Tesis Doctoral

Transferencia de aprendizaje mediante bosques de decisión

Goussies, Norberto Adrián

2014-11-28

Universidad de Buenos Aires

Facultad de Ciencias Exctas y Naturales

Departamento de Computación

Transferencia de aprendizaje mediante bosques de decisión

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área Cs. de la Computación

Norberto Adrián GOUSSIES

Directora de tesis: Marta E. Mejail

Consejera de estudios: Marta E. Mejail

Buenos Aires, 2014

Fecha de defensa: 28 de Noviembre del 2014

Transferencia de aprendizaje mediante bosques de decisión

Resumen Los bosques de decisión son una herramienta que se han popularizado para resolver diferentes tareas de visión por computadora. Sus principales ventajas son su alta eficiencia computacional, los resultados competitivos con el estado del arte que se obtienen al emplearlos y que son inherentemente clasificadores multiclase. Usualmente, para cada nueva tarea de visión por computadora donde se tiene que entrenar un bosque de decisión, un nuevo conjunto de entrenamiento debe ser confeccionado desde cero.

En esta tesis, presentamos un nuevo método de transferencia de aprendizaje que utiliza bosques de decisión y lo aplicamos para reconocer gestos y caracteres. El método propuesto extrae conocimiento de otras tareas de visión por computadora y lo aplica a una tarea destino, reduciendo el problema de crear nuevos conjuntos de entrenamiento.

Introducimos dos extensiones en el modelo de los bosques de decisión para poder transferir conocimiento de varias tareas de origen a una tarea destino. La primera es la ganancia de información mixta, que se puede interpretar como un regularizador basado en los datos. La segunda es la propagación de etiquetas, que infiere el estructura de la variedad del espacio de características. Demostramos que ambas extensiones son importantes para obtener altas tasas de reconocimiento.

Nuestros experimentos demuestran mejoras sobre los bosques de decisión tradicionales en el ChaLearn Gesture Challenge y en el conjunto de datos MNIST (Mixed National Institute of Standards and Technology) de dígitos escritos a mano. Además demostramos mejoras en tasas de reconocimiento en comparación con otros clasificadores del estado del arte.

Palabras clave: bosques de decisión, transferencia de aprendizaje, reconocimiento de gestos, reconocimiento de caracteres, propagación de etiquetas

Tranfer learning using decision forests

Abstract Decision forests are an increasingly popular tool in computer vision tasks. Their advantages include high computational efficiency, state-of-the-art accuracy and multi-class support. Usually for each new computer vision task where a decision forest has to be learned a new training set is collected from scratch.

In this thesis, we present a novel method for transfer learning which uses decision forests, and we apply it to recognize gestures and characters. This method extracts knowledge from previous computer vision tasks and applies it to the target task, thus reducing the problem of collecting new datasets.

We introduce two extensions into the decision forest framework in order to transfer knowledge from several source tasks to a given target task. The first one is mixed information gain, which is a data-based regularizer. The second one is label propagation, which infers the manifold structure of the feature space. We show that both of them are important to achieve higher recognition rates.

Our experiments demonstrate improvements over traditional decision forests in the ChaLearn Gesture Challenge and MNIST (Mixed National Institute of Standards and Technology) dataset of handwritten digits. Also, we show that the transfer learning decision forests compare favorably against other state-of-the-art classifiers.

Keywords: decision forests, transfer learning, gesture recognition, optical character recognition, label propagation

Agradecimientos

En primer lugar quisiera agradecer a Marta Mejail por haberme dado la oportunidad de ser su discípulo al haber sido mi directora de tesis de licenciatura y doctorado. Me dedicó una gran cantidad de tiempo y esfuerzo sin pedir nada a cambio, lo cual valoro mucho. Siempre estuvo dispuesta a aconsejarme sobre una variedad de temas, desde investigación hasta temas personales. Ha logrado crear y dirigir un grupo de investigación que genera un ambiente estimulante para invesitgar. Gracias a ella, he tenido oportunidades de formarme en congresos, pasantías y cursos que jamás había imaginado. En segundo lugar quisiera agradecer a todos los miembros del DC y particularmente a los miembros del grupo de investigación que siempre estuvieron presentes para charlar y aportar ideas sobre los temas de investigación. Pero con los que también compartí innumerables cenas, almuerzos y momentos especiales.

Además, quiero agradecer al CONICET por la beca de doctorado que me permitió realizar este trabajo de tesis y a la Facultad de Ciencias Exactas y Naturales y a la Universidad de Buenos Aires por el espacio y el ambiente de trabajo.

Finalmente quiero agradecer a Inés por todo el amor que me dió durante estos años y por haberme acompañado en los buenos y malos momentos. Por tenerme paciencia cada vez que le pedí tiempo de nuestra relación para dedicarle a la facultad.

Contents

List of Figures

Chapter 1

Introduction

Machine learning tools have achieved significant success in many computer vision tasks, including face detection [VJ04], object recognition [FGMR10], character recognition [LBBH98] and gesture recognition [GAJ+13]. Those tasks are often posed as a classification problem, namely identifying to which of a set of categories a new observation belongs. A major advantage of using machine learning tools is that they tend to deal robustly with the complexities found in real data.

Decision forests [Bre01, AC12] are an important type of classifiers that have recently gained popularity in the computer vision and machine learning community since they are a key component for the real-time human pose recognition method implemented in the Kinect [SFC+11]. Additionally, decision forests compare favorably with respect to other type classifiers [RNA08].

Certain properties make decision forests particularly interesting for computer vision problems. First, decision forests are multi-class classifiers; therefore it is not necessary to train several binary classifiers for a multi-class problem. Second, they are fast both to train and test. Finally, they can be parallelized, which makes them ideal for graphics processing unit (GPU) implementations [Sha08] and multi-core implementations. Additionally, decision forests as most of the supervised learning techniques, are learned from scratch using a training dataset collected for the task.

However, in many cases it is difficult to create new training datasets for

each new computer vision task. Although the problem remains unsolved, some progress has already been made in certain computer vision tasks, such as object recognition [FFFP06] and action recognition [SM11]. The key insight is to try to replicate the ability of the human brain, which is capable of learning new concepts applying previously acquired knowledge.

Transfer learning aims at extracting the knowledge from one or more source tasks, and apply that knowledge to a target task. As opposed to multi-task learning, rather than simultaneously learning the source and target tasks, transfer learning focus more on learning the target task. The roles of the source and target tasks are not symmetric [PY10]. The goal is to exploit the knowledge extracted from the source tasks so as to improve the generalization of the classifier in the target task.

Traditional supervised learning techniques usually require a large training set which must provide for each training example a label. Therefore some alternatives have emerged, in order to overcome this crucial limitation. An interesting approach that addresses this limitation is semi-supervised learning. In which, a small amount of labeled examples are combined with a large amount of unlabeled examples. A different approach is taken in transfer learning, where instead of using unlabeled examples, the training set is extended with examples of related tasks. The underlying idea is that it is possible to learn from different tasks by considering relationship among them and by trying to find functions that generalize well over all of the available tasks.

There are three expected benefits of using a transfer learning approach over a traditional machine learning approach when we compare its performance with respect to the number of training instances [TOC10]. The first expected benefit is that the transfer learning approach should have a higher start than a traditional machine learning approach. In other words, the performance of the transfer learning method when the number of training samples of the target task is one, or even zero, should be higher than the performance of the traditional machine learning techniques.

The other two expected advantages of using a transfer learning approach are to have a higher slope and a higher asymptote. Put differently, the trans-

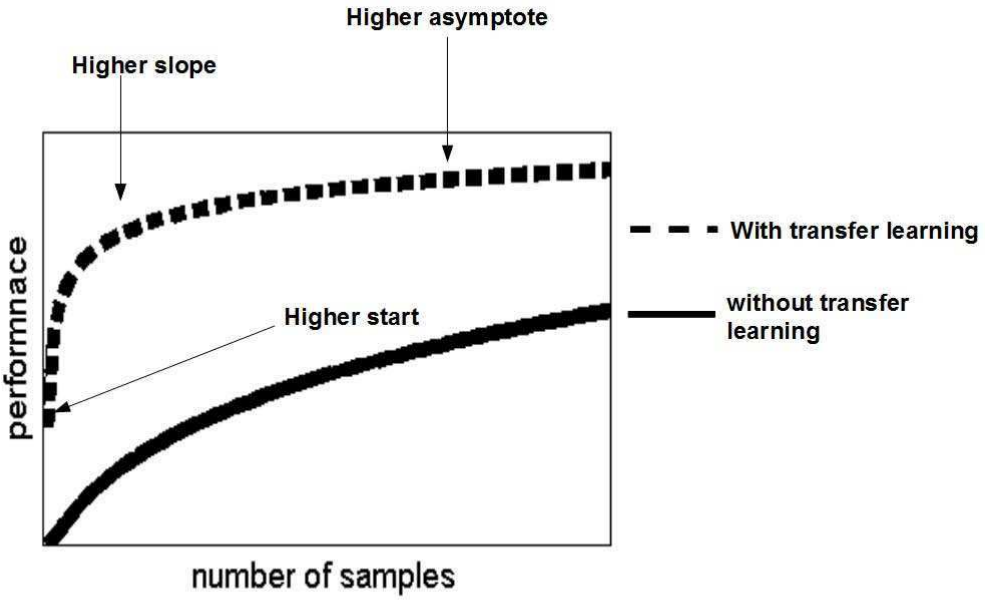


Figure 1.1: Expected advantages of transfer learning [TOC10].

fer learning approach should require less number of training samples to increase its performance and, at same time, when a lot of training samples are available the performance should be better. The expected advantages are depicted in Figure 1.1.

Many examples can be found in computer vision where transfer learning can be truly beneficial. One example is optical character recognition, which seeks to classify a given image into one of the characters of a given alphabet. Most methods have focused on recognizing characters from the English alphabet [LBBH98]. The recognition of characters from other alphabets, such as French, implies collecting a new training dataset [GA11]. In that case, it would be helpful to transfer the classification knowledge into the new t.

The need for transfer learning also arises in gesture recognition [GAJ+13], which aims at recognizing a gesture instance drawn from a gesture vocabulary. For example, a gesture vocabulary may consist of Italian gestures or referee signals. In this case, the classifier needs to predict the gesture of the vocabulary that corresponds to a given video. Again, it would be interest-

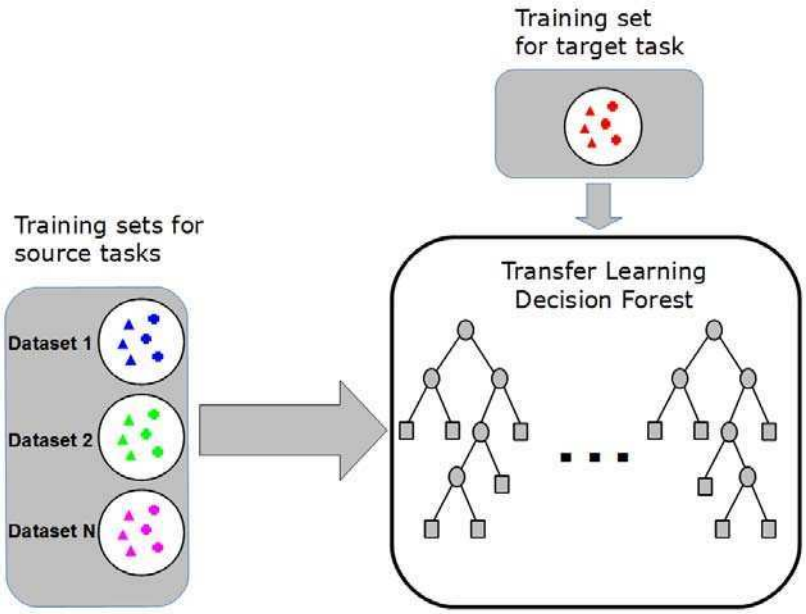


Figure 1.2: Learning processes for our novel transfer learning decision forests.

ing to improve the performance of a system by exploiting the knowledge acquired from similar vocabularies.

1.1 Contributions

In this thesis, we present a novel method for trasnfer learning which extends the decision forest framework [Bre01, AC12], and we apply it to transfer knowledge from multiple source tasks to a given target task. We introduce two mechanisms in order to transfer knowledge from the source tasks to the target task. The first one is mixed information gain, which is a data-based regularizer. The second one is label propagation, which infers the manifold structure of the feature space. The learning process for our novel transfer learning decision forests is illustrated in Figure 1.2. As we can see, our technique learns simultaneously from the source and target tasks. The decision forests are trained using the combined target and source training sets.

The first key contribution is to revise the criterion for finding the parameters of each internal node of the decision forests in the transfer learning

setting. The novel criterion exploits the knowledge from the source tasks and the target task to find the parameters for each internal node of the decision forests. The additional information penalizes split functions with a high information gain in the target task and a low information gain in the source tasks. We prove some relevant properties of the novel criterion. The properties address the question of how many source tasks are required to improve the performance using transfer learning decision forests.

The second key contribution is to propagate labels through leaves in order to infer the manifold structure of the feature space. The aim of this step is to assign a predictive model to the leaves without training samples of the target task after the trees of the decision forests are grown. We create a fully connected graph, for each tree in the forest, where the nodes are the leaves of the tree and the weight of each edge takes into account the training data reaching the leaves. An implicit assumption of this step is that nearby leaves should have similar predicitve models.

We extensively validate our approach in two challenging datasets. First, our experiments in the ChaLearn gesture challenge dataset [GAJ+12] show that our method does not have a uniform margin of improvement over all the tasks. However, we demonstrate that when there are source tasks related to the target task, we obtain greater improvements. Second, our experiments in the MNIST dataset [LBBH98] show that the transfer learning decision forests outperform the traditional decision forests for training datasets of different size. Moreover, the gap between both classifiers is larger when the size of the training dataset is small.

1.2 Related Work

In the following we will review transfer learning techniques which have been applied to computer vision problems. A recent survey [PY10] provides a comprehensive overview of the developments for classification, regression and clustering. In recent years, the computer vision community has become increasingly interested in using transfer learning techniques, especially for object recognition [LFW04,STFW05,FFFP06,BU05,TMF07,QCD08,

BT10, GLC11, SKFD10, TOC13].

A variety of methods have been proposed in the generative probabilistic setting [FFFP06, STFW05]. These models consider the relationships between different object parts during the training process. The key idea is to share some parameters or prior distributions between object categories, using the knowledge from known classes as a generic reference for newly learned models. The association of objects with distributions over parts scales linearly in [STFW05], while exponentially in [FFFP06].

Moreover, discriminative models have been extended to the transfer learning setting [TOC13, DYXY07, YD10, AZ11, LST11, TMF07]. [AZ11] and [TOC13] apply transfer learning to the support vector machines framework. During the training process of the target detector, the previously learned template is introduced as a regularizer into the cost function. [DYXY07] allow users to utilize a small amount of newly labeled data developing a framework based on boosting [FS97]. Later, [YD10] extends [DYXY07] for handling multiple sources.

An interesting idea, usually referred as feature transfer [PY10], is to extract from the source tasks a good representation for the target task. A framework for learning an object classifier from a single example is described in [Fin05]. The aim is to emphasize the relevant dimensions for classification using available examples of related tasks. Inspired by incremental learning, [GLC11] presents a technique that creates intermediate representations of data between the source and the target task as points on a manifold.

Assuming that the source tasks and the target tasks share some parameters or prior distributions for the hyperparameters of the models is referred as parameter or model transfer approach [PY10]. A method to transfer shape information across object classes is presented in [SGS09]. More recently, it has been proposed [GF12] to transfer the appearance of the objects, their location distribution within the image, and the context in which the objects are embedded exploiting the semantic hierarchy of ImageNet.

Instance transfer approaches [PY10] consider source and target data together during the training process. [LST11] augments the training data for each class by borrowing and transforming examples from other classes. A

method for learning new visual categories is described in [QCD08], using only a small subset of reference prototypes for a given set of tasks. As described earlier, [DYXY07, YD10] proposed a boosting-based algorithm that allows knowledge to be effectively transferred from old to new data. The effectiveness of the novel algorithm is analyzed both theoretically and empirically. In this thesis, we develop an instance transfer approach that exploits source and target data to find the parameters of each internal node of the decision forests.

Few researchers have addressed the problem of transfer learning using decision forests or trees. [LSSB09] extends random forests to semi-supervised learning. In order to incorporate unlabeled data a maximum margin approach is proposed, which is optimized using a deterministic annealing-style technique. [WZCG08] propose to treat each input attribute as extra task to bias each component decision tree in the ensemble. [PKZ13] propose a novel criterion for node splitting to avoid the rank-deficiency in learning density forests for lipreading. [wLGC07] learn a new task by traversing and transforming a decision tree previously learned for a related task. The transfer learning decision tree learns the target task from a partial decision tree model induced by ID3 [Qui86b]. In this thesis, we follow a different approach, first we consider the source and target data when we build each tree of the decision forests. Second, decision forests reduce the variance of the classifier aggregating the results of multiple random decision trees.

Our approach shares some features with the work by [FCGT12], who propose to transfer learning with boosted C4.5 decision trees. The main difference is that their method reduces the variance of the decision trees by means of boosting, which has been shown to be less robust against label noise when compared with decision forests [Bre01, LSSB09]. In addition, we use label propagation to learn the manifold structure of the feature space, and assign predictive models only to the leaves of the trees.

Transfer learning has been applied to the problem of optical character recognition. In [QSCP10] a method to learn classifiers from a collection of related tasks in which each task has its own label set is presented. The problem is formulated as one of maximizing the mutual information among

the label sets. The experiments on the MNIST dataset [LBBH98] show that jointly learning the multiple related tasks significantly improves the classification accuracy when the size of the training set is small. Using a different approach, a large margin of improvement is also achieved in [FCGT12].

There has been a growing interest in applying transfer learning techniques to gesture recognition. A method for transfer learning in the context of sign language is described in [FFW07]. A set of labeled words in the source and target data is shared so as to build a word classifier for a new signer on a set of unlabeled target words. A transfer learning method for Conditional Random Fields is implemented to exploit information in both labeled and unlabeled data to learn high-level features for gesture recognition in [LYZH10]. More recently, the ChaLearn Gesture Competition [GAJ+13] provided a benchmark of methods that apply transfer learning to gesture recognition. Several approaches submitted to the competition have been published [MNG13, Lui12, WRLD13].

1.3 Publications

The contributions presented in this thesis have been published in the following papers:

- **Norberto Goussies**, Sebastian Ubalde, Francisco Gómez Fernández, Marta Mejail. *Optical Character Recognition Using Transfer Learning Decision Forests* In Proceedings of the IEEE International Conference on Image Processing, 2014 (to appear)

- **Norberto Goussies**, Sebastian Ubalde, Marta Mejail. *Transfer Learning Decision Forests for Gesture Recognition* In Journal of Machine Learning Research, 2014 (accepted)

1.4 Organization of the thesis

This thesis is organized as follows. We discuss the decision forest model in Chapter 2. The novel algorithm for training a transfer learning decision

forest is described in Chapter 3, we illustrate its performance on some artificial data sets, and prove some properties of the mixed information gain. We present two applications to computer vision problems of our novel transfer learning decision forest in Chapter 4, and show that they achieve superior recognition rates when the training set is small. Finally, Chapter 5 details our conclusions.

Capítulo 1

Introducción

Las herramientas de aprendizaje automático han logrado un éxito significativo en muchas tareas de visión por ordenador incluyendo la detección de rostros [VJ04], reconocimiento de objetos [FGMR10], reconocimiento de caracteres [LBBH98] y reconocimiento de gestos [GAJ+13]. Dichas tareas se plantean a menudo como un problema de clasificación, es decir, identificar a cuál de un conjunto de categorías pertenece una nueva observación. Una ventaja importante del uso de herramientas de aprendizaje de automático es que tienden a tratar robustamente con las complejidades encontradas en datos reales.

Los bosques de decisión [Bre01, AC12] son un importante tipo de clasificadores que han ganado recientemente popularidad en la comunidad de visión por ordenador y aprendizaje automático, ya que son un componente clave en el método de reconocimiento de posturas en tiempo real implementado en la Kinect [SFC+11]. Además, los bosques de decisión se comparan favorablemente con respecto a otros tipos de clasificadores [RNA08].

Ciertas propiedades hacen que los bosques de decisión sean particularmente interesante para los problemas de visión por computador. En primer lugar, los bosques de decisión son clasificadores multiclase; por lo tanto no es necesario entrenar varios clasificadores binarios para un problema multiclase. En segundo lugar, son rápidos, tanto para entrenar como para clasificar. Por último, pueden ser paralelizados, lo que los hace ideales para la GPU [Sha08] y procesadores de múltiples núcleos. Además, los bosques de

decisión como la mayor parte de las técnicas de aprendizaje supervisado, se aprenden desde cero utilizando un conjunto de datos de entrenamiento recogidos para la tarea.

Sin embargo, en muchos casos, es difícil crear nuevos conjuntos de datos de entrenamiento para cada nueva tarea de visión por ordenador. Aunque el problema sigue sin resolverse, se han realizado algunos progresos en ciertas tareas de visión por ordenador, tales como el reconocimiento de objetos [FFFP06] y reconocimiento de acciones [SM11]. La idea clave es tratar de replicar la capacidad del cerebro humano, el cual es capaz de aprender nuevos conceptos utilizando conocimientos adquiridos previamente.

Los métodos de transferencia de aprendizaje tiene como objetivo extraer el conocimiento de una o más tareas de origen, y aplican ese conocimiento en una tarea de destino. En oposición a los métodos de aprendizaje multitarea, que aprenden simultáneamente las tareas de origen y destino, los métodos de transferencia de aprendizaje se centran más en el aprendizaje de la tarea de destino. Las funciones de las tareas de origen y destino no son simétricas [PY10]. El objetivo es aprovechar el conocimiento extraído de las tareas de origen a fin de mejorar la generalización del clasificador en la tarea destino.

Las técnincas de aprendizaje supervisado tradicionales, por lo general requieren un conjunto de entrenamiento grande que debe proporcionar para cada muestra de entrenamiento una etiqueta. Por lo tanto algunas alternativas han surgido, con el fin de superar esta limitación crucial. Un enfoque interesante que aborda esta limitación es el aprendizaje semisupervisado. En el cual, una pequeña cantidad de muestras etiquetadas se combinan con una gran cantidad de muestras no etiquetadas. Un enfoque diferente se toma en transferencia de aprendizaje, donde en lugar usar muestras no etiquetadas, el conjunto de entrenamiento se extiende con muestras de tareas relacionadas. La idea subyacente es que es posible aprender de las diferentes tareas, considerando la relación entre ellas y tratando de encontrar las funciones que generalizan conjuntamente para todas las tareas disponibles.

