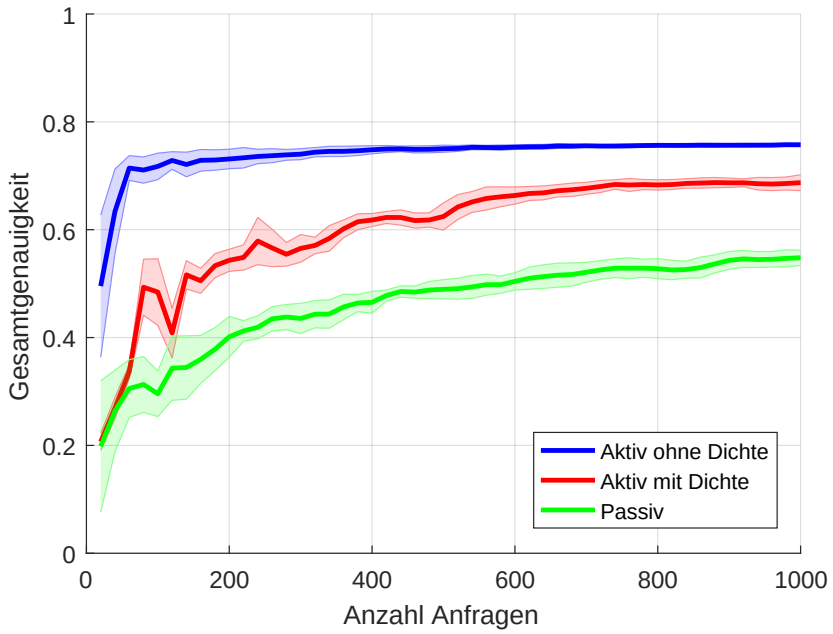


Abbildung 6.7: Die durchgezogenen Linien sind die Mittelwerte von zehn Wiederholungen des Experimentes auf dem Vaihingen-Datensatz. Die schattierten Flächen sind die zugehörigen Standardabweichungen. Beide Varianten, ohne Berücksichtigung der lokalen Dichte (blau) und mit Berücksichtigung (rot) sind besser als passives Lernen (grün). Es wird jedoch eine bessere Leistung erreicht, wenn die lokale Dichte nicht berücksichtigt wird.

6.7 Vergleich auf verschiedenen Datensätzen

In diesem Experiment wurden die Erkenntnisse der anderen Experimente vereint und die besten Parameterwerte auf verschiedenen Datensätzen angewendet und mit passivem Lernen verglichen.

❑ Ergebnis

Die erzielten Lernkurven des Experimentes sind in Abbildung 6.8 dargestellt. Die vorgestellte Methode ist auf allen Datensätzen signifikant besser als das passive Lernen ($p < 0,002$). Auf dem Datensatz Abenberg verbesserte sich die maximale Gesamtgenauigkeit um 16 Prozentpunkte, auf dem Potsdam-Datensatz um 12 Prozentpunkte und auf dem Datensatz Vaihingen um 18 Prozentpunkte. Die Lernrate steigerte sich ebenfalls erheblich, so dass die gleiche Qualität wie beim passiven Lernen im aktiven Fall mit deutlich weniger Anfragen erreicht werden kann. Im Fall vom Abenberg-Datensatz reduzierte sich der Aufwand um 92% und für die Datensätze Potsdam sowie Vaihingen um jeweils 96%. Tabelle 6.1 fasst diese Zahlen übersichtlich zusammen.

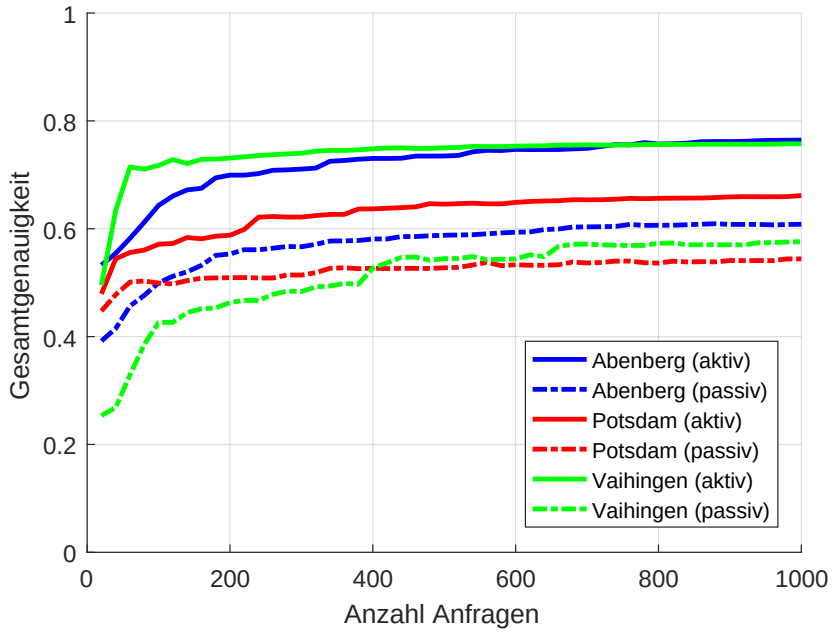


Abbildung 6.8: Lernkurven der vorgestellten SCHAL-Methode (durchgezogene Linien) im Vergleich mit passivem Lernen (gestrichelte Linien) auf drei verschiedenen Datensätzen (Abbildung nach [Wuttke et al., 2018]). Die dargestellten Kurven sind die Mittelwerte aus zehn Wiederholungen des Experimentes. Auf die Darstellung der Standardabweichungen wurde aus Gründen der Übersichtlichkeit verzichtet. Es ist festzustellen, dass die SCHAL-Methode auf allen Datensätzen besser ist als das passive Lernen.

Tabelle 6.1: Verbesserungen durch die SCHAL-Methode für drei Datensätze.

	Abenberg	**Potsdam**	**Vaihingen**
Passiv			
Anfragen	1.000	1.000	1.000
Genauigkeit [%]	60,8	54,4	57,6
Aktiv (Gleiche Anzahl Anfragen)			
Anfragen	1.000	1.000	1.000
Genauigkeit [%]	76,4	66,2	75,7
Verbesserung [pp]	15,6	11,8	18,1
Aktiv (Gleiche Genauigkeit)			
Anfragen	80	40	40
Genauigkeit [%]	61,2	54,4	63,4
Verbesserung [%]	92	96	96

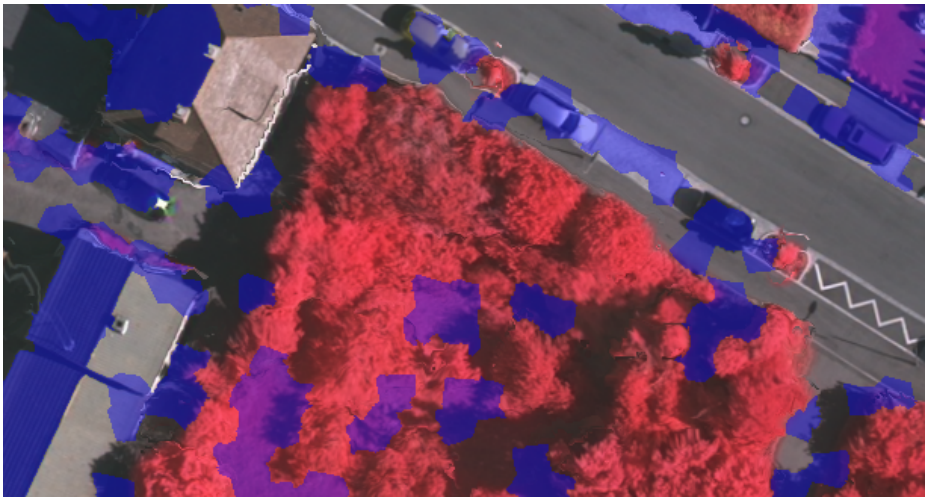


Abbildung 6.9: Detailansicht aus einem Orthofoto des Vaihingen-Datensatzes. Vegetation erscheint aufgrund der dargestellten Kanäle (nahes Infrarot, grün, blau) rötlich. Die blau hervorgehobenen Bereiche wurden falsch klassifiziert. Probleme bereiten vor allem flache Vegetation, schattige Dächer und Autos.

❑ Diskussion

Im Durchschnitt über die drei Datensätze ergibt sich durch den Einsatz der vorgestellten Methode eine Reduktion der Anzahl benötigter Trainingsbeispiele um 94,6% gegenüber dem passiven Lernen. Dies zeigt, dass die Methode nicht auf speziellen Merkmalen eines Datensatzes beruht, sondern auch verallgemeinert eingesetzt werden kann.

Abbildung 6.9 zeigt einen Detailausschnitt des Vaihingen-Datensatzes. Das Eingangsbild ist das Orthofoto bestehend aus dem Infrarot-, Rot- und Grünkanal. Hierdurch ist Vegetation rötlich dargestellt. Die blau überlagerten Bereiche zeigen falsch klassifizierte Pixel. Problematisch sind vor allem Bereiche mit Gras, schattige Dachhälften und Fahrzeuge.

Die falsch klassifizierten Grasflächen wurden mit dem Klassenlabel „Baum“ versehen. Dies ist durch die eingeschränkten Merkmale des Datensatzes erklärbar. In den vorhandenen drei Kanälen erscheinen Grasflächen sehr ähnlich zu Bäumen. Die Fehlklassifikationen in schattigen Bereichen sind ein häufiges Problem in der Fernerkundung [Shahtahmassebi et al., 2013]. Dächer in vollem Sonnenschein und im Schatten ergeben sehr unterschiedliche Spektren. Daher wird die schattige Dachhälfte nicht als die selbe Klasse wie die sonnige Dachhälfte erkannt. Die schlechte Klassifikationsleistung für die Klasse „Auto“ liegt an der zu geringen Menge an Anfragen nach entsprechenden Stichproben. Dies ist ein Indiz dafür, dass die Explorationsleistung der Methode für seltene Klassen verbessert werden sollte.

6.8 Vergleich verschiedener Methoden

Dieses Experiment vergleicht die vorgestellte SCHAL-Methode in ihren beiden Varianten (mit und ohne Berücksichtigung der lokalen Dichte) mit anderen, in Kapitel 2 vorgestellten, Methoden. Um den Vergleich zu vervollständigen, wird passives Lernen den aktiven Verfahren gegenübergestellt.

Wie im Abschnitt 4.5 beschrieben, erfordert die originale mcPAL-Methode quadratischen Aufwand. Sie konnte daher nicht direkt angewendet werden. Stattdessen wurde die Segmentierung des ersten Schrittes der SCHAL-Methode angewendet, um die Daten zu reduzieren. Anschließend wurde die originale mcPAL-Methode verwendet. Sie wird hier daher als multi-class PAL mit Segmentierung (mcPAL*) bezeichnet. Die Ergebnisse sind in Abbildung 6.10 dargestellt.

