PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

CATEGORY-LEVEL VISUAL OBJECT RECOGNITION USING NOVEL MACHINE LEARNING TECHNIQUES

BILLY MARK PERALTA MÁRQUEZ

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Doctor in Engineering Sciences

Advisor:
ÁLVARO SOTO

Santiago de Chile, November 2013

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

CATEGORY-LEVEL VISUAL OBJECT RECOGNITION USING NOVEL MACHINE LEARNING TECHNIQUES

BILLY MARK PERALTA MÁRQUEZ

Members of the Committee:
ÁLVARO SOTO
MIGUEL TORRES
DOMINGO MERY
JUAN CARLOS NIEBLES
RENATO SALINAS
CRISTIÁN VIAL

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Doctor in Engineering Sciences

Santiago de Chile, November 2013

To my family and friends

ACKNOWLEDGEMENTS

This thesis would not have been finished without the support and encouragement of several special people. Therefore, I would like to take this opportunity to show my gratitude to those who have assisted me in countless ways.

I would first like to express my greatest thanks to my supervisor Alvaro Soto. There was many times when I was lost and each time Alvaro was there to steer me towards the right path. His predisposition to offer me so much of his time, confidence and intellect is the major reason this thesis was completed.

I also would like to thank my friends for their appreciated help and cooperation. I have had the good fortune to study with some wonderful and authentic people: Alberto, who offer me the warmth of his family. Sebastian, who give me moments of interesting talkings. Christian, who gift me the freshness of his imagination. Karim, who give me valuable advices. Pablo, who offer me its confidence. Thanks to Adolfo, Anali, Cristian, Domingo, Esteban, Hans, Irene, Javier, Pepe, Stefan and Tomas. I apologize if I forget some name.

I specially would like to thank my family to whom I owe a great deal. To my father Reynaldo thank you for showing me the way of patience. My own curiosity provably comes from his recursive inquiries about the nature. To my brothers Roberto, Lissy, Patricia, Diego, Reynaldo and Sandra, you are always in my heart. Finally, to my mother Ana thank you for showing me the way of fortitude. She is a great example of courage and love.

Finally, I would like to thanks to the institutions that funded my studies and travels: CONICYT, DCC, DIPEI and VRAID.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

LIST OF FIGURES viii

LIST OF TABLES x

ABSTRACT xii

RESUMEN xiii

1. INTRODUCTION 1

1.1. Background 1

1.2. General Goals 4

1.3. Hypothesis 4

1.4. Thesis Work and Main Contributions 5

1.5. Document Structure 7

References 8

2. ENHANCING K-MEANS USING CLASS LABELS AND AN APPLICATION TO CODEBOOK GENERATION FOR OBJECT RECOGNITION 10

2.1. Introduction 10

2.2. Background 14

2.2.1. K-Means and K-Medoids 14

2.2.2. Related work 16

2.3. Labeled K-Means 18

2.4. Experiments and Results 22

2.4.1. Experiments in general datasets 22

2.4.2. Experiments in object datasets 32

2.5. Conclusions 34

References 39

3. EMBEDDED LOCAL FEATURE SELECTION WITHIN MIXTURE OF EXPERTS 45
3.1. Introduction 45
3.2. Background 47
3.2.1. Feature selection 47
3.2.2. L_1 regularization 48
3.3. Related work 50
3.4. Proposed Approach 54
3.4.1. Mixture of Experts 55
3.4.2. Regularized Mixture of Experts (RMoE) 57
3.5. Experiments 61
3.5.1. Synthetic datasets 62
3.5.2. Real datasets 65
3.6. Conclusions 70
References 71
4. ADAPTIVE HIERARCHICAL CONTEXTS FOR OBJECT RECOGNITION WITH CONDITIONAL MIXTURE OF TREES 77
4.1. Introduction 77
4.2. Hierarchical context 79
4.2.1. Prior Model 80
4.2.2. Measurement Model 80
4.3. Adaptive Hierarchical Mixture of Trees (AH-MoT) 81
4.3.1. Conditional mixture of trees 82
4.3.2. Inference 86
4.4. Experiments 87
4.5. Conclusions 89
References 93

5. CONCLUSIONS AND FUTURE RESEARCH 96
5.1. Conclusions 96
5.2. Future Research Topics 97
References 99

LIST OF FIGURES

2.1 Toy example comparing traditional and supervised clustering. There are two class labels denoted by clear and dark circles. We see that while classical clustering finds spatial clusters, supervised clustering finds clusters that are uniform with respect to class labels. 11

2.2 The codebook generated by K-Means associates patches only considering their distance. On the other hand, the use of labels in codebook generated by LK-Means tends to generate more discriminative patches. In the case of K-Means scheme, we can see that the second codeword in the second image leads to a not relevant patch of the image (see more details in main text). 13

2.3 Comparison of sensibility of adjusted mutual information (AMI) respect to parameter α. Figure 2.3(a) shows that the best result for Diabetes dataset is obtained with α equal to 0.9, AMI=0.092, which is almost the double than the result with K-Means, AMI=0.050. On contrast, Figure 2.3(b) shows that in case of Glass dataset, the best result is obtained with α equal to 0.1, AMI=0.171, which is slightly greater than performance with K-Means, AMI=0.168. These results show that the discriminativity of clustering is dependant of data and parameter α. 36

2.4 Optional caption for list of figures 37

2.5 Top six more discriminative codewords for K-Means and LK-Means. Each codeword is represented by the four nearest patches to each one. In general, LK-Means appears to obtain visually more discriminative codewords than K-Means as we can see in the third codeword. Although a powerful clasiffier algorithm could be able to separate the classes, the quality of codeworks can help to the classifier. 38

3.1 Mixture of experts scheme. 56

3.2 Average accuracy on real datasets for all tested algorithms with respect to the number of dimensions of the datasets. 67