Al comparar el rendimiento, en función del tamaño del conjunto de entrenamiento, de un método de transferencia de aprendizaje contra el ren-

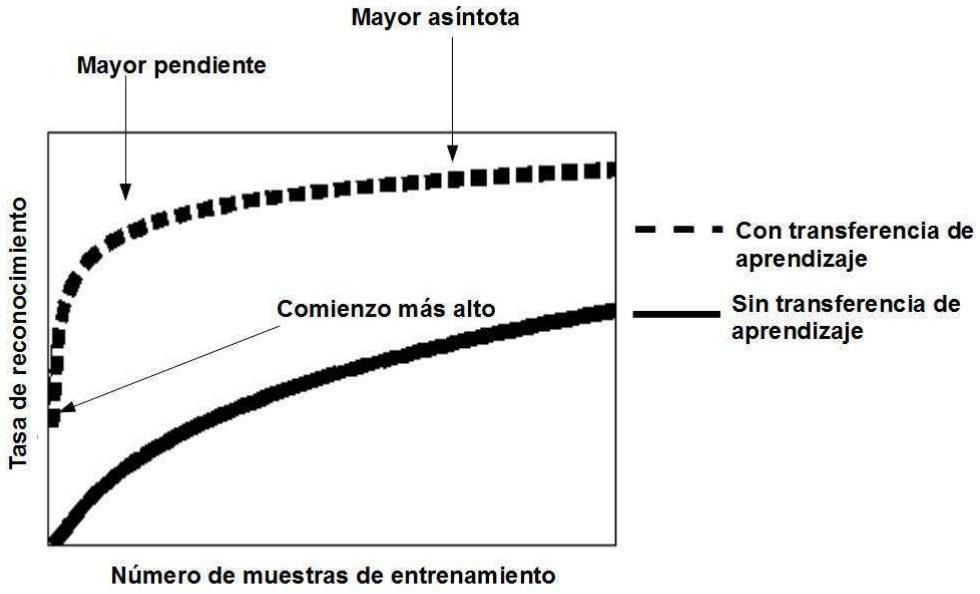


Figura 1.1: Ventajas esperadas de usar transferencia de aprendizaje [TOC10].

dimiento de un método de aprendizaje automático tradicional se esperan observar tres beneficios del primero por sobre el segundo [TOC10]. El primer beneficio esperado es que el método basado en transferencia de aprendizaje debería tener un comienzo más alto que el otro. En otras palabras, el rendimiento del método basado en transferencia de aprendizaje, cuando el número de muestras de entrenamiento de la tarea de destino es uno, o incluso cero, debería ser mayor que el rendimiento de las técnicas tradicionales de aprendizaje automático.

Las otras dos ventajas esperadas de la utilización de un enfoque de transferencia de aprendizaje son tener una mayor pendiente y tener una asíntota superior. Dicho de otra manera, el enfoque de transferencia de aprendizaje debería requerir menos muestras de entrenamiento para aumentar su rendimiento y, al mismo tiempo, cuando una gran cantidad de muestras de entrenamiento están disponibles el rendimiento debería ser mejor. Las ventajas esperadas se representan en la Figura 1.1.

Muchos ejemplos se pueden encontrar en visión por ordenador donde la

transferencia de aprendizaje puede ser realmente beneficiosa. Un ejemplo es el reconocimiento óptico de caracteres, que trata de clasificar una determinada imagen en uno de los caracteres de un alfabeto dado. La mayoría de los métodos se han centrado en el reconocimiento de caracteres de la alfabeto en inglés [LBBH98]. El reconocimiento de caracteres de otros alfabetos, como el francés, implica la recolección de un nuevo conjunto de datos de entrenamiento [GA11]. En ese caso, sería útil transferir el conocimiento en el clasificador que se entrena para la nueva tarea.

La necesidad de transferencia de aprendizaje también surge en el reconocimiento de gestos [GAJ+13], que tiene por objeto el reconocer una instancia de un gesto de un vocabulario de gestos. Por ejemplo, un vocabulario de gestos puede consistir en gestos italianos o señales de árbitros. En este caso, el clasificador necesita predecir el gesto del vocabulario que le corresponde a un vídeo dado. Una vez más, sería interesante mejorar el rendimiento de un sistema mediante la explotación del conocimiento adquirido de vocabularios similares.

1.1 Contribuciones

En esta tesis se presenta un nuevo método para transferir aprendizaje que extiende los bosques de decisión [Bre01, AC12], y lo aplicamos para transferir conocimiento de múltiples tareas origen a una tarea destino determinada. Introducimos dos mecanismos con el fin de transferir el conocimiento de las tareas de origen a la tarea de destino. El primero de ellos es la gananacia de información mixta, que es un regularizador basado en datos. El segundo es la propagación de etiquetas, que deduce la estructura de la variedad del espacio de características. El novedoso proceso de aprendizaje de nuestros bosques de decisión se ilustra en la Figura 1.2. Como podemos ver, nuestra técnica aprende al mismo tiempo de las tareas de origen y de destino. El bosque decisión es entrenado utilizando los conjuntos de entrenamiento de origen y destino de manera combinada.

La primera contribución clave es revisar el criterio para seleccionar los parámetros de cada nodo interno de los bosques de decisión para lograr que

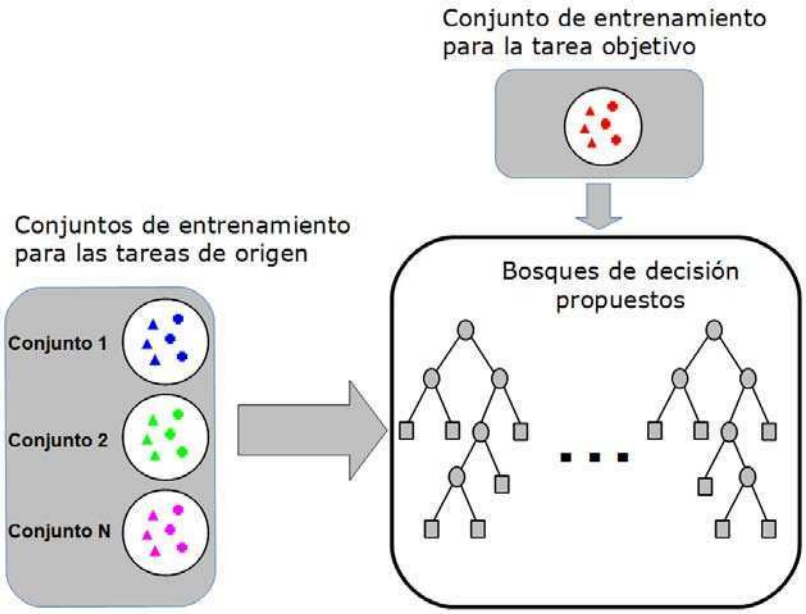


Figura 1.2: Proceso de aprendizaje del método propuesto.

se transfiera conocimiento. El nuevo criterio explota el conocimiento de las tareas de origen y la tarea de destino para encontrar los parámetros para cada nodo interno del bosque de decisión. La información adicional penaliza funciones de división con una ganancia de información grande en la tarea de destino y baja ganancia de información en las tareas de origen. Se probaron algunas propiedades relevantes del nuevo criterio. Las propiedades abordan la pregunta sobre cuántas tareas de origen se requieren para mejorar el rendimiento utilizando los bosques de decisión propuestos.

La segunda contribución clave es propagar las etiquetas a través de las hojas con el fin de inferir la estructura de la variedad del espacio de características. El objetivo de este paso es asignar un modelo predictivo a las hojas que no tienen muestras de entrenamiento de la tarea destino, después de haber entrenado cada uno de los árboles de decisión. Creamos un grafo completamente conectado, para cada árbol en el bosque, donde los nodos son las hojas del árbol y el peso de cada arista tiene en cuenta los datos de entrenamiento que llegan a las hojas. Un supuesto implícito de este paso es que las hojas cercanas deben tener modelos predicitvos similares.

Validamos extensamente nuestro enfoque en dos conjuntos de datos sumamente difíciles. En primer lugar, nuestros experimentos en el conjunto

de datos del desafío ChaLearn [GAJ+12] demuestra que nuestro método no tiene un margen uniforme de mejora con respecto a todas las tareas. Sin embargo, hemos demostrado que cuando hay tareas de origen relacionadas con la tarea de destino, se obtiene una mayor mejora. En segundo lugar, nuestros experimentos en el conjunto de datos MNIST [LBBH98] muestran que los bosques de decisión propuestos superan a los bosques de decisión tradicionales para conjuntos de entrenamiento de diferente tamaño. Además, la brecha entre ambos clasificadores es mayor cuando el tamaño del conjunto de datos de entrenamiento es pequeño.

1.2 Trabajo Relacionado

En lo que sigue vamos a revisar las técnicas de transferencia de conocimiento que han sido aplicadas a problemas de visión por ordenador. Un reciente estudio [PY10] proporciona un panorama general de los avances recientes para clasificación, regresión y agrupamiento. En los últimos años, la comunidad de la visión por ordenador se ha convertido cada vez más interesada en el uso de técnicas de transferencia de aprendizaje, sobre todo para el reconocimiento de objetos [LFW04, STFW05, FFFP06, BU05, TMF07, QCD08, BT10, GLC11, SKFD10, TOC13].

Una variedad de métodos que se han propuesto en la literatura se encuadran dentro del modelo probabilístico generativo [FFFP06, STFW05]. Estos modelos asumen la existencia de relaciones entre las diferentes partes de un objeto durante el proceso de entrenamiento. La idea clave es compartir algunos parámetros o distribuciones a priori entre las categorías de objetos, utilizando el conocimiento de clases conocidas como una referencia genérica para las nuevas clases. La asociación de objetos con las distribuciones de las partes escala de forma lineal en [STFW05], mientras que escala de manera exponencial en [FFFP06].

Por otra parte, los modelos discriminantes han sido adaptados al enfoque de transferencia de aprendizaje [TOC13, DYXY07, YD10, AZ11, LST11, TMF07]. [AZ11] y [TOC13] aplican transferencia de aprendizaje a las máquinas de soporte vectorial. Durante el proceso de entrenamiento de la tarea

destino, un modelo aprendido previamente se introduce como un regularizador en la función de costo. [DYXY07] permite a los usuarios utilizar una pequeña cantidad de datos etiquetados desarrollando un sistema basado en boosting [FS97]. Más tarde, [YD10] extiende [DYXY07] para incorporar múltiples tareas de origen.

Una idea interesante, normalmente conocida como transferencia de característcias [PY10], es la de extraer de las tareas de origen una buena representación para la tarea de destino. Un sistema para el entrenamiento de un clasificador de objetos utilizando un único ejemplo se describe en [Fin05]. La intención es hacer énfasis en las dimensiones relevantes para la clasificación mediante muestras disponibles de tareas relacionadas. Inspirado por el aprendizaje incremental [GLC11] presenta una técnica que crea representaciones intermedias de datos entre la tarea origen y la tarea de destino como puntos en una variedad.

Suponer que las tareas de origen y la tarea de destino comparten algunos parámetros o distribuciones a priori para los hiperparámetros de los modelos se conoce como el enfoque de transferencia por parámetros o por modelos [PY10]. Un método para transferir información de la forma a través de las clases de objetos se presenta en [SGS09]. Más recientemente, se ha propuesto [GF12] transferir la apariencia de los objetos, la distribución de la ubicación dentro de la imagen, y el contexto en el que se encuentran los objetos explotando la jerarquía semántica de ImageNet.

Los enfoques que transfieren instancias [PY10] consideran los datos de origen y de destino de manera conjunta durante el proceso de entrenamiento. [LST11] aumenta los datos de entrenamiento para cada clase transformando muestras de otras clases. Un método para el aprendizaje de nuevas categorías visuales se describe en [QCD08], que usa solamente un pequeño subconjunto de prototipos de referencia para un determinado conjunto de tareas. Como se describió anteriormente, [DYXY07, YD10] propuso un algoritmo basado en boosting que permite que el conocimiento sea transferido de manera efectiva de los antiguos a los nuevos datos. La eficacia del nuevo algoritmo se analiza teórica y empíricamente. En esta tesis, se desarrolla un enfoque de transferencia instancia que explota los datos de origen y de

destino para encontrar los parámetros de cada nodo interno del bosque de decisión.

Pocos investigadores han abordado el problema de transferencia de aprendizaje utilizando los bosques o árboles de decisión. [LSSB09] extiende los bosques de decisión a aprendizaje semi-supervisado. Con el fin de incorporar datos no etiquetados se propone un enfoque de margen máximo, que se optimiza usando una técnica de simulado recocido (*simulated annealing*). [WZCG08] propone tratar los atributos como si fuesen una tarea adicional para sesgar cada árbol de decisión. [PKZ13] propone un nuevo criterio para la dividisión de los nodos para evitar los problemas de deficiencia por rango durante el entrenamiento de los bosques de densidad utilizados para leer los labios. [wLGC07] propone un sistema que aprende una nueva tarea atravesando y transformando un árbol de decisión previamente aprendido para una tarea relacionada. El árbol de decisión es entrenado en la tarea destino utilizando un árbol de decisión parcial inducido por ID3 [Qui86b] . En esta tesis, seguimos un enfoque diferente, en primer lugar se consideran los datos de la tarea de origen y de la tarea destino en forma conjunta cuando construimos cada árbol del bosque de decisión. En segundo lugar, los bosques de decisión reducen la varianza del clasificador agregando los resultados de varios árboles de decisión.

Nuestro enfoque comparte algunas características con el enfoque presentando en [FCGT12], que propone transferir el aprendizaje utilizando una combinación de árboles de decisión C4.5 y boosting. La principal diferencia es que el método propuesto en [FCGT12] reduce la varianza de los árboles de decisión por medio de boosting, lo que ha demostrado ser menos robusto frente al ruido etiqueta en comparación con los bosques de desición propuestos en [Bre01, LSSB09]. Además, nuestro enfoque, utiliza la propagación de etiquetas para aprender la estructura de la variedad del espacio de características, y se asignan modelos predictivos sólo a las hojas de los árboles.

Se ha aplicado en el pasado transferencia de aprendizaje al problema de reconocimiento óptico de caracteres. En [QSCP10] se presenta un método para aprender clasificadores a partir de una colección de tareas relacionadas

en las que cada tarea tiene su propio conjunto de etiquetas. El problema se formula como la maximización de la información mutua entre los conjuntos de etiquetas. Los experimentos en el conjunto de datos MNIST [LBBH98] muestran que el aprendizaje de manera conjunta de múltiples tareas relacionadas mejora significativamente la precisión de la clasificación cuando el tamaño del conjunto de entrenamiento es pequeño. Usando un enfoque diferente, se consigue también un gran margen de mejora en [FCGT12].

Ha habido un creciente interés en la aplicación de técnicas de transferencia de aprendizaje para el reconocimiento de gestos. Un método de transferencia de aprendizaje en el contexto del lenguaje por señas se describe en [FFW07]. Un conjunto de palabras etiquetadas en los datos de origen y de destino es compartido con el fin de construir un clasificador de palabras para un nuevo gesticulardor en un conjunto de palabras destino sin etiquetas. Un método de transferencia de aprendizaje para campos aleatorios condicionales se implementa para explotar la información, tanto en los datos etiquetados y no etiquetados para aprender las características de alto nivel para el reconocimiento de gestos en [LYZH10]. Más recientemente, la competencia de gestos ChaLearn [GAJ+13] proporcionó un punto de referencia de los métodos de aprendizaje que se aplican a la transferencia de aprendizaje de gestos. Varios enfoques enviados a la competencia han sido publicados [MNG13, Lui12, WRLD13].

1.3 Publicaciones

Las contribuciones presentadas en esta tesis has sido publicadas en los siguientes artículos:

- **Norberto Goussies**, Sebastian Ubalde, Francisco Gómez Fernández, Marta Mejail. *Optical Character Recognition Using Transfer Learning Decision Forests* In Proceedings of the IEEE International Conference on Image Processing, 2014 (to appear)

- **Norberto Goussies**, Sebastian Ubalde, Marta Mejail. *Transfer Learning Decision Forests for Gesture Recognition* In Journal of Machine Learning

Research, 2014 (accepted)

1.4 Organización de la Tesis

Esta tesis está organizada de la siguiente manera. Se discute el modelo de bosque de decisión tradicional en el capítulo 2. El nuevo modelo de bosques de decisión se describe en el capítulo 3, ilustramos su rendimiento en algunos conjuntos de datos artificiales, y probamos algunas de las propiedades de la ganancia de información mixta. Se presentan dos aplicaciones a problemas de visión por ordenador de nuestro nuevo modelo de bosques de decisión en el capítulo 4, y demostramos que se logran tasas de reconocimiento superiores cuando el conjunto de entrenamiento es pequeño. Por último, el capítulo 5 detalla nuestras conclusiones.

Chapter 2

The Decision Forest Model

In this chapter we discuss the decision forest model [AC12, Bre01] for the classification problem. We start with general intuition of the decision forest model, then we formalize the classification problem. Finally, we describe the algorithms and some important theoretical results of the decision forest model. The decision forest model has the following advantages:

- they naturally handle classification problems with multiple classes
- the output is a probability
- they generalize well at testing
- they can be efficiently implemented in parallel.

2.1 The Intuition

Let us suppose we want develop a system that automatically recognizes the type of scene captured in a photograph. This problem can be expressed as a classification problem, where the input is a set of relevant features for the task and the output is a discrete, categorical label (*e.g.* alleyway, mountain, brewery, *etc*).

The approach followed by the decision forest model is to create a committee of slightly different decision trees. The decision trees perform good tests in a good order. For example, to identify if a photograph corresponds

to a mountain scene, a possible test that is relevant is whether or not the sky is present. The decision trees in the committee perform different tests in different order, but all of them return a predictive model. The prediction of the decision forest is computed combining the predictive models of all the decision trees.

2.2 Bayesian Decision Theory

The goal in a classification problem is to learn a classifier $f : \mathcal{X} \longrightarrow \mathcal{Y}$ that assigns to each input vector $\mathbf{x}$ of the input space $\mathcal{X} = \mathbb{R}^d$ a category $\mathbf{y}$ out of a finite number of discrete categories $\mathcal{Y} = \{1, \ldots, C\}$. Probability theory provides a consistent framework for the quantification and manipulation of uncertainty which arises in many ways in a classification problem. Therefore, we regard X, Y as random variables and the joint probability distribution $p(\mathbf{x}, \mathbf{y}) = P(X = \mathbf{x}, Y = \mathbf{y})$ contains a complete summary of the uncertainty associated with these variables, but it is unknown. Instead, the classifier $f(\mathbf{x})$ is learned using a training set $\mathcal{D} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathbb{R}^d, \mathbf{y} \in \mathcal{Y}\}$. We assume that $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$ are independent and identically distributed realizations of X, Y, following the joint distribution $p(\mathbf{x}, \mathbf{y})$.

The probability that the classifier $f(\mathbf{x})$ makes a mistake on a future realization from the distribution $p(\mathbf{x}, \mathbf{y})$ is given by the generalization error:

$$L(f) \triangleq P(f(X) \neq Y) \tag{2.1}$$

which defines the criterion according to which we will assess the quality of a learned f obtained from data.

The optimal Bayes classifier f^* [Was04], if the distribution $p(\mathbf{x}, \mathbf{y})$ is known, is given by:

$$f^*(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}|\mathbf{x}) \tag{2.2}$$

which predicts the label $\mathbf{y} \in \mathcal{Y}$ for which the probability $p(\mathbf{y}|\mathbf{x})$ is the highest. To explain why this decision procedure is optimal, let us calculate the probability of error when we make a prediction. Whenever we observe a particular $\mathbf{x}$ we have:

$$P(f^*(X) \neq Y | X = \mathbf{x}) = 1 - P(f^*(X) = Y | X = \mathbf{x}) = 1 - p(\mathbf{y}|\mathbf{x}). \tag{2.3}$$

Therefore, for a given $\mathbf{x}$ we minimize the probability of error by deciding the $\mathbf{y}$ that has the highest probability $p(\mathbf{y}|\mathbf{x})$. The optimal Bayes classifier f^* defined in (2.2) has the following property:

$$L(f^*) \leq L(f), \ \forall f. \tag{2.4}$$

The Bayes classifier f^* depends on unknowns quantities so we need to use the training data $\mathcal{D}$ to find some approximation to the Bayes rule.

As a consequence of the Bayes optimal classifier, the classification problem is broken down into two separate stages, the inference stage and the decision stage. In the former stage we use the training data $\mathcal{D}$ to learn a predictive model for the conditional distribution $p(\mathbf{y}|\mathbf{x})$. In the later stage we use these estimation of the posterior probabilities to make class assignments as follows:

$$f(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} \widehat{p}(\mathbf{y}|\mathbf{x}) \tag{2.5}$$

where $\widehat{p}(\mathbf{y}|\mathbf{x})$ is the estimation of the conditional distribution $p(\mathbf{y}|\mathbf{x})$. The difference between the error of the optimal Bayes classifier $L(f^*)$ and the error of the learned classifier $L(f)$ is the excess of error of the classifier f and it can be upper bounded as follows [DGL96]:

$$L(f) - L(f^*) \leq \sum_{\mathbf{y}\in\mathcal{Y}} \int_{\mathbb{R}^d} |p(\mathbf{y}|\mathbf{x}) - \widehat{p}(\mathbf{y}|\mathbf{x})| p(\mathbf{x}) d\mathbf{x}. \tag{2.6}$$

This upper bound states that if $\widehat{p}(\mathbf{y}|\mathbf{x})$ is close to the real a posteriori probability $p(\mathbf{y}|\mathbf{x})$ in $L1$-sense, then the error probability of decision f is near the optimal decision f^*. Even though this upper bound is not possible to compute in real life, since we do not know the real a posteriori probability $p(\mathbf{y}|\mathbf{x})$, it emphasizes the importance of estimating $p(\mathbf{y}|\mathbf{x})$.

2.3 Decision Trees

The goal of a decision tree F is to approximate a complex distribution $p(\mathbf{y}|\mathbf{x})$ using a divide and conquer strategy. The basic idea is to partition the instance space into a small number of regions, each of which allows to make

a reliable prediction by having a simple class distribution. The most compelling reason for using decision tree is to explain complicated data and to have a classifier that is easy to analyze and understand.

A decision tree F is a strictly binary tree in which each node k represents a subset R_k in the instance space $\mathbb{R}^d$ and all the leaves ∂F form a partition $\mathcal{P}$ of $\mathbb{R}^d$. Also, each node has exactly two or zero children. In addition, each leaf $k \in \partial F$ of a decision tree F has a predictive model associated with it: $p_F(\mathbf{y}|\mathbf{x} \in R_k)$. The internal nodes $k \in F^\circ$ of a decision tree have a linear split function: $h(\mathbf{x}, \boldsymbol{\theta}_k) = \mathbf{x} \cdot \boldsymbol{\theta}_k$, where $\boldsymbol{\theta}_k$ are the parameters of node k and, $\mathbf{x} \cdot \boldsymbol{\theta}_k$ is the inner product between the vectors $\mathbf{x}$ and $\boldsymbol{\theta}_k$. The subset represented by the left child k_L of node k is defined as $R_{k_L} = R_k^L = R_k^{\boldsymbol{\theta}_k^-} = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} \in R_k \wedge h(\mathbf{x}, \boldsymbol{\theta}_k) < 0\}$ and, similarly, we define $R_{k_R} = R_k^R = R_k^{\boldsymbol{\theta}_k^+} = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} \in R_k \wedge h(\mathbf{x}, \boldsymbol{\theta}_k) \geq 0\}$ as the subset represented by the right child k_R. The training set reaching node k is defined as $\mathcal{D}_k = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{D} | \mathbf{x} \in R_k\}$.

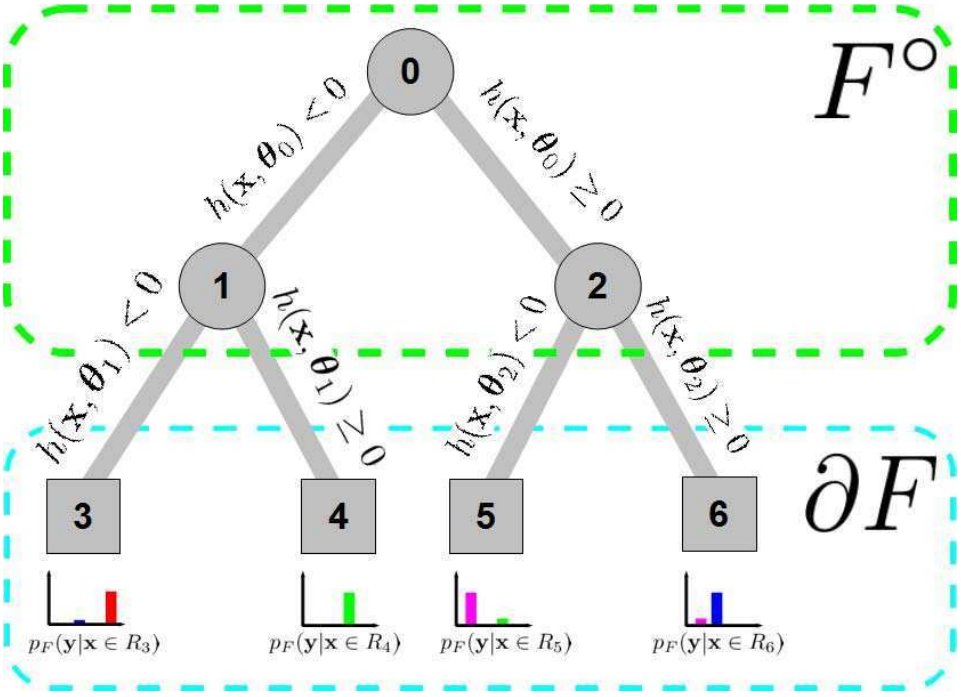
We illustrate the components of a decision tree in Figure 2.1 using a two dimensional toy example. The internal nodes are denoted with circles and they are inside the green box. The leaves or external nodes are denoted with squares and they are inside the light blue box. A decision tree is a tree where each internal node k stores a split function $h(\mathbf{x}, \boldsymbol{\theta}_k)$ to be applied to test instance. Additionally, each leaf k stores a predictive model $p_F(\mathbf{y}|\mathbf{x} \in R_k)$. The partition $\mathcal{P}$ defined by a decision tree F is composed of different regions whose boundaries are defined by the split functions of the internal nodes.