❑ Ergebnis

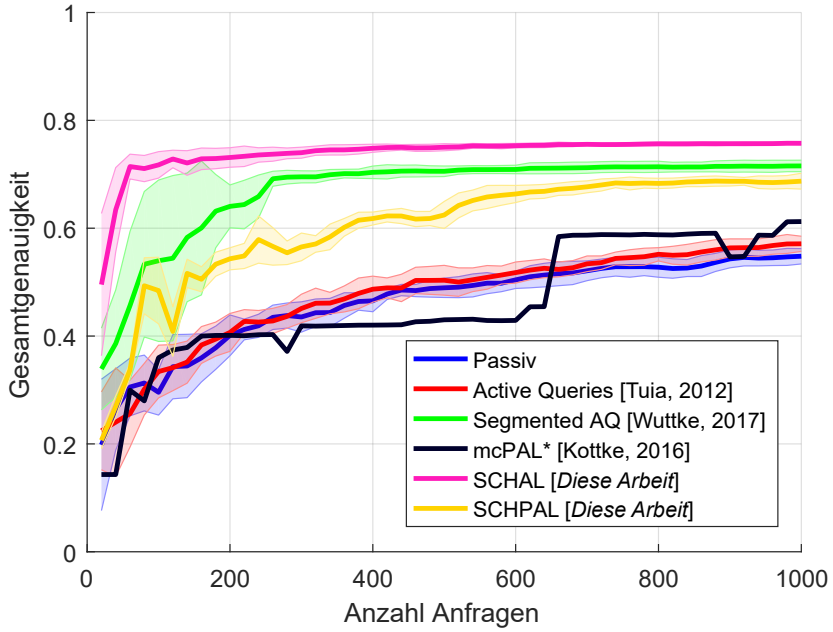


Abbildung 6.10: Vergleich der vorgestellten Methode, in den Varianten mit und ohne Berücksichtigung der lokalen Dichte (SCHPAL und SCHAL) auf dem Vaihingen-Datensatz (Abbildung nach [Wuttke et al., 2018]). Ebenso sind andere Methoden aus dem Stand der Wissenschaft dargestellt. Die durchgezogenen Linien sind die Mittelwerte von zehn Wiederholungen des Experimentes und die schattierten Flächen sind die zugehörigen Standardabweichungen. Die in dieser Arbeit vorgestellte SCHAL-Methode ist in ihrer Version ohne die Berücksichtigung der lokalen Dichte sowohl in maximal erreichter Gesamtgenauigkeit als auch in der Lernrate deutlich besser als die anderen Methoden.

Im Vergleich zu passivem Lernen, verbessern alle Methoden sowohl die erreichte maximale Gesamtgenauigkeit als auch die Lernrate. Die drei Methoden, die eine Segmentierung verwenden, zeigen deutlich bessere Leistungen als die Methoden ohne Segmentierung. Die mcPAL*-Methode ist deterministisch und muss daher nicht wiederholt ausgeführt werden. Ihr Ergebnis ist demzufolge ohne Standardabweichung dargestellt.

Nach einem paarweisen Vergleich aller sechs Methoden bestätigt der Wilcoxon-Vorzeichen-Rang-Test einen signifikanten Unterschied ($p < 0,01$) zwischen jedem Paar außer der mcPAL*-Methode und passivem Lernen ($p = 0,49$). Der Vollständigkeit halber sind alle p-Werte in Tabelle 6.2 aufgelistet.

Tabelle 6.2: Übersicht der statistischen p-Werte für die Nullhypothese, dass die Klassifikationsleistungen zweier Methoden der selben Verteilung entstammen.

	AQ	**SAQ**	**mcPAL***	**SCHAL**	**SCHPAL**
Passiv	0,001	0,002	0,49	0,002	0,002
AQ		0,002	0,006	0,002	0,002
SAQ			0,002	0,002	0,002
mcPAL*				0,002	0,002
SCHAL					0,002

Tabelle 6.3: Vergleich der untersuchten aktiven Methoden mit passivem Lernen.

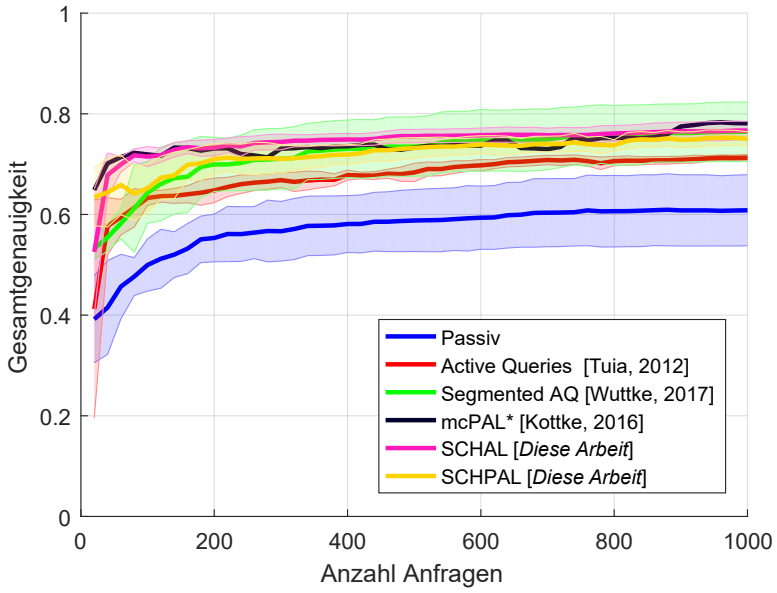
Methode	**Maximale Genauigkeit**	**Weniger Anfragen**
Passiv	55,7%	0%
Active Queries [Tuia et al., 2012]	57,1%	12%
Segmented Active Queries [Wuttke et al., 2017]	71,6%	86%
mcPAL* [Kottke et al., 2016]	61,2%	34%
SCHAL [*Diese Arbeit*]	**75,7%**	**96%**
SCHPAL [*Diese Arbeit*]	68,7%	76%

❑ Diskussion

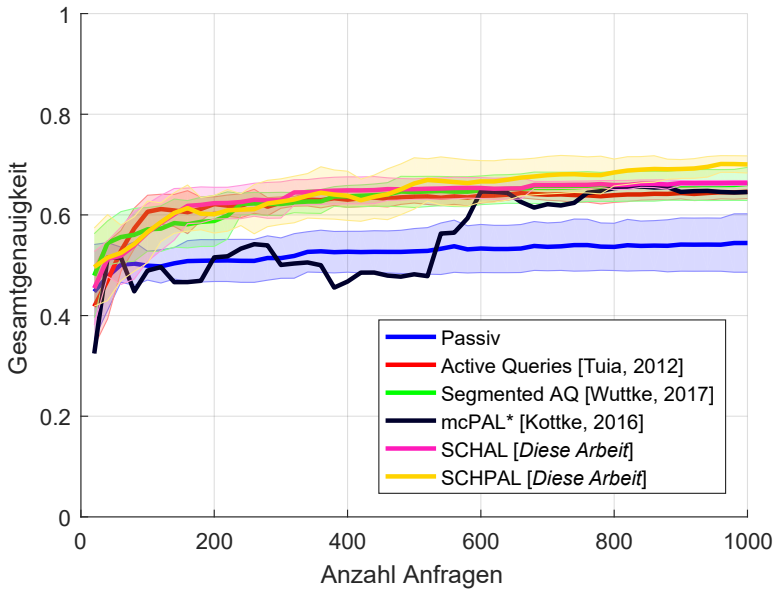
Die Verwendung von Segmentierung als ersten Schritt steigert die Leistung deutlich (vergleiche *active queries* mit *segmented active queries*). Ebenso steigert die Anpassung der verwendeten Distanzmaße die Leistung (vergleiche *segmented active queries* mit SCHAL). Die Berücksichtigung der lokalen Dichte hingegen, verschlechtert die Leistung (vergleiche SCHAL mit SCHPAL). Die maximalen Gesamtgenauigkeiten und die erreichten Verbesserungen der benötigten Anfragen im Vergleich zu passivem Lernen sind in Tabelle 6.3 aufgeführt.

Um auszuschließen, dass die beobachteten Verbesserungen nur mit einem speziellen Datensatz möglich sind, wurde das Experiment auf den anderen zwei Datensätzen ausgeführt. Abbildung 6.11 zeigt diese Ergebnisse. Die Verbesserungen sind nicht so stark ausgeprägt wie auf dem Vaihingen-Datensatz, es ist dennoch erkennbar, dass die Verwendung der Segmentierung als erster Schritt besser ist als andere aktive Lernmethoden, welche wiederum besser sind als passives Lernen.

Die Klassifikationsergebnisse nach Abschluss des Trainingsprozesses aller verglichenen Methoden sind in Abbildung 6.12 dargestellt. Das passive Lernen klassifiziert sehr viele Pixel als „Hintergrund“, so dass das Ergebnis sehr verrauscht wirkt. Die aktiven Methoden hingegen ergeben ein klareres Ergebnis mit Ausnahme der mcPAL-Methode, welche sehr häufig die Klassen „Baum“ und „Flache Vegetation“ verwechselt. Die Einführung der Segmentierung stabilisiert die Ergebnisse deutlich, wie im Vergleich zwischen den Methoden *active queries* und *segmented active queries* zu erkennen ist. Das beste Ergebnis liefert die vorgestellte Methode ohne Berücksichtigung der lokalen Dichte. Durch Verwendung der mcPAL-Rahmenstruktur werden mehr Pixel als „Hintergrund“ klassifiziert und die Klasse „Gebäude“ ist weniger klar gelabelt. Es kommt öfter zu einer Verwechslung mit der Klasse „Versiegelte Fläche“.



(a)



(b)

Abbildung 6.11: Vergleich der verschiedenen Methoden auf den Datensätzen Abenberg (a) und Potsdam (b). Die durchgezogenen Linien sind die Mittelwerte und die schattierten Flächen sind die Standardabweichungen von zehn Wiederholungen des Experimentes. Die aktiven Lernmethoden sind besser als passives Lernen. Die in dieser Arbeit vorgestellte SCHAL-Methode zeigt die besten Leistungen.

