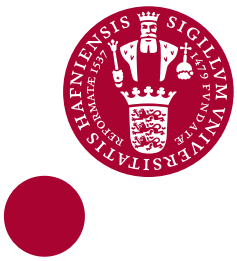


Transfer Learning for Image Segmentation by Combining Image Weighting and Kernel Learning

van Opbroek, Annegreet; Achterberg, Hakim C.; Vernooij, Meike W.; de Bruijne, Marleen

Published in:
IEEE Transactions on Medical Imaging

DOI:
10.1109/TMI.2018.2859478

Publication date:
2019

Document version
Peer reviewed version

Published as:
A. van Opbroek, H. C. Achterberg, M. W. Vernooij and M. de Bruijne, "Transfer Learning for Image Segmentation by Combining Image Weighting and Kernel Learning," in IEEE Transactions on Medical Imaging.
doi: 10.1109/TMI.2018.2859478

Transfer Learning for Image Segmentation by Combining Image Weighting and Kernel Learning

Annegreet van Opbroek, Hakim C. Achterberg, Meike W. Vernooij, Marleen de Bruijne

***Abstract*—Many medical image segmentation methods are based on supervised classification of voxels. Such methods generally perform well when provided with a training set that is representative of the test images to segment. However, problems may arise when training and test data follow different distributions, for example due to differences in scanners, scanning protocols, or patient groups. Under such conditions, weighting training images according to distribution similarity has been shown to greatly improve performance. However, this assumes that part of the training data is representative of the test data; it does not make unrepresentative data more similar.**

We therefore investigate kernel learning as a way to reduce differences between training and test data and explore the added value of kernel learning for image weighting. We also propose a new image weighting method that minimizes maximum mean discrepancy (MMD) between training and test data, which enables the joint optimization of image weights and kernel. Experiments on brain tissue, white matter lesion, and hippocampus segmentation show that both kernel learning and image weighting, when used separately, greatly improve performance on heterogeneous data. Here, MMD weighting obtains similar performance to previously proposed image weighting methods. Combining image weighting and kernel learning, optimized either individually or jointly, can give a small additional improvement in performance.

I. Introduction

The segmentation of biomedical images into the various tissues and structures forms a crucial step for both medical research and clinical practice. Automatic segmentation is important because manually segmenting three-dimensional images is very time consuming and prone to inter- and intra-observer variability. Many automatic segmentation methods are based on voxelwise supervised classification. Here, per voxel a decision is made as to the class it belongs to (which tissue or structure) by a classifier trained on a manually annotated training set. In brain MRI, such methods have for example been applied to whole-brain segmentation [19], [23], brain tissue segmentation [26], [9], [27], white matter lesion segmentation [3], [10], [14], and brain structure segmentation [12], [39], [27].

However, a disadvantage of supervised learning methods is a deterioration in performance in case of certain differences between training and test data, such as use of different scanners, imaging protocols, or differences in patient groups. Human observers can easily adapt to such differences. Supervised learning methods however, can struggle with differences between images, because they often result in a difference between the distributions of training and test samples in the feature space. Transfer learning[1] methods are designed to handle certain differences between training and test data, including differences in sample distributions [30]. Many transfer learning methods that have so far been presented in medical image segmentation are based on weighting training samples. This can be done by weighting individual samples (voxels) [40], [41], [15] or complete training images [42], [5]. Compared to sample weighting, image weighting has the advantage of requiring no labeled training data from the test scanner to handle differences between scanners.

These previously presented weighting methods only weight training samples *as is*, no steps are taken to reduce differences in the representation (the feature values) of training and test samples. An image weighting method can only either use a training image as is (give it a positive weight), or not use it (give it a weight of zero). We therefore propose to combine image weighting with a feature representation transfer step that 1) makes the data distributions more similar between training and test data and 2) separates the different classes as well as possible.

So far, few works have investigated explicit feature representation transfer for medical image segmentation. In machine learning and computer vision however, various methods have been developed. Many of them seek a linear transformation that minimizes distribution differences between training and test samples [30]. Since linear transformations are often not enough to overcome differences between datasets, transfor-

A. van Opbroek is with the Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology of the Erasmus MC - University Medical Center Rotterdam, 3000 CA Rotterdam, The Netherlands (e-mail: a.vanopbroek@erasmusmc.nl).

H.C. Achterberg is with the Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology of the Erasmus MC - University Medical Center Rotterdam.

M.W. Vernooij is with the Departments of Radiology and Epidemiology of the Erasmus MC - University Medical Center Rotterdam.

M. de Bruijne is with the Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology of the Erasmus MC - University Medical Center Rotterdam, and also with the Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen, Denmark. (e-mail: marleen.debruijne@erasmusmc.nl)

Part of the data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

[1] sometimes called *domain adaptation*, although some give different definitions for the two terms

mations are often performed in a high- (possibly infinitely-) dimensional kernel space (see for example the metric learning methods of [32], [25]). Alternatively, one may directly learn a kernel that reduces distribution differences. Pan et al. [29] for example, propose an unsupervised framework to learn a kernel that makes training and test distributions more similar by minimizing maximum mean discrepancy (MMD) between training and test samples. However, since the method is unsupervised, the learned kernel is not optimized for classification. Duan et al. [13] present a supervised method that learns a kernel that minimizes MMD between training and test samples while simultaneously minimizing a notion of classification error on some labeled test samples in the learned kernel space. Unfortunately, these two methods, as well as most kernel learning and metric learning methods developed so far, are computationally difficult to train on many samples (e.g. more than $1\,000$). In voxelwise classification however, many more samples are available for training and using them is likely to result in a better performance. The image weighting method of [42] for example, uses $50\,000$ samples *per image*.

In this paper, we propose two efficient kernel learning methods that can be used in conjunction with image weighting. Firstly, we discuss a multiple kernel learning (MKL) method that minimizes within-class distances and maximizes between-class distances. Secondly, we use this MKL method in the framework of Duan et al. [13], which adds an MMD term. This results in a kernel space that is suitable for classification and additionally makes training and test distributions more similar. Contrary to the methods of Pan et al. [29] and Duan et al. [13], the proposed methods can be trained on ten thousands of training samples per image. We investigate whether these kernel learning methods result in a better segmentation compared to using a standard Gaussian kernel. We also investigate whether these learned kernels improve performance when using image weighting. Further, we show that the MMD measure can be used to determine not only the kernel, but also the image weights. This way, image weights and kernel can be optimized jointly, which facilitates the optimization.

First, in Section II-B, we describe three image weighting methods: two weighting methods of [42], which minimize the Kullback-Leibler (KL) divergence and Bhatacharryya Distance (BD), and a new image weighting method that minimizes MMD. In Section II-C, we discuss the two MKL methods used: centered kernel alignment [7], and the MMD MKL framework of Duan et al. [13]. We also show in Section II-C2 how MMD image weighting and MMD MKL can be integrated in a joint optimization framework. We investigated the performance of image weighting, MKL, and the combination when training and testing on data from different datasets, which were acquired with different scanners and scanning protocols. Three medical image segmentation tasks were studied: brain tissue segmentation, white matter lesion segmentation, and hippocampus segmentation.

II. Methods

Image weighting is performed similar to Van Opbroek et al. [42], where each training image is given a weight based on minimizing a distance measure between the probability density functions (PDFs) of the training images and the test image to segment. Next, a training set is assembled based on the training images and their determined weights, by sampling voxels and their class labels according to the weight distribution of all training images. Consecutively, a support vector machine (SVM) classifier [8] is trained and test images are segmented by voxelwise classification. Kernel learning is performed for each test image individually and followed by classification with a kernel SVM [34] with the learned kernel.

A. Notation

We denote column vectors with bold small letters, e.g. $\boldsymbol{x}$, matrices with capital letters, e.g. X, and scalar values with small letters, e.g. m, θ. An exception is made for ϕ, which denotes a mapping into kernel space and K, which denotes a kernel ($K(\boldsymbol{x},\boldsymbol{y}) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{y})$) by means of the kernel trick). Indices of vectors and matrices are denoted with subscripts, e.g. $\boldsymbol{x}_i$, $M_{p,q}$. Superscripts are used for naming, e.g. n^{te} for the number of test samples and n^{tr} for the number of training samples per image.