4.1 An example of the results of our method with respect to a state-of-the-art model (Choi et al.,2010). We show the top six most confident detections for each model using the same underlying single object detector (Felzenszwalb et al, 2008). In the case of 4.1(b), the model uses fixed contextual relations that do not detect the car object and produce a wrong detection of a mountain object. In contrast, our model 4.1(c) uses global scene information to adaptively select a suitable component of a mixture of trees that embeds particular contextual relations that provide a correct detection of the car object and do not detect a phantom mountain object. 78

4.2 Modification of contextual objects relationships: (a) The model by Choi et al. (b) Proposed model, AH-MoT, that incorporates global scene information as a root element that influences object-to-object relationships. 83

4.3 Some detections considering a single tree model (Choi et al., 2010) and AH-MoT. Our model, AH-MoT, usually provides better detections than a single tree model. 90

4.4 Examples of component trees for AH-MoT with six trees in the OUTDOOR dataset. Positive and negative correlations are indicated respectively with blue and red lines. 91

4.5 Dependence of the top 11 more frequent objects. Figure 4.5(a) shows the relationships for a single tree. Figures 4.5(b) and 4.5(c) show two relationships in AH-MoT with 6 trees. 92

LIST OF TABLES

2.1 Datasets details. 23

2.2 Adjusted Mutual Information results for real datasets using 10-CV. In average, LK-Means usually outperforms K-Means and SRIDHCR with variable confidence. 26

2.3 Adjusted Variation of Information results for real datasets using 10-CV. In average, LK-Means usually outperforms competitors with variable confidence. 27

2.4 Mirkin distance (MD) results for real datasets using 10-CV. In half of cases, LK-Means usually outperforms K-Means and SRIDHCR with variable confidence. 28

2.5 Adjusted Rand Index (ARI) results for real datasets using 10-CV. In average, LK-Means usually outperforms K-Means and SRIDHCR with variable confidence. 29

2.6 Speed in seconds for real datasets using 10-CV. LK-Means is considerably faster than SRIDHCR. Even though K-Means is faster than LK-Means, the difference is not very high and is compensated by an increase in clusters quality. 31

2.7 Details of real object datasets. 33

2.8 Accuracy for real object datasets using 20-HoldOut. In average, LK-Means overcomes K-Means in almost all datasets and all configurations. The advantage of LK-Means is more clear for larger number of codewords (K>150). 34

3.1 Synthetic datasets used for experiments. 62

3.2 Accuracy on synthetic datasets using 30 hold-out partitions. RMoE(λ^*_ν,λ^*_ω) indicates average (std. deviation) classification accuracy for best parameters configuration. The pair (λ'_ν,λ'_ω) show the median of the best parameters obtained by 3-fold cross-validation inside the training set of each hold-out partition (see main text for details). 63

3.3 Average parameter dimensionality for results shown in Table 3.2. 63

3.4 Relative relevance assigned by RMoE to features used to generate class patterns in synthetic datasets using the score in Equation (3.15). Each cell indicates the

position in the ranking where, in average, it is possible to find a given percentage of the relevant dimensions (header of the respective column). 65

3.5 Real datasets used for experiments. 66

3.6 Accuracy on real datasets using 30 hold-out partitions. RMoE(λ^*_ν,λ^*_ω) indicates average (std. deviation) classification accuracy for best parameters configuration. The pair (λ'_ν,λ'_ω) show the median of the best parameters obtained by 3-fold cross-validation inside the training set of each hold-out partition (see main text for details). 66

3.7 Average dimensionality of parameters in real datasets. 68

3.8 Average execution time (in miliseconds) of mixture-of-experts and regularized mixture-of-experts for different datasets using in each case 100 independent executions. 69

3.9 Average accuracy in subset of Pascal-2007 dataset. The methods compared with RMoE are standard classification algorithms: MoE, SVM, Random Forest and 1NN 69

4.1 APR for OUTDOOR and SUN09 databases provided by the tested methods. Relative improvement with respect to Choi et al. is shown in parenthesis. . . . 88

ABSTRACT

Automatic visual recognition of generic objects is a highly relevant area of study. Nonetheless, intra-class and pose variations, as well as, background clutter and partial occlusion, are some of the main difficulties to achieve this goal. As an application case, robots with a robust visual system can achieve a high level of autonomy and a semantic understanding of their environments. Current state-of-the-art approaches to visual category-level object recognition are generally based on two main steps: generation of visual descriptors and training of visual classifiers using these descriptors and labeled images. Furthermore, these steps are usually complemented using techniques oriented to include contextual information in the models. In this thesis, we contribute to the area of visual recognition by proposing three techniques oriented to improve each of the previous steps, respectively. First, we introduce a technique for building visual descriptors based on a Bag-of-Words (BoWs) representation. In contrast to current approaches based on unsupervised clustering techniques, our proposal combines unsupervised and supervised information leading to more discriminative BoWs representations. Afterwards, we present a technique to improve the performance of current visual classifiers using a divide-and-conquer strategy based on a Mixture of Expert (MoE) approach. We innovate with respect to current MoE techniques by incorporating an embedded local feature selection scheme within each visual classifier. Finally, we propose an approach that exploits contextual information to improve the performance of object recognition techniques. We innovate with respect to state-of-the-art techniques by considering scene dependent contextual relations among object classes. We test the performance of all these techniques by applying them to common benchmark datasets. Our results validate our main hypotheses indicating improvements with respect to alternative state-of-the-art methods. This also shows that the ideas presented in this thesis represent a relevant contribution to the state-of-the-art of category-level object recognition.

Keywords: Machine learning, object recognition, codebooks of histograms, context based object recognition, embedded feature selection.