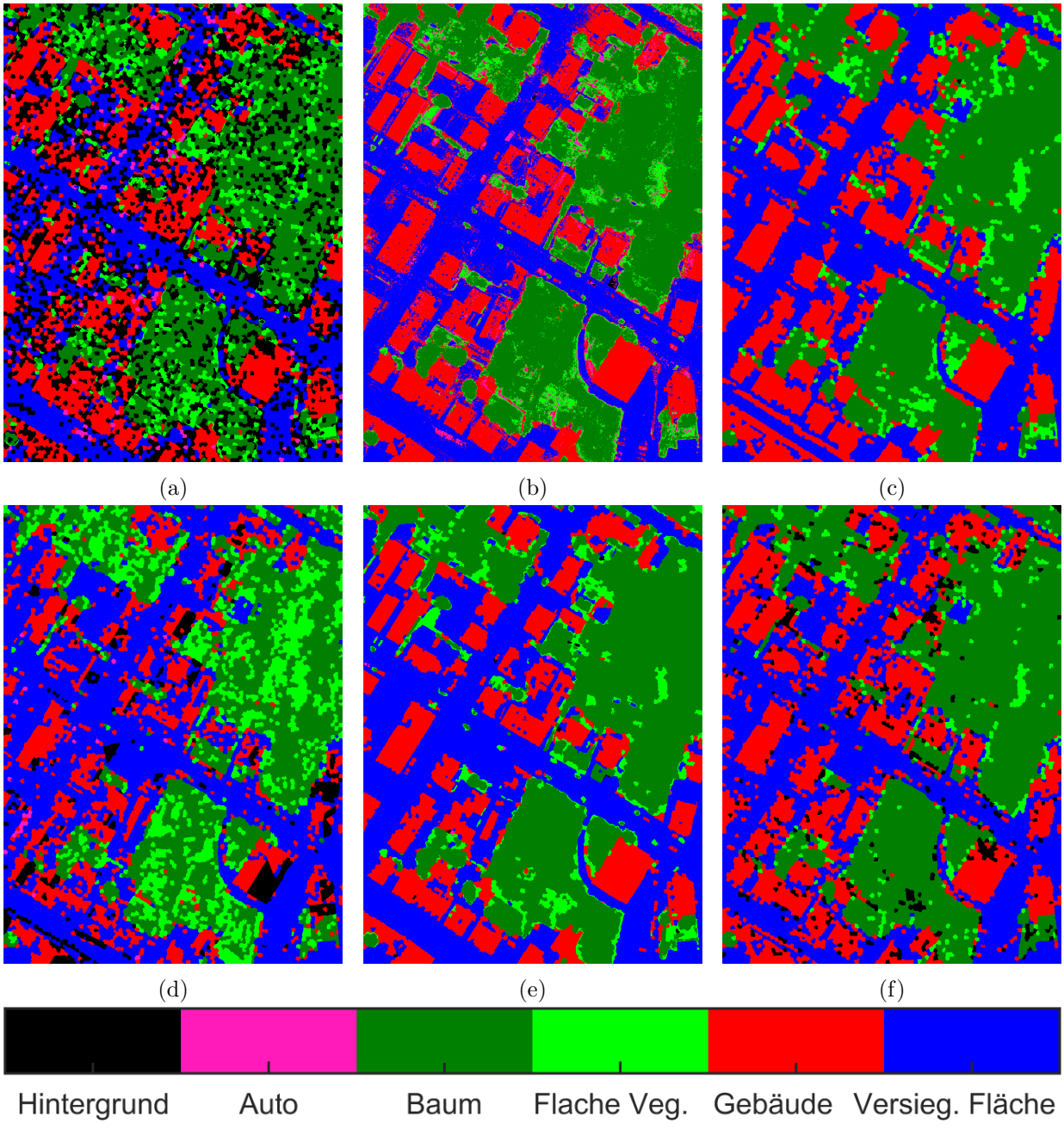


Abbildung 6.12: Vergleich der Klassifikationsergebnisse aller in dieser Arbeit untersuchten Methoden auf dem Vaihingen-Datensatz (Abbildung nach [Wuttke et al., 2018]). Passives Lernen (a) zeigt die schlechtesten Ergebnisse. Die Methoden des aktiven Lernens (b-f) erkennen alle am häufigsten auftretenden Klassen, zeigen jedoch Unterschiede darin wie zuverlässig die Klassen erkannt werden. (b) *active queries* [Tuia et al., 2012], (c) *segmented active queries* [Wuttke et al., 2017], (d) mcPAL* [Kottke et al., 2016], (e) [*diese Arbeit*] SCHAL, (f) [*diese Arbeit*] SCHPAL.

6.9 Allgemeine Diskussion

In Abschnitt 4.2.2 wurde über die durchgeführte Anpassung des SLIC-Algorithmus gesprochen. Durch Änderung des verwendeten Distanzmaßes kann die Erstellung der Segmente gesteuert werden. In Abbildung 6.13 ist hierzu der selbe Ausschnitt des Abenberg-Datensatzes mit drei verschiedene Segmentierungen dargestellt. Vor allem in der Mitte des Bildes ist der Unterschied zwischen den drei verwendeten Distanzmaßen zu erkennen. Das euklidische Maß führt dazu, dass die Segmentgrenzen zwar den spektralen Unterschieden zwischen Gras und Straße folgen, jedoch überwiegt der räumliche Anteil und führt insbesondere im linken Bereich zur Vereinigung von Regionen, die getrennt sein sollten. Der spektrale Winkel unter Verwendung von drei Kanälen führt hierbei zu besseren Ergebnissen. Die Region mit sehr dichtem Gras im mittleren Grasstreifen wird klar segmentiert. Der Bereich mit dünnerem Grasbewuchs wird jedoch immer noch fälschlicherweise mit der Straße zusammengefasst. Erst der Einsatz aller vier verfügbaren Kanäle führt zu einer sauberen Trennung von Gras und Straße. Das Bild ist immer noch stark übersegmentiert, dies ist jedoch aus den in Abschnitt 4.3.4 erläuterten Gründen erwünscht. Eine weitere Beobachtung ist, dass die Segmente im angepassten SLIC-Algorithmus deutlich unregelmäßiger und weniger kompakt sind. Dies spielt für die vorgestellte Methode jedoch keine Rolle, da das Endergebnis die Klassenlabel für Pixel und nicht für Segmente sind.



(a) (b) (c)

Abbildung 6.13: Vergleich der Segmentierungsergebnisse von drei verschiedenen Distanzmaßen des SLIC-Algorithmus. Original euklidisches Distanzmaß (a), spektraler Winkel mit drei Kanälen (b) und spektraler Winkel mit allen vier verfügbaren Kanälen (c).

Der Ablauf der vorgestellten Methode ist in Abbildung 6.14 anhand von Zwischenergebnissen dargestellt. Der Segmentierungsschritt überführt das Eingangsbild (a) in eine Menge von Repräsentantenvektoren. Jeder dieser Vektoren repräsentiert eines der Segmente (b). Anschließend erstellt der Clusterbildungsschritt einen Binärbaum als hierarchisches Clustering (c, d). Der Schritt des aktiven Lernens stutzt diesen Baum unter zu Hilfenahme der *Ground Truth* (e) bis das Anfragebudget erschöpft ist. Ein Mehrheitsentscheid überführt das optimierte Pruning in die Klassifikationskarte (f).

Für das bessere Verständnis des Trainingsvorgangs werden in Abbildung 6.15 die Klassifikationsergebnisse für den Vaihingen-Datensatz aus unterschiedlichen Trainingsstadien gezeigt. Zu Beginn des Trainings nach nur 20 Anfragen werden noch viele Pixel mit dem Klassenlabel „Hintergrund“ versehen, da noch nicht genügend Informationen für eine korrekte Klassifikation vorhanden sind. Ebenso wird allen Pixeln von flacher Vegetation das Klassenlabel „Baum“ gegeben. Dies ist ein Anzeichen dafür, dass die Klasse „Flache Vegetation“ noch nicht von der Methode entdeckt wurde. Der gleiche Effekt tritt bei den Klassen „Gebäude“ und „Versiegelte Fläche“ auf. Nach 100 Anfragen werden deutlich weniger Pixel der Klasse „Hintergrund“ zugeordnet und die bisher fehlenden Klassen wurden entdeckt. Am Ende des Trainingsprozesses nach 1.000 Anfragen sind weiterhin große Flächen der flachen Vegetation als „Baum“ falsch klassifiziert. Dies ist ein Zeichen

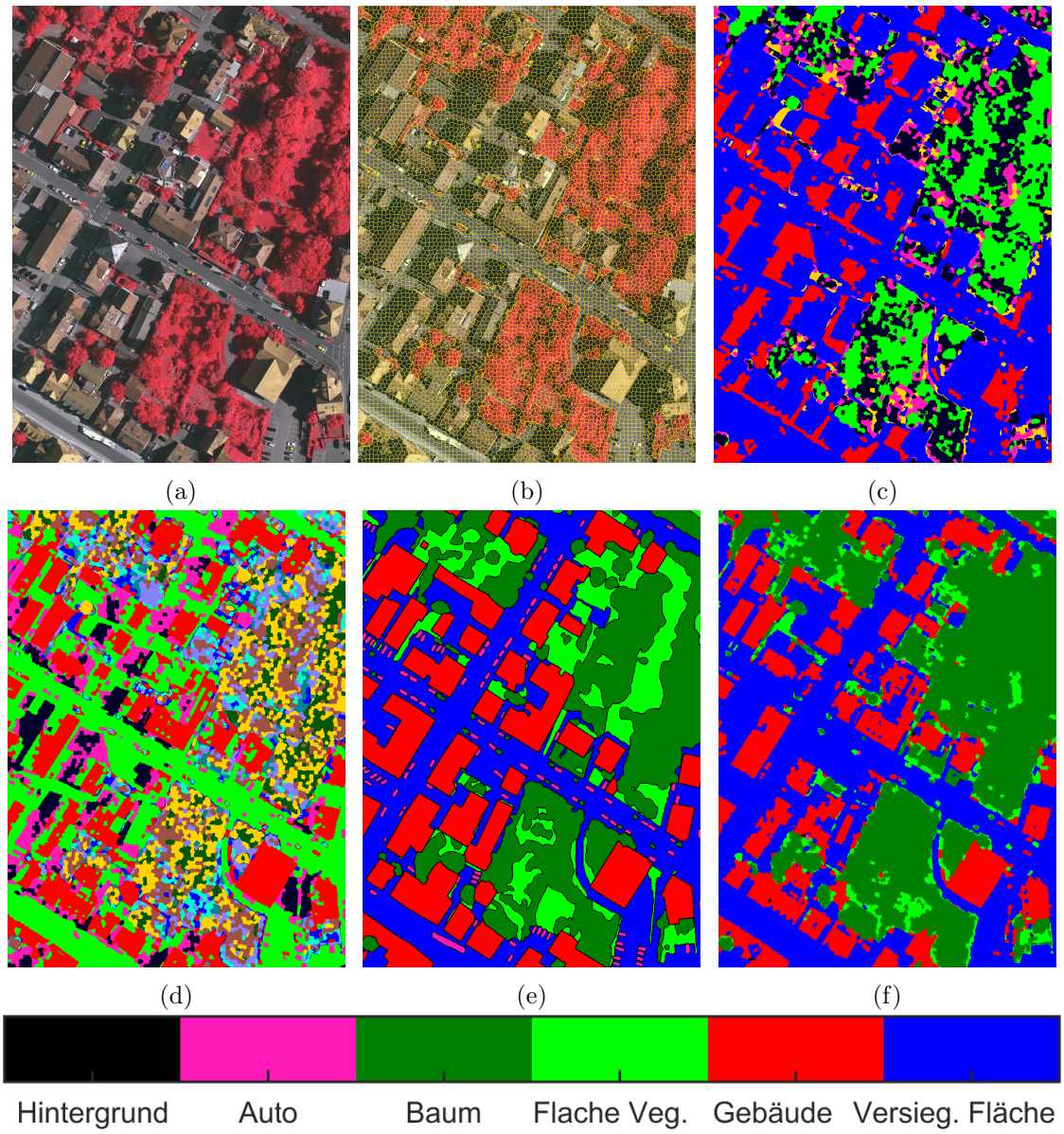


Abbildung 6.14: Ergebnisse der einzelnen Methodenschritte anhand des Vaihingen-Datensatzes (Abbildung nach [Wuttke et al., 2018]). (a) Eingangsbild, (b) Segmentierung, (c) Clusterbildung nach 10 Iterationen, (d) Clusterbildung nach 20 Iterationen, (e) *Ground Truth*, (f) Klassifikationsergebnis. Die Legende gilt nur für die Teilbilder e und f. Da das Clustering (c und d) unüberwacht stattfindet, existieren keine Klassenlabel für dessen Ergebnis.

dafür, dass die in den Daten vorhandenen Kanäle (infrarot, grün und blau) keine ausreichenden Merkmale zur vollständigen Trennung sind. Höheninformation zu verwenden ist an dieser Stelle sehr hilfreich, liegt jedoch außerhalb des Fokus dieser Arbeit.