A total of m training images are used, which may have different scanning characteristics. Every training image m_i provides n^{tr} randomly sampled training samples (voxels) $\boldsymbol{x}_j^{m_i}$ ($j = 1, 2, \ldots, n^{\text{tr}}$), where $\boldsymbol{x}_j^{m_i} \in \mathbb{R}^n$ denotes a vector containing a value for each of the n features. The $m \cdot n^{\text{tr}} \times n$ matrix of all training samples is denoted by X^{tr}. Each sample $\boldsymbol{x}_j^{m_i}$ has a label $y_j^{m_i} \in \mathbb{N}$. n^{te} test samples (voxels) $\boldsymbol{x}_j^{\text{te}}$, ($j = 1, 2, \ldots, n^{\text{te}}$) are acquired from the test image, for which we predict the label y_j^{te}. The $n^{\text{te}} \times n$ matrix of test samples is denoted with X^{te}. The training samples from image m_i follow the distribution $P_{m_i}(\boldsymbol{x})$ and the test samples follow distribution $P^{\text{te}}(\boldsymbol{x})$. Based on these distributions (or PDFs), m_i is given a weight w_{m_i}, resulting in a weight vector for all training images, $\boldsymbol{w} = [w_1, w_2, \ldots, w_m]^T$. These weights are non-negative and normalized such that $\sum_{m_i=1}^{m} w_{m_i} = |\boldsymbol{w}| = 1$.

B. Image Weighting

Giving each training image m_i a weight w_i results in a total training PDF that is a weighted sum of each of the individual training PDFs:

$$P^{\text{tr}}(\boldsymbol{x}) = \sum_{m_i}^{m} w_{m_i} P_{m_i}(\boldsymbol{x}). \tag{1}$$

The optimal weights $\boldsymbol{w}^*$ are chosen by minimizing a convex distance criterion between this total training PDF and the test PDF:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \text{DIST}_{\boldsymbol{w}}(P^{\text{tr}}, P^{\text{te}}). \tag{2}$$

1) PDF Weighting: The method presented by Van Opbroek et al. [42] first explicitly estimates the PDFs P_{m_i} (and, depending on the distance function, also P^{te}) by kernel density estimation and then uses a distance function on the estimated PDFs. These PDFs are approximated from the samples by

kernel density estimation with a Gaussian kernel, where the kernel parameter σ^S is determined with Silverman's rule [35]:

$$\sigma^S = \Big(\frac{4}{n+2}\Big)^{\frac{1}{n+4}} n^{\text{tr}\,\frac{-1}{n+4}} \sigma^{\text{tr}}. \tag{3}$$

Here, σ^{tr} equals the standard deviation of the training samples, averaged over all features. The chosen σ^S minimizes the mean integrated square error between the actual and the estimated PDF for a multivariate Gaussian kernel [35].

There are various methods to measure the distance between training and test PDFs. One of them, as presented in [42], is the Kullback-Leibler divergence (KL):

$$\begin{aligned} \text{KL}(P^{\text{te}}||P^{\text{tr}}) &= \int_{\mathcal{D}} P^{\text{te}}(\boldsymbol{x}) \log\Big(\frac{P^{\text{te}}(\boldsymbol{x})}{P^{\text{tr}}(\boldsymbol{x})}\Big) d\boldsymbol{x} && (4)\\ &\approx \frac{1}{n^{\text{te}}} \sum_{j=1}^{n^{\text{te}}} \log P^{\text{te}}(\boldsymbol{x}_j^{\text{te}}) && (5)\\ &\quad -\frac{1}{n^{\text{te}}} \sum_{j=1}^{n^{\text{te}}} \log\Big(\sum_{m_i=1}^{m} w_{m_i} P_{m_i}(\boldsymbol{x}_j^{\text{te}})\Big), \end{aligned}$$

where $\mathcal{D}$ is the domain of P^{te} and P^{tr}.

Note that the first term of Equation 5 does not depend on $\boldsymbol{w}$. So the optimal weights $\boldsymbol{w}$ are determined by maximizing $\sum_{j=1}^{n^{\text{te}}} \log\Big(\sum_{m_i=1}^{m} w_{m_i} P_{m_i}(\boldsymbol{x}_j^{\text{te}})\Big)$ under the constraints $\boldsymbol{w} \geq \mathbf{0}$ and $|\boldsymbol{w}| = 1$.

A second method to measure distances between PDFs that performed well in [42] is the Bhattacharyya distance (BD):

$$\begin{aligned} \text{BD}(P^{\text{te}}, P^{\text{tr}}) &= -\int_{\mathcal{D}} \sqrt{P^{\text{te}}(\boldsymbol{x}) P^{\text{tr}}(\boldsymbol{x})}\, d\boldsymbol{x} && (6)\\ &\approx -\frac{1}{n^{\text{tr}}} \sum_{\boldsymbol{x}_i \in \mathcal{D}} \sqrt{P^{\text{te}}(\boldsymbol{x}_i)} && (7)\\ &\quad \sqrt{\sum_{m_i=1}^{m} w_{m_i} P_{m_i}(\boldsymbol{x}_i)}, \end{aligned}$$

where the $\boldsymbol{x}_i$ are randomly drawn from $\mathcal{D}$. Similar to the KL, the criterion in Equation 7 is minimized for $\boldsymbol{w}$ subject to the constraints $\boldsymbol{w} \geq \mathbf{0}$ and $|\boldsymbol{w}| = 1$.

2) MMD Weighting: We propose a new image weighting method based on maximum mean discrepancy (MMD) minimization. This method can determine the image weights immediately from the samples and therefore does not require a PDF estimation step. MMD image weighting is similar to the MMD sample weighting method of Huang et al. [17], but weights groups of samples (images) instead of separate samples. The image weights $\boldsymbol{w}$ are determined by minimizing the MMD between all training samples X^{tr} and test samples X^{te}. The MMD between two datasets X and Y is defined as the distance between the means of the samples $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in Y$ in a kernel space ϕ:

$$\text{MMD}(X, Y) = ||\frac{1}{n^X} \sum_{i=1}^{n^X} \phi(\boldsymbol{x}_i) - \frac{1}{n^Y} \sum_{j=1}^{n^Y} \phi(\boldsymbol{y}_j)||^2, \tag{8}$$

where n^X and n^Y are the number of samples $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively.

The image weights are determined by minimizing the MMD between the training set X^{tr}, which are given a weight per image and the test set X^{te}:

$$\begin{aligned} &\text{MMD}(X^{\text{tr}}, X^{\text{te}}) = \\ &= ||\frac{1}{n^{\text{tr}}} \sum_{m_i=1}^{m} w_{m_i} \sum_{j=1}^{n^{\text{tr}}} \phi(\boldsymbol{x}_j^{m_i}) - \frac{1}{n^{\text{te}}} \sum_{i=1}^{n^{\text{te}}} \phi(\boldsymbol{x}_i^{\text{te}})||^2 \qquad (9)\\ &= \frac{1}{n^{\text{tr}2}} \boldsymbol{w}^T K^{\text{tr,tr}} \boldsymbol{w} - \frac{2}{n^{\text{tr}} n^{\text{te}}} \boldsymbol{w}^T \boldsymbol{k}^{\text{tr,te}} + \frac{1}{n^{\text{te}2}} k^{\text{te,te}}. \end{aligned}$$

Here, $K^{\text{tr,tr}}$ is an $m \times m$ matrix of inner products between training images, $\boldsymbol{k}^{\text{tr,te}}$ denotes a vector of length m of inner products between each of the training images and the test image, and $k^{\text{te,te}}$ is a single value giving the inner product of the test samples. Formulas for $K^{\text{tr,tr}}$, $\boldsymbol{k}^{\text{tr,te}}$, and $k^{\text{te,te}}$ are given in the supplementary materials, Equations 1-3 (supplementary materials are available in the supplementary files /multimedia tab).

Equation 9 can be written as

$$\text{MMD}(X^{\text{tr}}, X^{\text{te}}) = \boldsymbol{\omega}^T M \boldsymbol{\omega}, \tag{10}$$

where $\boldsymbol{\omega}^T = [\boldsymbol{w}^T, 1]$ and

$$M = \begin{bmatrix} \frac{1}{n^{\text{tr}2}} K^{\text{tr,tr}} & -\frac{1}{n^{\text{tr}} n^{\text{te}}} \boldsymbol{k}^{\text{tr,te}} \\ -\frac{1}{n^{\text{tr}} n^{\text{te}}} \boldsymbol{k}^{\text{tr,te}^T} & \frac{1}{n^{\text{te}2}} k^{\text{te,te}} \end{bmatrix}. \tag{11}$$

Given a kernel K, the optimal image weights can be found by minimizing Equation 10 for $\boldsymbol{\omega}$ under the constraints $\boldsymbol{w} \geq \mathbf{0}$, $|\boldsymbol{w}| = 1$.