RESUMEN

El reconocimiento automático visual de objetos genéricos es un área muy importante de estudio. Sin embargo, variaciones intra-clase y pose tanto como ruido de fondo de imagen y oclusiones parciales son algunas de las principales dificultades para lograr este objetivo. Como ejemplo de aplicación, robots con un sistema de visión confiable pueden obtener un mayor nivel de autonomía y comprensión semántica de sus entornos. Enfoques actuales del estado de arte para reconocimiento visual de categoría de objetos son generalmente basados en dos pasos principales: generación de descriptores visuales y entrenamiento de clasificadores visuales usando estos descriptores e imágenes etiquetadas. Además, estos pasos son usualmente complementados con técnicas orientadas a incluir informacion contextual en los modelos. En esta tesis, nosotros contribuimos al área de reconocimiento visual proponiendo tres técnicas orientadas a mejorar respectivamente cada uno de los pasos previos. Primero, introducimos una técnica para contruir descriptores visuales basados en representación de bolsa de palabras (BoWs). A diferencia de enfoques actuales basados en técnicas de clustering no supervisado, nuestra propuesta combina información no supervisada y supervisada conduciendo a representaciones de BoWs más discriminativas. Luego, presentamos una técnica para mejorar la performance de clasificadores visuales actuales usando un enfoque de divide-y-vencerás basado en el enfoque de Mixtura de Expertos (MoE). Nosotros innovamos con respecto al estado de arte de las actuales técnicas de MoE al incorporar un esquema de selección local embebida de características dentro cada clasificador visual. Finalmente, proponemos un enfoque que aprovechar la información contextual para mejorar la performance de las técnicas de reconocimiento de objetos. Nosotros innovamos en relacion a técnicas del estado de arte al considerar las relaciones contextuales entre clases de objetos como dependientes de la escena. Nosotros probamos la performance de todas estas técnicas al aplicarlas a bases de datos estándares de prueba. Nuestros resultados validan nuestras principales hipótesis mostrando mejoras en

relación a métodos alternativos del estado de arte. Esto también muestra que las ideas presentadas en esta tesis representan una contribución relevante en el área del reconocimiento de categoría de objetos.

Palabras Claves: Aprendizaje de máquina, reconocimiento de objetos, codebook de histogramas, selección embebida de variables , reconocimiento de objetos basado en contexto.

1. INTRODUCTION

1.1. Background

Today, technology is changing the way people produce and handle information. From business to science and engineering, there is a massive generation of huge databases storing valuable information. The opportunities of an automatic and effective mining of these new data sources are huge, however, finding relevant patterns in high dimensional data has become a difficult task (Mitchell, 1997). Furthermore, in many applications the relevancy of a pattern does not only depend on the observed data, but also depends on contextual information. As an example, in the area of computer vision is known that relevant visual features suitable to achieve a task such as human detection can drastically change if the target humans are walking in a crowded environment, swimming in a pool, or sitting behind a desk. Machine learning techniques have become the most attractive alternative to find these patterns and to improve these automatic systems.

In particular, we focus on applying machine learning techniques to the task of category-level object recognition using visual information. This is currently a major technological challenge and goal, as there is an urgent need to increase the level of semantic understanding that an automatic system can achieve from its environment. In particular, machine learning based approaches have been the most successful techniques to improve object recognition performance (Viola & Jones, 2004; Fei-Fei, 2005; Felzenszwalb, McAllester, & Ramanan, 2008). Nevertheless, there is still plenty of space for relevant contributions in this area.

In general terms, current state-of-the-art approaches to visual category-level object recognition are based on 2 main steps: i) Generation of visual descriptors and ii) Training of visual classifiers using suitable image descriptors and labeled images (Fergus, Perona, & Zisserman, 2003; Felzenszwalb et al., 2008). Also, these steps are usually complemented by technique oriented to include contextual information in the models, such as objects co-occurrence (Choi, Lim, Torralba, & Willsky, 2010). In this thesis, we contribute to the area

of visual recognition by presenting 3 techniques oriented to improve each of the previous steps, respectively. First, we present a method to improve current techniques to obtain visual descriptors using the so-called Bag of Visual Words (BoWs) approach. Then, we present a technique to improve the performance of current visual classifiers using a Mixture of Expert approach. Finally, we present a new adaptive scheme to include contextual information to improve the performance of object recognition techniques. Next, we present further details of each of these contributions.

In terms of generation of visual descriptors, a current popular approach is to build an image codebook using low level image cues, such as gray-level contrast information. The codebook is generally used in patch-based object recognition (Sivic & Zisserman, 2003; Fergus et al., 2003; Csurka, Bray, Dance, & Fan, 2004). Under this approach, an image is represented by a set of histograms of patches, where each histogram is associated to a particular spatial region of an image. This histogram representation is also known as a BoWs or "Bag-of-features" (Nowak, Jurie, & Triggs, 2006). A codebook is the complete set of codewords, where each codeword is a representative element of a set of similar patch descriptors. Then, this representation is used to train a visual classifier. This scheme has led to successful object recognition, showing robustness to partial occlusion and within-class shape variations (Sivic & Zisserman, 2003; Fergus et al., 2003; Csurka et al., 2004).

A usual option for generating visual codewords is to use clustering algorithms such as K-Means (Sivic & Zisserman, 2003; Csurka et al., 2004; Zhang, Marszalek, Lazebnik, & Schmid, 2007) or Gaussian mixture models (GMMs) (Dorkó & Schmid, n.d.; Larios et al., 2007; Perronnin, 2008). The advantages of using clustering algorithms for codebook generation are simplicity, low risk of overfitting, and computational efficiency. K-Means is popular in this context due to its simplicity and relative efficiency with respect to other options. Nonetheless, this algorithm ignores object labels which can help to find more suitable codebook to support object classification tasks. It is possible to find previous works that use object labels during the codebook generation (Winn, Criminisi, & Minka, 2005; Perronnin, 2008), however, labels are not directly combined with the clustering technique

or they are integrated with computationally demanding tools that do not offer the advantages of K-means. In this thesis, we present Labeled K-Means or LK-Means, a codeword generating approach that modifies the usual operation of the K-Means algorithm to include class labels during the clustering process.