Zum Vergleich sind in Abbildung 6.16 die Ergebnisse des Potsdam-Datensatzes in gleicher Weise aufbereitet. Im Gegensatz zum Vaihingen-Datensatz wurden hier direkt alle Hauptklassen entdeckt. Eine Ursache der deutlich besseren Leistung der Klassifikation der Klasse „Auto“ kann sein, dass es wesentlich mehr Fahrzeuge gibt. Jedoch ist die Verwechslung zwischen den Klassen „Gebäude“ und „Versiegelte Flächen“ deutlich stärker, was zu der insgesamt schlechteren maximalen Klassifikationsgenauigkeit im Vergleich mit dem Vaihingen-Datensatz führt.

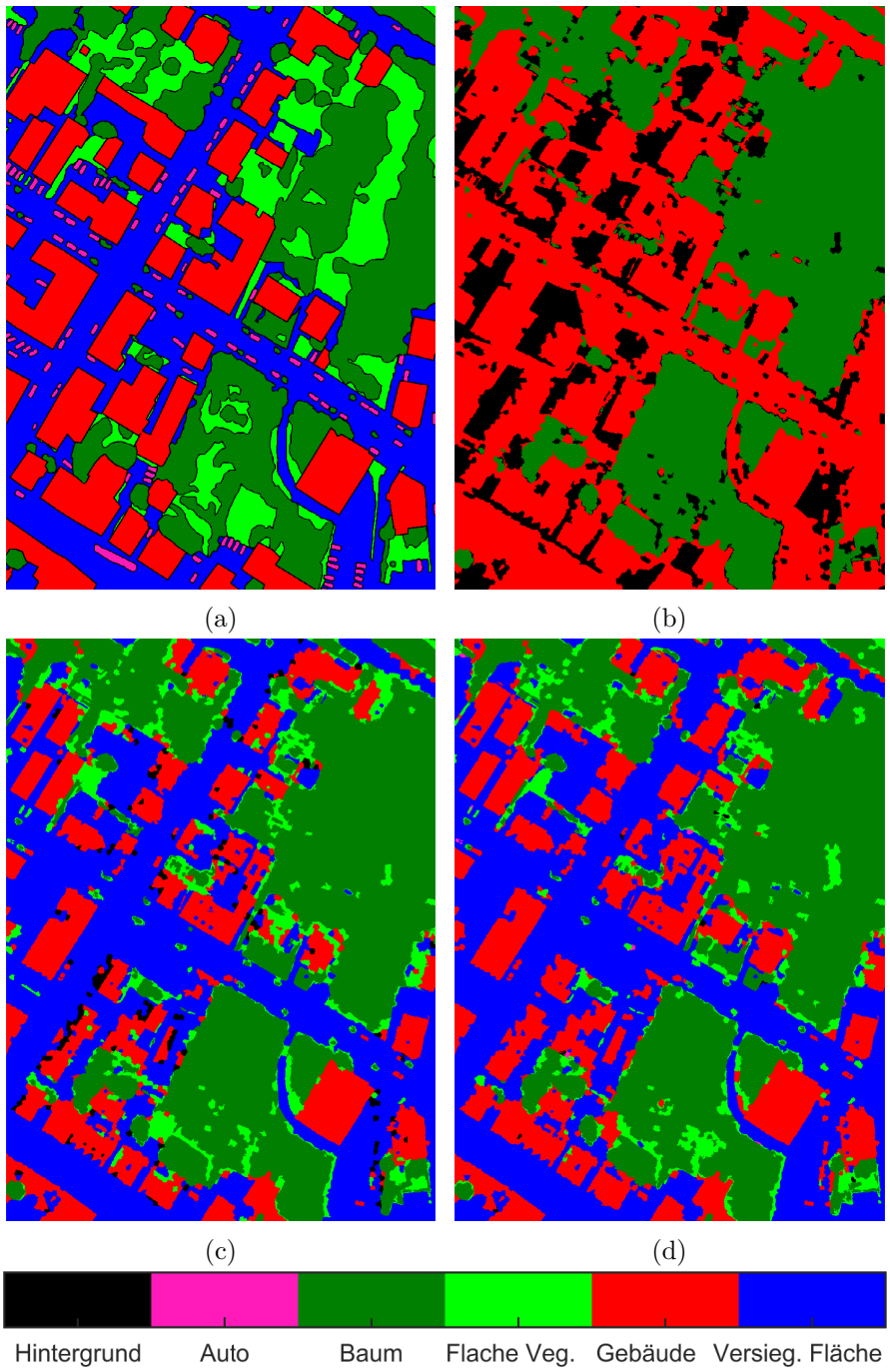


Abbildung 6.15: *Ground Truth* (a) und Klassifikationskarten (b-d) zu verschiedenen Zeitpunkten des Trainingsprozesses für den Vaihingen-Datensatz. Nach 20 Iterationen (b) wurden noch nicht alle Klassen entdeckt. Die Klassen „Gebäude“ und „Versiegelte Flächen“ sowie „Baum“ und “Flache Vegetation“ sind noch nicht voneinander getrennt. Nach 100 Anfragen (c) werden deutlich weniger Pixel als „Hintergrund“ klassifiziert und die Klassen werden besser getrennt. Nach 1.000 Anfragen (d) verbleibt ein Klassifikationsfehler von 25%. Dieser tritt hauptsächlich bei den Klassen „Flache Vegetation„ und „Auto“ auf.

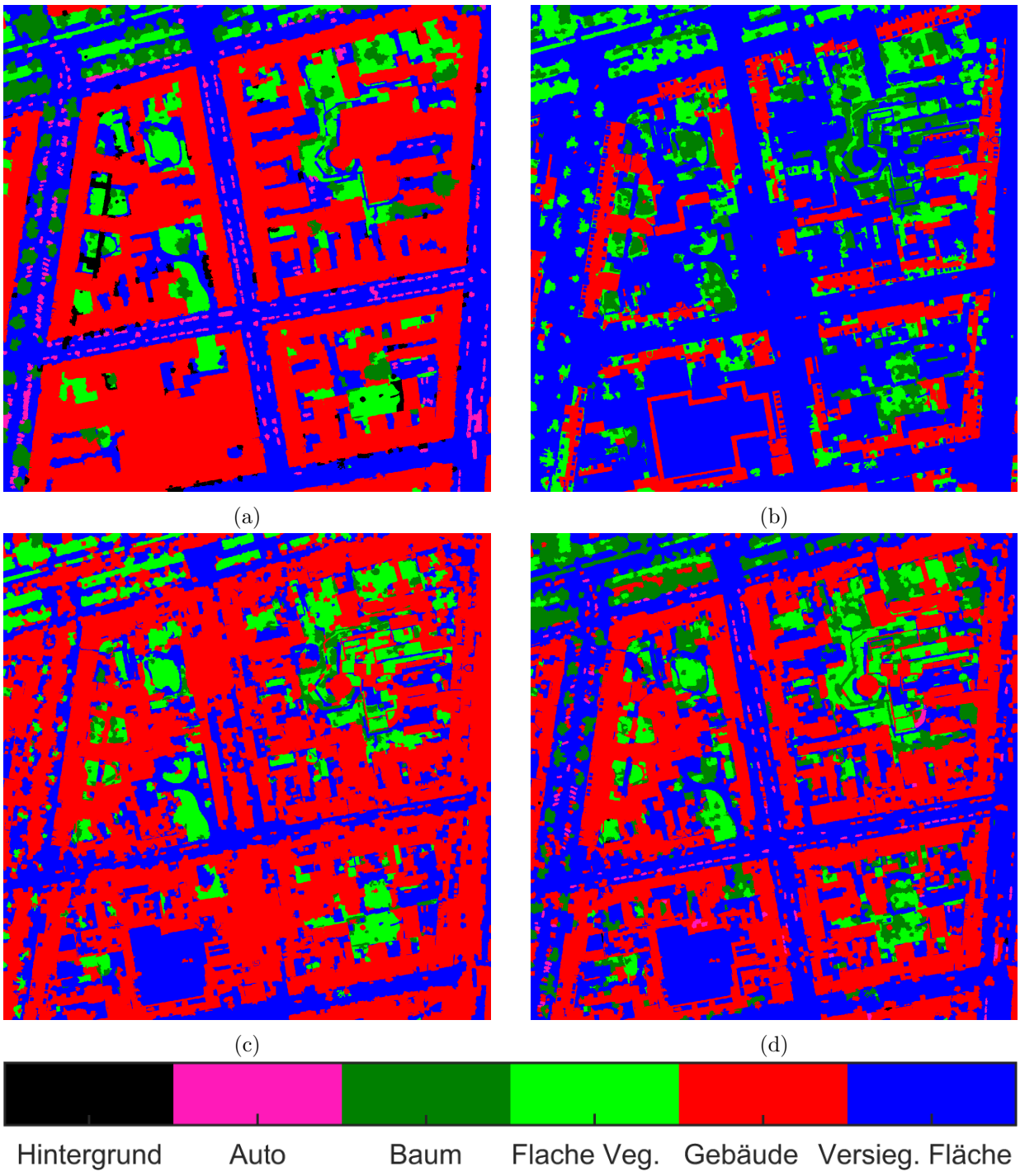


Abbildung 6.16: Klassifikationsergebnisse für den Potsdam-Datensatz während des Trainings (Abbildung nach [Wuttke et al., 2018]). *Ground Truth* (a) und Klassifikationsergebnisse des Potsdam-Datensatzes. Die Teilbilder zeigen die Klassifikationsergebnisse nach 20 (b), 100 (c) und 1.000 (d) Iterationen. Die Hauptklassen wurden frühzeitig entdeckt. Jedoch finden häufige Verwechslungen zwischen den Klassen „Gebäude“ und „Versiegelte Flächen“ statt.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

❑ Führt die Kombination von Segmentierung, Clusterbildung und aktivem Lernen zu effizienterem Training?

Zielstellung dieser Arbeit war es, Segmentierung, Clusterbildung und aktives Lernen zu kombinieren, so dass die Trainingseffizienz eines Verfahrens zur Landbedeckungsklassifikation gesteigert wird. Die vorgestellte Methode erreicht dieses Ziel auf zwei verschiedene Arten. Wird die zu erreichende Klassifikationsgüte festgehalten, reduziert die Methode die Anzahl der benötigten Trainingsbeispiele im Durchschnitt über drei Datensätze um 94,6%. Wird alternativ die Anzahl der verwendeten Trainingsbeispiele festgehalten, steigert die Methode die Klassifikationsgenauigkeit um 15,1 Prozentpunkte im Vergleich zum passiven Lernen.

❑ Welchen Beitrag leisten die einzelnen Schritte der Methode?

Den größten Einfluss auf diese Effizienzsteigerung hat der Einsatz des Segmentierungsschrittes. An zweiter Stelle steht der Schritt des aktiven Lernens. Der Schritt der Clusterbildung trägt zum Erfolg bei, jedoch in geringerem Ausmaß als die anderen beiden Schritte. Generell lässt sich sagen, dass die Standardabweichung der beobachteten Lernkurven sehr gering ist. Hieraus lässt sich schließen, dass die erzielten Ergebnisse präzise sind.

❑ Wie können Informationen über die lokale Dichte integriert werden und unterstützen sie den Lernprozess?