2.3.1 Objective Function

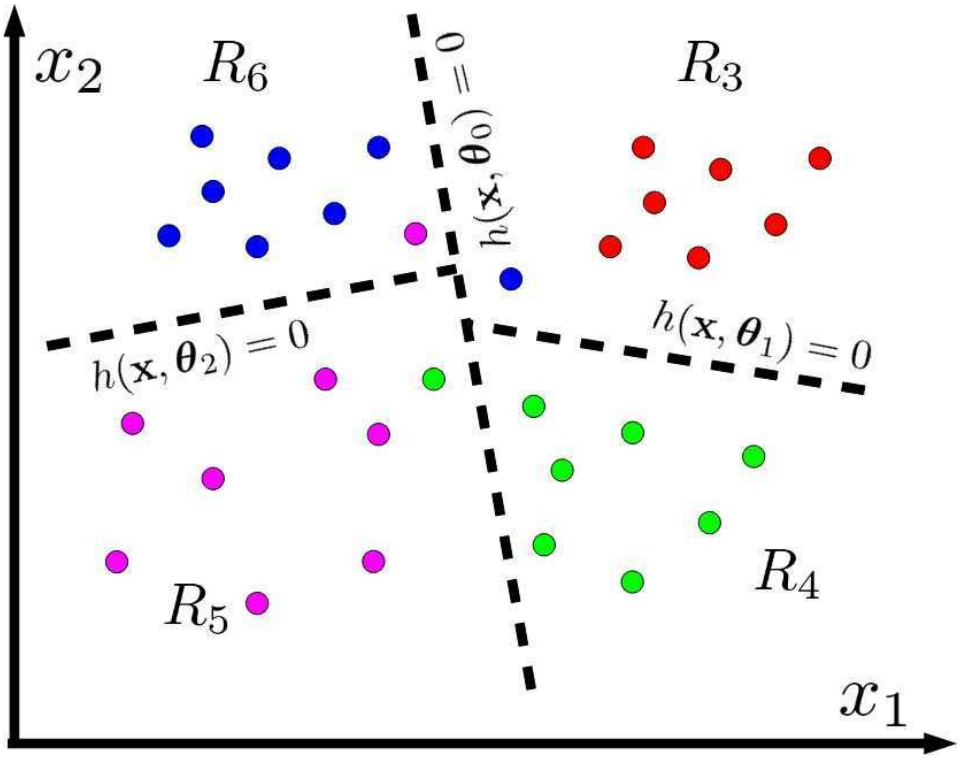
In the previous section we informally described the basic ideas behind the decision trees. In this section, we formalize those ideas in order to define an objective function to guide the learning process of a decision tree. Let's start by the notion of having a simple class distribution in a region R_k.

Given the training set $\mathcal{D}_k$ associated with the region R_k the empirical class distribution is defined as:

$$\widehat{p}(\mathbf{y}|\mathbf{x} \in R_k) = \widehat{p}_{\mathcal{D}_k}(\mathbf{y}) = \frac{1}{|\mathcal{D}_k|} \sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{D}_k} \delta_{\mathbf{y}'}(\mathbf{y}) \tag{2.7}$$



(a) A decision tree F with its different components. See text for description.



(b) Partition represented by the leaves of the decision tree on the left.

Figure 2.1: Decision tree components.

where $\delta_{\mathbf{y}'}(\mathbf{y})$ is the Kronecker delta.

Different measures to formalize the notion of a simple class distribution associated with a region R_k can be defined:

- Misclassification error:

$$\mathcal{E}(R_k) = \mathcal{E}(\widehat{p}_{\mathcal{D}_k}) = 1 - \max_{\mathbf{y}\in\mathcal{Y}} \widehat{p}_{\mathcal{D}_k}(\mathbf{y}) \tag{2.8}$$

- Gini index:

$$\mathcal{G}(R_k) = \mathcal{G}(\widehat{p}_{\mathcal{D}_k}) = \sum_{\mathbf{y}\in\mathcal{Y}} \widehat{p}_{\mathcal{D}_k}(\mathbf{y})(1 - \widehat{p}_{\mathcal{D}_k}(\mathbf{y})) \tag{2.9}$$

- Entropy:

$$\mathcal{H}(R_k) = \mathcal{H}(\widehat{p}_{\mathcal{D}_k}) = -\sum_{\mathbf{y}\in\mathcal{Y}} \widehat{p}_{\mathcal{D}_k}(\mathbf{y}) \log(\widehat{p}_{\mathcal{D}_k}(\mathbf{y})) \tag{2.10}$$

These measures are minimized when the distribution is peaked. For two classes, if p is the proportion of elements that belong to the first class, these three measures are $1 - \max(p, 1-p)$, $2p(1-p)$ and $-p\log p - (1-p)\log(1-p)$. We compare them in Figure 2.2. For three classes, the expression is more complex and we need to define p and q as the proportion of the first and second class respectively. We plot the level lines of the functions in Figure 2.3.

All three are similar, since they have the same global maximums and minimums, but entropy and the Gini index are differentiable, and hence more suitable to numerical optimization. In addition, entropy and the Gini index are more sensitive to changes in the node probabilities than the misclassiffication rate. Let's analyze in more detail the misclassification error, for the case of three classes and $q = 0.5$. We plot the values of the three measures when $q = 0.5$ in Figure 2.4. We see that the misclassification error is a constant line at 0.5, while the Gini index and the entropy take different values. The Gini index and the entropy are maximized when the class distribution is $(0.25, 0.5, 0.25)$ and minimized when the class distribution are $(0.5, 0.5, 0)$ or $(0, 0.5, 0.5)$. The former class distribution is less preferable since the probability of the three classes is different to zero. For this reason,

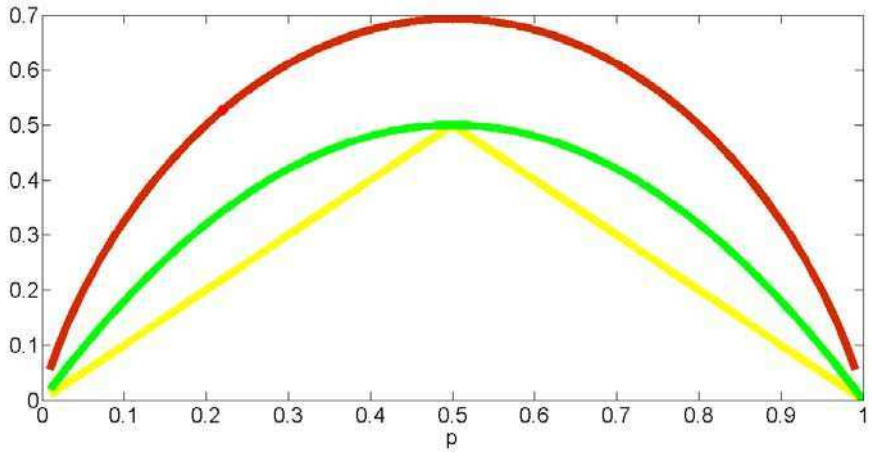


Figure 2.2: Comparison of misclassification error (yellow), Gini index (green) and entropy (red) for two classes.

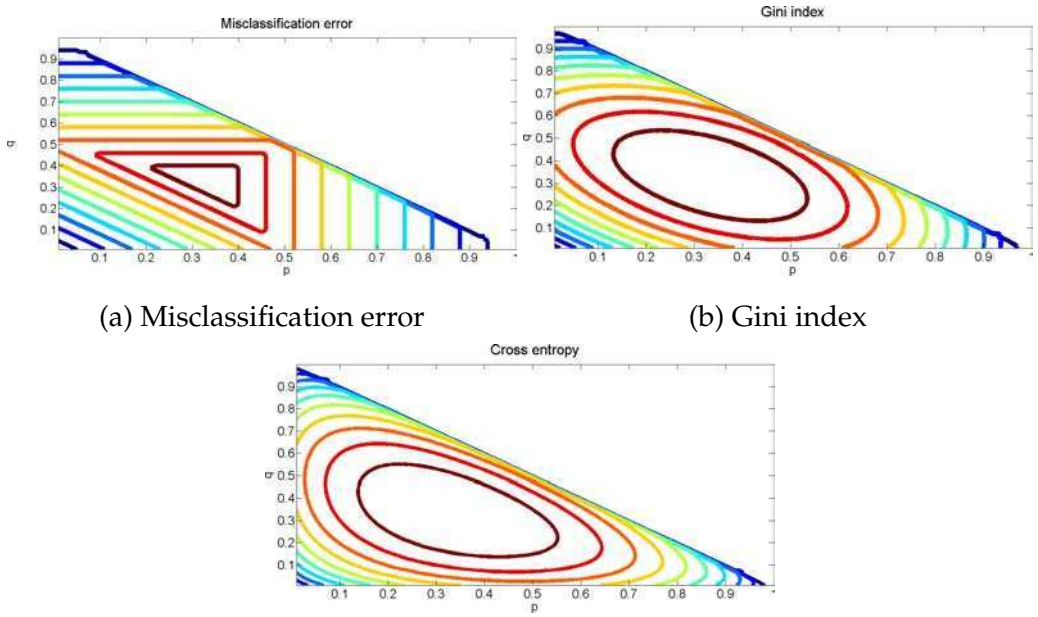
Gini index and entropy are the most common measures used when learning a decision tree. In this thesis we restrict our attention to the entropy measure, which is the one that has been analyzed more in depth in the literature.

A natural objective function that assigns high scores to decision trees with simple regions is the total information gain:

$$\mathcal{I}(F) = \mathcal{H}(R_{root}) - \mathcal{H}(F) = \mathcal{H}(\widehat{p}_{\mathcal{D}}) - \sum_{k \in \partial F} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathcal{H}(\widehat{p}_{\mathcal{D}_k}) \tag{2.11}$$

which is the difference between the entropy of the class distribution in the entire training set and the weighted average of the entropies in the leaves of the decision tree F. It is important to note that the total information gain is computed using the estimated class distributions $\widehat{p}$ and not the true distribution p. Regardless, many of the relevant properties that are true for the information gain based on the true distribution are also true for the total information gain computed using the estimated class distribution. For example it holds that $0 \leq \mathcal{I}(F) \leq \mathcal{H}(\widehat{p}_{\mathcal{D}})$.

The total information gain can be easily maximized growing a tree that has only one training sample in each leaf. A decision tree with that property would be very unstable since a small change in the training set would generate a big change in the structure of the tree. The problem is that the decision tree has a large number of leaves. Thus, we need to apply a constraint that limits the number of regions.



(a) Misclassification error

(b) Gini index

(c) Entropy

Figure 2.3: Comparison of misclassification error, gini index and entropy for three classes.

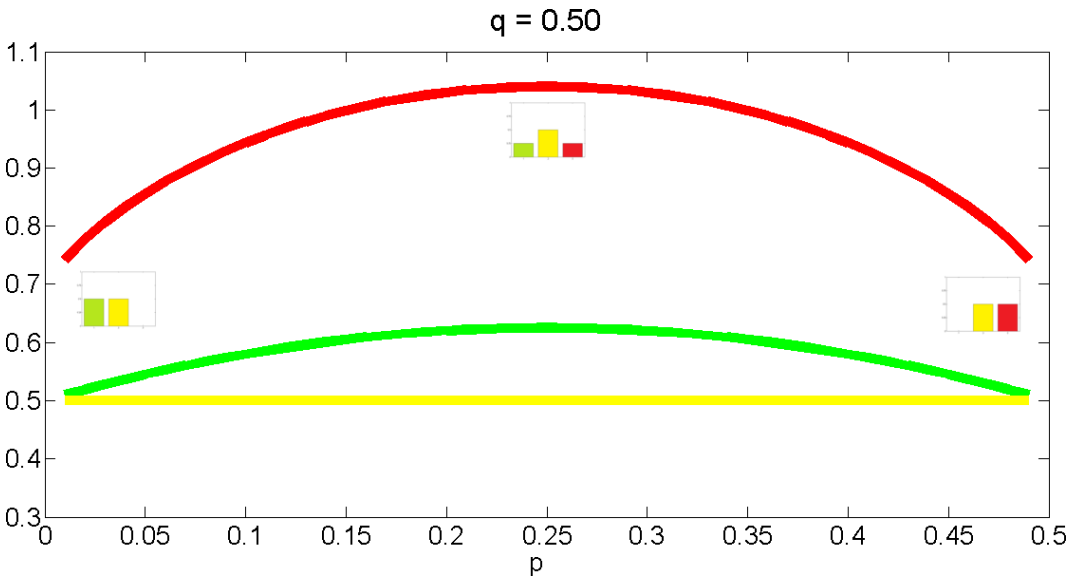


Figure 2.4: Comparison of misclassification error (yellow), gini index (green) and entropy (red) for three classes, when $q = 0.5$.

A possible way to limit the number of regions is penalizing the maximum number of tests:

$$\mathcal{D}(F) = \max_{k \in \partial F} d(k) \tag{2.12}$$

where $d(k)$ is the depth of the node k. The maximum number of tests does not take into account how many times each branch will actually be used but constraints all the branches equally. A less restrictive constraint to limit the number of regions is the average number of tests:

$$\mathcal{A}(F) = \sum_{k \in \partial F} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} d(k) \tag{2.13}$$

Combining the total information gain $\mathcal{I}(F)$ and the maximum number of tests $\mathcal{D}(F)$ we obtain an useful objective function:

$$F = \arg\max_{F:\mathcal{D}(F) \leq \kappa} \mathcal{I}(F) \tag{2.14}$$

where κ is a parameter of the learning algorithm that constraints the decision tree to have a small number of regions. The objective function of (2.14) is maximized when a decision tree partitions the instance space into a small number of regions, each having a simple class distribution.

2.3.2 Stepwise Uncertainty Reduction

The optimization problem defined in (2.14) express exactly the decision tree that we would like to find. However, it is very difficult to solve since it requires to simultaneously find the structure of the decision tree and the parameters of the internal nodes. There is a large body of work analyzing this optimization problem and several variants, see for example [GMP00] and the references therein.

The general conclusion is that solving the optimization problem in (2.14) is NP-complete [GMOS95] but good approximations can be achieved using a greedy method [MS95]. The main idea is to find the best parameters $\boldsymbol{\theta}_k$ of each node $k \in F$ separately, in a top-down fashion. In order to find the best parameters $\boldsymbol{\theta}_k$ of an isolated node $k \in F$ we maximize the information gain

of each node parameter $\boldsymbol{\theta}_k$ separately :

$$\mathcal{I}(\boldsymbol{\theta}_k) = \mathcal{I}(\mathcal{D}_k, \boldsymbol{\theta}_k) = \mathcal{H}(\widehat{p}_{\mathcal{D}_k}) - \frac{|\mathcal{D}_k^{\boldsymbol{\theta}_k^+}|}{|\mathcal{D}_k|}\mathcal{H}\left(\widehat{p}_{\mathcal{D}_k^{\boldsymbol{\theta}_k^+}}\right) - \frac{|\mathcal{D}_k^{\boldsymbol{\theta}_k^-}|}{|\mathcal{D}_k|}\mathcal{H}\left(\widehat{p}_{\mathcal{D}_k^{\boldsymbol{\theta}_k^-}}\right) \tag{2.15}$$

where $\mathcal{D}_k^{\boldsymbol{\theta}_k^+} = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_k | h(\mathbf{x}, \boldsymbol{\theta}_k) \geq 0\}$ is the training set reaching the right child and $\mathcal{D}_k^{\boldsymbol{\theta}_k^-} = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_k | h(\mathbf{x}, \boldsymbol{\theta}_k) < 0\}$ is the training set reaching the left child. The information gain defined in (2.15) is a simplification of the total information gain (2.11) for the case of a tree with only one internal node.

The pseudo-code of the greedy algorithm is presented in Algorithm 1, which takes as a parameter a training set $\mathcal{D}$ and a pointer to the root of the tree F. The procedure starts at the root and grows a tree recursively. For each node k, it checks if the stopping criteria is met. If it does, then the node is an external ode and a predictive model is estimated and stored as an array of size $|\mathcal{Y}|$ in $k.\widehat{p}$. Otherwise it searches for an optimal parameter for the split function $h(\mathbf{x}, \boldsymbol{\theta}_k)$ and splits the set $\mathcal{D}$ into two smaller sets $\mathcal{D}^L, \mathcal{D}^R$. Finally the procedure calls itself twice to learn the left and right sub-trees using $\mathcal{D}^L, \mathcal{D}^R$ respectively.

Algorithm 1 Step Uncertainty Reduction

Require: Training set $\mathcal{D}$

1: **function** TRAIN-DECISION-TREE($k, \mathcal{D}$)
2: **if** stopping criteria is met **then**
3: $k.\widehat{p} \leftarrow \widehat{p}_{\mathcal{D}}$
4: **return**
5: **end if**
6: $k.\boldsymbol{\theta} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{I}(\boldsymbol{\theta})$
7: $(\mathcal{D}^L, \mathcal{D}^R) \leftarrow split(\mathcal{D}, k.\boldsymbol{\theta})$
8: *Train-Decision-Tree*($k.left, \mathcal{D}^L$)
9: *Train-Decision-Tree*($k.right, \mathcal{D}^R$)
10: **end function**

The stopping criteria in line 2 depends on the constraint chosen to limit the number of regions of the tree. The simplest stopping criteria consist in

checking that the depth is larger than a given threshold and is the consistent with the constraint of Eq. (2.12). However in most cases, the stopping criteria is composed of several conditions, including the size of the training set reaching the node, the depth of the node and, the minimum information gain is not above a certain threshold [AC12].

2.3.3 Randomized Node Optimization

Different approaches have been proposed to maximize the information gain in line 6 of the Algorithm 1. The main problem is the high computational complexity of this step, since there are huge number of distinct hyperplanes that divide the training set into two non-overlapping subsets. An upper bound, derived in [MKS94] states that there are at most $2^d\binom{n}{d}$ since every subset of size d from the n points can define a d-dimensional hyperplane, and each such hyperplane can be rotated slightly in $2d$ directions to divide the set of d points in all possible ways. Thus, exhaustive search is not a feasible option to minimize the information gain. In [HKS93], the authors have shown that the problem of searching the best hyperplane using the number of misclassified examples is NP-hard.

A key aspect of decision forests is their approach to the maximization of information gain. Randomness is injected during the maximization of the information gain leading to de-correlation between individual tree predictions. As a consequence, the generalization of the decision forests is improved. The basic idea is to restrict the possible values of $\boldsymbol{\theta}_k$ to a small subset $\mathcal{T}_k \subset \mathbb{R}^d$ when maximizing the information gain of node k:

$$k.\boldsymbol{\theta} \leftarrow \arg\max_{\boldsymbol{\theta} \in \mathcal{T}_k} \mathcal{I}(\boldsymbol{\theta}). \tag{2.16}$$

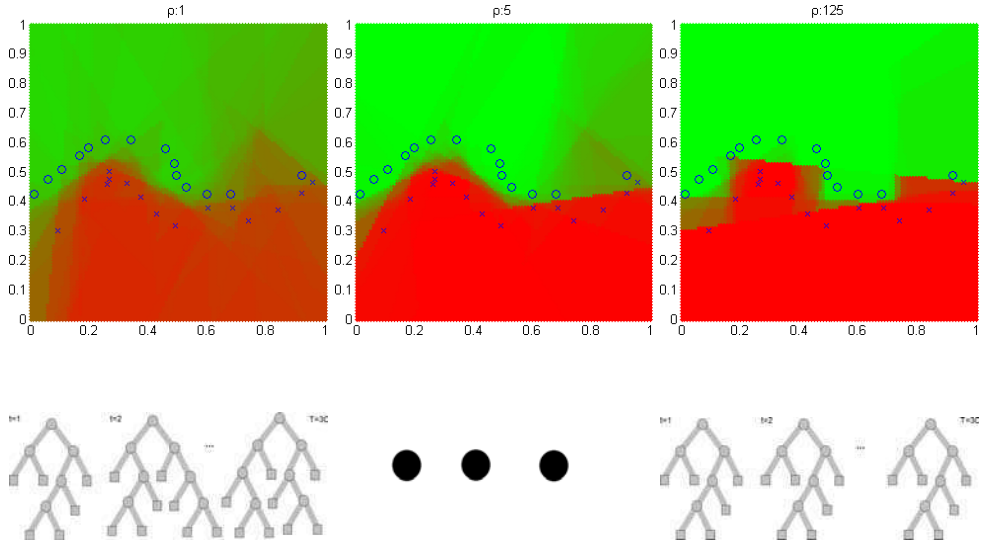
The size of the set $\mathcal{T}_k$ is fixed for all the nodes k in the forest and is noted as ρ. The elements of the set $\mathcal{T}_k$ are chosen at random for each node from the set of all possible parameters. To emphasize that the elements of the set $\mathcal{T}_k$ are chosen at random, we write when necessary, $\mathcal{T}_k(\Theta_k)$ where Θ_k is a random variable that defines elements of the set. Moreover, the random variable $\boldsymbol{\Theta}$ defines the elements of all the sets $\mathcal{T}_k$ in a tree $F(\boldsymbol{\Theta})$. Thus, in a tree $F(\boldsymbol{\Theta})$ the

random variable $\boldsymbol{\Theta}$ is used to determine the search spaces in all the nodes of the tree. Also, the random variable $\boldsymbol{\Theta}$ is assumed to be independent of the training data.

The parameter ρ controls the amount of randomness in a forest. For $\rho = 1$ we get the maximum randomness, as we increase ρ the randomness is reduced. The difference between a large and a small ρ is depicted in Figure 2.5 using a 2D toy example. We train a decision forest for different values of $\rho = 1, 5, 125$ using the points in blue as a two class training set. In this visualization the colour associated with each test point is a linear combination of the colours (red and green) corresponding to the two classes. The mixing weights are proportional to the posterior probabilities obtained using the decision forest. Thus, intermediate, mixed colours correspond to regions of high uncertainty and low predictive confidence whereas pure colors correspond to regions with high confidence.