C. Kernel Learning

We aim to find a kernel space that reduces differences between datasets. When searching for a kernel matrix, we would need to pose constraints to make sure that the found matrix is symmetric and positive semidefinite, in order for it to be a kernel (because of Mercer's theorem). A relatively easy way to find such a kernel matrix is with multiple kernel learning (MKL), where we start with a set of n^k pre-defined base kernels K_k^{b} ($k = 1, 2, \ldots, n^k$). The searched kernel is determined as the optimal linear combination of these base kernels:

$$K = \sum_{k=1}^{n^k} v_k K_k^{\text{b}}, \tag{12}$$

where the kernel weights $\boldsymbol{v} = [v_1, v_2, \ldots, v_{n_k}]^T \geq \mathbf{0}$. Since K is a linear combination of kernels, it is guaranteed to be a kernel as well.

We present two kernel learning methods: Centered Kernel Alignment [7] in Section II-C1 and the MMD kernel learning method that is similar to the method of Duan et al. [13] in Section II-C2. This method can be efficiently combined with MMD image weighting so that image and kernel weights can be jointly optimized.

1) Centered Kernel Alignment: Cortes et al. [7] presented a kernel learning method that optimizes the kernel weights $\boldsymbol{v}_k$ in Equation 12 by maximizing centered kernel alignment (CKA) between the learned kernel K and the *ideal kernel* K^{I}. The ideal kernel equals 1 if two samples have the same class label and -1 if they have different class labels. This corresponds to a situation in the kernel space ϕ where all samples with the same class label are at the same location and samples with different class labels are always a distance of 2 apart.[2]

First, the CKA method centers the distribution of samples in kernel space for all base kernels, which sets the expectation of the samples in the kernel space to zero. After obtaining the centered base kernels K_k^{bc} ($k = 1, 2, \ldots, n^k$), the weights $\boldsymbol{v}$ are chosen as to maximize the alignment between K and K^{I}:

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}} \frac{\langle \sum_{k=1}^{n^k} v_k K_k^{\mathrm{bc}}, K^{\mathrm{I}} \rangle_F}{|| \sum_{k=1}^{n^k} v_k K_k^{\mathrm{bc}} ||_F}, \tag{13}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius product and $||\cdot||_F$ denotes the Frobenius norm[3]. The expression in Equation 13 can be seen as the cosine of the angle between the learned kernel K and the ideal kernel K^{I} and equals one if and only if the two kernels are the same [16].

2) MMD Kernel Learning: Duan et al. [13] proposes to find a kernel K that minimizes 1) the distance between a (possibly weighted) training distribution and a test distribution, so that in the learned kernel space training samples appear similar to test samples, 2) some notion of classification error on the training samples, so that the learned kernel is suitable for classification. This results in the following optimization criterion:

$$K^* = \arg\min_{K} \theta \, \mathrm{DIST}_K(P^{\mathrm{tr}}, P^{\mathrm{te}}) + f_K(X^{\mathrm{tr}}, \boldsymbol{y}^{\mathrm{tr}}), \tag{14}$$

where DIST_K is a PDF distance function in the kernel space of K, f_K measures classification error in the kernel space of K, and θ is a trade-off parameter between the two terms.

For f_K, Duan et al. [13] uses either structural risk functional (used in support vector regression) or the Hinge loss (used in SVM), both of which are computationally very expensive if many training samples are used. CKA on the other hand, can be calculated very efficiently for a large number of training samples that consist of weighted subsets. We therefore use the CKA value, multiplied by -1, since CKA is maximized but Equation 14 is minimized. Similar to Duan et al. [13], we use the MMD as the distance function DIST_K in Equation 14.

From Equation 10 we can see how the MMD between training and test distributions can be calculated when the training images are weighted (by setting $\boldsymbol{\omega} = [\boldsymbol{w}^T, 1]^T$) or when no image weights are used (by setting $\boldsymbol{\omega} = [\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}, 1]^T$). Since the searched kernel K consists of a linear combination of base kernels, as in Equation 12, we construct a matrix M per base kernel, M_k, so that

$$\mathrm{MMD}(X^{\mathrm{tr}}, X^{\mathrm{te}}) = \sum_{k=1}^{n^k} v_k \boldsymbol{\omega}^T M_k \boldsymbol{\omega}. \tag{15}$$

When combining MMD kernel learning and MMD image weighting in this framework, we optimize

$$\begin{aligned} [\boldsymbol{w}^*, \boldsymbol{v}^*] &= \arg\min_{\boldsymbol{w},\boldsymbol{v}} \theta \, \mathrm{MMD}_{\boldsymbol{w},\boldsymbol{v}}(P^{\mathrm{tr}}, P^{\mathrm{te}}) \\ &\quad - \mathrm{CKA}_{\boldsymbol{v}}(X^{\mathrm{tr}}, \boldsymbol{y}^{\mathrm{tr}}) \qquad (16) \\ &= \arg\min_{\boldsymbol{w},\boldsymbol{v}} \theta \sum_{k=1}^{n^k} v_k \begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix}^T M_k \begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix} \\ &\quad - \frac{\langle \sum_{k=1}^{n^k} v_k K_k^{\mathrm{bc}}, K^{\mathrm{I}} \rangle_F}{|| \sum_{k=1}^{n^k} v_k K_k^{\mathrm{bc}} ||_F}. \qquad (17) \end{aligned}$$

Equation 17 follows from Equation 16 by plugging in the MMD definition from Equation 15 and the CKA definition from Equation 13.

For MMD kernel learning without MMD image weighting, Equation 17 is optimized for the kernel weights $\boldsymbol{v}$ only.

In the supplementary materials we discuss three issues that came up in the implementation and how these were solved (supplementary materials are available in the supplementary files /multimedia tab). The first considers numerical precision in calculation of the MMD, the second considers kernel normalization (which prevents the MKL from favoring kernels that shrink the feature space), and the third presents an efficient way to find the optimal value for θ.

III. Experiments

We performed experiments on voxelwise classification on three MRI brain segmentation tasks: brain tissue segmentation, white matter lesion (WML) segmentation, and hippocampus segmentation. For brain tissue segmentation, each voxel inside a manually annotated brain mask was classified as either white matter (WM), gray matter (GM), or cerebrospinal fluid (CSF). For WML segmentation, each voxel inside an automatically generated brain mask was classified as either WML or non-WML. For hippocampus segmentation, a region of interest (ROI) around the hippocampus was determined using multi atlas registration. Inside this ROI, every voxel was classified as hippocampus or non hippocampus. Classifications were performed with an SVM classifier on intensity features. In an extra experiment, we compared SVM performance with that of a random forest (RF) classifier.

For all three applications, we used data from different datasets, which were acquired with different scanners and scanning parameters. Each image was segmented once in leave-one-dataset-out cross validation, where the training data consisted of all images from different datasets than the test image.

This section describes the data, used features, and experimental setup.

[2] This can be seen by calculating $\mathrm{dist}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{||\phi(\boldsymbol{x_i}) - \phi(\boldsymbol{x_j})||^2} = \sqrt{K_{i,i} - 2K_{i,j} + K_{j,j}}$.

[3] $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_F = \mathrm{Trace}(\boldsymbol{x}^T \boldsymbol{y})$ and $||\boldsymbol{x}||_F = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_F}$

Table I
OVERVIEW OF THE DATASETS USED FOR BRAIN TISSUE (BT), WML, AND HIPPOCAMPUS (HC) SEGMENTATION. FOR HIPPOCAMPUS SEGMENTATION, A TOTAL OF 135 SCANS WERE USED, SEPARATED INTO 46 DATASETS OF IMAGES THAT WERE SCANNED WITH THE SAME SCANNER.