Des Weiteren wurde untersucht, ob das Integrieren von lokalen Dichteinformationen durch den Einsatz der PAL-Rahmenstruktur die Klassifikationsleistung steigert. Diese Vermutung bestätigte sich nicht. Stattdessen wurde festgestellt, dass diese Ergänzung die Klassifikationsgenauigkeit und Lerngeschwindigkeit reduziert. Es ist daher die Schlussfolgerung gezogen worden, dass vor einem Einsatz der PAL-Rahmenstruktur zur Landbedeckungsklassifikation mehr Forschungsarbeiten notwendig sind. Vielversprechend erscheinen Untersuchungen zur Struktur und Form der Cluster im Merkmalsraum. Alternativ können andere Herangehensweisen zum Integrieren der lokalen Dichte herangezogen werden, wie zum Beispiel in [Xu et al., 2007; Zhu et al., 2010]. Auf dem aktuellen Stand erzielt die vorgestellte Methode bessere Ergebnisse, wenn die lokalen Dichteinformationen nicht integriert werden.

❑ Wie unterscheiden sich die Ergebnisse der vorgestellte Methode im Vergleich zu Methoden nach dem Stand der Forschung?

Die in dieser Arbeit vorgestellte SCHAL-Methode reduziert die Trainingskosten stärker als die anderen untersuchten Methoden des aktiven Lernens. Ebenso ist die maximal erreichte Klassifikationsgenauigkeit höher. Die Segmente in den Klassifikationskarten sind deutlich größer und stärker verbunden im Vergleich zu den anderen Methoden des Stands der Forschung.

7.2 Ausblick

Mit Hilfe der vorgestellten Methode konnten die Eingangs gestellten Forschungsfragen beantwortet werden. Die Methode bleibt dabei sehr modular und flexibel. Jeder Schritt kann an spezifische Bedürfnisse angepasst werden. Zum Beispiel kann der SLIC-Algorithmus im ersten Schritt durch einen alternativen Segmentierungsalgorithmus ersetzt werden, ohne die darauf folgenden Schritte anpassen zu müssen. Ebenso kann im zweiten Schritt der *bisecting* k-Means Algorithmus durch einen anderes Clusterverfahren ersetzt werden, solang dieses Verfahren ein hierarchisches Clustering als Ergebnis produziert. Schließlich kann im dritten Schritt die Selektionsstrategie ausgetauscht werden, falls es für das Einsatzszenario erforderlich sein sollte, weitere Randbedingungen zu beachten. Ein interessanter Untersuchungsgegenstand ist beispielsweise die Frage, wie es bereits zu Beginn des Trainings möglich ist, wichtige Stichproben zu identifizieren. Das gilt insbesondere dann, wenn noch nicht alle Klassen entdeckt worden sind und gleichzeitig keine oder nur sehr wenige Label-Informationen vorliegen. Mögliche Lösungen hierfür können Methoden aus dem Gebiet der Empfehlungs-Systeme sein [Zhang et al., 2014]. Diese Herausforderung ist in der Literatur auch unter der Bezeichnung „Kaltstart-Problem“ bekannt [Maltz & Ehrlich, 1995].

Im Folgenden werden konkrete Bereiche genannt, um die vorgestellte Methode zu ergänzen, zu verbessern und zu erweitern. Hierzu werden drei Erfolg versprechende Ansätze erläutert, die unmittelbar an die vorgestellte Methode anknüpfen.

Ein erster Ansatz ist die Verwendung von Methoden zur Merkmalsextraktion. Abhängig von ihrer Art können diese Methoden entweder vor oder nach dem Segmentierungsschritt angewendet werden. Ein Beispiel für den Einsatz vor der Segmentierung ist das Konkatenieren neuer Merkmalsdimensionen. Hier könnte zum Beispiel die Integration des NDVIs in die zu segmentierenden Merkmalsvektoren die Leistung des SLIC-Algorithmus verbessern. Beispiele für den Einsatz nach der Segmentierung sind aus den Segmenten berechnete Merkmale wie Textureigenschaften [Berberoğlu et al., 2010] und Gestaltfaktoren [Michaelsen et al., 2016]. Diese neuen Merkmale können verwendet werden, um bessere Repräsentantenvektoren zu erzeugen. Ebenso ist es erfolgversprechend Nicht-Linearitäten aus den Daten zu entfernen, die durch variierende Lichtverhältnisse, verschiedene Blickwinkel oder Mehrfachreflexion des Umgebungslichts entstehen [Gross et al., 2015].

Ein zweiter Ansatz zur Erweiterung ist die Integration von teilüberwachtem Lernen [Chapelle et al., 2010]. Hierzu werden die Klassenlabel von Knoten mit sehr hoher Klassifikationssicherheit als gegeben angenommen. Ein Klassenlabel gilt als sicher, wenn der Unterschied zwischen der oberen $p^{OS}_{v,w}$ und unteren $p^{US}_{v,w}$ Unsicherheitsschranke sehr groß ist. Mit Hilfe dieser neuen Label-Informationen könnte eine SVM trainiert werden. Dieser Ansatz folgt der Annahme, dass Stichproben, die von der hier vorgestellten Methode als wichtig beurteilt werden, auch für das Training der SVM wichtig sind. Diese Annahme kann überprüft werden, indem untersucht wird, wie viele Trainingsbeispiele Stützvektoren der SVM werden. Ist dieser Anteil sehr gering, sind sehr viele Label-Informationen redundant und könnten als Trainingsbeispiele eingespart werden. Mit dieser Erweiterung könnte die vorgestellte Methode auch für Szenarien eingesetzt werden, bei denen ein Vorabtraining möglich ist und die Klassifizierung schon während der Erfassung der neuen Daten möglich sein muss. Ebenso könnte das Neu-Trainieren für den Einsatz auf weiteren Datensätzen entfallen.

Ein insbesondere für die Landbedeckungsklassifikation geeigneter Ansatz ist die Integration von Höheninformationen. Hierzu kann der räumliche Anteil $D_{spatial}$ (siehe Definition 2.6) des vom SLIC-Algorithmus verwendeten Distanzmaßes um eine dritte Komponente ergänzt werden. Dabei muss darauf geachtet werden, dass alle räumlichen Bestandteile korrekt zueinander skaliert sind. Ohne diese Skalierung kann es passieren, dass ein Bestandteil die anderen dominiert und das Ergebnis verzerrt wird. Die Höheninformationen stehen meist in der Einheit Meter zur Verfügung, die räumlichen Informationen zur Ausdehnung im Bild sind jedoch in der Einheit Pixel

angegeben. Der verbindende Faktor ist die Bodenauflösung, welche zur Skalierung herangezogen werden kann. Es bleibt zu untersuchen, ob durch die neuen Höheninformationen die erwartete, verbesserte Trennung der Klassen „Baum“ und „Flache Vegetation“ sowie „Gebäude“ und „versiegelte Fläche“ erreicht wird. Eine ähnliche Verbesserung ist das Erweitern des spektralen Anteils $D_{spektral}$ (siehe Definition 2.5) auf eine größere Anzahl von Kanälen, so dass auch Hyperspektraldaten verwendet werden können. Diese Erweiterung ist durch die in der vorliegenden Arbeit präsentierte Verwendung des spektralen Winkels D_{SA} (siehe Definition 3.23) mathematisch sehr einfach möglich. Es ist sehr interessant in Zukunft zu untersuchen, ob diese Erweiterung eine Verbesserung der Ergebnisse bewirkt.

Zusammenfassend lässt sich feststellen, dass die in dieser Arbeit vorgestellte Methode erfolgreich drei sehr verschiedene Ansätze des maschinellen Lernens kombiniert: Segmentierung, Clusterbildung und aktives Lernen. Somit stellt sie ein wertvolles Werkzeug zur Landbedeckungsklassifikation dar.

Eigene Veröffentlichungen

Im Rahmen der Untersuchungen zur vorliegenden Arbeit entstanden durch eigene Arbeiten folgende Veröffentlichungen:

- [Wuttke et al., 2012]: Ein Verfahren zur Reduzierung des Trainingsaufwands für die Landbedeckungsklassifikation wird vorgestellt. Es visualisiert die aktuelle Klassifikationssicherheit des Lernverfahrens und ermöglicht es dem Anwender gezielt die Stichproben auszuwählen, welche für das Training am nützlichsten sind. In Verbindung mit einem k-nächster-Nachbar-Verfahren lässt sich die Anzahl benötigter Trainingsbeispiele um 80% reduzieren.

- [Wuttke et al., 2014]: Untersuchungen zeigen, dass die Eignung der Selektionsstrategie von der Wahl des Klassifikators abhängt. Für einen Maximum-Likelihood-Klassifikator wurden drei verschiedene Selektionsstrategien untersucht und bewertet.

- [Wuttke et al., 2015]: Eine Verbundmethode wird vorgestellt und der Einfluss verschiedener äußerer und innerer Faktoren untersucht. Äußere Faktoren sind die Verfügbarkeit der Stichproben und die Verteilung der Merkmalsvektoren. Innere Faktoren sind die Wahl des Klassifikators und der Selektionsstrategie.

- [Wuttke et al., 2016]: Aktives Lernen mit Hilfe von *uncertainty sampling* in Verbindung mit SVMs ist eine Herausforderung. Es wurden verschiedene Fehlerquellen untersucht und mögliche Lösungsvorschläge diskutiert.

- [Wuttke et al., 2017]: Die vorgestellte Methode verwendet einen Segmentierungsschritt als Vorbereitung vor dem aktiven Lernen. Dies führt zu einer starken Steigerung der Klassifikationsgüte und Lernrate, so dass deutlich weniger Trainingsbeispiele benötigt werden.

- [Wuttke et al., 2018]: Die Verwendung des spektralen Winkels anstatt des Euklidischen Distanzmaßes steigert die Klassifikationsgüte. Acht Experimente untersuchen die verschiedenen Parameter der vorgestellten Methode.

Ebenso fand eine unterstützende Arbeit als Mitautor folgender Veröffentlichungen statt:

- [Schilling et al., 2013]: Es wurde eine Multi-Sensor-Plattform vorgestellt, welche die aufgenommenen Daten in Echtzeit zur Bodenstation überträgt. Unterstützung wurde im Bereich der Datenaufbereitung und -verarbeitung für Klassifikationsaufgaben gegeben.

- [Lenz et al., 2014]: Ein Verfahren zur automatischen Boresight-Kalibrierung von Pushbroom-Sensoren wurde vorgestellt. Unterstützung fand bei der direkten Georeferenzierung durch Vorwärtsprojektion statt.

- [Gross et al., 2015]: Diese Arbeit stellt eine Methode zur Transformation von Hyperspektraldaten vor. Mit Hilfe von Referenzspektren können nicht-lineare Effekte in den Daten reduziert werden. Unterstützung wurde bei der anschließenden Klassifikation unter Einsatz des spectral angle mapper (SAM) gegeben.

Literaturverzeichnis

Abe N, Mamitsuka H (1998) Query learning strategies using boosting and bagging. In: *ICML'98 Proceedings of the 15th International Conference on Machine learning* 1–9. San Francisco, CA, USA: Morgan Kaufmann.

Achalakul T, Taylor S (2001) Real-time multi-spectral image fusion. *Concurrency and Computation: Practice and Experience*, 13 (12): 1063–1081.

Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11): 2274–2281.

Akbari H, Shea Rose L, Taha H (2003) Analyzing the land cover of an urban environment using high-resolution orthophotos. *Landscape and Urban Planning*, 63 (1): 1–14.

Al-Amri SS, Kalyankar NV, Khamitkar SD (2010) Image segmentation by using threshold techniques. *Journal of Computing*, 2 (5): 83–86.

Albert L, Rottensteiner F, Heipke C (2017) A higher order conditional random field model for simultaneous classification of land cover and land use. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130: 63–80.

Ali M, Clausi D (2001) Using the Canny edge detector for feature extraction and enhancement of remote sensing images. In: *IGARSS'01 Proceedings of the International Geoscience and Remote Sensing Symposium*, volume 5 2298–2300. Piscataway, NJ, USA: IEEE.