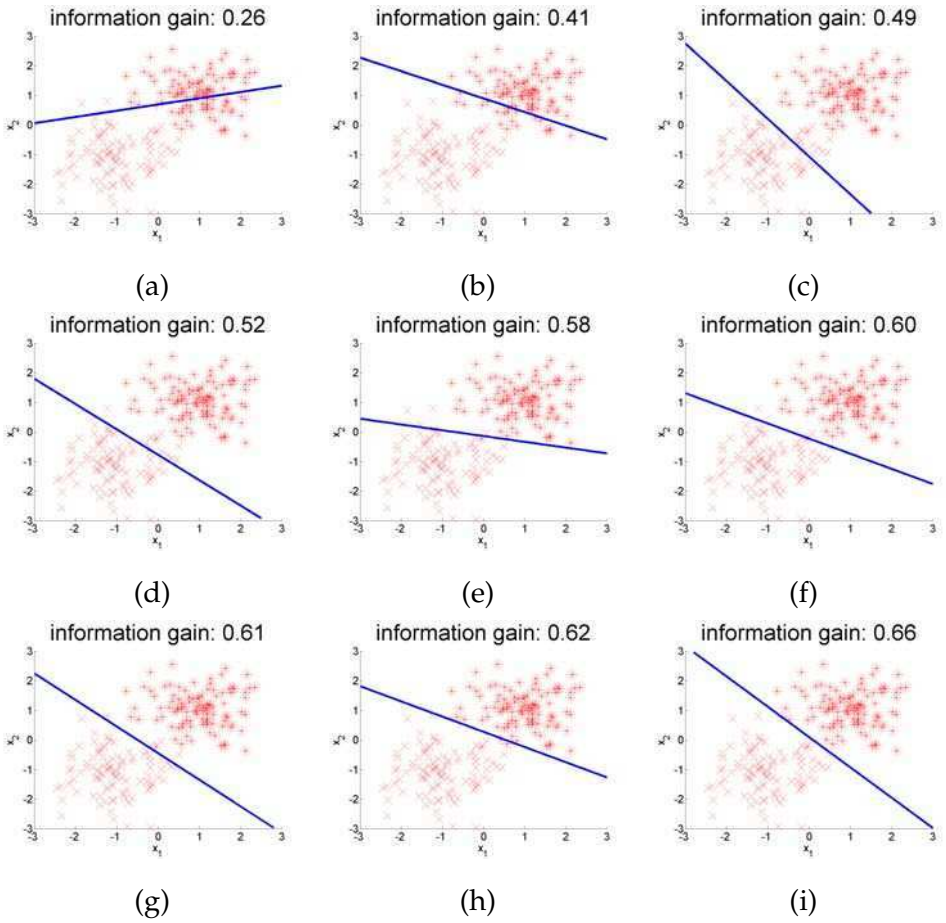
Additionally, Figure 2.5 shows the structure of the trees for each value of ρ. We can observe that there is a high diversity of trees when $\rho = 1$ and as we increase ρ that diversity is lost. This is because when we maximize the information gain using a large ρ we find the similar hyperplanes for each node in different trees of the forest. Obtaining only one type of tree when ρ is large enough, which corresponds to choosing the best $\boldsymbol{\theta}$ for each node. We can observe that, in this case (Figure 2.5c), the forest behaves like a single decision tree. Moreover, even when there are no errors in the training set, the decision boundary is not smooth hence we expect to have a larger generalization error.

Figure 2.6 illustrates the maximization of the information gain using the exhaustive search method. Let us suppose that we have $2D$ training sample with only two labels and we are searching for the optimal parameters $\boldsymbol{\theta}$ of the first node of a decision tree and that $\rho = 9$. In Figs. 2.6a - 2.6i we show nine random splits of the training data-set and their corresponding information gain. In this case the exhaustive search, after comparing the information gain of each of the nine random splits, will choose the split in Fig. 2.6i, since it is the one that has the largest information gain.



(a) $\rho = 1$ (b) $\rho = 5$ (c) $\rho = 125$

Figure 2.5: Decision forest for different values of ρ.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2.6: Information gain associated with nine different splits (blue lines) for a simple $2D$ dataset.

2.3.4 Decision Tree Testing

Given an input vector $\mathbf{x}$ and a decision tree F the class distribution $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ associated with $\mathbf{x}$ is defined as:

$$\widehat{p}_F(\mathbf{y}|\mathbf{x}) = \sum_{k \in \partial F} \widehat{p}_{\mathcal{D}_k}(\mathbf{y}) 1_{R_k}(\mathbf{x}) \tag{2.17}$$

where the function 1_A is the indicator function defined as:

$$1_A(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in A \\ 0 & \text{if } \mathbf{x} \notin A \end{cases} \tag{2.18}$$

Since the regions R_k represented by the leaves of the tree F form a partition of the instance space $\mathbb{R}^d$ only one term in the sum of (2.17) will be different to zero. Thus, if $l_F : \mathbb{R}^d \longrightarrow \partial F$ is the function that, given an input vector $\mathbf{x}$ returns the leaf such that $\mathbf{x} \in R_{l_F(x)}$ we can rewrite $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ as follows:

$$\widehat{p}_F(\mathbf{y}|\mathbf{x}) = \widehat{p}_{\mathcal{D}_{l_F(x)}}(\mathbf{y}) \tag{2.19}$$

Therefore, the class distribution $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ for an input vector $\mathbf{x}$ is the empirical class distribution of the samples of the training set reaching the same node as $\mathbf{x}$. The expression for $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ given in (2.19) is simpler to compute than the one given in (2.17) since we only use the empirical class distribution of the node $l_F(\mathbf{x})$. The pseudo-code for computing $l_F(\mathbf{x})$ is presented in Algorithm 2, which takes as a parameter a pointer to the root of the decision tree F and the input vector $\mathbf{x}$. The procedure starts at the root of the decision tree and evaluates its split function. Depending on the result of the binary test the current node pointer is updated with a reference to the right or left child. This process is repeated until the current node is a leaf. Using the result of Algorithm 2 it is easy to obtain the empirical class distribution $\widehat{p}_{\mathcal{D}_{l_F(x)}}(\mathbf{y})$ since it was stored in the node as an array during the training stage.

2.4 Ensemble Model

A decision forest $\mathcal{F} = \{F(\Theta_1), \ldots, F(\Theta_T)\}$, is defined as an ensemble of T decision trees F that are trained independently (and possible in parallel).

Algorithm 2 Given $\mathbf{x}$ and F find the leaf associated with $\mathbf{x}$

Require: instance vector $\mathbf{x}$ and pointer to the root of decision tree F

```
1:  function FIND-LEAF(x, root)
2:      current_node ← root
3:      while current_node is not a leaf do
4:          if h(x, θ_current_node) < 0 then
5:              current_node ← current_node.left
6:          else
7:              current_node ← current_node.right
8:          end if
9:      end while
10:     return current_node
11: end function
```

The difference between the different trees in the forest is obtained by the degree of randomness ρ and the random variable $\mathbf{\Theta}$ used to grow each tree (Section 2.3.3). A test sample $\mathbf{x}$ is pushed simultaneously through all trees in the forest, using Algorithm 2, obtaining a prediction model $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ for each tree. Then, the prediction of the forest is computed combining all the forest predictions:

$$\widehat{p}_{\mathcal{F}}(\mathbf{y}|\mathbf{x}) = \frac{1}{T}\sum_{F\in\mathcal{F}} \widehat{p}_F(\mathbf{y}|\mathbf{x}). \tag{2.20}$$

The bias of the class distribution $\widehat{p}_{\mathcal{F}}(\mathbf{y}|\mathbf{x})$ is the same as the bias of the class distribution of each individual tree $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ since the expected value of the average of identically distributed random variables is the same as the expected value of any of the random variables. And, if the decision trees are grown sufficiently deep then, the bias of its class distribution $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ is low. As a consequence, if the decision trees of a decision forest are grown sufficiently deep the bias of its class distribution $\widehat{p}_{\mathcal{F}}(\mathbf{y}|\mathbf{x})$ is low.

On the other hand, the variance of the class distribution of decision forest $\widehat{p}_{\mathcal{F}}(\mathbf{y}|\mathbf{x})$ has a more complex expression. Let σ^2 be the variance of the class distribution of each individual decision tree, keeping the training set fixed and let α be the correlation between class distribution of individual decision

trees. The variance of the class distribution $\widehat{p}_{\mathcal{F}}(\mathbf{y}|\mathbf{x})$ is given by:

$$\sigma^2\alpha + \frac{1-\alpha}{T}\sigma^2 \tag{2.21}$$

As more decision trees are added to the decision forest, the second term vanishes, but the first term remains. Thus, the variance is reduced proportionally to α. The key idea is, therefore, to grow decision forests reducing the correlation without increasing the variance too much.

2.5 Related Work

The framework described in this chapter contains some of the most common components in the literature of decision forests and trees. In this section we review some of the most relevant ideas in the area. One of the seminal works on decision trees is the Classification and Regression Trees (CART) book [BFSO84] where the authors explain how to apply decision trees for classification and regression problems. A well known result [HR76] is that constructing optimal binary decision trees is NP-Complete, thus the training algorithm in [BFSO84] finds an approximation.

Many variants of decision trees have been introduced since [BFSO84]. Much of the work has concentrated on the type of split function in each internal node. A common choice is to use axis-parallel linear splits [BFSO84, Qui86a, Qui93] in which a threshold and a feature dimension are chosen and samples are assigned to the branches accordingly. These trees can be extremely fast at test time and finding the optimal threshold, once the feature has been determined, is also efficiently solved. The disadvantage is that this type of split is usually very limited thus, deeper decision trees are required.

The majority of the tree construction methods use a linear split function [MKS94, AC12, BU95, MKS+11, TD05] in which samples are assigned to the branches depending on which side of a hyperplane they fall in. In this case, the hyperplanes are not necessary parallel to any axis in the feature space. Additionally, the internal nodes of a decision tree can partition collinear data which is not possible with axis parallel linear splits. Unfor-

tunately, finding the best hyperplane that splits a training set is computationally more demanding than finding the best axis-parallel split [MKS94]. Thus, much research has focused on this issue. A randomized hill-climbing algorithm is proposed in [MKS94] which is an improvement over [BFSO84]. An interesting bound on the approximation error between the optimal hyperplane and a the one found using a random based search is obtained in [GMOS95]. Nevertheless, as explained in [HTF03] smalls errors propagate in decision trees due its hierarchical nature, thus even a small error in the root can generate large overall errors. A large number of theoretical properties concerning the consistency of decision trees are proven in [DGL96].

Decision forests were proposed by different authors as a method to obtain higher accuracies than the ones obtained with a single tree. An ensemble of multiple decision trees is constructed systematically by pseudo-randomly selecting a subsets of features [Ho98]. A new approach to shape recognition is proposed in [AG97], in which multiple trees are grown to recognize shapes. An extensive experimental comparison between decision forests and several other classification and regression techniques is given in [Bre01,CKY08]. The experiments show that decision forests achieve higher accuracies in general.

The statistical properties of the decision forests are still under active investigation and a few interesting results are known. Although the random forest algorithm appears simple, it is difficult to analyze and theoretical work focuses on stylized versions of the algorithm used in practice. A major difference with decision trees is that consistency of decision trees is proved letting the number of observations in each terminal node become large [DGL96]. However decision forests are generally built to have a small number of training samples in each terminal node and a large number of trees.

Much research has focused on analyzing the properties of the regression decision forests. A main result is the connection between regression random forests and weighted Layered Nearest Neighbor (LNN) [LJ06]. Additionally, [BD10] study the consistency of uniformly weighted LNN re-

gression estimates and discuss their link with random forests estimates. A short draft [Bre04] presenting a simple model for decision forest served as a good starting point to study the statistical properties of regression decision forests. Later, a deeper analysis has been presented in [Bia12], expanding the ideas of [Bre04]. Another variant of regression decision forest is presented in [DMdF14] with a proof of consistency and an experimental comparison between the algorithm used in practice and the presented in the paper.

The extension of these results to the classification decision forests is not trivial, consistency of two simple random forest classifiers is proven in [BDL08]. But the consistency of the algorithm proposed in [Bre01] has not been established. Moreover, [BDL08] shows that for some pathological distributions the algorithm proposed in [Bre01] is not consistent.

2.6 Resumen

En este capítulo se introdujo el modelo de bosques de decisión para el problema de clasificación. Se utilizó como punto de partida la teoría de decisión bayesiana, la cual fue explicada en la Sección 2.2. En la cual se formalizaron los conceptos de clasificador f, error de generalización $L(f)$ y probabilidad a posteriori $\widehat{p}(\mathbf{y}|\mathbf{x})$. Además se explicaron algunas relaciones entre los mismos.

Posteriormente, en la Sección 2.3, los diferentes elementos de los bosques de decisión fueron introducidos. Primero se explicó cómo entrenar los árboles de decisión, dado un conjunto de entrenamiento $\mathcal{D}$. Para eso, se analizaron diferentes funciones objetivo y se introdujo el algoritmo de reducción de incerteza por pasos (Sección 2.3.2). El cual es un método goloso que se utiliza para encontrar simultáneamente la estructura del árbol de decisión y los parámetros de sus nodos. El método busca maximizar la ganancia de información de cada nodo por separado. Pero encontrar el parámetro óptimo para cada nodo es difícil computacionalmente, dado que el espacio de posibles parámetros de nodos es exponencial en el tamaño de la cantidad de muestras de entrenamiento. Por lo tanto, se explicó el método de optimiza-

ción al azar de nodos (Sección 2.3.3) que evalúa solamente un subconjunto del conjunto de posibles parámetros.

El método de optimización al azar de nodos junto con el modelo de ensamble (Sección 2.4) son los dos elementos fundamentales de los bosques de decisión. El modelo de ensamble es el mecanismo utilizado para reducir la varianza que tienen los árboles de decisión. El mecanismo entrena varios arboles de decisión en paralelo y en forma independiente, para luego promediar sus predicciones. Cuando la correación de las predicciones de los árboles de decisión es baja, la ecuación (2.21) muestra que la varianza se reduce. El método de optimización al azar de nodos es utilizado para reducir la correlación de las predicciones de los arboles de decisión. Los árboles de decisión sufren el problema de tener una alta varianza debido a su estructura jerárquica y por ende es improtante utilizar un buen método para reducir la varianza de sus predicciones.

Chapter 3

Transfer Learning Decision Forests

In this chapter we present our novel extensions to the decision forest framework in order to transfer knowledge from several source tasks to a given target task. We start by introducing the mathematical notation and then we explain our extensions. We theoretically analyze one of extensions and show that it improves the estimation of the information gain.

3.1 The Intuition

In the previous chapter we described how to use the decision forest framework to develop systems that automatically solve computer vision tasks. The framework developed in the previous chapter require to collect a training set from scratch for each task. Gathering those data-sets can be extremely time consuming and might have a significant impact on the overall cost of the final system. In order to mitigate the impact on the costs, it is usually the case that only small training sets are collected. However, machine learning techniques are not well suited for these cases.

A possible approach to avoid those problems is to attempt to design machine learning techniques that more closely imitates the human behavior. It is well known that the learning ability of humans improves progressive over time. Furthermore, learning new tasks is usually simpler after learning a previous related one. The reason of this, is that new concepts are not learned in isolation, but considering connections to what is already known [TOC13].

However, the decision forest framework presented in the previous chapter can not improve over time since each task is learned separately. In this chapter we present a novel method of transfer learning which extends the decision forest framework. The method presented in this chapter combines the training sets of several tasks in order to find the parameters of the internal and external nodes of the decision forest. The main idea is to combine the training sets of the source and target tasks during the random node optimization to obtain parameters that are more general than the ones obtained when we consider only the training set of the target task.

For example, suppose that we want to improve the system that automatically recognizes the type of scene captured in a photograph, explained in the previous chapter. Given a feature vector extracted from an image we want to decide if the photograph corresponds to a scene of a windmill / mobile house / barn. Since the sky is present frequently in the photographies of a barn or of a windmill it is a strong feature, thus we propose to combine the training set of tasks windmill and barn to achieve higher accuracies. The key idea is that since windmill and barn share some common features, the parameters of the nodes found combining the training sets do not overfit.

3.2 Mathematical Framework

We introduce the formal notation that will be used in this chapter. We are interested in solving the classification task T using the knowledge of the source tasks $S_1, \dots S_N$ in order to improve classification accuracy in the target task T. All the tasks share the same feature space $\mathbb{R}^d$ but have different label space; the label space of the target task is noted $\mathcal{Y}$ and the label space of the source tasks is noted $\mathcal{Y}_1, \dots, \mathcal{Y}_N$ respectively. In addition, we have a training set for the target task $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \mathbb{R}^d,\ \mathbf{y}_i \in \mathcal{Y}\}$ and a training set for each source task $\mathcal{D}^j = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \mathbb{R}^d,\ \mathbf{y}_i \in \mathcal{Y}_j\}$. The goal is to find a decision forest $\mathcal{F} = \{F_1, \dots, F_T\}$, defined as an ensemble of T decision trees F, which minimizes the classification error using the training sets for the source tasks $\mathcal{D}^1, \dots \mathcal{D}^N$ and the training set for the target set $\mathcal{D}$.

3.3 Training

The training algorithm of a decision forest $\mathcal{F}$ consists in training each of the trees $F \in \mathcal{F}$ independently, introducing a certain level of randomness in the training process in order to de-correlate individual tree predictions and improve generalization. In this chapter each node has a training set for each task associated with it, as opposed to the previous chapter where each node has only one training set. Additionally some training sets in this chapter might be empty whereas in the previous chapter the training set in each node never was empty.

We grow each tree using the step uncertainty reduction algorithm 1. But, in this chapter, we adapt the procedure for optimizing the parameters $\boldsymbol{\theta}_k$ for each node $k \in F^\circ$ to the transfer learning setting [PY10]. The difference between the classification decision forests, described in Chapter 2, and the transfer learning decision forests presented in this chapter is the objective function. In the former, the information gain is used to find the best parameters, taking into account only one task. By contrast, in this chapter we use the mixed information gain function, described in Section 3.4, that uses the combined training set of the target and source tasks to find the parameters of each internal node.

The partition $\mathcal{P}$ defined by the leaves ∂F after making a tree F grow might contain regions R with no training samples of the target task T_0. Therefore, we cannot define a predictive model for those regions. In order to overcome this issue we infer the labels from the regions that have training samples of task T_0, as described in Section 3.5.

3.4 Mixed Information Gain

We propose to find parameters $\boldsymbol{\theta}_k$ for each internal node $k \in F^\circ$ in order to obtain a partition $\mathcal{P}$ of the feature space $\mathbb{R}^d$ such that, in each region $R \in \mathcal{P}$, the training samples of each task have the same label. This aims at improving the generalization capabilities of each tree independently, since each region $R \in \mathcal{P}$ is found using more training samples, and is more gen-

eral because it is encouraged to split the training samples of several tasks simultaneously.

In this chapter, the parameters $\boldsymbol{\theta}_k$ of each internal node $k \in F^\circ$ are found maximizing the information gain of the target task T and source task $S_1, \ldots, S_N$ simultaneously:

$$\boldsymbol{\theta}_k^* = \arg\max_{\boldsymbol{\theta}_k \in \mathcal{T}_k} (1-\gamma)\mathcal{I}(\mathcal{D}_k, \boldsymbol{\theta}_k) + \gamma \sum_{n=1}^{N} p_{n,k}\mathcal{I}(\mathcal{D}_k^n, \boldsymbol{\theta}_k) \tag{3.1}$$

where γ is a scalar parameter that weights the two terms, $\mathcal{T}_k \subset \mathbb{R}^d$ is a small subset of the instance space available when training the internal node $k \in F^\circ$, and $p_{n,k}$ is the fraction of samples of the source task S_n in the samples reaching the node k, $p_{n,k} = \frac{|\mathcal{D}_k^k|}{\sum_{j=1}^{N} |\mathcal{D}_k^j|}$. The set $\mathcal{D}_k$ are the training samples of the target task T reaching the node k and the sets $\mathcal{D}_k^n$ are the training samples of the source tasks S_n reaching the node k.

The maximization of (3.1) is achieved using randomized node optimization, Section 2.3.3. We perform an exhaustive search over subset $\mathcal{T}_k$ of the feature space parameters $\mathbb{R}^d$. The size of the subset is a training parameter noted as $\rho = |\mathcal{T}_k|$. The randomized node optimization is a key aspect of the decision forest model, since it helps to de-correlate individual tree predictions and to improve generalization.

The first term of the objective function in (3.1) is the information gain associated with the training samples reaching node k for the target task T. This term encourages the objective function to find parameters $\boldsymbol{\theta}_k$, that define a split function $h(\mathbf{x}, \boldsymbol{\theta}_k)$, such that the training set of the target task T reaching the descendants of the node k have low entropy. Conversely, the second term in (3.1) encourages the objective function to find parameters $\boldsymbol{\theta}_k$ that makes the training samples of source tasks $S_1, \ldots, S_N$ reaching the descendant nodes of k as pure as possible.

As a result of the weighted combination of the two terms, the objective function in (3.1) penalizes split functions $h(\mathbf{x}, \boldsymbol{\theta}_k)$ with a high information gain in the target task T and a low information gain in the source tasks $S, \ldots, S_N$. The key idea is that those splits might have a high information gain in the target task T only because the training set for task T is limited,

and if we choose them the generalization performance will decrease.

Which is why the objective function (3.1) can be interpreted as a regularized version the classical information gain function. However, the regularization is achieved by using additional data, in contrast to other regularization methods that assume that the parameters should be close to zero, like L2 regularization [Mur12]. The advantage of using additional data to regularize a problem is that the solution is not restricted to be smooth, which is important in many cases.

3.4.1 Properties

In this section we discuss some theoretical properties of our mixed information gain formulation. The properties in this section address the question of how many source tasks are required to obtain transfer learning. With this aim, we prove a property that states that many simple source tasks can be combined to obtain a more complex source task. Moreover, we explain that additional data has a positive effect in estimating the entropy associated to a set.

The first key property of our work is an alternative representation of the second term in (3.1), which shows that many simple tasks can be combined to obtain a more complex task. Given the source tasks $S_1, \ldots, S_N$ we can create a new task $S^{1,\ldots,N}$ by concatenating together all the label sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_N$, denoted by $\mathcal{Y}^{1,\ldots,N} = \oplus_{n=1}^{N} \mathcal{Y}_n$. The training set of each individual source task is combined to obtain $\mathcal{D}_k^{1,\ldots,N} = \cup_{n=1}^{N} \mathcal{D}_k^n$. Therefore, multiple small training sets can be combined to obtain a large training set for a more complex task. In the following theorem we show the relation between the information gain $\mathcal{I}(\mathcal{D}_k^{1,\ldots,N}, \boldsymbol{\theta}_k)$ and the information gain of each $\mathcal{I}(\mathcal{D}_k^1, \boldsymbol{\theta}_k), \ldots, \mathcal{I}(\mathcal{D}_k^N, \boldsymbol{\theta}_k)$.