Dataset	Source	# images	Sequences	Field Strength	Scanner	Voxel size (mm^3)
BT1	RSS [20]	6	T1	1.5T	GE	$0.49 \times 0.49 \times 0.80$
BT2	RSS [21]	12	HASTE-Odd	1.5T	Siemens	$1.25 \times 1 \times 1$
BT3	MRBrainS Challenge [26]	5	T1	3T	Philips	$0.958 \times 0.958 \times 3.0$
BT4	IBSR [44]	18	T1	1.5T	Unknown	$0.84 \times 0.84 \times 1.5$ to $1 \times 1 \times 1.5$
BT5	IBSR [44]	20	T1	1.5T	10x Siemens, 10x GE	$1 \times 3.1 \times 1$
WML1	RSS [20]	20	T1,PD,FLAIR	1.5T	GE	$0.49 \times 0.49 \times 0.80$
WML2	MS Lesion Challenge [37]	20	T1,T2,FLAIR	3T	Siemens	$0.5 \times 0.5 \times 0.5$
WML3	MS Lesion Challenge [37]	20	T1,T2,FLAIR	3T	Siemens Allegra	$0.5 \times 0.5 \times 0.5$
HC1-27	ADNI	1 to 10	T1	1.5T	Various	$1 \times 1 \times 1$
HC28-46	ADNI	1 to 10	T1	3T	Various	$1 \times 1 \times 1$

A. Data

All data used for the experiments is summarized in Table I.

For brain tissue segmentation (BT), we used a total of 61 images from 5 datasets with corresponding manual segmentations. Two datasets from the Rotterdam Scan Study (RSS) [21], [20], one from the MRBrainS Challenge [26], and two from the Internet Brain Segmentation Repository (IBSR) [44]. The images of datasets BT1, BT2, and BT3 were acquired with a single scanner, BT4 and BT5 were acquired with multiple scanners (but which images are from the same scanner is unknown). All images have a manual segmentation of the brain mask and tissues. Images from BT3 and BT4 have the cerebellum included in the brain mask, images from BT1, BT2, and BT5 do not. The images from BT2, which are Haste-Odd images, have inverted intensities compared to the T1 images from the other studies. Therefore, the voxel values in the images in BT2 were inverted prior to calculation of the features.

For WML segmentation, a total of 40 images with manual segmentations from 3 datasets were used. One from the RSS [20] and two from the MS Lesion Segmentation Challenge of MICCAI 2008 [37]. The brain extraction tool (BET) [36] was used to generate brain masks. For the calculation of features, we treated the PD images of WML1 to be the same modality as the T2 images of WML2 and WML3, since they are visually similar.

For the hippocampus experiments, we used MR images from the Harmonized Protocol (HarP)[4]. This dataset consists of 135 T1-weighted images of the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset [28][5] with manually annotated hippocampi [1]. These 135 images were scanned at 34 sites, with a total of 46 different scanners. We split these images up into datasets of images that were scanned with the same scanner. All images were rigidly registered to MNI space with $1 \times 1 \times 1$ mm^3 voxel size. We used the brain extraction tool (BET) [36] to generate brain masks.

[4]http://www.hippocampal-protocol.net/

[5]The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). For up-to-date information, see http://www.adni-info.org.

1) Preprocessing: All images were corrected for intensity non-uniformity with the N4 method [38]. Next, all image intensities were normalized by rescaling the 4-96th percentile to the 0-1 interval inside the brain mask.

B. Features

For each sequence and at each voxel, the following features were calculated:

- Intensity
- Intensity after convolution with a Gaussian kernel with varying σ_F
- The gradient magnitude of the intensity after convolution with a Gaussian kernel with varying σ_F
- The Laplacian of the intensity after convolution with a Gaussian kernel with varying σ_F
- Only for brain tissue: spatial features

For brain tissue segmentation $\sigma_F = 1, 2, 4$ mm^3 was chosen, for WML $\sigma_F = 0.5, 1, 2$ mm^3 was chosen because of the overall smaller voxel sizes, and for hippocampus segmentation $\sigma_F = 1, 2.2,$ and 5 mm^3 was chosen, similar to the hippocampus segmentation method of [39].

For brain tissue segmentation, spatial features were added. For each voxel, cylindrical coordinates (R, θ_F, and z) were calculated in the following way: first, every brain mask was scaled to $[-1, 1]$ in each direction and the origin O was set as the middle of the brain mask. Here, R equals the distance of the voxel to O, z indicates its position in the cranial-caudal direction, and θ_F equals the positive angle ($\in [0, \pi]$) between the line through the voxel and O and the anterior-posterior axis. For WML, no spatial features were used since lesions can appear at different locations between training and test images. For hippocampus segmentation, no spatial features were used because spatial information is encoded in the selected ROI.

This resulted in a total of 13 features for the brain tissue experiments, 30 features for the WML experiments, and 10 features for the hippocampus experiments. These features were used for both the kernel learning and the image weighting.

For all applications, all features were normalized to zero mean, unit variance per image. No feature reduction was used.

C. Experimental Setup

For each of the three applications, the images of each dataset were segmented by leave-one-dataset-out cross vali-

Table II
OVERVIEW OF COMPARED METHODS. "KERNEL" INDICATES THE USED KERNEL LEARNING METHOD (GAUSSIAN OR MKL WITH CKA MAXIMIZATION OR MMD MINIMIZATION). "WEIGHTING" INDICATES THE USED IMAGE WEIGHTING METHOD (EQUAL WEIGHTING OR WEIGHTING DETERMINED BY KL, BD, OR MMD MINIMIZATION). "OPTIMIZATION" INDICATES WHAT WAS OPTIMIZED (THE KERNEL K AND/OR THE IMAGE WEIGHTS $\boldsymbol{w}$) AND, FOR METHODS THAT USE BOTH KERNEL LEARNING AND IMAGE WEIGHTING, WHETHER THE OPTIMIZATION WAS PERFORMED SEPARATELY (SEP.) OR JOINTLY.

Cat.	Method	Kernel	Weighting	Optimization
0	K0 W0	Gaussian	Equal	-
W	K0 W-KL	Gaussian	KL	$\boldsymbol{w}$
W	K0 W-BD	Gaussian	BD	$\boldsymbol{w}$
W	K0 W-MMD	Gaussian	MMD	$\boldsymbol{w}$
K	K-CKA W0	CKA	Equal	K
K	K-MMD W0	MMD	Equal	K
K&W	K-CKA W-KL	CKA	KL	K, $\boldsymbol{w}$ sep.
K&W	K-CKA W-BD	CKA	BD	K, $\boldsymbol{w}$ sep.
K&W	K-CKA W-MMD	CKA	MMD	K, $\boldsymbol{w}$ sep.
K&W	KW-MMD	MMD	MMD	K, $\boldsymbol{w}$ jointly

dation: by training on all images from different datasets. For hippocampus segmentation, all voxels within an ROI around the hippocampus were classified as either hippocampus or non-hippocampus. For brain tissue and WML segmentation, voxels within the brain mask were classified as white matter, gray matter, CSF and as WML and non-WML respectively. The sampling of voxels in train and test data is described in more detail in Section III-C2.

1) Compared Methods: We compared the performance of 10 methods, which can be separated into four categories: Cat. 0, a baseline method that uses neither image weighting nor kernel learning; Cat. W, three image weighting methods, KL, BD, and MMD; Cat. K, two multiple kernel learning (MKL) methods, CKA, which corresponds to Equation 14 with $\theta = 0$ and MMD, which corresponds to Equation 14 with $\theta \neq 0$; Cat. K&W, four methods that combine MKL and image weighting: CKA combined with KL, BD, and MMD image weighting, and MMD kernel learning combined with MMD image weighting, where the kernel and weight optimization was performed jointly, as in Equation 16. All compared methods are summarized in Table II. When no MKL was used, a Gaussian kernel was used with kernel parameter γ_G determined with cross validation.

For MKL, we used a total of 60 base kernels, consisting of the 4 types of kernels used by Duan et al. [13]: Gaussian kernel, Laplacian kernel, inverse square distance kernel, and inverse distance kernel. For each kernel type, we used 15 different values for the kernel parameter $\gamma = 10\hat{}[-8, -7, \ldots, 5, 6]$.

Performance of all classifiers was measured in terms of classification error for the brain tissue and WML experiments. For the hippocampus experiments, performance was measured in Dice overlap [11]. In all three applications, significance of differences between two methods was measured with a paired two-tailed t-test with the significance threshold at $P = 0.05$.

For the hippocampus experiments, we applied separate classifiers for the left and right hippocampus, which outperformed a joint classifier. The performance on the hippocampus segmentation was averaged over left and right hippocampus.

For all SVM classifiers, we used an implementation in LibSVM [4]. The optimization of the kernel and image weights was performed with the interior-reflective Newton method [6]. After the optimization, kernel weights and image weights below 0.01 were set to zero.