Ambati V, Vogel S, Carbonell JG (2010) Active learning and crowd-sourcing for machine translation. In: *LREC'10 Proceedings of the 7th International Conference on Language Resources and Evaluation* 2169–2174. Pittsburgh: Institute for Software Research at Carnegie Mellon University.

Anderson JR (1976) *A land use and land cover classification cystem for use with remote sensor data.* Number 964 in Geological Survey Professional Papers. Washington: U.S. Government Printing Office.

Angluin D (1988) Queries and concept learning. *Machine Learning*, 2 (4): 319–342.

Arthur D, Vassilvitskii S (2006) How slow is the k-means method? In: *SCG'06 Proceedings of the 22nd annual Symposium on Computational Geometry* 144–153. New York: Association for Computing Machinery.

Atlas LE, Cohn D, Ladner R (1990) Training connectionist networks with queries and selective sampling. In: *NIPS'89 Proceedings of the 2nd International Conference on Neural Information Processing Systems* 566–573. San Francisco: Morgan Kaufmann.

Bailly A, Chapel L, Tavenard R, Camps-Valls G (2017) Nonlinear time-series adaptation for land cover classification. *IEEE Geoscience and Remote Sensing Letters*, 14 (6): 896–900.

Balcan MF, Beygelzimer A, Langford J (2006) Agnostic active learning. In: *ICML'06 Proceedings of the 23rd international conference on Machine learning* 65–72. New York, NY, USA: Association for Computing Machinery.

Balcan MF, Blum A (2005) A PAC-style model for learning from labeled and unlabeled data. In: Auer P, Meir R (eds) *Colt 2005*, volume 3559 of *Lecture Notes in Computer Science* 111–126. Springer, Berlin, Heidelberg.

Bao Q, Guo P (2004) Comparative studies on similarity measures for remote sensing image retrieval. In: *IEEE International Conference on Systems, Man and Cybernetics* 1112–1116. Piscataway, NJ, USA: IEEE.

Berberoğlu S, Akin A, Atkinson PM, Curran PJ (2010) Utilizing image texture to detect land-cover change in Mediterranean coastal wetlands. *International Journal of Remote Sensing*, 31 (11): 2793–2815.

Berberoğlu S, Lloyd CD, Atkinson PM, Curran PJ (2000) The integration of spectral and textural information using neural networks for land cover mapping in the Mediterranean. *Computers & Geosciences*, 26 (4): 385–396.

Beucher S, Lantuejoul C (1979) Use of watersheds in contour detection. In: *Proceedings of the International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation* 17–21. Rennes, France: CCETT,.

Beygelzimer A, Langford J, Tong Z, Hsu DJ (2010) Agnostic active learning without constraints. In: *NIPS'10 Proceedings of the 23rd International Conference on Neural Information Processing Systems* 199–207. Red Hook, NY, USA: Curran Associates.

Blanzieri E, Melgani F (2008) Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Transactions on Geoscience and Remote Sensing*, 46 (6): 1804–1811.

Bloodgood M, Vijay-Shanker K (2009) A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In: *CoNLL'09 Proceedings of the 13th Conference on Computational Natural Language Learning* 39–47. Stroudsburg, PA, USA: Association for Computational Linguistics.

Blum M, Floyd RW, Pratt V, Rivest RL, Tarjan RE (1973) Time bounds for selection. *Journal of Computer and System Sciences*, 7 (4): 448–461.

Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (11): 1222–1239.

Breiman L (1996) Bagging predictors. *Machine Learning*, 24 (2): 123–140.

Breiman L (2001) Random forests. *Machine Learning*, 45 (1): 5–32.

Brits R, Engelbrecht AP, van den Bergh F (2002) A niching particle swarm optimizer. In: *Proceedings of the 4th Conference on Simulated Evolution And Learning* 692–696. Singapore: Nanyang Technological University.

Bruzzone L, Carlin L (2006) A multilevel context-based system for classification of very high spatial resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 44 (9): 2587–2600.

Bruzzone L, Marconcini M (2010) Domain adaptation problems: a DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (5): 770–787.

Campbell C, Cristianini N, Smola A (2000) Query learning with large margin classifiers. In: *ICML'00 Proceedings of the 17th International Conference on Machine Learning* 111–118. San Francisco, CA, USA: Morgan Kaufmann.

Campbell JB, Wynne RH (2012) *Introduction to remote sensing.* New York: Guilford Publications, 5th edition.

Carleer AP, Debeir O, Wolff E (2005) Assessment of very high spatial resolution satellite image segmentations. *Photogrammetric Engineering & Remote Sensing*, 71 (11): 1285–1294.

Cebron N (2008) *Aktives Lernen zur Klassifikation großer Datenmengen mittels Exploration und Spezialisierung.* PhD thesis, Universität Konstanz, Konstanz.

Cesa-Bianchi N, Gentile C, Vitale F, Zappella G (2010) Active learning on graphs via spanning trees. In: *NIPS'10 Proceedings of the Workshop On Networks Across Disciplines of the 23rd International Conference on Neural Information Processing Systems* 1–6. Red Hook, NY, USA: Curran Associates.

Chapelle O, Schölkopf B, Zien A (2010) *Semi-supervised learning.* Cambridge, Massachusetts: The MIT Press, 1st edition.

Cohn D, Atlas L, Richard L (1994) Improving generalization with active learning. *Machine Learning*, 15 (2): 201–221.

Colomina I, Molina P (2014) Unmanned aerial systems for photogrammetry and remote sensing: a review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92: 79–97.

Congalton RG (1991) A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, 37 (1): 35–46.

Cortes C, Vapnik VN (1995) Support-vector networks. *Machine Learning*, 20 (3): 273–297.

Cramer M (2010) The DGPF-test on digital airborne camera evaluation overview and test design. *PFG Photogrammetrie, Fernerkundung, Geoinformation*, 2010 (2): 73–82.

Dagan I, Engelson SP (1995) Committee-based sampling for training probabilistic classifiers. In: *ICML'95 Proceedings of the 12th International Conference on Machine Learning* 150–157. San Francisco, CA, USA: Morgan Kaufmann.

Dasgupta S (2005) Coarse sample complexity bounds for active learning. In: *NIPS'05 Proceedings of the 18th International Conference on Neural Information Processing Systems* 235–242. Cambridge, MA, USA: MIT Press.

Dasgupta S, Hsu DJ (2008) Hierarchical sampling for active learning. In: *ICML'08 Proceedings of the 25th international conference on Machine learning* 208–215. New York, NY, USA: Association for Computing Machinery.

Dean AM, Smith GM (2003) An evaluation of per-parcel land cover mapping using maximum likelihood class probabilities. *International Journal of Remote Sensing*, 24 (14): 2905–2920.

Demir B, Bovolo F, Bruzzone L (2012) Updating land-cover maps by classification of image time series: a novel change-detection-driven transfer learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 51 (1): 300–312.

Demir B, Minello L, Bruzzone L (2014) Definition of effective training sets for supervised classification of remote sensing images by a novel cost-sensitive active learning method. *IEEE Transactions on Geoscience and Remote Sensing*, 52 (2): 1272–1284.

Demir B, Persello C, Bruzzone L (2011) Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49 (3): 1014–1031.

Dey V, Zhang Y, Zhong M (2010) A review on image segmentation techniques with remote sensing perspective. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (7A): 31–42.

Dheeru D, Karra Taniskidou E (2017) *UCI Machine Learning Repository.* Irvine, CA, USA: University of California, Irvine, School of Information and Computer Sciences.

Di Gregorio A, Jansen LJM (2000) *Land cover classification system (LCCS) : classification concepts and user manual*. Rome: Food and Agriculture Organization of the United Nations.

Durbha SS, King RL, Younan NH (2011) Evaluating transfer learning approaches for image information mining applications. In: *IGARSS'11 Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* 1457–1460. Piscataway, NJ, USA: IEEE.

Ester M, Kriegel HP, Sander J (1999) Knowledge discovery in spatial databases. In: *KI'99 Proceedings of the 23rd Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence* 61–74. Berlin: Springer.

Estivill-Castro V (2002) Why so many clustering algorithms. *ACM SIGKDD Explorations Newsletter*, 4 (1): 65–75.

Fedorov V (1972) *Theory of optimal experiments*. New York, NY, USA: Academic Press, 1st edition.

Foody GM, COX DP (1994) Sub-pixel land cover composition estimation using a linear mixture model and fuzzy membership functions. *International Journal of Remote Sensing*, 15 (3): 619–631.

Freund Y, Schapire RE (1995) A desicion-theoretic generalization of on-line learning and an application to boosting. In: Goos G, Hartmanis J, Leeuwen J, Carbonell JG, Siekmann J, Vitányi P (eds) *EuroCOLT'95 European Conference on Computational Learning Theory*, volume 904 of *Lecture Notes in Computer Science* 23–37. Berlin: Springer.

Fukunaga K, Hostetler L (1975) The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21 (1): 32–40.

Gao BC (1995) Normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment*, 2480 (58): 2257–2266.

Geneletti D, Gorte BGH (2003) A method for object-oriented land cover classification combining Landsat TM data and aerial photographs. *International Journal of Remote Sensing*, 24 (6): 1273–1286.

Gjertsen A (2007) Accuracy of forest mapping based on Landsat TM data and a kNN-based method. *Remote Sensing of Environment*, 110 (4): 420–430.

Gong P, Howarth PJ (1990) The use of structural information for improving land-cover classification accuracies at the rural-urban fringe. *Photogrammetric Engineering & Remote Sensing*, 56 (1): 67–73.

Gross W, Borchardt S, Middelmann W (2013) Evaluation of spectral unmixing using nonnegative matrix factorization on stationary hyperspectral sensor data of specifically prepared rock and mineral mixtures. In: Beyerer J, León FP, Längle T (eds) *OCM'13 - Optical Characterization of Materials - conference proceedings* 169–178. Karlsruhe: KIT Scientific Publishing.

Gross W, Wuttke S, Middelmann W (2015) Transformation of hyperspectral data to improve classification by mitigating nonlinear effects. In: *WHISPERS'15 Proceedings of the 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing* 1–4. Piscataway, NJ, USA: IEEE.

Hake G (2002) *Kartographie: Visualisierung raum-zeitlicher Informationen*. Berlin: De Gruyter, 8th edition.

Hanneke S (2007) A bound on the label complexity of agnostic active learning. In: *ICML'07 Proceedings of the 24th international conference on Machine learning* 353–360. New York, NY, USA: Association for Computing Machinery.

Hanneke S (2014) *Theory of disagreement-based active learning*. Now Foundations and Trends, 1st edition.

Har-Peled S, Sadri B (2005) How fast is the k-means method? *Algorithmica*, 41 (3): 185–202.

Haraké L, Schilling H, Blohm C, Hillemann M, Lenz A, Becker M, Keskin G, Middelmann W (2016) Concept for an airborne real-time ISR system with multi-sensor 3D data acquisition. In: *SPIE'16 Proceedings of the conference on Electro-Optical and Infrared Systems: Technology and Applications XIII* 9987–9985. Bellingham, WA, USA: SPIE.

Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28 (1): 100–108.

Hartung J, Elpelt B (1984) *Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik.* München: Oldenbourg.

Harvey NR, Theiler J, Brumby SP, Perkins S, Szymanski JJ, Bloch JJ, Porter RB, Galassi M, Young AC (2002) Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 40 (2): 393–404.