Theorem 1. *Let $S_1, \ldots, S_N$ be N tasks and, $\mathcal{D}^1, \ldots, \mathcal{D}^N$ their respectively training sets and, let $p_n = \frac{|\mathcal{D}^n|}{\sum_{k=1}^{N} |\mathcal{D}^k|}$ be the fraction of training samples of task S_n in the training set $\mathcal{D}^{1,\ldots,N} = \cup_{n=1}^{N} \mathcal{D}^n$. Then we have:*

$$\mathcal{I}(\mathcal{D}^{1,\ldots,N}, \boldsymbol{\theta}) = \sum_{n=1}^{N} p_n \mathcal{I}(\mathcal{D}^n, \boldsymbol{\theta}) \tag{3.2}$$

Proof. The theorem is proved at [FCGT12], for completeness we include here the proof. Given $\mathbf{y} \in \mathcal{Y}_n$, let's develop the relationship between $\widehat{p}_{\mathcal{D}^{1,\dots,N}}(\mathbf{y})$ and $\widehat{p}_{\mathcal{D}^n}(\mathbf{y})$. By the definition of estimated probabilities we have that:

$$\widehat{p}_{\mathcal{D}^{1,\dots,N}}(\mathbf{y}) = \frac{1}{|\mathcal{D}^{1,\dots,N}|} \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{D}^{1,\dots,N}} \delta_{\mathbf{y}'}(\mathbf{y}) \tag{3.3}$$

since the training sets are disjoints we have:

$$\frac{1}{|\mathcal{D}^{1,\dots,N}|} \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{D}^{1,\dots,N}} \delta_{\mathbf{y}'}(\mathbf{y}) = \frac{1}{\sum_{n=1}^{N}|\mathcal{D}^n|} \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{D}^n} \delta_{\mathbf{y}'}(\mathbf{y}) \tag{3.4}$$

$$= \frac{1}{\sum_{k=1}^{N}|\mathcal{D}^k|}\frac{|\mathcal{D}^n|}{|\mathcal{D}^n|} \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{D}^n} \delta_{\mathbf{y}'}(\mathbf{y}) = \frac{p_n}{|\mathcal{D}^n|} \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{D}^n} \delta_{\mathbf{y}'}(\mathbf{y}) \tag{3.5}$$

using the definition of $\widehat{p}_{\mathcal{D}^n}(\mathbf{y})$ we have that:

$$\widehat{p}_{\mathcal{D}^{1,\dots,N}}(\mathbf{y}) = p_n \widehat{p}_{\mathcal{D}^n}(\mathbf{y}). \tag{3.6}$$

Now, let's show the following relationship between the entropy of the combined training set and the entropies of its disjoint subsets:

$$\mathcal{H}(\widehat{p}_{\mathcal{D}^{1,\dots,N}}) = \mathcal{H}(p_1,\dots,p_n) + \sum_{k=1}^{N} p_k \mathcal{H}(\widehat{p}_{\mathcal{D}^k}) \tag{3.7}$$

By definition of entropy we have:

$$\mathcal{H}(\widehat{p}_{\mathcal{D}^{1,\dots,N}}) = -\sum_{k=1}^{N}\sum_{\mathbf{y}\in\mathcal{Y}_k} \widehat{p}_{\mathcal{D}^{1,\dots,N}}(\mathbf{y}) \log\left(\widehat{p}_{\mathcal{D}^{1,\dots,N}}(\mathbf{y})\right) \tag{3.8}$$

using Eq. (3.6) we obtain:

$$= -\sum_{k=1}^{N}\sum_{\mathbf{y}\in\mathcal{Y}_k} p_k\widehat{p}_{\mathcal{D}^k}(\mathbf{y}) \log\left(p_k\widehat{p}_{\mathcal{D}^k}(\mathbf{y})\right) \tag{3.9}$$

applying the properties of the logarithms:

$$= -\sum_{k=1}^{N}\sum_{\mathbf{y}\in\mathcal{Y}_k} p_k\widehat{p}_{\mathcal{D}^k}(\mathbf{y}) \log p_k - \sum_{k=1}^{N}\sum_{\mathbf{y}\in\mathcal{Y}_k} p_k\widehat{p}_{\mathcal{D}^k}(\mathbf{y}) \log \widehat{p}_{\mathcal{D}^k}(\mathbf{y}) \tag{3.10}$$

and after some algebraic manipulations:

$$= \mathcal{H}(p_1,\dots,p_N) + \sum_{k=1}^{N} p_k \mathcal{H}\left(\widehat{p}_{\mathcal{D}^k}\right) \tag{3.11}$$

The Eq. (3.7) is a generalization of the grouping rule of the entropy [CT06]. Finally, let's prove the property, by definition of information gain we have that:

$$\begin{aligned} \mathcal{I}(\mathcal{D}^{1,\dots,N}, \boldsymbol{\theta}) &= \mathcal{H}(\widehat{p}_{\mathcal{D}^{1,\dots,N}}) \\ &- \frac{|\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^+}|}{|\mathcal{D}^{1,\dots,N}|}\mathcal{H}\left(\widehat{p}_{\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^+}}\right) \\ &- \frac{|\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^-}|}{|\mathcal{D}^{1,\dots,N}|}\mathcal{H}\left(\widehat{p}_{\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^-}}\right) \end{aligned} \tag{3.12}$$

applying Eq. (3.7) in each term:

$$\begin{aligned} &= \mathcal{H}(p_1, \dots, p_n) + \textstyle\sum_{k=1}^{N} p_k \mathcal{H}(\widehat{p}_{\mathcal{D}^k}) \\ &- \frac{|\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^+}|}{|\mathcal{D}^{1,\dots,N}|}\left(\mathcal{H}(p_1, \dots, p_n) + \textstyle\sum_{k=1}^{N} p_k \mathcal{H}(\widehat{p}_{\mathcal{D}^{k,\boldsymbol{\theta}^+}})\right) \\ &- \frac{|\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^-}|}{|\mathcal{D}^{1,\dots,N}|}\left(\mathcal{H}(p_1, \dots, p_n) + \textstyle\sum_{k=1}^{N} p_k \mathcal{H}(\widehat{p}_{\mathcal{D}^{k,\boldsymbol{\theta}^-}})\right) \end{aligned} \tag{3.13}$$

after some algebraic manipulations, we obtain:

$$= \sum_{k=1}^{N} p_k \left(\mathcal{H}(\widehat{p}_{\mathcal{D}^k}) - \frac{|\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^+}|}{|\mathcal{D}^{1,\dots,N}|}\mathcal{H}(\widehat{p}_{\mathcal{D}^{k,\boldsymbol{\theta}^+}}) - \frac{|\mathcal{D}^{1,\dots,N,\boldsymbol{\theta}^-}|}{|\mathcal{D}^{1,\dots,N}|}\mathcal{H}(\widehat{p}_{\mathcal{D}^{k,\boldsymbol{\theta}^-}})\right) \tag{3.14}$$

by definition of information gain:

$$= \sum_{k=1}^{N} p_k \mathcal{I}(\mathcal{D}^k, \boldsymbol{\theta}). \tag{3.15}$$

□

This theorem relates the information gain of several source tasks $S_1, \dots, S_N$ to the information gain of another source task $S^{1,\dots,N}$. An important consequence of this equation is that we can combine the training set of simpler tasks $S_1, \dots, S_N$ to obtain a larger training set for another source task $S^{1,\dots,N}$. Therefore, increasing the number of training samples per source task or the number of source tasks has a similar effect. Increasing the number of training samples is beneficial for most of the machine learning methods and, as we will show later in this chapter, our method is not the exception.

Let's explore in more detail how the combination of the information gain of tasks $S_0, \dots, S_N$ for finding the optimal parameters $\boldsymbol{\theta}_k$ improves the generalization properties of the decision forests. The parameters $\boldsymbol{\theta}_k$ are found using an empirical estimation of the entropy $\mathcal{H}(\mathcal{D}_k)$ of the training samples

$\mathcal{D}_k$ reaching node k and its children. Consequently, errors in estimating entropy can result in very different trees. Tighter bounds for the expected entropy are found by increasing the number of training samples, as explained in Theorem 2.

Theorem 2. *Let p be a probability distribution on $\mathbb{R}^d \times \mathcal{Y}$ such that the marginal distribution over $\mathcal{Y}$ is a categorical distibution with parameters $p_1, \ldots, p_{|\mathcal{Y}|}$, and suppose $\mathcal{D}_K = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_K, \mathbf{y}_K)\}$ is the set generated by sampling K times from $\mathbb{R}^d \times \mathcal{Y}$ according to p. Let $\mathcal{H}(p) = -\sum_{y=1}^{|\mathcal{Y}|} p_y \log(p_y)$ be the entropy of distribution p. Then $\mathbb{E}(\mathcal{H}(\mathcal{D}_K)) + \sum_{\mathbf{y}\in\mathcal{Y}} p_\mathbf{y} \log\left(1 + \frac{1-p_\mathbf{y}}{Kp_\mathbf{y}}\right) \leq \mathcal{H}(p) \leq \mathbb{E}(\mathcal{H}(\mathcal{D}_K))$.*

Proof. First, we prove $\mathbb{E}(\mathcal{H}(\mathcal{D}_K)) + \sum_{\mathbf{y}\in\mathcal{Y}} p_\mathbf{y} \log\left(1 + \frac{1-p_\mathbf{y}}{Kp_\mathbf{y}}\right) \leq \mathcal{H}(p)$

By definition of the empirical entropy and linearity of the expectation, we have:

$$\mathbb{E}(\mathcal{H}(\mathcal{D}_K)) = -\mathbb{E}\left[\sum_{\mathbf{y}\in\mathcal{Y}} \hat{p}_{\mathcal{D}_K}(\mathbf{y}) \log(\hat{p}_{\mathcal{D}_K}(\mathbf{y}))\right] = -\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{\mathcal{D}_K}(\mathbf{y}) \log(\hat{p}_{\mathcal{D}_K}(\mathbf{y}))\right] \tag{3.16}$$

Using the definitions of the empirical histogram $\hat{p}_{\mathcal{D}_K}(\mathbf{y})$ and the expectation:

$$-\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{\mathcal{D}_K}(\mathbf{y}) \log(\hat{p}_{\mathcal{D}_K}(\mathbf{y}))\right] = -\sum_{\mathbf{y}\in\mathcal{Y}} \sum_{j=0}^{K} p\left(\hat{p}_{\mathcal{D}_K}(\mathbf{y}) = \frac{j}{K}\right) \frac{j}{K} \log \frac{j}{K} \tag{3.17}$$

Assuming that the samples are iid, then:

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} \sum_{j=0}^{K} \binom{K}{j} p_\mathbf{y}^j (1-p_\mathbf{y})^{K-j} \frac{j}{K} \log \frac{j}{K} \tag{3.18}$$

Note that, in this equation, $p_\mathbf{y}$ is the true probability of distribution p. After some algebraic manipulations, we obtain the following:

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_\mathbf{y} \sum_{j=0}^{K-1} \binom{K-1}{j} p_\mathbf{y}^j (1-p_\mathbf{y})^{K-1-j} \log \frac{j+1}{K} \tag{3.19}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_\mathbf{y} \sum_{j=0}^{K-1} p\left(\hat{p}_{\mathcal{D}_K}(\mathbf{y}) = \frac{j}{K}\right) \log \frac{j+1}{K} \tag{3.20}$$

Applying Jensen's inequality for the convex function $-\log(x)$, we obtain:

$$\geq -\sum_{\mathbf{y}\in\mathcal{Y}} p_\mathbf{y} \log\left(\sum_{j=0}^{K-1} p\left(\hat{p}_{\mathcal{D}_K}(\mathbf{y}) = \frac{j}{K}\right) \frac{j+1}{K}\right) \tag{3.21}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \frac{(K-1)p_{\mathbf{y}}+1}{K} \tag{3.22}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(p_{\mathbf{y}} + \frac{1-p_{\mathbf{y}}}{K}\right) \tag{3.23}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(p_{\mathbf{y}} \left(1 + \frac{1-p_{\mathbf{y}}}{Kp_{\mathbf{y}}}\right)\right) \tag{3.24}$$

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log p_{\mathbf{y}} - \sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(1 + \frac{1-p_{\mathbf{y}}}{Kp_{\mathbf{y}}}\right) \tag{3.25}$$

$$= \mathcal{H}(p) - \sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log \left(1 + \frac{1-p_{\mathbf{y}}}{Kp_{\mathbf{y}}}\right) \tag{3.26}$$

Now we prove $\mathcal{H}(p) \leq \mathbb{E}(\mathcal{H}(\mathcal{D}_K))$.

By definition of the emprirical entropy and linearity of the expectation, we have:

$$\mathbb{E}(\mathcal{H}(\mathcal{D}_K)) = -\mathbb{E}\left[\sum_{\mathbf{y}\in\mathcal{Y}} \hat{p}_{\mathcal{D}_K}(\mathbf{y}) \log(\hat{p}_{\mathcal{D}_K}(\mathbf{y}))\right] = -\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{\mathcal{D}_K}(\mathbf{y}) \log(\hat{p}_{\mathcal{D}_K}(\mathbf{y}))\right] \tag{3.27}$$

Applying Jensen's inequality for the convex function $x \log x$, we obtain the following:

$$\leq -\sum_{\mathbf{y}\in\mathcal{Y}} \mathbb{E}\left[\hat{p}_{\mathcal{D}_K}(\mathbf{y})\right] \log(\mathbb{E}\left[\hat{p}_{\mathcal{D}_K}(\mathbf{y})\right]) \tag{3.28}$$

Since $\mathbb{E}\left[\hat{p}_{\mathcal{D}_K}(\mathbf{y})\right] = p_{\mathbf{y}}$, we have:

$$= -\sum_{\mathbf{y}\in\mathcal{Y}} p_{\mathbf{y}} \log(p_{\mathbf{y}}) = \mathcal{H}(p) \tag{3.29}$$

□

Theorem 2 shows that the empirical entropy $\mathcal{H}(\mathcal{D}_K)$ is closer to the entropy of the distribution p when the training set is larger, since when $K \to \infty$, $\log\left(1 + \frac{1-p_{\mathbf{y}}}{Kp_{\mathbf{y}}}\right) \to 0$. Therefore, if we assume that the source tasks are related to the target task (i.e. both have a similar distribution p). Theorem 1 tell us that we can combine the training sets of the different tasks, increasing the number of training samples. Finally, using Theorem 2, we can conclude that the mixed information gain (3.1) finds parameters $\boldsymbol{\theta}_k$ that achieve lower generalization errors than the traditional information gain $\mathcal{I}(\mathcal{D}_k, \boldsymbol{\theta}_k)$ since more training samples are taken into account.

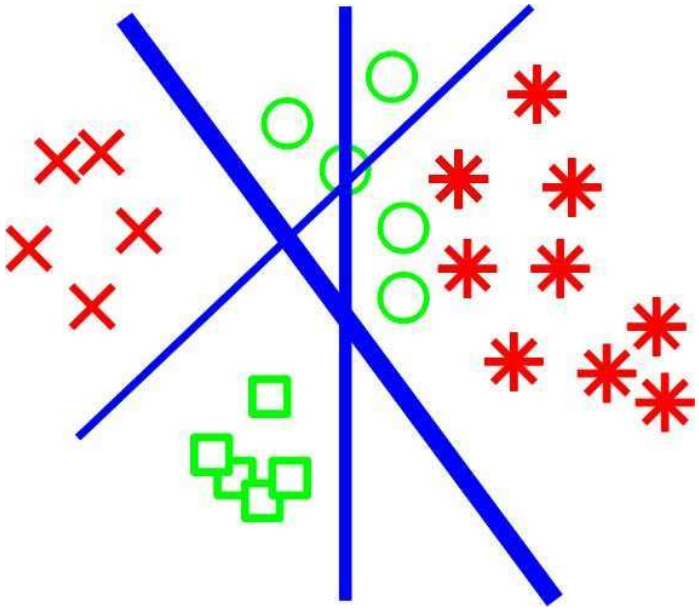
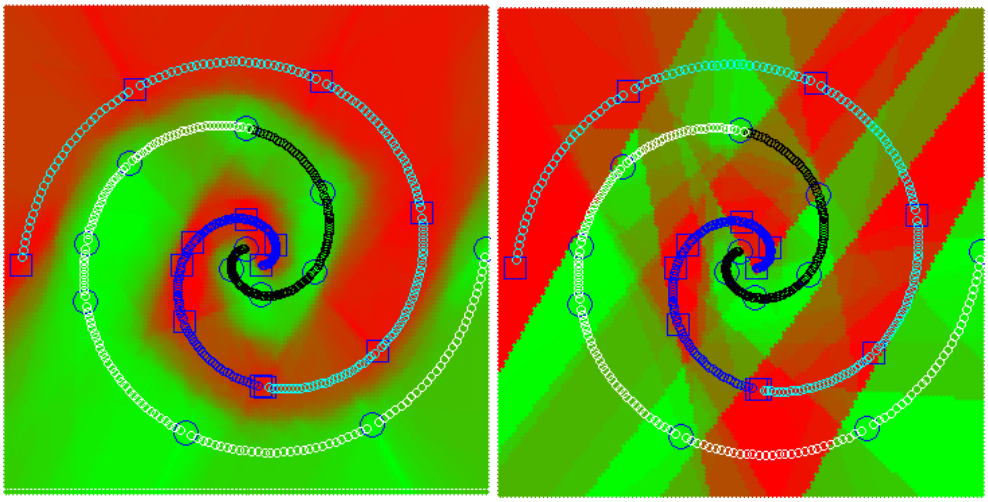


Figure 3.1: Illustration of mixed information gain on a toy problem in which there are two tasks, each with two labels. The thickness of the blue lines indicates the mixed information gain of the split (all the splits have the same information gain). The target task T has two green labels ($\mathcal{Y} = \{\times, *\}$) and the source task S_1 has two red labels ($\mathcal{Y}_1 = \{\bigcirc, \square\}$).

To understand how the mixed information gain works, Figure 3.1 considers a toy problem with two tasks, each with two labels. Estimating the information gain of a split with only a few training samples of the target task is difficult since there are a lot of possible splits with the same empirical information gain but different generalization capabilities. Our goal is to discover which split to use, and we intend to choose the one with the best generalization capability. When, in our formulation, we use the additional training samples from the source tasks to compute the information gain of a split, some of the splits are penalized for having a low information gain in the source task and, thus, allows us to find a split with increased generalization.

One of the main drawbacks of decision trees is the high variance of the resulting classifier. A small change in the training data can often result in a very different series of splits. The major reason for this instability is the hierarchical nature of the process: the effect of an error in the top split is



(a) (b)

Figure 3.2: (a) Output classification of a transfer learning decision forests, tested on all points in a rectangular section of the feature space. The color associated with each test point is a linear combination of the colors (red and green) corresponding to the two labels (□, ◯) in the target task. The training data for the target task is indicated with big markers and the training data for the source task is indicated with small markers. (b) Output classification of a decision forest tested in the same feature space section as before but trained using only data for the target task.

propagated down to all the splits below it [HTF03]. Decision forests [Bre01] build a large collection of de-correlated decision trees, and hence reduce the variance averaging the prediction of each of them. The mixed information gain is a complementary approach for reducing their variance which increases the generalization of each tree independently. It is important to note that the mixed information preserves the diversity of the forests, which is essential to improve the generalization error. The random nature of the random node optimization [AC12] used to optimize (3.1) allows us to keep a high diversity among the trees.

Figures 3.2a and 3.2b compare the output classification on all the points in a rectangular section of the feature space for a decision forest classifier and for our transfer learning decision forest classifier. Both decision forests were trained with the same maximum depth $D = 8$, and have the same

number of trees $|\mathcal{F}| = 100$. The dataset for the target and source task is organized in the shape of a two-arm spiral. We can see that the classification decision forests have serious generalization problems since, even when all the training data of the target task is correctly classified, the spiral structure is not predicted accurately. In contrast, the spiral structure is predicted by the transfer learning decision forests as shown in Figure 3.2a.

3.5 Label Propagation

For each leaf $k \in \partial F$ of each tree $F \in \mathcal{F}$, we must have a predictive model $\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_k)$ that estimates the probability of label $\mathbf{y} \in \mathcal{Y}$ given a previously unseen test input $\mathbf{x} \in R_k \subseteq \mathbb{R}^d$. This poses a problem when we make each tree grow using the mixed information gain because we may end up with leaves $k \in \partial F$ that have no training samples of the target task T to estimate the predictive model $\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_k)$. In this work we use label propagation to assign a predictive model $\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_k)$ to those leaves.

We are given a set of leaves $\mathcal{U} \subseteq \partial F$ without training samples of the target task T and a set of leaves $\mathcal{L} \subseteq \partial F$ with training samples of the target task T. The goal is to obtain a predictive model $\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_k)$ for the leaves $k \in \mathcal{U}$ avoiding the propagation of labels through low density regions but, at the same time, propagating labels between nearby leaves. We construct a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \partial F$ is the vertex set and $\mathcal{E}$ is the edge set with each edge $\mathbf{e}_{ij} \in \mathcal{E}$ representing the relationship between nodes $i, j \in \partial F$.

Each edge $\mathbf{e}_{ij} \in \mathcal{E}$ is weighted taking into account the training samples of tasks $T, S_1 \ldots, S_N$. For each leaf $k \in \partial F$ we define the estimated mean $\boldsymbol{\mu}_k$ and estimated covariance Σ_k using the training samples reaching the node:

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{D}_k^+|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_k^+} \mathbf{x} \tag{3.30}$$

$$\Sigma_k = \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_k^+} \sum_{(\mathbf{x}',\mathbf{y}') \in \mathcal{D}_k^+} (\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x}' - \boldsymbol{\mu}_k)^T \tag{3.31}$$

where $\mathcal{D}_k^+$ is the union of the training sets of the target and source tasks reaching node k. We use the estimated mean $\boldsymbol{\mu}_k$ and estimated covariance

Σ_k to define the weight between two nodes $\mathbf{e}_{ij} \in \mathcal{E}$:

$$\mathbf{e}_{ij} = \frac{1}{2}\left(d_{ij}^T \Sigma_i d_{ij} + d_{ij}^T \Sigma_j d_{ij}\right) \tag{3.32}$$

where $d_{ij} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ is the difference between the estimated mean of the leaves $i, j \in \partial F$. Weight $\mathbf{e}_{ij} \in \mathcal{E}$ is the symmetric Mahalanobis distance. We use it to discourage the propagation of labels through low density regions. For each node $k \in \mathcal{U}$ we find the shortest path in graph $\mathcal{G}$ to all the nodes in $\mathcal{L}$. Let $s_k^* \in \mathcal{L}$ be the node with the shortest path to node k. We assign the predictive model $\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_{s_k^*})$ to $\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_k)$.

Label propagation methods are usually at least quadratic $\mathcal{O}(n^2)$ in terms of the number of training samples, making them slow when a large number of training samples is available. We avoid this problem by propagating the predictive model of the leaves, instead of propagating the labels of the training samples.

We illustrate the behavior of label propagation in Figure 3.3 using a 2D toy example. We consider the same two-arm spiral problem of Figure 3.2 which has data that follow a complex structure. We show the predictive models for the regions of two randomly grown trees before and after propagating labels. We observe that the predictive models are propagated following the intrinsic structure of the data, as a consequence of taking into account the training data of each region.

3.5.1 Testing

The predictive model of all the trees $F \in \mathcal{F}$ is combined to produce the final prediction of the forest:

$$\widehat{p}_{\mathcal{F}}(\mathbf{y}|\mathbf{x}) = \frac{1}{|\mathcal{F}|}\sum_{F \in \mathcal{F}} \widehat{p}_F(\mathbf{y}|\mathbf{x}). \tag{3.33}$$

Let $l_F : \mathbb{R}^d \rightarrow \partial F$ be the function that, given a sample $\mathbf{x} \in \mathbb{R}^d$, returns the leaf such that $\mathbf{x} \in R_{l_F(\mathbf{x})}$. The prediction for a tree F is:

$$\widehat{p}_F(\mathbf{y}|\mathbf{x}) = \widehat{p}_F\left(\mathbf{y}|\mathbf{x} \in R_{l_F(\mathbf{x})}\right). \tag{3.34}$$

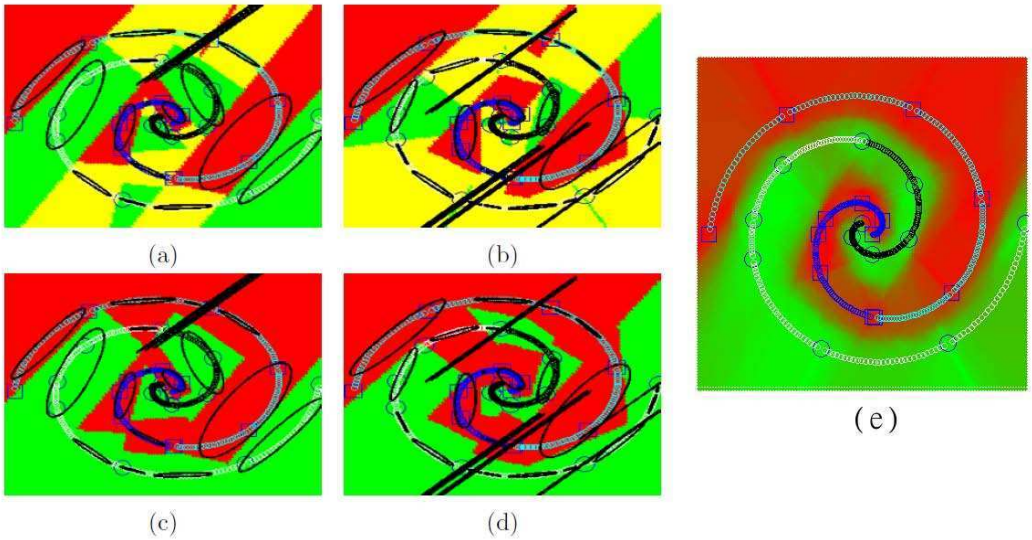


Figure 3.3: Illustration of the label propagation procedure between regions, as before the training data for the target task is indicated with big markers and the training data for the source task is indicated with small markers. The ellipses in black are the isocontours of a Gaussian distribution learned by maximum likelihood for each region using the training samples in the region. (a, b) show the predictive model for two different trees $F \in \mathcal{F}$ before propagating labels. The color associated with each region is a linear combination of the colors (red and green) corresponding to the two labels ($\square$, $\bigcirc$) in the target task. The yellow regions are the ones without training data of the target task. (c, d) show the predictive model after the label propagation. (e) Output classification of the final transfer learning decision forests.