We additionally performed experiments with a random forest (RF) classifier instead of an SVM, since RF is more commonly used in medical image segmentation than SVM. Similar to previous work on image weighting for medical image segmentation [5], we used 100 trees and otherwise default parameters. We used the Matlab *TreeBagger* implementation. Since RFs cannot handle kernels directly, we used Kernel PCA [33] for the experiments in Categories K and K&W. Here, the d principal components with the most variance were extracted from kernel space and used as features for the training and test datasets. We chose d equal to the number of features in the original data (13 for brain tissue, 30 for WML, and 10 for hippocampus).

2) Training and Test sets: For brain tissue segmentation, training and test sets were composed by uniform random selection of voxels within the brain mask.

For WML segmentation, only voxels with a normalized FLAIR intensity above 0.75 were selected for training and testing, since WMLs appear bright on the FLAIR images. The threshold of 0.75 was chosen in a trade-off to discard non-WML voxels while maintaining WML voxels. As a result, most CSF voxels and some GM voxels were excluded, while almost all WM and WML voxels were maintained. Next, WML and non-WML voxels were sampled disproportionally into the training and test sets, so that per dataset, 20% of the selected voxels consisted of WML voxels. This adaptation was chosen since the prior probability of a voxel being WML is so low (1.61% for WML1, 1.31% for WML2, and 0.22% for WML3) that classifiers would be likely to choose to classify all voxels as non-WML voxels. Contrary to the classification, the kernel learning and image weights were determined on an proportionally sampled dataset.

For hippocampus segmentation, all training images from different scanners were non-rigidly registered to the test image. To select test samples, the atlases of the training images were transformed accordingly and an ROI was constructed consisting of all voxels for which at least 10% of the atlases vote it to be hippocampus. Similarly, training samples were selected by non-rigidly registering all training images to each other and creating an ROI, of which the voxels were used as training voxels. The registrations were performed with the Elastix registration toolbox [24] by maximizing normalized mutual information, using the registration settings of [2].

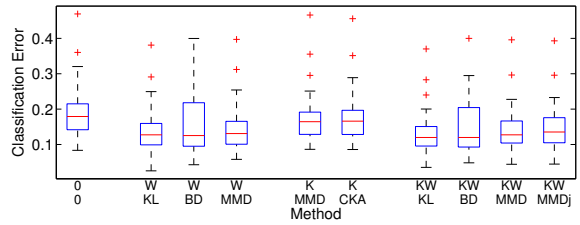
For all applications, the training sets for MKL and image weighting consisted of 1 000 randomly sampled voxels per image (for both the training images and the test image).

The training sets for the classifiers were constructed by first setting all weights $w < 0.01$ to zero, followed by randomly sampling 10 000 voxels from all training images together, according to the weight vector $\boldsymbol{w}$. For the unweighted classifiers, all training images were weighted equally. For hippocampus segmentation, the test set consisted of all voxels

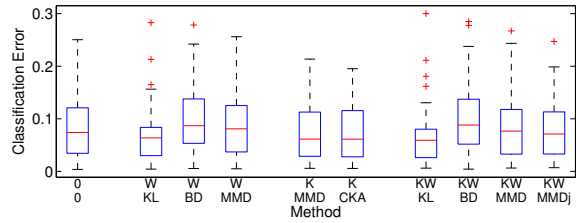
within the ROI. For brain tissue and WML segmentation, the test sets consisted of 10 000 randomly selected voxels per test image.

3) Parameter Settings: For the methods in category 0 and W, (no MKL), the kernel parameter γ_G and the SVM parameter C needed to be set. For the CKA kernel learning methods (both with and without image weighting), only the SVM parameter C was needed. For the MMD MKL methods (K-MMD W0 and KW-MMD) the MMD trade-off parameter θ and the SVM parameter C were needed. These parameters were optimized with leave-one-dataset-out grid search on all training images. So for every test dataset, the optimal parameters were found by classifying the voxels of each training image by a classifier trained on all other training images that were from different datasets. For computational reasons, all three parameters were tuned without image weighting.

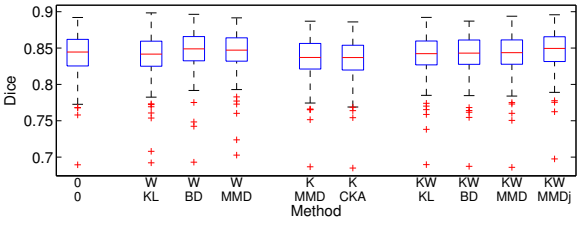
IV. Results



(a) Tissue Segmentation



(b) WML Segmentation



(c) Hippocampus Segmentation

Figure 1. Boxplot of the segmentation results in Table III. The x-axis shows respectively the method category and the method used for image weighting or kernel learning.(a) brain tissue segmentation, (b) WML segmentation, (c) hippocampus segmentation.

Table III shows the performance of all 10 methods on brain tissue, WML, and hippocampus segmentation by SVM classification. Figure 1 shows an accompanying boxplot of the results. For brain tissue and WML segmentation, a boxplot of the results per dataset can be found in the supplementary materials (supplementary materials are available in the supplementary files /multimedia tab). Note that for hippocampus segmentation, the differences between the methods were much smaller than for the other two applications. This is probably because the images are much more homogeneous between datasets than for the other two applications, since ADNI aims to acquire data at different sites with similar protocols.

Table III
Performance of all methods on the three applications with an SVM classifier, averaged over all images of all datasets. Performance on Tissue and WML segmentation is in classification error (%), performance on hippocampus segmentation is in Dice overlap (%). The best method and all methods that are not significantly worse, are in bold. $*$ indicates a significant difference between K-CKA W-MMD and KW-MMD.

Cat.	Method	Brain Tissue	WML	Hippocampus
0	K0 W0	19.03	8.40	84.0
W	K0 W-KL	13.36	**7.01**	83.8
W	K0 W-BD	14.99	9.84	**84.6**
W	K0 W-MMD	14.09	8.53	84.4
K	K-CKA W0	17.31	**7.58**	83.4
K	K-MMD W0	17.28	**7.62**	83.5
K&W	K-CKA W-KL	**12.86**	**6.84**	83.9
K&W	K-CKA W-BD	14.80	9.92	84.1
K&W	K-CKA W-MMD	13.95$*$	8.42	84.1
K&W	KW-MMD	14.32	8.13	**84.6**$*$

We will first discuss the results on image weighting (Cat. W), followed by kernel learning (Cat. K), and the combination of the two (Cat. K&W).

A. Image Weighting

For all three applications, the baseline method was significantly outperformed by a weighting method. Which weighting method performed best differed between applications. For WML segmentation, KL weighting performed considerably better than other weighting methods, as was also observed and discussed in [42]. This was likely caused by the very small percentage of WML voxels, which caused the weighting method to weight according to overall image similarity rather than WML similarity. Note that WML voxels have very high intensity in the FLAIR scan, while other voxels do not, so the WML voxels are located in a location with low $P(\boldsymbol{x})$. KL weighting probably outperforms BD and MMD weighting since it focuses more on parts of the distribution with a low $P(\boldsymbol{x})$, which is caused by the log in Equation 5 (differences between training and test distributions in locations where P_{m_i} is small contribute more to the KL than in locations where P_{m_i} is large). For brain tissue and hippocampus segmentation, the differences were smaller between the three weighting methods, where KL performed best for brain tissue segmentation and BD for hippocampus segmentation. MMD was second best in both applications.

Figure 2 shows per weighting method and per application the average distribution of weights over the training images. The average number of training images equals 46 for tissue, 15 for WML, and 130 for hippocampus segmentation. By all three methods, the majority of training images were given a weight of zero, i.e. were not used in the training of the classifier. The average number of non-zero image weights was about 9,

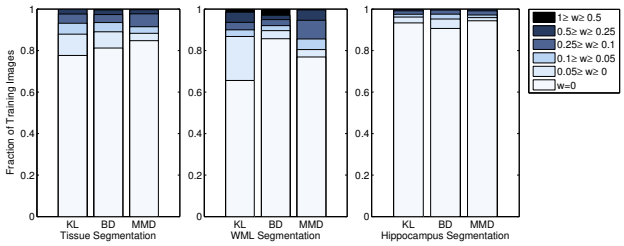


Figure 2. Average distribution of weights over all training images on each of the three applications. KL, BD, and MMD respectively give the weight distributions for weighting according to W-KL, W-BD, and W-MMD.