Hasanzadeh M, Kasaei S (2008) Fuzzy image segmentation using membership connectedness. *EURASIP Journal on Advances in Signal Processing*, 2008 (1): 1–13.

Hasanzadeh M, Kasaei S (2010) A multispectral image segmentation method using size-weighted fuzzy clustering and membership connectedness. *IEEE Geoscience and Remote Sensing Letters*, 7 (3): 520–524.

Hazewinkel M (2002) *Encyclopaedia of mathematics.* Berlin: Springer, 1st edition.

Hird J, Montaghi A, McDermid G, Kariyeva J, Moorman B, Nielsen S, McIntosh A (2017) Use of unmanned aerial vehicles for monitoring recovery of forest vegetation on petroleum well sites. *Remote Sensing*, 9 (5): 413–432.

Hirschmuller H (2008) Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 (2): 328–341.

Hu W, Huang Y, Wei L, Zhang F, Li H (2015) Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015 (1): 1–12.

Huang C, Davis LS, Townshend JRG (2002) An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23 (4): 725–749.

Huo LZ, Tang P, Zhang Z, Tuia D (2015) Semisupervised classification of remote sensing images with hierarchical spatial similarity. *IEEE Geoscience and Remote Sensing Letters*, 12 (1): 150–154.

International Organization for Standardization (2008) *Colorimetry – part 4: CIE 1976 L*a*b* colour space.* ISO 11664-4:2008 (CIE S 014-4/E:2007), Technical report.

Jain AK (1989) *Fundamentals of digital image processing.* Englewood Cliffs, N.J., USA: Prentice-Hall, 17th edition.

Jain AK, Dubes RC (1988) *Algorithms for clustering data.* Englewood Cliffs, N.J., USA: Prentice-Hall.

Kääriäinen M (2006) Active learning in the non-realizable case. In: *ALT'06 Proceedings of the 17th international conference on Algorithmic Learning Theory*, volume 4264 of *Lecture Notes in Computer Science* 63–77. Berlin: Springer.

Kaiser P, Wegner JD, Lucchi A, Jaggi M, Hofmann T, Schindler K (2017) Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55 (11): 6054–6068.

Kashef R, Kamel MS (2009) Enhanced bisecting k-means clustering using intermediate cooperation. *Pattern Recognition*, 42 (11): 2557–2569.

Kavzoglu T, Mather PM (2003) The use of backpropagating artificial neural networks in land cover classification. *International Journal of Remote Sensing*, 24 (23): 4907–4938.

Keshava N (2004) Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries. *IEEE Transactions on Geoscience and Remote Sensing*, 42 (7): 1552–1565.

Kindermann R, Snell JL (1980) *Markov random fields and their applications*, volume 1. Providence, Rhode Island: American Mathematical Society.

King RD, Rowland J, Oliver SG, Young M, Aubrey W, Byrne E, Liakata M, Markham M, Pir P, Soldatova LN, Sparkes A, Whelan KE, Clare A (2009) The automation of science. *Science*, 324 (5923): 85–89.

King RD, Whelan KE, Jones FM, Reiser PGK, Bryant CH, Muggleton SH, Kell DB, Oliver SG (2004) Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427 (6971): 247–252.

Klausmann P, Fries S, Willersinn D, Stilla U, Thoennessen U (1999) Application-oriented assessment of computer vision algorithms. In: Jähne B (ed) *Handbook of computer vision and applications*, volume 3chapter 7, 133–152. San Diego, Calif.: Academic Press.

Knuth DE (1976) Big omicron and big omega and big theta. *ACM SIGACT News*, 8 (2): 18–24.

Kolmogorov V (2006) Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (10): 1568–1583.

Kottke D, Krempl G, Lang D, Teschner J, Spiliopoulou M (2016) Multi-class probabilistic active learning. In: *ECAI'16 Proceedings of the 22nd European Conference on Artificial Intelligence* 586–594. Amsterdam: IOS Press.

Krempl G, Kottke D, Lemaire V (2015) Optimised probabilistic active learning (OPAL). *Machine Learning*, 100 (2-3): 449–476.

Krempl G, Kottke D, Spiliopoulou M (2014a) Probabilistic active learning: a short proposition. In: *ECAI'14 Proceedings of the 21st European Conference on Artificial Intelligence* 1049–1050. Amsterdam: IOS Press.

Krempl G, Kottke D, Spiliopoulou M (2014b) Probabilistic active learning: towards combining versatility, optimality and efficiency. In: *DS'14 Proceedings of the 17th International Conference on Discovery Science*, volume 8777 of *Lecture Notes in Computer Science* 168–179. Cham: Springer.

Krishnamurthy V (2002) Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE Transactions on Signal Processing*, 50 (6): 1382–1397.

Krishnanand KN, Ghose D (2005) Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In: *SIS'2005 Proceedings of the IEEE Swarm Intelligence Symposium* 84–91. IEEE.

Kruse FA, Lefkoff AB, Boardman JW, Heidebrecht KB, Shapiro AT, Barloon PJ, Goetz AFH (1993) The spectral image processing system (SIPS)-interactive visualization and analysis of imaging spectrometer data. *Remote Sensing of Environment*, 44 (2-3): 145–163.

Kuffer M, Pfeffer K, Sliuzas R (2016) Slums from space - 15 years of slum mapping using remote sensing. *Remote Sensing*, 8 (6): 455–483.

Kühn F, Oppermann K, Hörig B (2004) Hydrocarbon index - an algorithm for hyperspectral detection of hydrocarbons. *International Journal of Remote Sensing*, 25 (12): 2467–2473.

Kumar B, Dikshit O (2015) Integrating spectral and textural features for urban land cover classification with hyperspectral data. In: *JURSE'15 Proceedings of the Joint Urban Remote Sensing Event* 1–4. IEEE.

Kussul N, Lavreniuk M, Skakun S, Shelestov A (2017) Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14 (5): 778–782.

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature*, 521 (7553): 436–444.

Lee S, Crawford MM (2004) Hierarchical clustering approach for unsupervised image classification of hyperspectral data. In: *IGARSS'04 Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* 941–944. Piscataway, NJ, USA: IEEE.

Lee S, Crawford MM (2005) Unsupervised multistage image classification using hierarchical clustering with a bayesian similarity measure. *IEEE Transactions on Image Processing*, 14 (3): 312–320.

Lenz A, Schilling H, Perpeet D, Wuttke S, Gross W, Middelmann W (2014) Automatic in-flight boresight calibration considering topography for hyperspectral pushbroom sensors. In: *IGARSS'14 Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* 2981–2984. Piscataway, NJ, USA: IEEE.

Lewis DD, Catlett J (1994) Heterogeneous uncertainty sampling for supervised learning. In: *ICML'94 Proceedings of the 11th International Conference on Machine Learning* 148–156. San Francisco, CA, USA: Morgan Kaufmann.

Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: *SIGIR'94 Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* 3–12. New York, NY, USA: Springer.

Lloyd SP (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2): 129–137.

Long C, Hua G (2015) Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In: *ICCV'15 Proceedings of the 15th IEEE International Conference on Computer Vision* 2839–2847. Piscataway, NJ, USA: IEEE.

Lu L, Huang Y, Di L, Hang D (2017) A new spatial attraction model for improving subpixel land cover classification. *Remote Sensing*, 9 (4): 360–374.

Ma L, Crawford MM, Tian J (2010) Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48 (11): 4099–4109.

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* 281–297. Berkeley, CA, USA: University of California Press.

Mahalanobis PC (1936) On the generalised distance in statistics. *Proceedings of the National Institute of Science of India*, 12: 49–55.

Makantasis K, Karantzalos K, Doulamis A, Doulamis N (2015) Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: *IGARSS'15 Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* 4959–4962. Piscataway, NJ, USA: IEEE.

Maltz D, Ehrlich K (1995) Pointing the way. In: *CHI'95 Proceedings of the SIGCHI Conference on Human factors in computing systems* 202–209. New York, NY, USA: Association for Computing Machinery.

Marcal ARS, Castro L (2005) Hierarchical clustering of multispectral images using combined spectral and spatial criteria. *IEEE Geoscience and Remote Sensing Letters*, 2 (1): 59–63.

Marmanis D, Datcu M, Esch T, Stilla U (2016) Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13 (1): 105–109.

Maulik U, Saha I (2010) Automatic fuzzy clustering using modified differential evolution for image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48 (9): 3503–3510.

McCallum A, Nigam K (1998) Employing EM in pool-based active learning for text classification. In: *ICML'98 Proceedings of the 15th International Conference on Machine learning* 350–358. San Francisco, CA, USA: Morgan Kaufmann.

Michaelsen E, Muench D, Arens M (2016) Searching remotely sensed images for meaningful nested gestalten. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41 (B3): 899–903.

Mohan M, Pathan SK, Narendrareddy K, Kandya A, Pandey S (2011) Dynamics of urbanization and its impact on land-use/land-cover. *Journal of Environmental Protection*, 2 (9): 1274–1283.

Montgomery DC (2013) *Design and analysis of experiments*. Hoboken, NJ: Wiley, 8th edition.

Mu X, Hu M, Song W, Ruan G, Ge Y, Wang J, Huang S, Yan G (2015) Evaluation of sampling methods for validation of remotely sensed fractional vegetation cover. *Remote Sensing*, 7 (12): 16164–16182.

Muñoz-Marí J, Tuia D, Camps-Valls G (2012) Semisupervised classification of remote sensing images with active queries. *IEEE Transactions on Geoscience and Remote Sensing*, 50 (10): 3751–3763.

Myint SW, Gober P, Brazel A, Grossman-Clarke S, Weng Q (2011) Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sensing of Environment*, 115 (5): 1145–1161.

Ngo LT, Mai DS, Nguyen MU (2012) GPU-based acceleration of interval type-2 fuzzy c-means clustering for satellite imagery land-cover classification. In: *ISDA'12 Proceedings of the 12th International Conference on Intelligent Systems Design and Applications* 992–997. Piscataway, NJ, USA: IEEE.

Nicodemus FE (1965) Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 4 (7): 767–775.

Okujeni A, van der Linden S, Jakimow B, Rabe A, Verrelst J, Hostert P (2014) A comparison of advanced regression algorithms for quantifying urban land cover. *Remote Sensing*, 6 (7): 6324–6346.

Olsson F, Tomanek K (2009) An intrinsic stopping criterion for committee-based active learning. In: *CoNLL'09 Proceedings of the 13th Conference on Computational Natural Language Learning* 138–146. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pal SK, Ghosh A, Shankar BU (2000) Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation. *International Journal of Remote Sensing*, 21 (11): 2269–2300.

Pan SJ, Yang Q (2010) A Survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22 (10): 1345–1359.

Patel VM, Gopalan R, Li R, Chellappa R (2015) Visual domain adaptation: a survey of recent advances. *IEEE Signal Processing Magazine*, 32 (3): 53–69.

Paul A, Rottensteiner F, Heipke C (2016) Iterative re-weighted instance transfer for domain adaptation. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3 (3): 339–346.

Rahman H, Verstraete MM, Pinty B (1993) Coupled surface-atmosphere reflectance (CSAR) model: 1. model description and inversion on synthetic data. *Journal of Geophysical Research*, 98 (D11): 20779–20789.

Robila SA (2005) Using spectral distances for speedup in hyperspectral image processing. *International Journal of Remote Sensing*, 26 (24): 5629–5650.

Rodríguez-Galiano VF, Abarca-Hernández F, Ghimire B, Chica-Olmo M, Atkinson PM, Jeganathan C (2011) Incorporating spatial variability measures in land-cover classification using random forest. *Procedia Environmental Sciences*, 3: 44–49.