In terms of visual classifiers, we believe that it is important to enhance current approaches with more flexible schemes that can adaptively select relevant visual cues. As an example, consider the case of recognizing a tennis ball and a regular pencil. While the tennis ball can be easily characterized in a visual subspace of round-shapes and yellowish-colors, the pencil has a more suitable characterization in a subspace given by long-straight-shapes as visual cues. Currently, most standard classification algorithms do not explicitly consider feature selection schemes. Instead, common approaches are based on off-line feature selection or dimensionality reduction techniques that globally select some visual dimensions.

In this thesis, we present a strategy for adaptive feature selection based on a "divide and conquer" approach, where a complex problem is divided into multiple simpler problems. In particular, we extend the Mixture of Expert technique (MoE) by adding to its regular operation an embedded feature selection scheme. MoE technique (Jacobs, Jordan, Nowlan, & Hinton, 1991) is a probabilistic version of the "divide-and-conquer" strategy. MoE divides the data into multiple regions where each region has its own classifier (Jacobs et al., 1991). Under this scheme, predictions of the experts are weighted using a global model known as gate function. Unfortunately, MoE does not make feature selection which can be very helpful to deal with complex visual recognition tasks. In this thesis, we present Regularized Mixture of Experts or RMoE, an extension of MoE that allows us to adaptively select features in high dimensional visual domains.

In terms of contextual information, current approaches for category-based object recognition only consider fixed contextual relations among objects. Under such scheme, the probability of seeing a person and a dog under an office scene is the same that under a public park scene. Clearly, this is not correct because the last case is more likely. In this

sense, contextual relations among objects do depend of the type of scene being analyzed. We hypothesize that inference about latent relationships between scene and inter-object co-occurrence can improve object recognition performance. Accordingly, in this thesis we present an adaptive hierarchical approach based on a Mixture of Trees or AH-MoT which is able to infer scene context to adaptively mix different possible contextual relations among objects.

1.2. General Goals

The general goal of this thesis is to contribute to the state-of-the-art of category-level object recognition techniques by developing algorithms oriented to enhance the different steps of the common object recognition pipeline. In particular, we propose: i) LK-Means: a discriminative scheme for the extraction of low level image descriptors that is able to include object labels during the codebook generation, ii) RMoE: a regularized version of MoE that provides an embedded feature selection scheme to adaptively select suitable visual features, iii) AH-MoT, a hierarchical approach based on a mixture of trees that is able to adaptively use contextual information to improve object recognition performance.

1.3. Hypothesis

The general hypothesis of this thesis is conformed by three hypotheses. First, LK-Means, a discriminative scheme for the extraction of low level image descriptors, is able to improve the quality of clusters of K-Means for object recognition tasks. Second, RMoE, a regularized version of MoE with embedded local feature selection, is able to improve the classification accuracy of MoE by selecting relevant dimensions. Third, AH-MoT, a hierarchical approach based on a mixture of trees, is able to improve object recognition performance given by a single tree.

1.4. Thesis Work and Main Contributions

The first part of this thesis describes LK-Means a novel supervised clustering method based on K-Means. The proposed method is used to generate a codebook for an object recognition application. An standard method of generation of codebooks is to apply K-Means algorithm to visual dataset because of its speed, however, this algorithm generates clusters only considering unsupervised information. The main idea is that the combination of unsupervised and supervised information can lead us to more informative clusters. On the other hand, the use of an algorithm similar to K-Means can allow us to overcome the speed and integration limitations of alternatives discriminative approaches for codeword generation. We believe that the combination of a class-dependent discriminative score with a generative clustering score can help us to find most informative clusters. The proposed method has been implemented and tested using standard object datasets. Results show that LK-Means is able to outperform several alternative solutions.

The second part of the thesis describes RMoE, a method based on an ensemble classifier with an embedded feature selection scheme. The contributed method is based on a mixture-of-experts model that incorporates a local feature selection using $L1$ regularization. An usual method of ensemble classifiers is the mixture-of-experts where multiple local classifiers are combines considerins weights given by a set of gate functions. Nonetheless, this algorithm consider all attributes in all local experts and gates functions. Our fundamental intuition consists of the belief that particular subsets of dimensions, or subspaces, are usually more appropriate to classify certain input instances. Our experiments show a notorious improving of accuracy in relation to traditional mixture-of-experts model and good feature selection capabilities for the experts classifiers.

Finally, the third part of the thesis describes AH-MoT, a new approach to use contextual information for boosting object recognition performance. A regular way to model the inter-object context information is using a tree-structured graphical model due to its easyness to learning and inference process, which can be considered a fixed context model. In particular, we use the flexibility of a mixture model to provide adaptive contextual relations

among objects. Our proposed method learns adaptive conditional relationships among objects according to the scene information. We accomplish this by introducing a probabilistic graphical model based on a conditional mixture of trees (Meila & Jordan, 2001). Each tree represents some local relationship between objects. Experiments are performed using two standard object datasets where we compare our method with an alternative state-of-the-art technique. Our results show the advantages of our adaptive contextual scheme.

Accordingly, the main contributions of this thesis are:

- To build a algorithm that jointly considers unsupervised and supervised information for codebook representation of images for object recognition applications. The main advantage of this method is to use all available information under an efficient clustering scheme. In our knowledge, LK-Means is the first technique where it is proposed a supervised clustering algorithm using a algorithmic mechanism similar to K-Means.
- To build a method based on mixture-of-experts where the parameters are learned simultaneously with the relevant features for each local expert and gate. The main advantage of this method is the join training of classifiers under an embedded feature selection scheme. In our knowledge, RMoE is the first work where it is proposed an embedded feature selection inside a Mixture-of-experts for classifier tasks.
- To build a method for context-based object recognition that considers latent relationships between scene and inter-object co-occurrence under a probabilistic framework. The main advantage of this method is the adaptability of co-occurrence relationships according to scene type. In our knowledge, AH-MoT is the first technique where it is proposed the adaptive use of multiple contexts.
- To implement and test each of the mentioned methods using real data, showing significant advantages with respect to alternative state-of-the-art approaches.