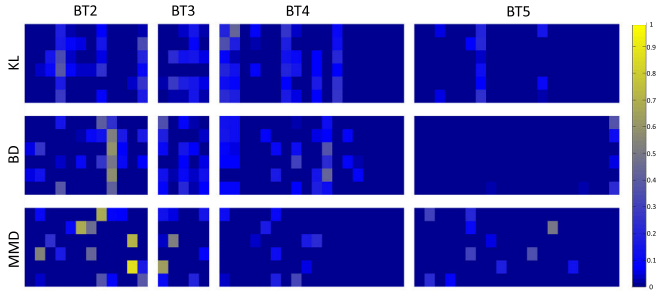


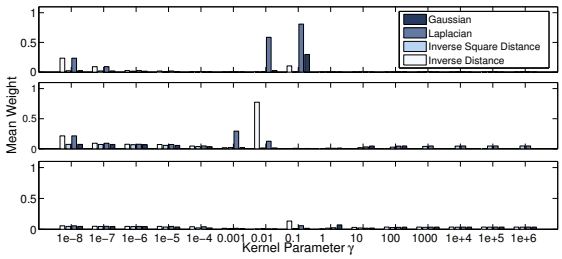
Figure 3. Distribution of weights for KL, BD and MMD weighting on the six images of BT1 by training on BT2-5. Rows indicate the 6 test images, columns indicate the 55 training images, ordered by dataset. Lighter pixels indicate higher weights (highest weight was 0.43).

3, and 10 for tissue, WML, and hippocampus segmentation respectively. For brain tissue and hippocampus segmentation, the KL and BD weighting methods seem to give a bit more non-zero weights than the MMD method, which could be beneficial since it has a regularization effect. For WML segmentation, there seems to be a negative correlation between the percentage of non-zero weights and the classification error. This could be explained by the fact that lesions are only a tiny percentage of the number of voxels, hence can hardly be distinguished in the PDFs. Therefore, more regularization in the form of small (non-zero) weights can be beneficial.

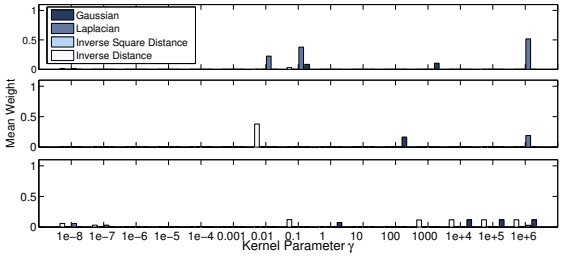
Figure 3 shows the distribution of weights for the three weighting methods when testing on the six images of the BT1 dataset. All three weighting methods gave most weight to the training images of BT2, but the specific images that were given a non-zero weight differed per test image and per weighting method. This can be seen from the correlation between the weight vectors given by the three methods, which was 0.05 between KL and BD, 0.02 between KL and MMD and 0.07 between BD and MMD, which is rather low. When we look at the correlation between total weights per training dataset however, we find much higher correlation: 0.60, 0.15, and 0.71. Here, we see that the MMD weights resembled the BD weights much better than the KL weights. Also note that the KL and BD weighting method both found one or more training images that were given a high weight for all six test images, resulting in similar weight distributions between test images (correlation between test images 0.62 for KL and 0.60 for BD). The MMD weighting method on the other hand, gave weight distributions that differ much more between test images (correlation 0.05). This behavior difference between KL/BD and MMD was also observed in the other experiments and could indicate that MMD weights more according to specific image characteristics (such as differences between subjects), whereas KL and BD weights more according to appearance differences between scanners. This point is discussed further in the Discussion.

B. Kernel Learning

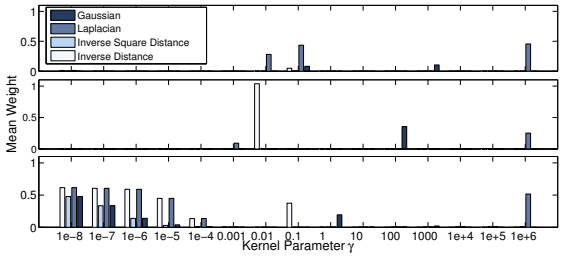
In brain tissue and WML segmentation, kernel learning significantly outperformed the baseline method. For hippocampus segmentation, it performed similar to the baseline. The difference between the two kernel learning methods (CKA and MMD) was quite small and not significantly different.



(a) K_{CKA}



(b) K_{MMD}



(c) KW_{MMD}

Figure 4. Average distribution of kernel weights on the three applications (top: tissue, middle: WML, bottom: hippocampus).

Figures 4(a) and (b) show the kernels that were selected by the CKA and MMD kernel learning method respectively for each of the three applications. Although performance was similar, the distribution of kernel weights was quite different between the two methods. Most kernels that were given a high weight in K-CKA were also given a high weight in K-MMD. K-MMD additionally gave high weights to kernels with the highest kernel parameter (10^6), i.e. kernels that focus on differences between samples that are very close by in feature space. We think this is an artifact caused by the additional

constraint that the samples need to be normalized in the resulting kernel space (Equation 10 of the supplementary material). This point is discussed further in the supplementary material (supplementary materials are available in the supplementary files /multimedia tab). K-CKA on the other hand, gave an additional low weight to almost all kernels. This might be because of a premature stopping of the optimization. Although these small weights hardly influenced the outcome, we set all kernel weights below 0.01 to zero to speed up the calculation of the kernels used for classification.

C. Combining Kernel Learning and Image Weighting

Image weighting generally outperformed kernel learning when used individually, especially for brain tissue segmentation. Combining the two approaches overall gave a small additional improvement in performance, except for hippocampus segmentation, where performance was similar between image weighting with and without kernel learning.

Jointly weighting kernels and images (KW-MMD) gave no clear difference in performance compared to individual kernel learning and MMD image weighting (K-CKA W-MMD). Figures 4(b) and (c) show that for brain tissue and WML segmentation, both methods selected the same base kernels, although the weights were slightly different for WML segmentation. For the hippocampus experiments, KW-MMD gave high weights to low-parameter kernels (i.e. focuses on differences between all samples). However, this behavior appeared only in about one third of segmented images and did not appear to significantly influence the performance. We think this is because of a premature stopping in the optimization, similarly as with K-CKA, discussed above. We also studied the distribution of W-MMD image weights when combining image weighting and MKL; these were very similar to those calculated without MKL for all three applications.

D. Random Forest Classification

Table IV
PERFORMANCE OF ALL METHODS ON THE THREE APPLICATIONS WITH A RF CLASSIFIER, AVERAGED OVER ALL IMAGES OF ALL DATASETS. PERFORMANCE ON TISSUE AND WML SEGMENTATION IS IN CLASSIFICATION ERROR (%), PERFORMANCE ON HIPPOCAMPUS SEGMENTATION IS IN DICE OVERLAP (%). THE BEST METHOD AND ALL METHODS THAT ARE NOT SIGNIFICANTLY WORSE, ARE IN BOLD. $*$ INDICATES A SIGNIFICANT DIFFERENCE BETWEEN K-CKA W-MMD AND KW-MMD.

Cat.	Method	Brain Tissue	WML	Hippocampus
0	K0 W0	19.87	**8.05**	84.8
W	K0 W-KL	**14.48**	**7.31**	84.2
W	K0 W-BD	16.60	11.79	84.2
W	K0 W-MMD	15.74	9.41	84.0
K	K-CKA W0	20.82	**7.79**	79.3
K	K-MMD W0	20.90	**8.00**	79.6
K&W	K-CKA W-KL	16.19	**7.22**	85.0
K&W	K-CKA W-BD	18.04	10.37	**85.2**
K&W	K-CKA W-MMD	17.60	**8.33**	**85.2**
K&W	KW-MMD	17.11*	**8.43**	**85.2**

Table IV shows the results when using a random forest (RF) classifier instead of an SVM. When comparing Table IV with Table III, we see no large differences between the baseline for both classifiers. The supplementary material gives a side-by-side table (Table II) of performance with the two classifiers, to facilitate comparison (supplementary materials are available in the supplementary files /multimedia tab). Image weighting generally brought some more improvement for SVM than for RF. For both classifiers, KL image weighting generally performed best, which for RF gave a significant improvement for tissue segmentation and a non-significant improvement for WML segmentation. For hippocampus, no increase in performance was found, which is not remarkable, since the differences between the different methods with SVM were also very small for this application. Kernel learning performed much worse for RF than for SVM for tissue and hippocampus segmentation. This is probably because we applied Kernel PCA to extract features for RF, while SVM works directly in the kernel space. Ideally, one would like to extract enough features to describe almost all the variance (e.g. 99%), but in our experiments, the variance in the learned kernel spaces equals 1 in every direction (Equation 11 of the supplementary material). As a result, around 1 000 features were needed (depending on the application and dataset) to describe 99% of the variance. RF performed poorly for such high numbers of features. For tissue segmentation, the combination of kernel learning and image weighting also performed much worse with RF than SVM, because of this same effect. For the other experiments, RF and SVM performed similarly.