Finally, let $k \in \partial F$ be the leaf that is reached by sample $\mathbf{x} \in \mathbb{R}^d$. The class distribution for that leaf is:

$$\widehat{p}_F(\mathbf{y}|\mathbf{x} \in R_k) = \begin{cases} \widehat{p}_{\mathcal{D}_k}(\mathbf{y}) & \textbf{if } \mathcal{D}_k \neq \emptyset \\ \widehat{p}_{\mathcal{D}_{k^*}}(\mathbf{y}) & \textbf{otherwise} \end{cases} . \tag{3.35}$$

Thus, $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ is the empirical histogram of the training samples of the target task T reaching node $l_F(\mathbf{x})$ if any. Otherwise, $\widehat{p}_F(\mathbf{y}|\mathbf{x})$ is the empirical histogram associated with the node that has the shortest path to $l_F(\mathbf{x})$.

3.6 Resumen

En este capítulo se presentaron las extensiones a los bosques de decisión que proponemos en la tesis. Las extensiones propuestas permiten que los bosques de decisión sean entrenados utilizando varias tareas fuente para mejorar el desempeño en la tarea destino. La primera extensión propuesta modifica la función objetivo que se optimiza para encontrar los parámetros de cada nodo del bosque de decisión. La segunda extensión propuesta es propagar las etiquetas a través de las hojas con el fin de inferir la estructura de la variedad del espacio de características.

La función objetivo propuesta realiza un promedio ponderado entre la ganancia de información de los datos de las tareas origen y la tarea destino. Como resultado, la nueva función objetivo penaliza parámetros que generan una ganancia de información alta en la tarea de destino y baja en las tareas de origen. El motivo por el cual es razonable penzalizar esos parámetros es que su alta ganancia de información en la tarea de destino puede ser que se deba simplemente a que no hay suficientes muestras de entrenamiento. De esta manera, se extraen las regularidades presentes en las tareas de origen y las utiliza en la tarea destino.

Adicionalmente, en la sección 3.4 hemos demostrado algunas propiedades relevantes de la nueva función objetivo. Las propiedades demostradas intentar responder, parcialmente, cuál es el número de tareas de origen que se requieren para mejorar el rendimiento utilizando los nuevos bosques de decisión propuestos. La primera propiedad establece que muchas de las ta-

reas de origen se pueden combinar para obtener una tarea origen más compleja. La segunda propiedad explica cómo los datos adicionales tienen un efecto positivo en la estimación de la entropía asociada a un conjunto de datos.

El objetivo de la segunda extensión propuesta, es asignar un modelo predictivo a las hojas sin muestras de entrenamiento de la tarea de destino, después de haber entrenado cada uno de los árboles del bosque decisión. Un supuesto implícito de este paso es que las hojas cercanas deben tener modelos predictivos similares. Se define la distancia entre cada par de hojas de un árbol de decisión utilizando los datos de todas las tareas que llegan a esa hoja. De esta forma, se tiene en cuenta la estructura de los datos y se desalienta la propagación de etiquetas a través de zonas de baja densidad.

Chapter 4

Applications

In this chapter we use the transfer learning decision forests in two different applications. The first application is the problem of recognizing gestures from a small number of training samples. The second application is the problem of recognizing characters having only a limited number of training samples. In both cases, we compare the results obtained using transfer learning decision forests with the ones obtained using decision forests and the ones previously reported in other works.

4.1 Gesture Recognition

Gesture recognition is one of the open challenges in computer vision. There are several reasons that make gesture recognition particularly challenging. First, gestures can be composed of multiple visual cues, for example, movements of fingers and lips, facial expressions, body pose. In addition, there exist technical limitations such as insufficient spatial or temporal resolution and unreliable depth cues. Therefore, it is difficult to reliably track the hand, head and body parts, and achieve 3D invariance.

There is a big number of potential applications for this problem, including surveillance, smart-homes, rehabilitation, entertainment, animation and human–robot interaction and sign language recognition just to mention a few. The task of gesture recognition is to determine the gesture label that best describes a gesture instance, even when performed by different people,

from various viewpoints and in spite of large differences in manner and speed.

To reach that goal, many approaches combine vision and machine learning tools. Computer vision tools are employed to extract features that provide robustness to distracting cues and that, at the same time, are discriminative. Machine learning is used to learn a statistical model from those features, and to classify new examples using the models learned. This poses a problem in gesture recognition since it is difficult to collect big datasets to learn statistical models. Therefore, in this work we perform experiments aimed at showing that our transfer learning decision forests are useful to mitigate this problem.

Recently, the ChaLearn competition [GAJ+12] provided a challenging dataset to evaluate whether transfer learning algorithms can improve their classification performance using similar gesture vocabularies. The dataset is organized into batches, with only one training example of each gesture in each batch. The goal is to automatically predict the gesture labels for the remaining gesture sequences (test examples). The gestures of each batch are drawn from a small vocabulary of 8 to 12 unique gestures, when we train a classifier to predict the labels of a target batch (or task) T we use the training samples of T and of the other batches $S_1, \ldots, S_N$.

Figure 4.1 shows some representative frames for actions in the batches devel01 and devel02. Some aspects of the data-set are intentionally easy, for example, the camera is fixed and gestures are separated by returning to a resting position. However, some other aspects of the data-set include challenges that are expected to be present in the real-life, for example, some parts of the body may be occluded and between batches there are variations in background, clothing, skin color, lighting, temperature and resolution.

Each batch of the ChaLearn competition includes 100 recorded gestures grouped in sequences of 1 to 5 gestures performed by the same user [GAJ+12]. To get sufficient spacial resolution, only the upper body was of the user was captured. Since the skeleton tracker of the Kinect, at the time of data collection, could not handle partial body occlusion the ChaLearn dataset does not contains any skeleton information. There is only one gesture in the train-

Figure 4.1: Sample frames from different actions in the devel01 and devel02 batches of the ChaLearn dataset.

ing sequences, but there might be more than one gesture in the testing sequences. Therefore, in order to use the method described in this section we need to temporally segment the testing sequencies. To this end, we use the Dynamic Time Warping (DTW) implementation given by the organizers.

In this section, we describe the features and the classifiers used to validate our approach, as well as their application to the ChaLearn competition [GAJ+12]. We review the temporal segmentation of the testing sequences in Section 4.1.1, the features for recognizing the label of each temporally segmented video are described in Section 4.1.2, the classifier for each temporally segmented video is presented in Section 4.1.3, and we evaluate the system in Section 4.1.4. Figure 4.2 shows the major components of our system and their interaction.

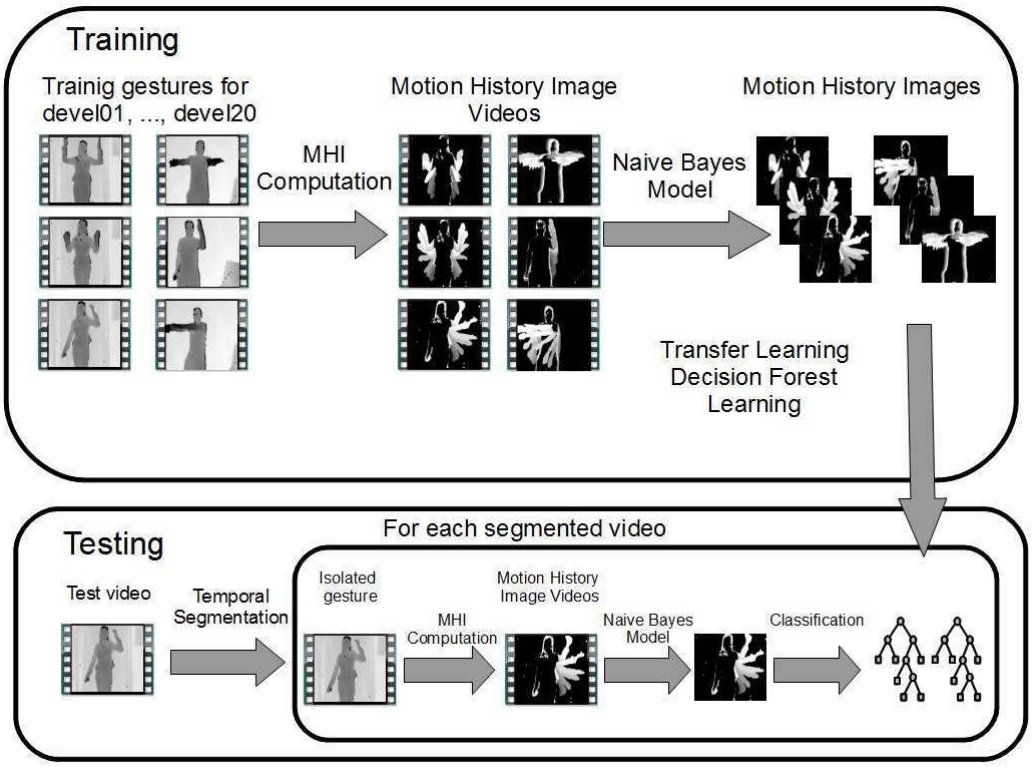


Figure 4.2: Major components of our system and their interaction.

4.1.1 Temporal Segmentation

The testing videos, unlike the training videos, contains a sequence of gestures performed by the same user. The number of gestures in a testing video is between one and five and, the subjects in the videos return to a neutral position between two consecutive gestures. The method described here and, provided by the organizers of the challenge make use of this constraint in their approach. The desired output of this step is to obtain the number of gestures in a test video and the start and end time of each gesture. The temporal segmentation is applied to each of the testing videos of the ChaLearn Gesture Challenge in order to isolate them and classify each gesture separately in the next step.

Let V be a testing video with M frames and let $\mathcal{V}_1, \ldots, \mathcal{V}_{|\mathcal{Y}|}$ be the training videos of a given target task T. First, the resting position is estimated using the firsts frames of each training video. Then, the estimated frame of the resting position and the training videos $\mathcal{V}_1, \ldots, \mathcal{V}_{|\mathcal{Y}|}$ are concatenated

together to obtain a long video $\mathcal{B}$ with J frames, where the last frame is the frame containing the estimated resting position. Finally, each frame of the testing video V and the large video $\mathcal{B}$ are preprocessed and compared using the negative euclidean distance and a J-by-M matrix L is built. The column i of the matrix L compares the frame i of the testing video V with all the J frames of $\mathcal{B}$.

An optimal warping path is then computed using the DTW algorithm and defining special conditions on the local transitions allowed in the path. The local transitions encourages the optimal path to go through the resting position frame when a gesture finishes and a new one starts. Additionally, the transitions allows to skip frame or to freeze a frame in order to perform an elastic matching of the frames. More importantly, the local transitions are designed to enforce that the optimal warping path can not jump between different frames of the training videos, except for the first and last frames.

4.1.2 Motion History Images

Given a depth video $\mathcal{V}$ where $\mathcal{V}(x, y, t)$ is the depth of the pixel with coordinates (x, y) at the tth frame. We compute the Motion History Image (MHI) [BD96, BD01, ATKI12] for each frame using the following function:

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } |\mathcal{V}(x, y, t) - \mathcal{V}(x, y, t-1)| \geq \xi \\ \max(0, H_\tau(x, y, t-1) - 1) & \text{otherwise} \end{cases} \tag{4.1}$$

where τ defines the temporal extent of the MHI, and ξ is a threshold employed to perform the foreground/background segmentation at frame t. The result is a scalar-valued image for each frame of the original video $\mathcal{V}$ where pixels that have moved more recently are brighter. MHI H_τ represents the motion in an image sequence in a compact manner, the pixel intensity is a function of the temporal history of motion at that point. A common problem when computing MHI H_τ using the color channel is the presence of textured objects in the image sequence; here we use the depth video $\mathcal{V}$ to overcome this issue. This is a relevant problem in gesture recognition, because, as a result of the clothes texture, the MHI is noisy [ATKI12].

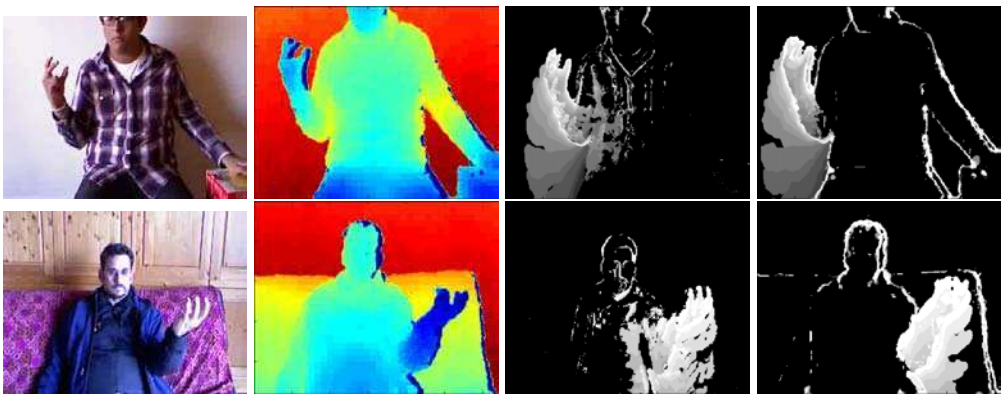


Figure 4.3: Comparison of the MHI computed using the depth channel or the RGB channel for two different training videos of the ChaLearn competition. The first two columns show the RGB channel and the depth channel (heat map), whereas the third and fourth columns show the MHI computed using the RGB channel and the MHI computed using the depth channel, respectively.

An interesting property of the MHI is that it is sensitive to the direction of motion; hence it is well-suited for discriminating between gestures with an opposite direction. An advantage of the MHI representation is that a range of times may be encoded in a single frame, and thus, the MHI spans the time scale of human gestures. After computing MHI H_τ we reduce the spatial resolution of each frame to $\omega_1 \times \omega_2$ pixels. Then, we flatten the MHI for each frame and obtain a feature $\mathbf{x} \in \mathbb{R}^{\omega_1\omega_2}$.

Figure 4.3 contrasts the result of computing the MHI using the RGB channel with the one obtained using the depth channel. In the first row, we see that the clothes texture generates noise in the MHI computed using the RGB channel. In the second row, the MHI of the RGB channel is noisy because of the shadow from the moving arm. Both problems are avoided using the depth channel for computing the MHI. The parameters to compute the MHI in all the cases were $\tau = 15$, and $\xi = 30$. Table 4.1 shows the classification error in the test set of the *devel*11 batch of the ChaLearn competition, after training a decision forest with the following parameters $D = 8, T = 50$.

$\tau \setminus \xi$	16	24	32	40
1	32.61 ± 0.14 %	32.61 ± 0.24 %	30.35 ± 0.22 %	29.26 ± 0.26 %
4	**30.43 ± 0.17 %**	31.52 ± 0.15 %	29.26 ± 0.15 %	28.17 ± 0.19 %
8	**30.43 ± 0.13 %**	**27.35 ± 0.16 %**	**28.09 ± 0.14 %**	**27.06 ± 0.18 %**
12	32.12 ± 0.23 %	32.61 ± 0.29 %	34.78 ± 0.31 %	29.35 ± 0.33 %
16	33.72 ± 0.28 %	32.61 ± 0.29 %	34.78 ± 0.25 %	30.43 ± 0.31 %

Table 4.1: Classification error in the test set of the *devel*11 batch for different combination of MHI parameters. In all the experiments we leave the the spatial resolution of each frame fixed to $\omega_1 \times \omega_2 = 16 \times 12$.

4.1.3 Naive Bayes

A main research trend in gesture recognition is to train Hidden Markov Models (HMMs) and theirs variants [BWK+04, KZL12], in order to exploit the temporal relation of a gesture. A drawback of this approach is that many training samples are required to train the large number of parameters of an HMM. Additionally, recognition rates might not improve significantly [LZL08]. This limitation has been recognized in [BWK+04] and a two-stage classifier was proposed to obtain one-shot learning.

Since in the ChaLearn competition [GAJ+12] there is only one labeled training sample of each gesture, we use the Naive Bayes model which has a smaller number of parameters than HMM. We use transfer learning decision forests to predict the probability that each frame will be part of a given gesture. We combine the predictions of the transfer learning decision forests for each frame using the naive bayes model. An advantage of the Naive Bayes assumption is that it is not sensitive to irrelevant frames (the probabilities for all the labels will be quite similar).

Given a video $\mathcal{V}$ of an isolated gesture, we want to find its label $\mathbf{y} \in \mathcal{Y}$. Assuming that the class prior $p(\mathbf{y})$ is uniform we have:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}|\mathcal{V}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} p(\mathcal{V}|\mathbf{y}) \tag{4.2}$$

Let $\mathbf{x}_1, \ldots, \mathbf{x}_M$ denote the MHI for each frame of a video $\mathcal{V}$ with M frames. We assume the Naive Bayes model, i.e. that the features $\mathbf{x}_1, \ldots, \mathbf{x}_M$ are i.i.d.

given the label $\mathbf{y}$, namely:

$$p(\mathcal{V}|\mathbf{y}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_M|\mathbf{y}) = \prod_{m=1}^{M} p(\mathbf{x}_m|\mathbf{y}) = \prod_{m=1}^{M} p(\mathbf{y}|\mathbf{x}_m)\frac{p(\mathbf{x}_m)}{p(\mathbf{y})} \tag{4.3}$$

We compute the probability $p(\mathbf{y}|\mathbf{x}_m)$ using our proposed transfer learning decision forest $\mathcal{F}$. The dataset for training the forest $\mathcal{F}$ consists of all the frames in each training video in the target task T and source tasks $S_1, \ldots, S_N$. We propose to use the frames of the training videos in the source tasks to obtain a better classifier for each frame.

Taking the logarithm in (4.3) and ignoring the constant terms we obtain the following decision rule:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathcal{V}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} \sum_{m=1}^{M} \log\left(p_{\mathcal{F}}(\mathbf{y}|\mathbf{x}_m)\right) \tag{4.4}$$

Note that we use the same forest $\mathcal{F}$ for computing the label distribution of all the frames in video $\mathcal{V}$. For this reason, given a frame $\mathbf{x}$, we expect distribution $p_{\mathcal{F}}(\mathbf{y}|\mathbf{x})$ to be multi-modal, which is an issue for several statistical methods. However, since the transfer learning decision forests have a predictive model for each leaf of their trees, they can deal with this type of distribution without major problems.

Figure 4.4 compares the classification error when predicting the label of a frame $p(\mathbf{y}|\mathbf{x})$ with the classification error when predicting the label of a video $p(\mathbf{y}|\mathcal{V})$, for different combinations of training parameters in the devel11 batch. We observe that the maximum depth D has a larger impact to predict the label of a video than the number of trees $|\mathcal{F}|$. Moreover, the classification error when predicting the label of a frame is greater than the classification error when predicting the label of a video. This means, as expected, that some frames are more discriminative than others, and that the misclassification of some frames is not a decisive factor for classifying a video correctly.

4.1.4 Experiments

In this section, we evaluate the transfer learning decision forests on the ChaLearn Gesture Challenge. First, we compare the results obtained for

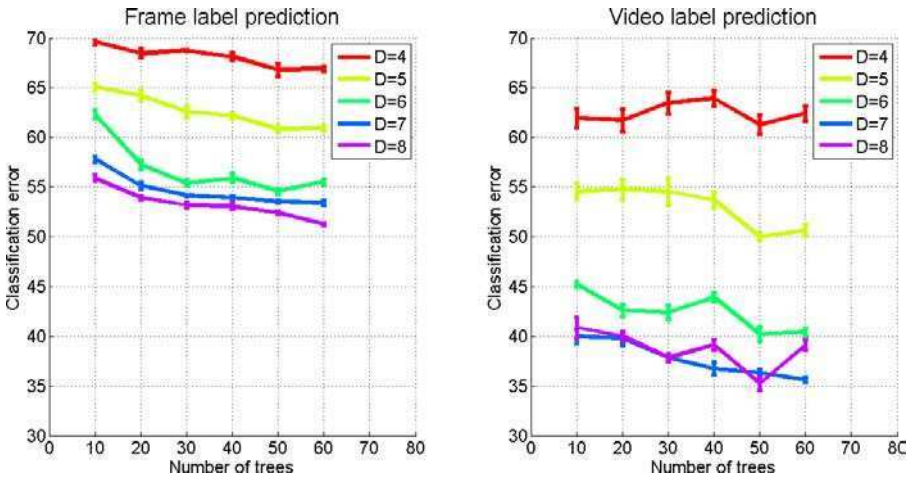


Figure 4.4: Effect of the training parameters for the frame label classification error $p(\mathbf{y}|\mathbf{x})$ (left) and video label classification error $p(\mathbf{y}|\mathcal{V})$ (right) in the devel11 batch using the transfer learning decision forests.

different parameters of the transfer learning decision forests, and then we compare these results with the ones reported in related works. For the MHI computation in this section, we set the temporal extent $\tau = 8$, the threshold $\xi = 25$, and reduce the spatial resolution of each frame to $\omega_1 \times \omega_2 = 16 \times 12$ pixels. All the results showed in this section are computed following the same procedure. First, we train the classifier with the training videos of the corresponding batches and then we compute the classification error using the testing videos of the batch.

Transfer Decision Learning Parameters

To obtain a general idea of the effect of the training parameters, Figure 4.5 evaluates the classification error for different combinations of training parameters. We report the average classification error obtained in the *devel* batches. We use the temporal segmentation of the videos provided by the ChaLearn competition organizers. The experiments show that when the mixing coefficient γ is between 25% and 50%, the classification error is the smallest. This means that we obtain improvements when transferring knowledge from related tasks but, nevertheless, we still need to make the decision trees grow using information of the target task.

It is important to remark that when $\gamma = 0$ we are not using the training

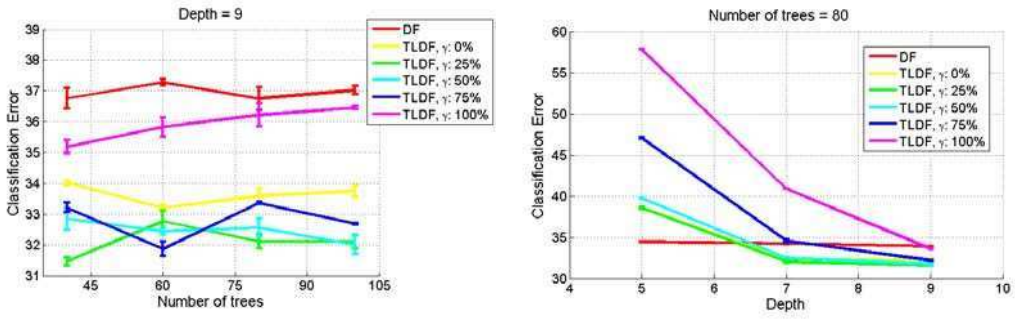


Figure 4.5: Comparison of the classification error using different combination of training parameters.

data of the source tasks and our mixed information gain simplifies to the usual information gain, thus, only the label propagation extension is being used. The classification error for the case $\gamma = 0$ indicates that we achieve an improvement using the label propagation alone. We obtain an additional improvement when γ is between 25% and 75%, therefore we can conclude that both extensions are important to reduce the classification error.