Rodríguez-Galiano VF, Ghimire B, Rogan J, Chica-Olmo M, Rigol-Sanchez JP (2012) An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67: 93–104.

Rouse JW, Haas RH, Schell JA, Deering DW (1974) Monitoring vegetation systems in the Great Plains with ERTS. In: *Proceedings of the 3rd Earth Resources Technology Satellite-1 Symposium. Volume 1: Technical Presentations, section A* 309–317. Washington, DC, USA: NASA.

Roy N, McCallum A (2001) Toward optimal active learning through Monte Carlo estimation of error reduction. In: *ICML'01 Proceedings of the 18th International Conference on Machine learning* 441–448. San Francisco, CA, USA: Morgan Kaufmann.

Rüdiger C, Calvet JC, Gruhier C, Holmes TRH, Jeu RAM, Wagner W (2009) An intercomparison of ERS-Scat and AMSR-E soil moisture observations with model simulations over France. *Journal of Hydrometeorology*, 10 (2): 431–447.

Schilling H, Lenz A, Gross W, Perpeet D, Wuttke S, Middelmann W (2013) Concept and integration of an on-line quasi-operational airborne hyperspectral remote sensing system. In: *SPIE'13 Proceedings of the Electro-Optical Remote Sensing, Photonic Technologies, and Applications VII; and Military Applications in Hyperspectral Imaging and High Spatial Resolution Sensing* 8897–8906. Bellingham, WA, USA: SPIE.

Schindler K (2012) An overview and comparison of smooth labeling methods for land-cover classification. *IEEE Transactions on Geoscience and Remote Sensing*, 50 (11): 4534–4545.

Schohn G, Cohn D (2000) Less is more: active learning with support vector machines. In: *ICML'00 Proceedings of the 17th International Conference on Machine Learning* 839–846. San Francisco, CA, USA: Morgan Kaufmann.

Schölkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. *Neural Computation*, 12 (5): 1207–1245.

Schreier G, Dech S, Diedrich E, Maass H, Mikusch E (2008) Earth observation data payload ground segments at DLR for GMES. *Acta Astronautica*, 63 (1-4): 146–155.

Schwarz G (1978) Estimating the dimension of a model. *The Annals of Statistics*, 6 (2): 461–464.

Senthilnath J, Omkar SN, Mani V, Tejovanth N, Diwakar PG, Shenoy AB (2012) Hierarchical clustering algorithm for land cover mapping using satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5 (3): 762–768.

Settles B (2009) *Active learning literature survey.* Madison: University of Wisconsin, Technical report.

Settles B (2012) *Active learning.* Morgan & Claypool Publishers.

Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: *EMNLP'08 Proceedings of the Conference on Empirical Methods in Natural Language Processing* 1070–1079. Stroudsburg, PA, USA: Association for Computational Linguistics.

Settles B, Craven M, Ray S (2008) Multiple-instance active learning. In: *NIPS'07 Proceedings of the 21st International Conference on Neural Information Processing Systems* 1289–1296. Red Hook, NY, USA: Curran Associates.

Seung HS, Opper M, Sompolinsky H (1992) Query by committee. In: *COLT'92 Proceedings of the 5th Annual Workshop on Computational Learning Theory* 287–294. New York, NY, USA: Association for Computing Machinery.

Shahtahmassebi A, Yang N, Wang K, Moore N, Shen Z (2013) Review of shadow detection and de-shadowing methods in remote sensing. *Chinese Geographical Science*, 23 (4): 403–420.

Souvannavong F, Merialdo B, Huet B (2005) Partition sampling: an active learning selection strategy for large database annotation. *IEE Proceedings - Vision, Image, and Signal Processing*, 152 (3): 347–355.

Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: *KDD'00 Proceedings of the Workshop on Text Mining at the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 599–627. New York, NY, USA: Association for Computing Machinery.

Stephenne N, Beaumont B, Hallot E, Lenartz F, Lefebre F, Lauwaet D, Poelmans L, Wolff E (2017) Exposure and vulnerability geospatial analysis using earth observation data in the city of Liege, Belgium. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4 (1/W1): 149–156.

Stuckens J, Coppin PR, Bauer ME (2000) Integrating contextual information with per-pixel classification for improved land cover classification. *Remote Sensing of Environment*, 71 (3): 282–296.

Student (1908) The probable error of a mean. *Biometrika*, 6 (1): 1–25.

Taubenböck H, Esch T, Felbier A, Wiesner M, Roth A, Dech S (2012) Monitoring urbanization in mega cities from space. *Remote Sensing of Environment*, 117: 162–176.

Tong S, Koller D (2000a) Active learning for parameter estimation in bayesian networks. In: *NIPS'00 Proceedings of the 13th International Conference on Neural Information Processing Systems* 626–632. Cambridge, MA, USA: MIT Press.

Tong S, Koller D (2000b) Support vector machine active learning with applications to text classification. In: *ICML'00 Proceedings of the 17th International Conference on Machine Learning* 287–295. San Francisco, CA, USA: Morgan Kaufmann.

Tran T, Julian J, Beurs K (2014) Land cover heterogeneity effects on sub-pixel and per-pixel classifications. *ISPRS International Journal of Geo-Information*, 3 (2): 540–553.

Tuia D, Muñoz-Marí J, Camps-Valls G (2012) Remote sensing image segmentation by active queries. *Pattern Recognition*, 45 (6): 2180–2192.

Tuia D, Persello C, Bruzzone L (2016) Domain adaptation for the classification of remote sensing data. *IEEE Geoscience and Remote Sensing Magazine*, 4 (2): 41–57.

Tuia D, Volpi M, Copa L, Kanevski MF, Munoz-Mari J (2011) A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5 (3): 606–617.

Vert JP, Tsuda K, Schölkopf B (2004) A Primer on Kernel Methods. In: Schölkopf B, Tsuda K, Vert JP (eds) *Kernel methods in computational biology* 35–70. Cambridge: MIT Press.

Vlachos A (2008) A stopping criterion for active learning. *Computer Speech & Language*, 22 (3): 295–312.

Volpi M, Tuia D (2017) Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55 (2): 881–893.

Wagner W, Fröhlich J, Wotawa G, Stowasser R, Staudinger M, Hoffmann C, Walli A, Federspiel C, Aspetsberger M, Atzberger C, Briese C, Notarnicola C, Zebisch M, Boresch A, Enenkel M, Kidd R, Beringe A, Hasenauer S, Naeimi V, Mücke W (2014) Addressing grand challenges in earth observation science. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2 (7): 81–88.

Wainwright MJ, Jaakkola TS, Willsky AS (2005) MAP estimation via agreement on trees. *IEEE Transactions on Information Theory*, 51 (11): 3697–3717.

Walker JS, Blaschke T (2008) Object-based land-cover classification for the Phoenix metropolitan area: optimization vs. transportability. *International Journal of Remote Sensing*, 29 (7): 2021–2040.

Ward JH (1963) Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58 (301): 236–244.

Weng Q (2012) Remote sensing of impervious surfaces in the urban areas: requirements, methods, and trends. *Remote Sensing of Environment*, 117: 34–49.

Werninghaus R, Buckreuss S (2010) The TerraSAR-X mission and system design. *IEEE Transactions on Geoscience and Remote Sensing*, 48 (2): 606–614.

Wilcoxon F (1945) Individual comparisons by ranking methods. *Biometrics Bulletin*, 1 (6): 80–83.

Witten IH, Frank E, Hall MA, Pal CJ (2017) *Data mining*. Cambridge, MA: Morgan Kaufmann Publisher, 4th edition.

Wuttke S, Middelmann W, Stilla U (2014) Bewertung von Strategien des aktiven Lernens am Beispiel der Landbedeckungsklassifikation. In: *DGPF Tagungsband der 34. Wissenschaftlich-Technischen Jahrestagung der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation*, volume 23 1–10. Potsdam: Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF) e.V.

Wuttke S, Middelmann W, Stilla U (2015) Concept for a compound analysis in active learning for remote sensing. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40 (3/W2): 273–279.

Wuttke S, Middelmann W, Stilla U (2016) Active learning with svm for land cover classification - what can go wrong? In: *iKNOW'16 Proceedings of the Workshop on Active Learning: Applications, Foundations and Emerging Trends* 9–16. Aachen: CEUR Workshop Proceedings.

Wuttke S, Middelmann W, Stilla U (2017) Improving active queries with a local segmentation step and application to land cover classification. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4 (1/W1): 165–173.

Wuttke S, Middelmann W, Stilla U (2018) Improving the efficiency of land cover classification by combining segmentation, hierarchical clustering, and active learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP (99): 1–16.

Wuttke S, Schilling H, Middelmann W (2012) Reduction of training costs using active classification in fused hyperspectral and LiDAR data. In: *SPIE'12 Proceedings of the 18th Conference on Image and Signal Processing for Remote Sensing*, volume 8537 9–17. Bellingham, WA, USA: SPIE.

Xu Z, Akella R, Zhang Y (2007) Incorporating diversity and density in active learning for relevance feedback. In: *ECIR'07 Proceedings of the 29th European Conference on Information Retrieval*, volume 4425 of *Lecture Notes in Computer Science* 246–257. Berlin: Springer.

Zhang M, Tang J, Zhang X, Xue X (2014) Addressing cold start in recommender systems. In: *SIGIR'14 Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* 73–82. ACM.

Zhou W, Huang G, Troy A, Cadenasso ML (2009) Object-based land cover classification of shaded areas in high spatial resolution imagery of urban areas: a comparison study. *Remote Sensing of Environment*, 113 (8): 1769–1777.

Zhou Y, Wang YQ (2008) Extraction of impervious surface areas from high spatial resolution imagery by multiple agent segmentation and classification. *Photogrammetric Engineering & Remote Sensing*, 74 (7): 857–868.

Zhu J, Wang H, Tsou BK, Ma M (2010) Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on Audio, Speech, and Language Processing*, 18 (6): 1323–1331.

Zhu X (2008) *Semi-supervised learning literature survey.* Madison: University of Wisconsin, Technical report.

Zhu X, Lafferty J, Ghahramani Z (2003) Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: *ICML'03 Proceedings of the 20th International Conference on International Conference on Machine Learning* 912–919. Palo Alto, CA, USA: AAAI Press.

Zhu XX, Tuia D, Mou L, Xia GS, Zhang L, Xu F, Fraundorfer F (2017) Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5 (4): 8–36.

Lebenslauf

Name	Sebastian Wuttke
Geburtstag und -ort	15.01.1984 in Eisenach
Wohnort	Luise-Riegger-Str. 58 76137 Karlsruhe



Ausbildung/Tätigkeit

1990 – 1994	Staatliche Grundschule 3 in Erfurt
1994 – 2002	Gymnasialteil der Kooperativen Gesamtschule „Am Schwemmbach“ in Erfurt Abschluss: Abitur
2002 – 2003	Grundwehrdienst im 8. Sanitätsregiment 13 in Halle und im Standortsanitätszentrum Erfurt
2003 – 2009	Studium der Informatik an der Universität Karlsruhe (TH), Abschluss: Diplom-Informatiker
2009 – 2017	Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung IOSB in Ettlingen
2013 – 2018	Doktorand an der Technischen Universität München, Ingenieurfakultät Bau Geo Umwelt, Photogrammetrie und Fernerkundung
seit 2018	Software Consultant bei TNG Technology Consulting GmbH in Karlsruhe und München