E. Computational Requirements

All experiments were performed as single-core jobs on a Linux cluster from 2014 with AMD Opteron 6376 (2.3GHz) CPUs. The methods without kernel learning required 2 to 3GB of memory, the kernel learning methods required about 6GB. Table 1 of the supplementary material gives the average computation time for SVM classification of one test image for each of the applications and methods (supplementary materials are available in the supplementary files /multimedia tab). On average, the baseline method took about 8 to 14 seconds to classify one test image, for both SVM and RF. The different weighting methods W-KL, W-BD, W-MMD ranged from 16 seconds to just over 4 minutes, depending on the application and the dataset, where larger training datasets required more computation time. These methods all use gradient descent and the calculation time is therefore dependent on the difficulty of finding the optimal solution. BD was the overall fasted method, followed by MMD, and KL.

All methods that incorporate kernel learning took much more calculation time. Here, calculation time scales quadratically with the number of training images used. For the WML experiments (20 to 30 training images), the kernel learning took about 45 minutes. For brain tissue segmentation (41 to 56 training images) it took about 2 hours and 40 minutes. For hippocampus segmentation (128 to 134 training images) this was about 73 hours and 10 minutes. Calculation of the 60 base kernels per training image is what makes these methods so expensive, so using fewer base kernels would speed them up considerably. Using Kernel PCA to extract 10, 13, or 30

features took between 10 and 30 seconds and depends heavily on the number of extracted features.

V. Discussion and Conclusion

Image weighting significantly improved performance over weighting all training images equally for all three studied applications: voxelwise brain tissue, white matter lesion (WML), and hippocampus segmentation. This convincing benefit of image weighting corresponds to previous findings [42], [5]. For WML segmentation, which had highly unbalanced classes, Kullback-Leibler (KL) weighting performed much better than other weighting methods, which was also observed in [42]. Here, KL weighting presumably outperformed other weighting methods since it gives more importance parts of the distribution with small prior distribution $P(\boldsymbol{x})$. For brain tissue and hippocampus segmentation, the best methods differed and differences between the three weighting methods were relatively small.

The new maximum mean discrepancy (MMD) image weighting has the advantage over KL and Bhattacharyya distance (BD) weighting that no intermediate PDF estimation is required. This is especially beneficial when many features are used, as the complexity of PDF estimation scales quadratically with the number of features. On the other hand, in the current implementation of MMD image weighting, calculation time scales quadratically ($\mathcal{O}(n^2)$) with the number of training samples used for the optimization (because of calculation of the MMD matrix M in Equation 11), whereas KL and BD weighting are $\mathcal{O}(n)$. It might therefore be worthwhile to investigate a method to speed up MMD calculation, such as [45], which is $\mathcal{O}(n)$.

We noticed that for different test images of the same scanner, training images were assigned similar weight vectors by KL and BD image weighting, but quite different weight vectors by MMD weighting. Both KL and BD measure density differences in feature space. MMD on the other hand, measures distances between all samples, which corresponds to the work that is required to shift samples (in feature space) to transform a training distribution into the test distribution. KL and BD do not take this distance between shifted samples into account. This could result in KL and BD often finding some training images (presumably with average tissue class sizes) that are given a high weight for all test images. MMD on the other hand, is likely more influenced by the sizes of the different distribution peaks (in our examples, the three different brain tissues) and therefore might weight more according to subject differences than KL and BD.

MKL with any of the two presented methods significantly improved performance over the baseline (a Gaussian kernel) for brain tissue and WML segmentation and performed similar to the baseline for hippocampus segmentation. For all three applications, image weighting resulted in a larger improvement than MKL, although this increase was not significant for WML segmentation. The proposed combination of image weighting and MKL overall gave an additional improvement over image weighting alone, except for hippocampus segmentation, where it performed similarly.

We found no clear difference in performance between MKL with only the centered kernel alignment (CKA) method of Cortes et al. [7] and using and additional MMD term. Note that CKA MKL is determined on training data only, hence, finds a kernel space that discriminates between classes and generalizes well between the training datasets. Adding the MMD term should give a kernel space that additionally reduces differences between training and test distributions. We showed that the two MKL methods do indeed generate a different kernel space by selecting different kernel weights, but classification performance using the two methods was very similar. Note that for both methods there is no guarantee that the resulting kernel space and image weights are suitable for classification of the test samples, since no labeled samples from the test distribution were available for training. The additional MMD term might therefore give a kernel space that reduces distribution differences between datasets compared to CKA MKL, but has more class overlap. Adding the MMD term does have the advantage of enabling joint optimization of kernel and image weights, by using MMD image weighting. Besides being more efficient, this has the advantage of resulting in a joint optimum for kernel and image weights. On the other hand, the MMD term would require to additionally set the trade-off parameter θ (Equation 16). But it might be possible to use a set value (e.g. $\tilde{\theta} = 1$ in Equation 11 of the supplementary material), since we found performance to be stable over a large range of values for θ.

Overall, our results on a variety of different tasks convincingly show that combining kernel learning and image weighting is beneficial for across-scanner segmentation. The proposed method requires a set of training images (we used 20 to 134 training images) with different characteristics, so the image weighting method can select the most useful images. We experimented with voxelwise classification, but the proposed methods might as well be applied to classification of super voxels or patches. It can also be applied to image classification problems such as computer-aided diagnosis by determining image weights based on the voxel distributions. In the presented experiments, only a small number of classes was predicted (two or three), but the methods can also be applied to problems with a larger number of classes, such as separation of different brain structures or subfields. This likely results in similar image weights as segmentation of the same image into other classes, since the weighting is unsupervised and weights according to similarity between complete images. For cases where classes are small (for example in segmentation of brain structures, subfields, or WMLs), it might be interesting to investigate whether it is possible to give more importance to similarity of specific classes or class boundaries.

Our method is easily applicable in practice, for example in multi-center studies, since it requires no labeled data from the test distribution. As shown in [42], image weighting can be beneficial even if some manually annotated images from the test scanner are available. Although this setting was not tested in this paper, we believe the combination of kernel learning and image weighting can also be beneficial here. In this case, the labeled test data can additionally be used to ensure that the learned kernel is suitable for classification of test samples, by

maximizing CKA on the test samples rather than the training samples.

We compared the added value for image weighting and kernel learning only for SVM classification with Gaussian scale space features. However, both image weighting and kernel learning can also be applied to other segmentation approaches, using different features, different classifiers, and different pre- and post-processing. Image weighting can straightforwardly be applied into other frameworks, especially in the form used in this paper, where training samples are selected based on the image weights. Decreasing train-test PDF distances could also be translated into a deep learning framework, by incorporation of KL, BD or MMD between training and test data in the loss function. The added value of image weighting has also been shown in other types of methods, such as in multi-atlas selection and in patch-based fusion schemes such as [43], which performs atlas selection per voxel based on minimizing KL between training and test patches around the voxel. As shown, kernel learning could be extended to non-kernel classifiers (such as random forest) with Kernel PCA [33]. However, in our experiments, performance was not always good, probably because of a variance normalization constraint in the learned kernel spaces, which resulted in very many components needing to be extracted to describe the data well. How to solve this issue (for example with a different constraint in kernel space, or a different way to extract features from the kernel space) would be an interesting direction for further research. Investigating other methods for feature representation transfer and the combination with image weighting would also be interesting. Transforming feature representations could for example also be performed by a contrast synthesis method such as [31], [18]. For deep learning, one could for example think of transferring representation between different datasets by generating a joint representation layer (see [22] for an example on how this can be done).

To conclude, the combination of image weighting and feature representation transfer through kernel learning appears to be a promising method for supervised segmentation using heterogeneous training data different from the test dataset. The good results on different tasks indicates that the proposed methods can be applied to a wide range of medical image segmentation tasks.

VI. Acknowledgments

This research was performed as part of the research project 'Transfer learning in biomedical image analysis' which is financed by The Netherlands Organization for Scientific Research (NWO).