Finally, these contributions are evidenced in the next serie of papers:

- B.Peralta and A. Soto. *Mixing hierarchical contexts for object recognition*. Proceedings of the 16th Iberoamerican Congress conference on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 2011, pp 232-239.
- B.Peralta, P. Espinace, and A. Soto. *Adaptive hierarchical contexts for object recognition with conditional mixture of trees*. Proceedings of the British Machine Vision Conference, 2012, 121.1-121.11.
- B.Peralta, P. Espinace, and A. Soto. *Enhancing K-Means using class labels*. ISI Journal Intelligent Data Analysis, 2013.
- B.Peralta and A. Soto. *Embedded local feature selection within Mixture of Experts*. ISI Journal Information Sciences, 2014.

1.5. Document Structure

The rest of this thesis is organized as follows: Chapter 2 presents LK-Means. Chapter 3 presents RMoE. Chapter 4 presents AH-MoT. Chapter 5 presents the main conclusion of this thesis and future avenues of research. The chapters of this thesis are self-contained, thus, they can be read independently.

References

Choi, M., Lim, J., Torralba, A., & Willsky, A. (2010). Exploiting hierarchical context on a large database of object categories. In *Conference on Computer Vision and Pattern Recognition (cvpr)* (pp. 129–136).

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, 1–22.

Dorkó, G., & Schmid, C. (n.d.). *Object class recognition using discriminative local features* (Tech. Rep. No. RR-5497). INRIA - Rhone-Alpes.

Fei-Fei, L. (2005). A bayesian hierarchical model for learning natural scene categories. In *Conference on Computer Vision and Pattern Recognition* (pp. 524–531).

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Conference on Computer Vision and Pattern Recognition.*

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Conference on Computer Vision and Pattern Recognition* (pp. 264–271).

Jacobs, R., Jordan, M., Nowlan, S., & Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 79-87.

Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., ... Dietterich, T. (2007). Automated insect identification through concatenated histograms of local appearance features. In *Workshop on Applications of Computer Vision.*

Meila, M., & Jordan, M. (2001). Learning with mixtures of trees. *Journal of Machine Learning*, 1–48.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Nowak, E., Jurie, F., & Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision.*

Perronnin, F. (2008). Universal and adapted vocabularies for generic visual categorization. *Pattern Analysis and Machine Intelligence*, 1243–1256.

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision* (pp. 1470–1477).

Viola, P., & Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 137–154.

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *International conference on computer vision* (pp. 1800–1807).

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal on Computer Vision*, 213–238.

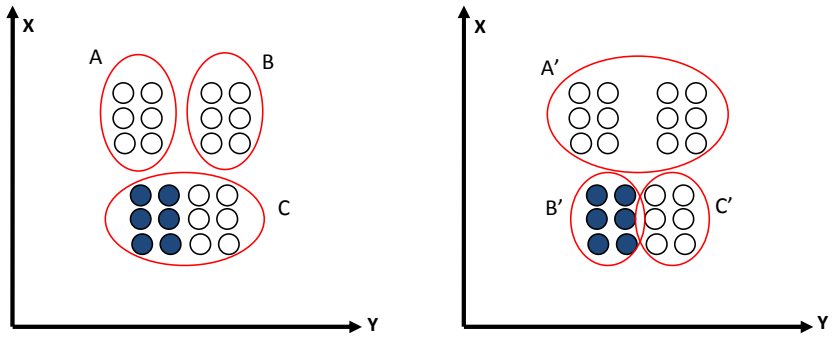
2. ENHANCING K-MEANS USING CLASS LABELS AND AN APPLICATION TO CODEBOOK GENERATION FOR OBJECT RECOGNITION

2.1. Introduction

Techniques to divide a set of N data instances into K groups are known as clustering algorithms. Clustering algorithms are commonly used in an unsupervised learning framework where the goal is to minimize an error function with respect to a given distance metric, for example, intra-cluster distance. A robust model for clustering is a mixture of Gaussians that has the ability to capture complex relationships among the data using a sound statistical approach (Bishop, 2007). Despite its advantages, this technique tends to be slow mainly due to the calculation of a covariance matrix. A faster and simpler clustering technique is K-Means algorithm that uses a hard assignment of data points to clusters and assumes a spherical covariance. While the simplicity of K-Means is one of the main reasons of its popularity (Wu et al., 2007), its execution speed with respect to alternative clustering techniques is also a desirable feature for intensive clustering tasks, e.g., the acquisition of codewords for visual recognition (Jurie & Triggs, 2005).

In contrast to traditional clustering, supervised clustering is applied to labeled data. Here, the goal is to find clusters with high purity, where purity of a cluster is defined as the percentage of data in a cluster that belongs to its most frequent class. Figure 2.1 shows an illustrative toy example corresponding to a 2D dataset with three spatial clusters and two classes. After applying both, unsupervised and supervised clustering, unsupervised clustering ignores class labels (Figure 2.1.a), while supervised clustering generates clusters that focus on a given class (Figure 2.1.b).

We believe that the ideal situation shown in Figure 2.1 is far from common. Natural clusters arising in real datasets are generally not totally homogeneous with respect to class labels, but they usually combines data instances from different classes. As a consequence, we believe that a strategy that combines a class-dependent discriminative score with a traditional generative clustering score can help to find most informative clusters. In



(a) Traditional clustering (b) Supervised clustering

FIGURE 2.1. Toy example comparing traditional and supervised clustering. There are two class labels denoted by clear and dark circles. We see that while classical clustering finds spatial clusters, supervised clustering finds clusters that are uniform with respect to class labels.

this respect, the benefits of combining supervised and unsupervised information has been explored before in (Lasserre, Bishop, & Minka, 2006).