The maximum depth of the trees is a highly relevant parameter for the transfer learning decision forests, and has some influence for the classification decision forests. As expected, the greater the maximum depth, the smaller the classification error. It is interesting to observe that the difference in the classification error between different values of the mixing coefficients γ is reduced when the maximum depth is increased.

Figure 4.6 shows the confusion matrices for the classifiers of the transfer learning decision forests (TLDFs) and the decision forests (DFs) in the batches *devel*06, *devel*11 and *devel*14. To train the TLDFs, we set the number of trees $T = 50$, the maximum depth $D = 8$, the mixing coefficient $\gamma = 25\%$, and the size of the subset $|\mathcal{T}| = 50$. In these batches the TLDFs classifier shows improvements over the DFs classifier. The improvement is not uniform for all the gestures of the batches, but only for some of them. This is because not all the gestures can benefit from the training data of the source tasks. Only the gestures that have, at least, one similar gesture in a source task show improvements.

The confusion matrix for the *devel*06 batch in Figure 4.6 shows significant

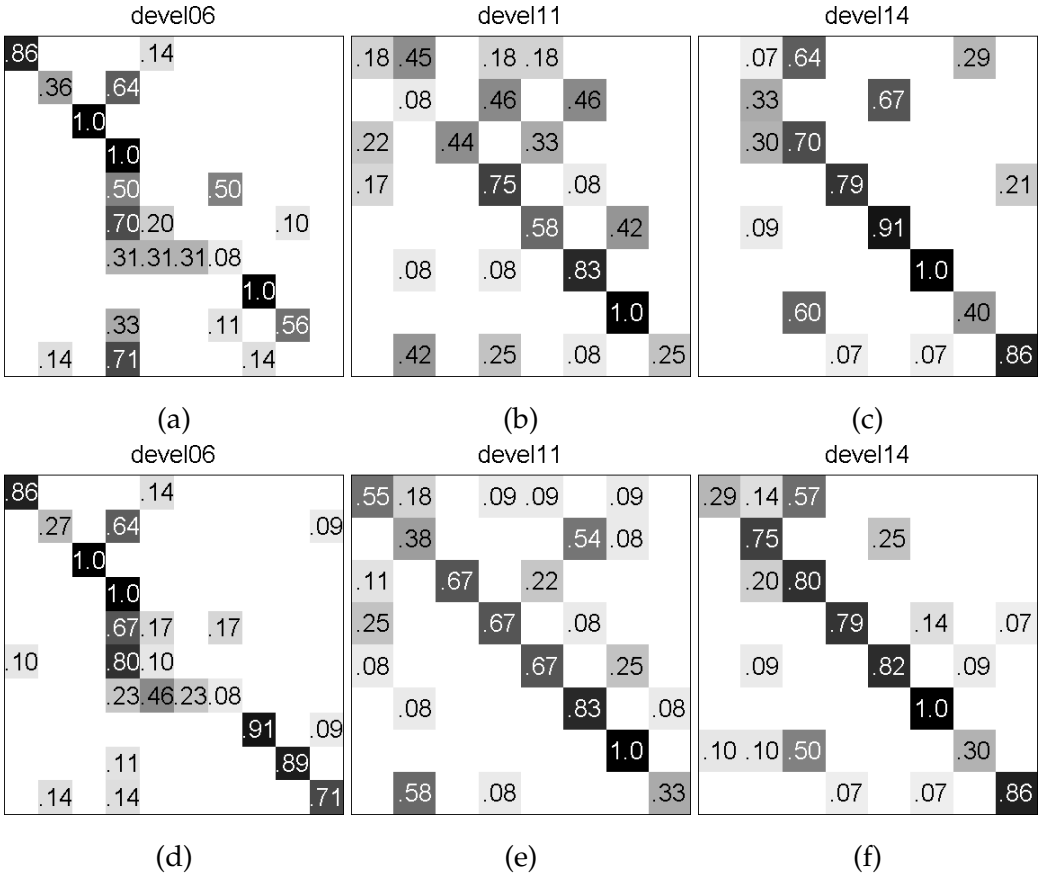


Figure 4.6: Comparison of the confusion matrices obtained using the DF (a), (b), (c) and TLDF (d), (e), (f) classifiers on the *devel*06, *devel*11 and *devel*14 batches.

improvements in the classification of the last gesture. Figure 4.7 shows a representative image of that gesture and similar gestures in the *devel*13 and *devel*15 batches. The person in front of the camera moves the left hand to a fixed position and then shows a similar pattern of the fingers, for all these gestures. The frames of these gestures are usually found in the same leaf after training the decision forest.

Devel and Final Data

Table 4.2 compares our results for the development batches of the ChaLearn Challenge with the ones previously reported in [Lui12, MNG13], using the evaluation procedure of the competition from [GAJ+12]. To train the TLDFs,

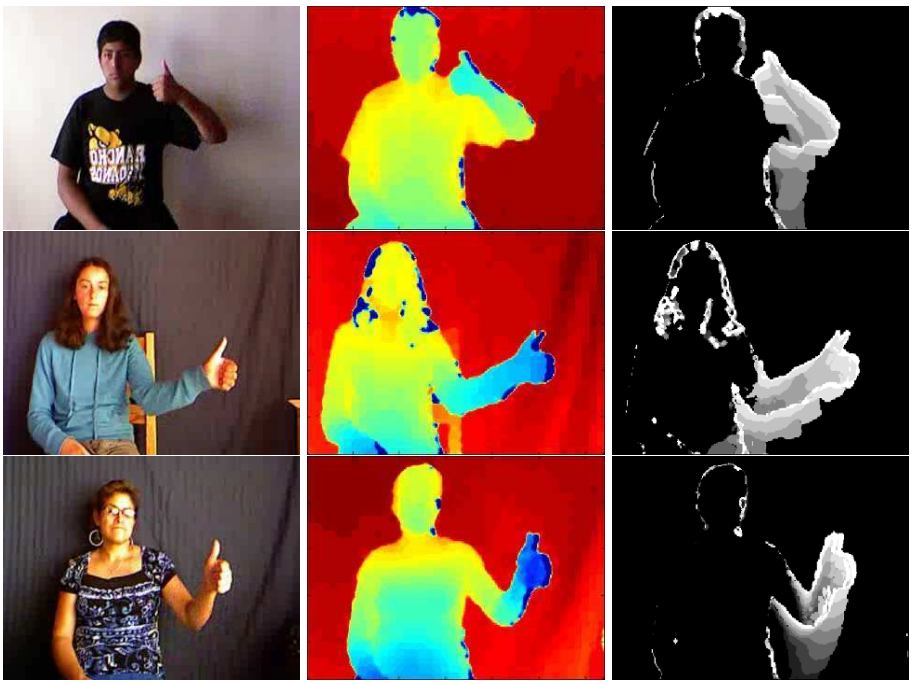


Figure 4.7: Similar gestures in different batches. The first, second and third rows show a gesture in the *devel*06, *devel*13 and *devel*15 batches respectively. The first column shows the RGB image for a representative frame of the video, the second column shows the corresponding depth image and the last column shows the MHI.

we set the number of trees $T = 50$, the maximum depth $D = 8$, the mixing coefficient $\gamma = 25\%$, and the size of the search space $|\mathcal{T}| = 50$. As shown in Table 4.2, for most batches, our transfer learning decision forests obtain improvements over the DFs, and for some batches, they obtain the smallest errors.

Table 4.3 compares our results for the final evaluation data with the final results of the competition from [GAJ+13]. The approach of the Joewan team is described in [WRLD13]. They propose 3D EMoSIFT, a novel feature which fuses RGB-D data and is invariant to scale and rotation. Most of the other teams have not described their approach in a publication.

	Principal motion	[Lui12]	[MNG13]	DF	TLDF
devel01	6.67%	–	13.33%	4.44%	**3.89%**
devel02	33.33%	–	35.56%	28.89%	**25.00%**
devel03	71.74%	–	71.74%	65.22%	**62.50%**
devel04	24.44%	–	**10.00%**	25.56%	13.89%
devel05	**2.17%**	–	9.78%	3.26%	4.89%
devel06	43.33%	–	**37.78%**	48.89%	45.00%
devel07	23.08%	–	18.68%	19.78%	**14.29%**
devel08	10.11%	–	**8.99%**	17.98%	10.11%
devel09	19.78%	–	**13.19%**	19.78%	15.38%
devel10	56.04%	–	**50.55%**	59.34%	60.99%
devel11	**29.35%**	–	35.87%	42.39%	39.13%
devel12	21.35%	–	22.47%	23.60%	**19.10%**
devel13	12.50%	–	**9.09%**	19.32%	25.00%
devel14	39.13%	–	28.26%	45.65%	**27.71%**
devel15	40.22%	–	**21.74%**	26.09%	31.52%
devel16	34.48%	–	31.03%	31.03%	**27.01%**
devel17	48.91%	–	**30.43%**	53.26%	45.11%
devel18	44.44%	–	40.00%	40.00%	**38.33%**
devel19	60.44%	–	**49.45%**	60.44%	54.95%
devel20	39.56%	–	**35.16%**	46.15%	67.22%
Avg.	33.15%	**24.09%**	28.73%	34.14%	31.55%

Table 4.2: Comparison of reported results using the Levenshtein distance.

Team	Private score set on final set #1	For comparison score on final set #2
alfnie	0.0734	0.0710
Joewan	0.1680	0.1448
Turtle Tamers	0.1702	0.1098
Wayne Zhang	0.2303	0.1846
Manavender	0.2163	0.1608
HIT CS	0.2825	0.2008
Vigilant	0.2809	0.2235
Our Method	0.2834	0.2475
Baseline method 2	0.2997	0.3172

Table 4.3: ChaLearn results of round 2.

4.2 Character Recognition

Optical character recognition, seeks to classify a given image into one of the characters of a given alphabet. Most methods have focused on recognizing characters from the English alphabet [LBBH98]. The recognition of characters from other alphabets, such as French, implies collecting a new training data-set [GA11]. Gathering those data-sets can be extremely time consuming and might have a significant impact on the overall cost of the final system. Moreover, machine learning techniques are not well suited for the case of very small training sets. In this section, we apply our novel method for transfer learning to the optical character recognition problem.

4.2.1 Experiments

The MNIST [LBBH98] data-set has been used to compare the transfer learning results of [QSCP10, FCGT12]. A small sample of the training set is used to simulate the situation when only a limited number of labeled examples is available. For each digit $0 \ldots 9$, we consider a binary task where label $+1$ means that the example belongs to the digit associated with the respective task, and label -1 means the opposite. We randomly choose 100 training

samples for each task and test them on the 10,000 testing samples. The experiments are repeated ten times.

To obtain a general idea of the effect of the training parameters, Figures 4.8 and 4.9 evaluates for each digit the recognition rates for different combinations of training parameters of the Transfer Learning Decision Forests (TLDFs) and Decision Forests (DFs). The experiments show that TLDFs achieve higher recognition rates when compared with DFs and that when the mixing coefficient γ is greater than 25%, the recognition rate is the highest. This means that we obtain improvements when transferring knowledge from related tasks. The difference in the recognition rate is not large when the mixing coefficient γ is greater than 25%. In addition, the maximum depth of the trees is a highly relevant parameter for the TLDFs. As expected, the greater the maximum depth, the higher the recognition rate.

Table 4.4 compares the recognition rates of the TLDFs and [QSCP10, FCGT12]. We train the TLDFs with $D = 6, T = 40, \gamma = 50\%$, and we do not apply any preprocessing to the sample images. The experiments show that our approach achieves better results than state-of-the-art methods in terms of transfer learning.

To analyze the influence of the number of training samples, we compare the classification error of the TLDFs with the classification error of the DFs. Figure 4.10 plots the classification error as a function of the number of training samples for each classifier. As we did previously, we compute the classification error using the 10000 test samples of the MNIST dataset. We see that the classification error of the TLDF is smaller than that of the DF. In addition, it is interesting to note that the gap between both classifiers is larger when the number of training samples is smaller, thus suggesting that the TLDF is more suitable than DF for small training samples.

4.3 Resumen

En este capítulo aplicamos los bosques de decisión propuestos en el capítulo anterior a dos problemas de visión por ordenador. En primer lugar, utilizamos los bosques de decisión en el problema de reconocimiento de gestos.

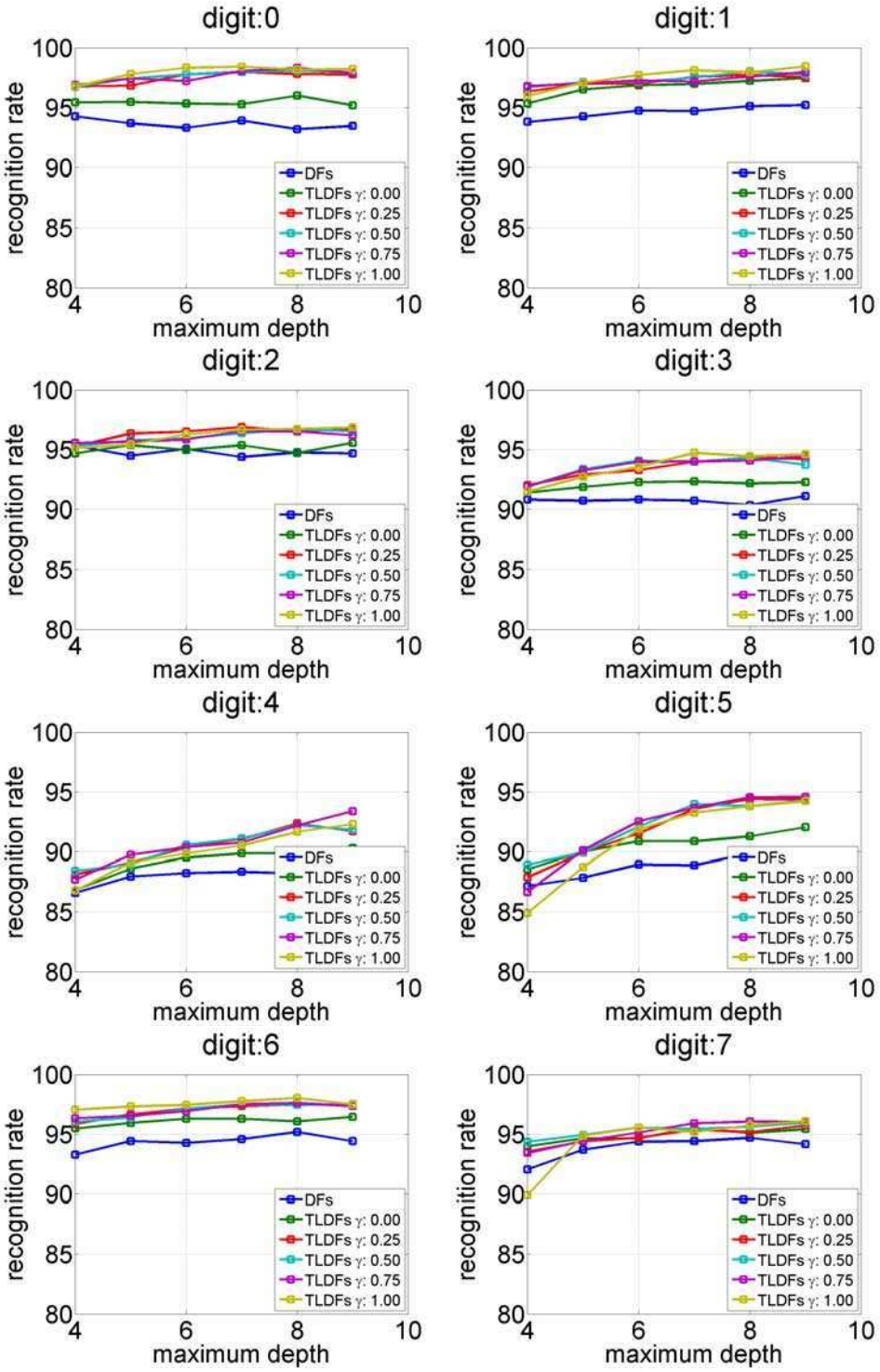


Figure 4.8: Recognition rates for different combination of TLDFs and DFs parameter (digits 0-7). In all the cases the number of trees is 40.

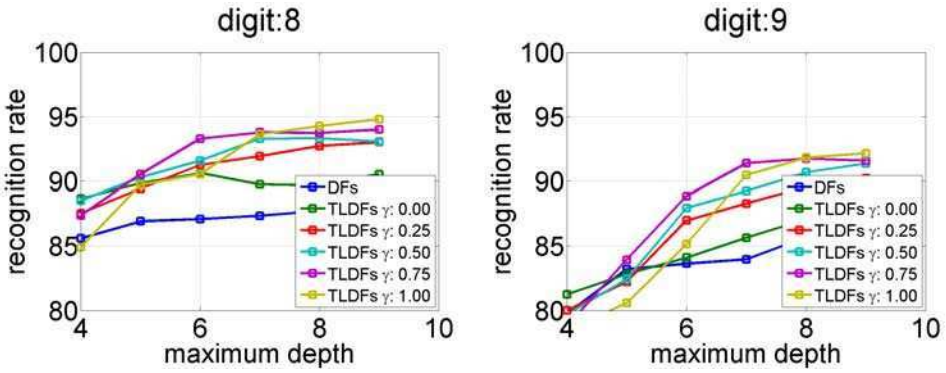


Figure 4.9: Recognition rates for different combination of TLDFs and DFs parameter (digits 8 and 9). In all the cases the number of trees is 40.

	Adaboost [FCGT12]	MTL [QSCP10]	MT-Adaboost [FCGT12]	Our approach
1/-1	91.77±1.89%	96.80±1.91%	96.80±0.56%	**97.23±0.44%**
2/-2	83.14±2.35%	69.95±2.68%	86.87±0.68%	**96.74±0.31%**
3/-3	82.96±1.24%	74.18±5.54%	87.68±1.04%	**93.29±0.96%**
4/-4	83.98±1.41%	71.76±5.47%	**90.38±0.71%**	90.10±1.23%
5/-5	78.42±0.69%	57.26±2.72%	84.25±0.73%	**92.79±1.62%**
6/-6	88.95±1.60%	80.54±4.53%	92.88±0.90%	**97.35±0.45%**
7/-7	87.11±0.90%	77.18±9.43%	92.81±0.57%	**95.55±1.39%**
8/-8	77.51±1.90%	65.85±2.50%	85.28±1.73%	**91.99±1.30%**
9/-9	81.84±1.85%	65.38±6.09%	**86.90±1.26%**	84.76±1.67%
0/-0	93.66±1.29%	97.81±1.01%	97.14±0.42%	**98.05±0.28%**
Avg.	84.93%	75.67%	90.10%	**93.78%**

Table 4.4: Comparison of recognition rates in the MNIST data-set.

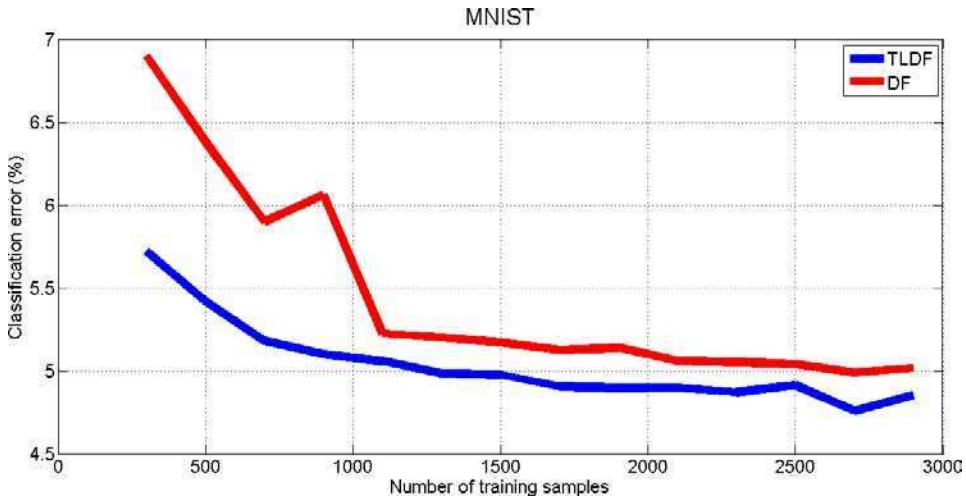


Figure 4.10: This figure evaluates the classification error as a function of the number of training samples.

Posteriormente utilizamos los bosques de decisión en el problema de reconocimiento de carácteres. En ambos casos, comparamos los resultados obtenidos utilizando los bosques de decisión propuestos con los bosques de decisión tradicionales.

Para evaluar y comparar los resultados obtenidos en el problema de reconocimiento de gestos usamos el conjunto de datos de la competencia ChaLearn. El cual está diseñado para evaluar y comparar métodos de transferencia de aprendizaje. El conjunto de datos consta de varios lotes de gestos, en los cuales hay un solo video de entrenamiento de cada gesto. El objetivo es predecir el gesto de los restantes videos del lote. En los videos de entrenamiento hay solamente un gesto por video, pero en el resto de los videos puede haber más de un gesto por video. Por lo tanto, el resto de los videos son segmentados temporalmente usando enfoque propuesto por los organizadores.

Para reconocer gestos desarrollamos un sistema que utiliza características basadas en imágenes de historia del movimiento del video. Para calcular cada imagen de de historia de movimiento se utiliza la ecuación (4.1), la cual combina la segmentación de movimiento de la imagen actual con la imagen de de historia de movimiento calculada previamente. Se obtene un modelo predictivo para cada imagen de un video utilizando los bosques de decisión.

Luego, las predicciones para cada imagen, se combinan utilizando la suposición de que son independientes, obteniendo un clasificador Bayesiano ingenuo.

Los experimentos mostraron que nuestro método no tiene un margen uniforme de mejora en todos los lotes. Sin embargo, hemos demostrado que cuando hay tareas de origen relacionadas con la tarea de destino, se obtiene una mejora. Además, los experimentos mostraron que las mejores tasas de clasificación se obtienen cuando el coeficiente de mezcla se encuntra entre 25 % y 50 %. Esto significa, que obtenemos mejoras cuando transferimos el conocimiento de las tareas relacionadas, pero, sin embargo, todavía tenemos que entrenar utilizando la información de la tarea de destino. Finalmente, los experimentos también mostraron que ambas extensiones son importantes a fin de obtener los errores de clasificaciń más pequeños.

El otro problema importante de visión por ordenador que hemos considerado en este capítulo es el de reconocimiento óptico de caracteres. Para ello hemos realizado experimentos utilizando el conjunto de datos MNIST [LBBH98]. En este caso, hemos sido capaces de demostrar que los bosques de decisión propuestos en esta tesis superan a los bosques de decisión tradicionales en conjuntos de entrenamiento de diferentes tamaños. En particular, la brecha entre ambos clasificadores es mayor cuando el tamaño del conjunto de entrenamiento es pequeño, lo que sugiere que los bosques de decisión propuestos son más adecuados cuando se tienen pocas muestras de entrenamiento. Además, comparamos los bosques de decisión propuestos con otros métodos del estado del arte en transferencia de aprendizaje y los experimentos demostraron que nuestro enfoque logra mejores resultados.

Chapter 5

Conclusions and Perspective

A desirable feature of machine learning methods is to transfer information from one context to another, in order to achieve higher performance with smaller training datasets. The underlying motivation is the fact that people can extract knowledge learned from previous tasks to solve new tasks faster or with better solutions. However, a large part of the machine learning and computer vision literature focuses on obtaining impressive results using classifiers trained on large datasets specifically collected for the task at hand. An important type of such classifiers are the decision forests, which have gained popularity recently.

In this thesis we have extended the decision forests classifiers to the transfer learning setting. We developed an instance-based transfer approach that exploits source and target data to improve the accuracy of the predictions of the novel classifier. Our approach uses the instances of the source and target datasets to train the decision forests.