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

[1] M. Boccardi, M. Bocchetta, F.C. Morency, D.L. Collins, M. Nishikawa, R. Ganzola, M.J. Grothe, D. Wolf, A. Redolfi, M. Pievani, et al. Training labels for hippocampal segmentation based on the EADC-ADNI harmonized hippocampal protocol. *Alzheimer's & Dementia*, 11(2):175–183, 2015.

[2] E.E. Bron, R.M.E. Steketee, G.C. Houston, R.A. Oliver, H.C. Achterberg, M. Loog, J.C. Swieten, A. Hammers, W.J. Niessen, M. Smits, et al. Diagnostic classification of arterial spin labeling and structural MRI in presenile early stage dementia. *Human Brain Mapping*, 35(9):4916–4931, 2014.

[3] A. Carass, S. Roy, A. Jog, J.L Cuzzocreo, E. Magrath, A. Gherman, J. Button, J. Nguyen, F. Prados, C.H. Sudre, et al. Longitudinal multiple sclerosis lesion segmentation: resource & challenge. *NeuroImage*, 148(1):77–102, 2017.

[4] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

[5] V. Cheplygina, A. Van Opbroek, M.A. Ikram, M.W. Vernooij, and M. De Bruijne. Asymmetric similarity-weighted ensembles for image segmentation. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 273–277. IEEE, 2016.

[6] T.F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[7] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(3):795–828, 2012.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] R. De Boer, H.A. Vrooman, M.A. Ikram, M.W. Vernooij, M.M.B. Breteler, A. Van der Lugt, and W.J. Niessen. Accuracy and reproducibility study of automatic MRI brain tissue segmentation methods. *NeuroImage*, 51(3):1047–1056, 2010.

[10] R. De Boer, H.A. Vrooman, F. Van der Lijn, M.W. Vernooij, M.A. Ikram, A. Van der Lugt, M. Breteler, and W.J. Niessen. White matter lesion extension to automatic brain tissue segmentation on MRI. *NeuroImage*, 45(4):1151–1161, 2009.

[11] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[12] V. Dill, A.R. Franco, and M. S. Pinho. Automated methods for hippocampus segmentation: the evolution and a review of the state of the art. *Neuroinformatics*, 13(2):1–18, 2015.

[13] L. Duan, I.W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):465–479, 2012.

[14] E. Geremia, O. Clatz, B.H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache. Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390, 2011.

[15] M.l Goetz, C. Weber, B. Stieltjes, and K. Maier-Hein. Learning from small amounts of labeled data in a brain tumor classification task. *Advances in Neural Information Processing Systems*, 2014.

[16] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(7):2211–2268, 2011.

[17] J. Huang, A. Gretton, K.M. Borgwardt, B. Schölkopf, and A.J. Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pages 601–608. NIPS, 2007.

[18] J.E. Iglesias, E. Konukoglu, D. Zikic, B. Glocker, K. Van Leemput, and B. Fischl. Is synthesizing MRI contrast useful for inter-modality analysis? In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, pages 631–638. Springer, 2013.

[19] J.E. Iglesias, C.Y. Liu, P.M. Thompson, and Z. Tu. Robust brain extraction across datasets and comparison with publicly available methods. *Medical Imaging, IEEE Transactions on*, 30(9):1617–1634, 2011.

[20] M.A. Ikram, A. Van der Lugt, W.J. Niessen, P.J. Koudstaal, G.P. Krestin, A. Hofman, D. Bos, and M.W. Vernooij. The Rotterdam Scan Study: design update 2016 and main findings. *European journal of epidemiology*, 30(12):1299–1315, 2015.

[21] M.A. Ikram, H.A. Vrooman, M.W. Vernooij, F. Van der Lijn, A. Hofman, A. Van der Lugt, W.J. Niessen, and M.M.B. Breteler. Brain tissue volumes in the general elderly population: The Rotterdam Scan Study. *Neurobiology of Aging*, 29(6):882–890, 2008.

[22] K. Kamnitsas, C. Baumgartner, C. Ledig, V. Newcombe, J. Simpson, A. Kane, D. Menon, A. Nori, A. Criminisi, D. Rueckert, et al. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In *International Conference on Information Processing in Medical Imaging*, pages 597–609. Springer, 2017.

[23] J. Kleesiek, G. Urban, A. Hubert, D. Schwarz, K. Maier-Hein, M. Bendszus, and A. Biller. Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. *NeuroImage*, 129(1):460–469, 2016.

[24] S. Klein, M. Staring, K. Murphy, M.A. Viergever, and J.P.W. Pluim. Elastix: a toolbox for intensity-based medical image registration. *Medical Imaging, IEEE Transactions on*, 29(1):196–205, 2010.

[25] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

[26] A.M. Mendrik, K.L. Vincken, H.J. Kuijf, M. Breeuwer, W.H. Bouvy, J. De Bresser, A. Alansary, M. De Bruijne, A. Carass, A. El-Baz, et al. MRBrainS challenge: online evaluation framework for brain image segmentation in 3T MRI scans. *Computational Intelligence and Neuroscience*, 2015:1–16, 2015.

[27] P. Moeskops, M.A. Viergever, A.M. Mendrik, L.S. De Vries, M.J.N.L. Benders, and I. Išgum. Automatic segmentation of MR brain images with a convolutional neural network. *Medical Imaging, IEEE Transactions on*, 35(5):1252–1261, 2016.

[28] S.G. Mueller, M.W. Weiner, L.J. Thal, R.C. Petersen, C.R. Jack, W. Jagust, J.Q. Trojanowski, A.W. Toga, and L. Beckett. Ways toward an early diagnosis in Alzheimer's disease: the Alzheimer's Disease Neuroimaging Initiative (ADNI). *Alzheimer's & Dementia*, 1(1):55–66, 2005.

[29] S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on*, 22(2):199–210, 2011.

[30] S.J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[31] S. Roy, A. Carass, and J. Prince. A compressed sensing approach for MR tissue contrast synthesis. In *Information Processing in Medical Imaging*, pages 371–383. Springer, 2011.

[32] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *Computer Vision–ECCV 2010*, pages 213–226, 2010.

[33] B. Schölkopf, A. Smola, and K.R. Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[34] B. Scholkopf and A.J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[35] B.W. Silverman. *Density estimation for statistics and data analysis*, volume 26. Chapman & Hall/CRC, 1986.

[36] S.M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, 2002.

[37] M. Styner, J. Lee, B. Chin, M. Chin, O. Commowick, H. Tran, S. Markovic-Plese, V. Jewells, and S. Warfield. 3D segmentation in the clinic: A grand challenge II: MS lesion segmentation. *Midas Journal*, pages 1–6, 2008.

[38] N.J. Tustison, B.B. Avants, P.A. Cook, Y. Zheng, A. Egan, P.A. Yushkevich, and J.C. Gee. N4ITK: improved N3 bias correction. *Medical Imaging, IEEE Transactions on*, 29(6):1310–1320, 2010.

[39] F. Van der Lijn, M. De Bruijne, S. Klein, T. Den Heijer, Y.Y. Hoogendam, A. Van der Lugt, M. Breteler, and W.J. Niessen. Automated brain structure segmentation based on atlas registration and appearance models. *Medical Imaging, IEEE Transactions on*, 31(2):276–286, 2012.

[40] A. Van Engelen, A.C. Van Dijk, M.T.B. Truijman, R. Van T Klooster, A. Van Opbroek, A. Van der Lugt, W.J. Niessen, M.E. Kooi, and M. De Bruijne. Multi-center MRI carotid plaque component segmentation using feature normalization and transfer learning. *Medical Imaging, IEEE Transactions on*, 34(6):1294, 2015.

[41] A. Van Opbroek, M.A. Ikram, M.W. Vernooij, and M. De Bruijne. Transfer learning improves supervised image segmentation across imaging protocols. *Medical Imaging, IEEE Transactions on*, 34(5):1018, 2015.

[42] A. Van Opbroek, M.W. Vernooij, M.A. Ikram, and M. De Bruijne. Weighting training images by maximizing distribution similarity for supervised segmentation across scanners. *Medical Image Analysis*, 24(1):245–254, 2015.

[43] H. Wang, J.W. Suh, S.R. Das, J.B. Pluta, C. Craige, P. Yushkevich, et al. Multi-atlas segmentation with joint label fusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):611–623, 2013.

[44] A.J. Worth. The Internet Brain Segmentation Repository (IBSR), 2009.

[45] J. Zhao and D. Meng. FastMMD: Ensemble of circular discrepancy for efficient two-sample test. *Neural Computation*, 27(6):1345–1372, 2015.