Eick et al. (Eick, Zeidat, & Zhao, 2004) enumerate several applications of supervised clustering, such as dataset compression, distance metric learning, or classification refinement, among others. As an example, supervised clustering can be used to identify customers profiles according to ordinal measures (e.g. age, salary, marital status) by finding clusters that are homogeneous with respect to their buying behavior in terms of particular product categories (labels). Further uses of supervised clustering can be found in the genetics and financial domains (Sinnkkonen, Kaski, & Nikkila, 2002).

Algorithms for supervised clustering usually have the form of a K-Medoids algorithm. Due to the use of medoids, this method is more resistant to outliers than K-Means, but it has the drawback of being considerably slower. In particular, assuming a fixed number of iterations, K-Medoids has quadratic complexity in the number of data instances, while in the case of K-Means this complexity is only linear (Bishop, 2007).

In relation to object recognition context, these clustering algorithms are applied to generate codebooks usually in patch-based methods (Fergus, Perona, & Zisserman, 2003; Csurka, Bray, Dance, & Fan, 2004; Jurie & Triggs, 2005; Winn, Criminisi, & Minka, 2005;

Marszaek & Schmid, 2006). In this scheme, an image is represented by a set of histograms of patches, where each histogram is usually associated to a particular spatial zone of an image. This histogram representation is often known as a "Bag-of-features" (Nowak, Jurie, & Triggs, 2006). A codeword is a representative element of a set of similar patch descriptors. Then, this representation is used as input for a classification algorithm. This scheme has led to successful object recognition, showing robustness to partial occlusion and within-class shape variations.

Within this approach, the codeword generation step works as follows: a number of small patches are cropped either around interest points, by random sampling, or using a fixed grid over each image. Then, histograms are built and codewords are obtained by means of a clustering algorithm such as K-Means or agglomerative clustering, where the means of clusters are the desired codewords. Usually, the distribution of patches is different for different classes, for example, patches showing a wheel are more likely to exist in images of motorbikes than in images of airplanes. Hence, it is reasonable to think that classes of images can be determined by evaluating the distribution of patches extracted there. An example of scheme for using the proposed clustering for codebook generation in object recognition applications is presented in Figure 2.2.

A common option for generating visual codewords is the use of an unsupervised clustering algorithm such as K-Means (Sivic & Zisserman, 2003a; Csurka et al., 2004; J. Zhang, Marszalek, Lazebnik, & Schmid, 2007), Gaussian mixture models (GMM) (Dorkó & Schmid, 2005; Larios et al., 2007; Perronnin, 2008), Mean-Shift (Jurie & Triggs, 2005), or hierarchical clustering (Agarwal, Awan, & Roth, 2004). The advantages of using clustering algorihtms for codewords generation are simplicity, low risk of overfitting, and computational efficiency. Some works use label information for the codebook generation (Winn et al., 2005; Moosmann, Triggs, & Jurie, 2007; Perronnin, 2008; Yang, Jin, Sukthankar, & Jurie, 2008; W. Zhang, Surve, Fern, & Dietterich, 2009), however, in these works, labels are used independently from the generation of codewords or they are included in methods that are significantly more complex than K-Means.

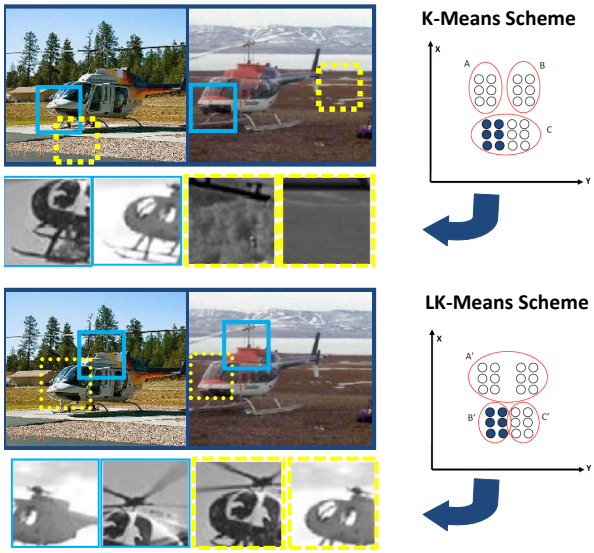


FIGURE 2.2. The codebook generated by K-Means associates patches only considering their distance. On the other hand, the use of labels in codebook generated by LK-Means tends to generate more discriminative patches. In the case of K-Means scheme, we can see that the second codeword in the second image leads to a not relevant patch of the image (see more details in main text).

In this work, we present a new method for supervised clustering that is based on two main hypotheses: i) A combination of supervised and unsupervised information can lead us to most informative clusters, and ii) The use of a K-Means type of algorithm can allow us to overcome the speed limitations of classical K-Medoids based supervised clustering methods. We show evidence for these hypotheses in Section 2.4. Following these ideas, the main contributions of this chapter are: i) Presenting LK-Means, a new supervised clustering algorithm that modifies K-Means cost function to incorporate supervised information, ii) Empirical evidence showing that our method outperforms K-Means and a K-Medoid supervised clustering method as measured by distinct metrics commonly used to access clustering quality, and iii) Empirical evidence showing that our method is more efficient in terms of execution time than a classical K-Medoid supervised clustering methods.

The rest of this chapter is organized as follows. Section 2.2 describes two baseline methods,K-Means and K-Medoids, and previous relevant previous works. Section 2.3

presents LK-Means, the proposed supervised clustering approach. Section 2.4 presents and discusses our experimental results. Finally, Section 2.5 presents our main conclusions and future avenues of research.