Few researchers have addressed the problem of transfer learning using hierarchical classifiers in the past. We introduced two extensions that make our approach distinctive. The first extension is the mixed information gain (Section 3.4). The second extension is the label propagation through the leaves of the decision trees (Section 3.5).

The first key contribution is the novel objective function for finding the parameters of the internal nodes in the decision trees. The aim of this extension is to transfer information from one context to another. In order to

achieve this goal the mixed information gain penalizes split functions with a high information gain in the target task and a low information gain in the source tasks. In this way, the algorithm applies the regularities present in the source tasks in the target task.

The second key contribution is to propagate labels through leaves in order to infer the manifold structure of the feature space. The aim of this step is to assign a predictive model to the leaves without training samples of the target task, after the trees of the decision forest are grown. An implicit assumption of this step is that nearby leaves should have similar predictive models.

Additionally, in Section 3.4 we have proved some relevant properties of the mixed information gain. The properties proved address the question of how many source tasks are required to improve the performance using transfer learning decision forests. The first property states that many source tasks can be combined to obtain a more complex source task. The second property explain how additional data has a positive effect in estimating the entropy associated to a dataset.

We have applied our novel transfer learning decision forests to two challenging computer vision problems. First, we have applied our transfer learning decision forests to the gesture recognition problem. It is difficult to collect large datasets for gesture recognition, therefore it is very interesting to design classifiers that can achieve high recognition rates with small training datasets. We validated our gesture recognition system using the ChaLearn gesture challenge dataset [GAJ+12].

The experiments showed that our method does not have a uniform margin of improvement over all the tasks. However, we demonstrated that when there are source tasks related to the target task, we obtain greater improvements. Additionally, the experiments showed that the best classification rates are obtained when the mixing coefficient is between 25% and 50%. This means that we obtain improvements when we transfer knowledge from related tasks but, nevertheless, we still need to train using the information of the target task. Finally, the experiments also showed that both extensions are important in order to obtain smaller classification errors.

The other important computer vision problem that we considered in this thesis is optical character recognition. To this end we performed experiments using the MNIST dataset [LBBH98]. Here, we were able to show that the transfer learning decision forests outperform the traditional decision forests for training datasets of different size. Moreover, the gap between both classifiers is larger when the size of the training dataset is small, thus suggesting that the transfer learning decision forests are more suitable for small training datasets. We compared our transfer learning decision forests with other state-of-the art methods and the experiments showed that our approach achieves better results in terms of transfer learning.

A main problem of the transfer learning setting is negative transfer. Which happens when the learner performs worse when source tasks are included. Little research work has been published on this topic and the approach proposed in this thesis could benefit of avoiding this issue. It has been suggested in the literature that a possible approach to overcome this problem is to define a suitable transferability measure to select relevant source tasks to extract knowledge from learning the target tasks.

The extension proposed to the decision forests in this thesis is a instance-based transfer learning approach. Thus, our method shares some of the limitations that other instance-based transfer learning approaches have. A main limitation of these type of approaches is that there is no intermediate representation to transfer, therefore in the case of having a lot of tasks the number of instances to transfer might become excessively large and the computational cost to train a classifier might be high.

There are some alternative approaches to transfer knowledge in the decision forest framework, not explored in this thesis, that might be interesting to analyze. An interesting alternative would be to replace the random node optimization by some optimization approach that takes into account the source tasks. For example, by modifying the way the random subset of node parameters is generated.

Recently, transfer learning techniques have been applied successfully in many real-world applications. It would be interesting to explore some of the other problems where transfer learning techniques have been applied suc-

cessfully to further validate our approach. In particular, it would be interesting to apply the transfer learning decision forests in other competitions. For example the ECML/PKDD-2006 discovery challenge, is a transfer learning competition where the task is to handle personalized spam filtering and generalization across related learning tasks.

Finally, the approach presented here assumes that the source and target tasks have the same feature space. However, in many applications, we may wish to transfer knowledge across domains or tasks that have different feature spaces. This type of transfer learning setting is usually referred as heterogeneous transfer learning and it would be interesting to explore how to adapt the random forest framework to this context.

During this thesis, we have tried to push forward the state-of-the-art methods for transfer learning. We think that this is a relevant area of research because it is impractical to collect huge data-sets for every new task we want to solve. However, the theoretical analysis concerning the novel method proposed in this thesis allows us to conclude that we achieve low generalization errors when the training set is large. This is a fundamental assumption when the generalization error of a classifier is analyzed and, is ubiquitous in the machine learning literature. In the future, it would be interesting to explore new ways to analyze theoretically the error made by a classifier when the number of training samples is limited.

Capítulo 5

Conclusiones y Perspectiva

Una característica deseable de los métodos de aprendizaje automático es la posibilidad de transferir información de un contexto a otro, con el fin de lograr un mayor rendimiento con conjuntos de entrenamiento más pequeños. La motivación subyacente es el hecho de que la gente puede extraer conocimiento aprendido de las tareas anteriores para resolver nuevas tareas más rapidamente o encontrando una mejor solución. Sin embargo, una gran parte de la literatura de aprendizaje automático y visión por ordenador se centra en la obtención de resultados impresionantes utilizando clasificadores entrenados en grandes conjuntos de datos recopilados específicamente para la tarea en cuestión. Un tipo importante de tales clasificadores son los bosques de decisiones, que han ganado popularidad recientemente.

En esta tesis hemos presentado un nuevo método de transferencia de aprendizaje que extiende los bosques de decisión. El método propuesto es un enfoque de transferencia basado en instancias que explota los datos de origen y de destino para mejorar la precisión de las predicciones del nuevo clasificador. Nuestro enfoque utiliza las instancias de los conjuntos de datos de origen y de destino para entrenar a los bosques de decisión.

Pocos investigadores han abordado el problema de transferencia de aprendizaje usando clasificadores jerárquicos en el pasado. Hemos introducido dos extensiones que hacen que nuestro enfoque distintivo. La primera extensión es la ganancia de información mixta (Sección 3.4). La segunda extensión es la propagación de etiquetas a través de las hojas de los árboles de

decisión (Sección 3.5).

La primera contribución clave es la nueva función objetivo para seleccionar los parámetros de los nodos internos en los bosques de decisión. El objetivo de esta función es lograr que se transferencia información de un contexto a otro. A fin de lograr este objetivo, la ganancia de información mixta penaliza funciones de división con una ganancia de información alta en la tarea destino y una ganancia de información baja en las tareas de origen. De esta manera, el algoritmo extrae las regularidades presentes en las tareas de origen y las utiliza en la tarea objetivo.

La segunda contribución clave es propagar etiquetas a través de las hojas con el fin de inferir la estructura de la variedad del espacio de características. El objetivo de este paso es asignar un modelo predictivo a las hojas sin muestras de entrenamiento de la tarea de destino, después de haber entrenado cada uno de los árboles del bosque decisión. Un supuesto implícito de este paso es que las hojas cercanas deben tener modelos predictivos similares.

Adicionalmente, en la sección 3.4 hemos demostrado algunas propiedades relevantes de la ganancia de información mixta. Las propiedades demostradas intentar responder, parcialmente, cuál es el número de tareas de origen que se requieren para mejorar el rendimiento utilizando los nuevos bosques de decisión propuestos. La primera propiedad establece que muchas de las tareas de origen se pueden combinar para obtener una tarea origen más compleja. La segunda propiedad explica cómo los datos adicionales tienen un efecto positivo en la estimación de la entropía asociada a un conjunto de datos.

Hemos aplicado los bosques de decisión, propuestos en esta tesis, a dos problemas de visión por ordenador desafiantes. En primer lugar, hemos aplicado nuestros bosques de decisión al problema de reconocimiento de gestos. Es difícil reunir grandes conjuntos de datos para el entrenar y validar sistemas de reconocimiento de gestos, por lo que es muy interesante diseñar clasificadores que puedan alcanzar altas tasas de reconocimiento con pequeños conjuntos de entrenamiento. Hemos validado nuestro sistema de reconocimiento de gestos usando los datos de la competencia de gestos

ChaLearn [GAJ+12].

Los experimentos mostraron que nuestro método no tiene un margen uniforme de mejora con respecto a todas las tareas. Sin embargo, hemos demostrado que cuando hay tareas de origen relacionadas con la tarea de destino, se obtiene una mejora. Además, los experimentos mostraron que las mejores tasas de clasificación se obtienen cuando el coeficiente de mezcla se encuentra entre 25 % y 50 %. Esto significa, que obtenemos mejoras cuando transferimos el conocimiento de las tareas relacionadas, pero, sin embargo, todavía tenemos que entrenar utilizando la información de la tarea de destino. Finalmente, los experimentos también mostraron que ambas extensiones son importantes a fin de obtener los errores de clasificación más pequeños.

El otro problema importante de visión por ordenador que hemos considerado en esta tesis es el de reconocimiento óptico de caracteres. Para ello hemos realizado experimentos utilizando el conjunto de datos MNIST [LBBH98]. En este caso, hemos sido capaces de demostrar que los bosques de decisión propuestos en esta tesis superan a los bosques de decisión tradicionales en conjuntos de entrenamiento de diferentes tamaños. En particular, la brecha entre ambos clasificadores es mayor cuando el tamaño del conjunto de entrenamiento es pequeño, lo que sugiere que los bosques de decisión propuestos son más adecuados cuando se tienen pocas muestras de entrenamiento. Además, comparamos los bosques de decisión propuestos con otros métodos del estado del arte en transferencia de aprendizaje y los experimentos demostraron que nuestro enfoque logra mejores resultados.

Un problema que algunos métodos de transferencia de aprendizaje suelen tener es el de transferencia negativa. Que sucede cuando el método de aprendizaje reduce su tasa de reconocimiento cuando se incluyen tareas de origen. Poco trabajo de investigación se ha publicado sobre este tema y el enfoque propuesto en esta tesis podría beneficiarse de evitar este problema. Se ha sugerido en la literatura que un posible enfoque para superar este problema es definir una medida de transferencia adecuada para seleccionar la tareas origen relevantes de las cuales extraer conocimiento que pueda ser

utilizado en las tareas de destino.

La extensión propuesta para los bosques de decisión en esta tesis se puede clasificar como transferencia de aprendizaje basado en instancias. Por lo tanto, nuestro método comparte algunas de las limitaciones que otros enfoques de transferencia de aprendizaje basado en instancias tienen. Una limitación principal de este tipo de enfoques es que no existe una representación intermedia para transferir, por lo tanto, en el caso de tener una gran cantidad de tareas, el número de instancias para transferir podría llegar a ser excesivamente grande y el costo computacional para entrenar a un clasificador puede ser alto.

Hay algunos enfoques alternativos para transferir conocimiento usando bosques de decisión que no fueron explorados en esta tesis, y podrían ser interesantes de analizar. Una alternativa es la de modificar el proceso de optimización de parámetros de los nodo por algún enfoque de optimización que tiene en cuenta las tareas de origen. Por ejemplo, al modificar la forma se genera el subconjunto aleatorio de parámetros de nodo.

Recientemente, las técnicas de transferencia de aprendizaje se han utilizado con éxito en muchas aplicaciones del mundo real. Sería interesante explorar algunos de los otros problemas en los que las técnicas de aprendizaje de transferencia se han aplicado con éxito para validar aún más nuestro enfoque. En particular, sería interesante aplicar los bosques de decisión propuestos en esta tesis en otras competencias. Por ejemplo en la competencia ECML / PKDD-2006, en la cual la tarea consiste en personalizar el filtrado de spam.

Por último, el enfoque que aquí se presenta supone que las tareas de origen y de destino tienen el mismo espacio de características. Sin embargo, en muchas aplicaciones, es posible que se desee transferir conocimiento a través de dominios o tareas que tienen diferente espacios de características. Este tipo de transferencia de aprendizaje es conocido en la literatura como transferencia de aprendizaje heterogéneo y sería interesante explorar cómo adaptar el los bosques de decisión a este contexto.

Durante esta tesis, hemos tratado de avanzar el estado del arte en transferencia de aprendizaje. Creemos que este es un área de investigación re-

levante, ya que no es práctico crear conjuntos de entrenamiento grandes para cada nueva tarea que se desee resolver. Sin embargo, el análisis teórico del nuevo método propuesto en esta tesis nos permite concluir que logramos bajos errores de generalización cuando el conjunto de entrenamiento es grande. Esta es una suposición fundamental cuando se analiza el error de generalización de un clasificador y, es omnipresente en la literatura de aprendizaje automático. En el futuro, sería interesante explorar nuevas formas de analizar teóricamente el error cometido por un clasificador cuando el número de muestras de entrenamiento es limitado.

Bibliography

[AC12] Ender Konukoglu Antonio Criminisi, Jamie Shotton, *Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning*, Foundations and Trends in Computer Graphics and Vision **7** (2012), no. 2-3, 81–227.

[AG97] Yali Amit and Donald Geman, *Shape quantization and recognition with randomized trees*, Neural computation **9** (1997), no. 7, 1545–1588.

[ATKI12] Md. Atiqur Rahman Ahad, J. K. Tan, H. Kim, and S. Ishikawa, *Motion history image: its variants and applications*, MVA (2012).

[AZ11] Y. Aytar and A. Zisserman, *Tabula rasa: Model transfer for object category detection*, CVPR, 2011.

[BD96] A. Bobick and J. Davis, *An appearance-based representation of action*, ICPR 1996, 1996, pp. 307–312.

[BD01] ———, *The recognition of human movement using temporal templates*, TPAMI (2001).

[BD10] Gérard Biau and Luc Devroye, *On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification*, Journal of Multivariate Analysis **101** (2010), no. 10, 2499–2518.

[BDL08] Gérard Biau, Luc Devroye, and Gábor Lugosi, *Consistency of random forests and other averaging classifiers*, The Journal of Machine Learning Research **9** (2008), 2015–2033.

[BFSO84] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*, Chapman and Hall/CRC, 1984.

[Bia12] Gérard Biau, *Analysis of a random forests model*, The Journal of Machine Learning Research **98888** (2012), no. 1, 1063–1095.

[Bre01] Leo Breiman, *Random forests*, Machine Learning (2001).

[Bre04] Leo Breiman, *Consistency for a simple model of random forests*, Statistical Department, University of California at Berkeley. Technical Report (2004), no. 670.

[BT10] Alessandro Bergamo and Lorenzo Torresani, *Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach*, NIPS, 2010, pp. 181–189.

[BU95] Carla E Brodley and Paul E Utgoff, *Multivariate decision trees*, Machine learning **19** (1995), no. 1, 45–77.

[BU05] Evgeniy Bart and Shimon Ullman, *Cross-generalization: learning novel classes from a single example by feature replacement*, CVPR, 2005.

[BWK+04] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and J. M. Brady, *A linguistic feature vector for the visual interpretation of sign language*, ECCV 2004, 2004.

[CKY08] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina, *An empirical evaluation of supervised learning in high dimensions*, Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 96–103.

[CT06] Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, Wiley-Interscience, 2006.

[DGL96] Luc Devroye, László Györfi, and Gabor Lugosi, *A probabilistic theory of pattern recognition*, Springer, 1996.

[DMdF14] Misha Denil, David Matheson, and Nando de Freitas, *Narrowing the gap: Random forests in theory and in practice*, ICML, 2014.

[DYXY07] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu, *Boosting for transfer learning*, Proceedings of the 24th international conference on Machine learning (New York, NY, USA), ICML '07, ACM, 2007, pp. 193–200.

[FCGT12] Jean Baptiste Faddoul, Boris Chidlovskii, Rémi Gilleron, and Fabien Torre, *Learning Multiple Tasks with Boosted Decision Trees*, ECML/PKDD, 2012.

[FFFP06] Li Fei-Fei, Rob Fergus, and Pietro Perona, *One-shot learning of object categories*, TPAMI **28** (2006), 594–611.

[FFW07] Ali Farhadi, David Forsyth, and Ryan White, *Transfer learning in sign language*, CVPR, 2007, pp. 1–8.

[FGMR10] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan, *Object detection with discriminatively trained part-based models*, TPAMI **32** (2010), no. 9, 1627–1645.

[Fin05] Michael Fink, *Object classification from a single example utilizing class relevance metrics*, NIPS, 2005, pp. 449–456.

[FS97] Yoav Freund and Robert E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. Comput. Syst. Sci. **55** (1997), no. 1, 119–139.

[GA11] Emmanuele Grosicki and Haikal El Abed, *Icdar 2011 - french handwriting recognition competition*, ICDAR, 2011, pp. 1459–1463.

[GAJ+12] Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, Ben Hamner, and Hugo Jair Escalante, *Chalearn gesture challenge: Design and first results*, Workshop on Gesture Recognition and Kinect Demonstration Competition, 2012.

[GAJ+13] Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, Hugo Jair Escalante, and Ben Hamner, *Results and analysis of the chalearn gesture challenge 2012*, WDIA, Lecture Notes in Computer Science, vol. 7854, Springer, 2013, pp. 186–204.

[GF12] Matthieu Guillaumin and Vittorio Ferrari, *Large-scale knowledge transfer for object localization in imagenet*, CVPR, 2012, pp. 3202–3209.

[GLC11] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa, *Domain adaptation for object recognition: An unsupervised approach*, ICCV, 2011, pp. 999–1006.

[GMOS95] Michael T. Goodrich, Vincent Mirelli, Mark Orletsky, and Jeffery Salowe, *Decision tree construction in fixed dimensions: Being global is hard but local greed is good*, Tech. report, 1995.

[GMP00] Michelangelo Grigni, Vincent Mirelli, and Christos H. Papadimitriou, *On the difficulty of designing good classifiers*, SIAM J. Comput. **30** (2000), no. 1, 318–323.

[HKS93] David Heath, Simon Kasif, and Steven Salzberg, *Induction of oblique decision trees*, IJCAI, 1993, pp. 1002–1007.

[Ho98] Tin Kam Ho, *The random subspace method for constructing decision forests*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **20** (1998), no. 8, 832–844.

[HR76] Laurent Hyafil and Ronald L. Rivest, *Constructing optimal binary decision trees is np-complete*, Information Processing Letters (1976), 15–17.

[HTF03] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, Springer, 2003.

[KZL12] A. Kurakin, Zhengyou Zhang, and Zicheng Liu, *A real-time system for dynamic hand gesture recognition with a depth sensor*, EUSIPCO 2012, 2012, pp. 1980–1984.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 1998, pp. 2278–2324.

[LFW04] Kobi Levi, Michael Fink, and Yair Weiss, *Learning from a small number of training examples by exploiting object categories*, CVPRW, 2004, pp. 96–102.

[LJ06] Yi Lina and Yongho Jeon, *Random forests and adaptive nearest neighbors*, Journal of the American Statistical Association **101** (2006), 578–590.

[LSSB09] Christian Leistner, Amir Saffari, Jakob Santner, and Horst Bischof, *Semi-supervised random forests*, ICCV 2009, 2009, pp. 506–513.

[LST11] Joseph J. Lim, Ruslan Salakhutdinov, and Antonio Torralba, *Transfer learning by borrowing examples for multiclass object detection*, NIPS, 2011.

[Lui12] Yui Man Lui, *Human gesture recognition on product manifolds*, JMLR **13** (2012), 3297−3321.

[LYZH10] Jie Liu, Kai Yu, Yi Zhang, and Yalou Huang, *Training conditional random fields using transfer learning for gesture recognition*, ICDM, 2010, pp. 314 – 323.

[LZL08] Wanqing Li, Zhengyou Zhang, and Zicheng Liu, *Graphical modeling and decoding of human actions*, MMSP, 2008, pp. 175–180.

[MKS94] Sreerama K. Murthy, Simon Kasif, and Steven Salzberg, *A system for induction of oblique decision trees*, Journal of Artificial Intelligence Research **2** (1994), 1–32.

[MKS+11] Bjoern H Menze, B Michael Kelm, Daniel N Splitthoff, Ullrich Koethe, and Fred A Hamprecht, *On oblique random forests*, Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 453–469.

[MNG13] Manavender R. Malgireddy, Ifeoma Nwogu, and Venu Govindaraju, *Language-motivated approaches to action recognition*, JMLR **14** (2013), 2189−2212.

[MS95] Sreerama Murthy and Steven Salzberg, *Decision tree induction: How effective is the greedy heuristic?*, In Proceedings of the First International Conference on Knowledge Discovery and Data Mining, Morgan Kaufmann, 1995, pp. 222–227.

[Mur12] Kevin P Murphy, *Machine learning: a probabilistic perspective*, Cambridge, MA, 2012.

[PKZ13] Yuru Pei, Tae-Kyun Kim, and Hongbin Zha, *Unsupervised random forest manifold alignment for lipreading*, ICCV, 2013.

[PY10] Sinno Jialin Pan and Qiang Yang, *A survey on transfer learning*, IEEE Transactions on Knowledge and Data Engineering **22** (2010), no. 10, 1345–1359.

[QCD08] Ariadna Quattoni, Michael Collins, and Trevor Darrell, *Transfer learning for image classification with sparse prototype representations*, CVPR, 2008, pp. 1 – 8.

[QSCP10] Novi Quadrianto, Alexander J Smola, Tiberio Caetano, and S.V.N. Vishwanathanand James Petterson, *Multitask Learning without Label Correspondences*, NIPS, 2010.

[Qui86a] J. R. Quinlan, *Introduction of decision trees*, Machine Learning (1986), 81–105.

[Qui86b] J. Ross Quinlan, *Induction of decision trees*, Machine Learning **1** (1986), no. 1, 81–106.

[Qui93] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers, 1993.

[RNA08] Caruana R., Karampatziakis N., and Yessenalina A, *An empirical evaluation of supervised learning in high dimensions*, ICML 2008, 2008, pp. 96–103.

[SFC+11] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake, *Real-time human pose recognition in parts from a single depth image*, CVPR 2011, 2011, pp. 1297–1304.

[SGS09] Michael Stark, Michael Goesele, and Bernt Schiele, *A shape-based object class model for knowledge transfer*, ICCV, 2009, pp. 373–380.

[Sha08] Toby Sharp, *Implementing decision trees and forests on a gpu*, ECCV 2008, 2008, pp. 595–608.

[SKFD10] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, *Adapting visual category models to new domains*, ECCV (4), 2010, pp. 213–226.

[SM11] Hae Jong Seo and Peyman Milanfar, *Action recognition from one example*, TPAMI **33** (2011), no. 5, 867–882.

[STFW05] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky, *Learning hierarchical models of scenes, objects, and parts*, ICCV, 2005, pp. 1331–1338.

[TD05] Peter J Tan and David L Dowe, *Mml inference of oblique decision trees*, AI 2004: Advances in Artificial Intelligence, Springer, 2005, pp. 1082–1088.

[TMF07] Antonio Torralba, Kevin P. Murphy, and William T. Freeman, *Sharing visual features for multiclass and multiview object detection*, TPAMI **29** (2007), no. 5, 854–869.

[TOC10] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo, *Safety in numbers: Learning categories from few examples with multi model knowledge transfer*, 2010.

[TOC13] ———, *Learning categories from few examples with multi model knowledge transfer*, TPAMI (2013).

[VJ04] Paul A. Viola and Michael J. Jones, *Robust real-time face detection*, IJCV **57** (2004), no. 2, 137–154.

[Was04] Larry Wasserman, *All of statistics*, Springer, 2004.

[wLGC07] Jun won Lee and Christophe Giraud-Carrier, *Transfer learning in decision trees*, IJCNN, 2007.

[WRLD13] Jun Wan, Qiuqi Ruan, Wei Li, and Shuang Deng, *One-shot learning gesture recognition from rgb-d data using bag of features*, Journal of Machine Learning Research **14** (2013), 2549–2582.

[WZCG08] Qing Wang, Liang Zhang, Mingmin Chi, and Jiankui Guo, *Mt-forest: Ensemble decision trees based on multi-task learning*, ECAI 2008, 2008, pp. 122–126.

[YD10] Yi Yao and Gianfranco Doretto, *Boosting for transfer learning with multiple sources*, CVPR, 2010, pp. 1855 – 1862.