2.2. Background

2.2.1. K-Means and K-Medoids

K-Means algorithm is one of the most popular clustering techniques. This algorithm partitions N data instances into K clusters, where the number of clusters K has to be known a priori. Specifically, given a dataset X with N data instances $x_i \in \boldsymbol{R}^d$, $i \in [1 \dots N]$, K-Means algorithm partitions X into K cluster C_k, $k \in [1 \dots K]$, by minimizing the following cost function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \delta_{nk} \left\| x_n - u_k \right\|^2 , \tag{2.1}$$

where the indicator function δ_{nk} is given by :

$$\delta_{nk} = \begin{cases} 1 & x_n \in C_k \\ 0 & \text{otherwise,} \end{cases}$$

$\| \cdot \|$ refers to L2-norm and u_k corresponds to the mean of cluster k.

Optimal parameters u_k are found by minimizing Equation 2.1 using a gradient descent approach. This results in an iterative procedure that alternates between assigning data instances to clusters centers, and re-estimating cluster centers given the new assignations. Convergence to a local minimum of Equation 2.1 is granted by the gradient descent type of exploration and the finite set of possible assignations of data instances to clusters. In particular, assuming a fixed number of iterations and dimensions, the computational complexity of K-Means is $O(NK)$. Algorithm 1 summarizes the main steps of the K-Means algorithm.

Algorithm 1 : K-Means algorithm.

1: Randomly select K data instances as initial means.
2: Associate each data instance with the cluster of its nearest mean and calculate the cost function using Equation 2.1.
3: Calculate the new means as the centroids of the K new partitions.
4: Repeat steps 2 and 3 until there is no change in the cost evaluation (or the cost change is below a suitable threshold).

While K-Means for fixed number of iterations and dimensions has a linear computational complexity with respect to the number of data instances, the computation of the centroids is sensible to outliers (Bishop, 2007). To alleviate this problem, the K-Medoids algorithm uses a more robust procedure to find the cluster centers, but this procedure has a quadratic complexity with respect to the number of data instances. In particular, K-Medoids minimizes a score that is similar to the one used by K-means, but it uses a more general distance metric $\nu(x, x')$ between data instances x and x', as shown in Equation 2.2. An example of metric $\nu(x, x')$ is the Euclidean distance, used in K-Means, or the Jaccard distance, commonly used in applications related to transactional databases (Markov & Larose, 2007).

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \delta_{nk} \nu(x_n, u_k) \tag{2.2}$$

In contrast to K-means, K-Medoids minimizes Equation 2.2 with respect to parameters u_k by calculating a matrix that stores the distances between all pair of data instances. Specifically, initially K-Medoids randomly chooses a set of K data instances as the initial set of K medoids and calculates the distance matrix between all the data instances. Afterwards, it replaces each medoid with all non-medoid points and calculates all the possible configurations cost according to Equation 2.2. Next, it chooses as new medoids the ones corresponding to the configuration with lowest cost. Finally, the method repeats the search over the non-medoids elements until the medoids do not change. The procedure is summarized in Algorithm 2.

Algorithm 2 : K-Medoid algorithm.

1: Randomly select *k* data instances as initial medoids.
2: Associate each data instance to its most similar medoid and calculate the cost using Equation 2.2.
3: **for** each medoid *m* **do**
4: **for** each non-medoid *o* **do**
5: Swap *m* and *o* and compute the cost of the configuration.
6: **end for**
7: **end for**
8: Select the set of elements corresponding to the configuration with lowest cost.
9: Repeat 3 through 8 until there is no change in the set of medoids.

Assuming a fixed number of iterations and dimensions, the computational complexity of K-Medoids is $O(K(N-K)^2)$. This implies that in general K-Medoids is more expensive than K-Means.

2.2.2. Related work

Semi-supervised clustering uses labeled and unlabeled data to find clusters that maximize a score related to cluster purity with respect to known class labels. Semi-supervised clustering methods can be divided into two groups: Similarity based methods and search-based methods (Basu, Bilenko, & Mooney, 2003). Similarity-based methods use a modified distance function that considers the labels of classified examples and then uses a traditional clustering algorithm. On the other hand, search-based methods modify clustering algorithms themselves to accommodate for labeled instances, but do not change the distance function (Blum & Mitchell, 1998).

In terms of supervised clustering, all available records have labels. Tishby et al. propose an agglomerative clustering algorithm (Tishby. & Slonim, 2000) using the notion of "information bottleneck" (Tishby, Pereira, & Bialek, 1999). This technique minimizes the information loss of the clustering related to a class conditional distribution. Embrechts et al. (Demiriz, Benett, & Embrechts, 1999) propose a genetic algorithm for a version of K-Means where the goal of the search process is to obtain clusters that minimize cluster dispersion and cluster impurity. Cohn et al. (Cohn, Caruana, & McCallum, 2003) modify

the popular EM algorithm for incorporating similarity and dissimilarity constraints. They assume the presence of a human oracle that guides the clustering process. Basu et. al. (Basu et al., 2003) modify the K-means algorithm to cope with class knowledge. They use a careful initialization based on the neighborhood of the data instances.

Sinkkonen et al. (Sinnkkonen et al., 2002) propose a method called discriminative clustering that minimizes distortion within clusters. In their work, distortion is related to the loss of mutual information among classes and clusters, which is caused by representing each cluster by a prototype. This technique seeks to produce clusters that are internally as homogeneous as possible with respect to a class conditional distribution. The resulting minimization is complex and they have to resort to approximations or simulated annealing methods to find suitable solutions.

Jordan et al. (Xing, Ng., Jordan, & Russell, 2003) (and similarly Shental et al. (Bar-Hillel, Hertz, Shental, & Weinshall, 2003)) transform training examples into constraints based on the observation that instances of different classes should have a distance larger than a given threshold. Then, they derive a modified distance metric that minimizes the distance between data instances considering the constraints. Finally, they use a K-Means algorithm in conjunction with the modified distance metric to compute clusters.

Eick et. al. (Eick et al., 2004) formally introduce the term supervised clustering. Their work proposes supervised versions of some clustering algorithms, such as K-Medoids and divisive clustering. In particular, the SRIDHCR algorithm (Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Restart) shows good performance in their experiments when compared to alternative techniques, thus, we choose this method as the baseline for comparison in our work.

Ye et. al. (Ye, Z.Zhao, & Wu, 2008) present a discriminative version of K-Means. They simultaneously solve linear discriminant analysis (LDA) and K-Means optimization using matrix algebra. An advantage of this method is that it makes a feature transformation using LDA properties. For each iteration, their method needs to solve an optimization

problem using linear search. Unfortunately, they do not show any measure of the speed of their method.

In relation to extensions of K-Means, Deelers and Auwatanamongkol (Deelers & Auwatanamongkol, 2007) propose a scheme to initialize the K-Means algorithm using a recursive strategy that, considering the data axis with highest variance, progressively divides the data until they obtain a suitable number of clusters. Shanmugasundaram and Sukumaran (Shanmugasundaram & Sukumaran, 2010) introduce a related scheme to initialize the K-Means algorithm, where they divide the data into two smaller cells considering the data axis with highest variance and keeping the two cells as far apart as possible. This procedure is repeated until one can obtain a prefixed number of clusters. Kumar et al. (Kumar, Puran, & Dhar, 2011) enhance the K-Means algorithm by considering particular data structures (red-black tree and min-heap) that allow them to reduce computational time. These previous works are valuable in terms of improving the initialization and time processing capabilities of the traditional k-means algorithm, however, these works do not consider labeled data as in our technique. In this sense, these techniques can be considered as complementary to our work.

In a related research task, Lasserre et al. (Lasserre et al., 2006) propose the idea of a convex combination of unsupervised and supervised information in machine learning. They introduce a Bayesian framework to combine unlabeled and labeled data, where they find that under limited training data, the best performance is given by a combination of both views. Here, we also follow a similar idea but in the context of a supervised version of the K-Means algorithm leading to a different optimization problem and solution. As shown by our experiments, our proposed strategy provides several advantages with respect to alternative techniques for supervised clustering.

2.3. Labeled K-Means

Following Eick et. al. (Eick et al., 2004), several supervised clustering methods follow a K-Medoids approach that is very time consuming. Inspired by (Lasserre et al., 2006), we

propose LK-Means, a K-Means like algorithm with a modified cost function that considers a convex combination of both, a class-dependent and non-class-dependent cost functions.

We assume a labeled dataset X with N training instances (x_i, y_i), where $x_i \in R^d$, $y_i \in [1, \ldots, L]$, and $i \in [1 \ldots N]$. We assume that the clustering problem requires K clusters. LK-Means replaces the tradicional K-Means cost function in Equation (2.1) with the following function:

$$J(u_k^l, \delta_{nk}^l) = \sum_{n=1}^{N} \left[\alpha \sum_{k=1}^{K} \sum_{l=1}^{L} \delta_{nk}^l \left\| x_n - u_k^l \right\|^2 \rho_k^l + (1-\alpha) \sum_{k=1}^{K} \delta_{nk} \left\| x_n - u_k \right\|^2 \right] \quad (2.3)$$

where δ_{nk}^l refers to the supervised indicator function that assigns instance x_n to mean u_k^l, which in turn corresponds to the mean of data instances in cluster k with label l. ρ_k^l represents a prior factor for data instances with label l inside cluster k, δ_{nk} refers to the unsupervised indicator functions, and u_k corresponds to the mean of all data instances in cluster k. Equation (2.3) represents a convex combination, where parameter α in the range $[0, 1]$ manages the balance between the supervised and unsupervised clustering scores.

In particular, prior factor ρ_k^l for data instances with label l inside cluster k is defined as:

$$\rho_k^l = \frac{\sum_{n=1}^{N} \delta_{nk}^l}{\sum_{n=1}^{N} \delta_{nk}} \quad (2.4)$$

ρ_k^l represents the confidence of label l in cluster k, with values in the range $[0, 1]$. When this weight is near one, cluster k tends to contain only elements with label l. In the opposite case, when this weight is near zero, cluster k tends to contain no elements with label l.

The unsupervised indicator function δ_{nk} for data instance x_n and cluster C_K is defined as:

$$\delta_{nk} = \begin{cases} 1 & \text{if} \quad x_n \in C_k \\ 0 & \text{otherwise} \end{cases}$$

In terms of each unsupervised mean u_k, it is defined as the weighted mean over all supervised means u_k^l for the corresponding cluster k:

$$u_k = \sum_{l=1}^{L} \rho_k^l u_k^l \tag{2.5}$$

To find the optimal parameters: δ_{nk}^l and u_k^l, we minimize Equation (2.3) using a block coordinate descent approach that resembles the operation of the K-Means algorithm. Specifically, we alternate optimizations of Equation (2.3), first with respect to δ_{nk}^l and then with respect to u_k^l. Following the K-Means terminology, we call these steps **assignment** and **update**-steps, respectively. We refer now to each of these steps.

In terms of the **assignment-step**, cost function J in Equation (2.3) considers each data instance n in separate terms of the main sum, therefore, we can independently optimize J with respect to each indicator δ_{nk}^l. Furthermore, in the **assignment-step** we fix the value of the supervised means u_k^l and, as a consequence, we also fix the values of the unsupervised means u_k. As a result, the assignment that minimizes the cost function J is given by:

$$\delta_{nk}^l = \begin{cases} 1 & \text{if} \quad k =_j \left[\alpha \delta_{nj}^l \left\|x_n - u_j^l\right\|^2 \rho_j^l + (1-\alpha)\delta_{nj} \left\|x_n - u_j\right\|^2\right] \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

In terms of initialization, initial values for the supervised indicator functions δ_{nk}^l are calculated using a Laplace smoothing. We use this procedure to avoid empty values for the supervised means vectors which can appear in the case of clusters without elements of the corresponding class. In this case, the supervised mean of a missing class is given by the unsupervised cluster because all elements have a value near to zero.

Specifically, we apply a Laplace smoothing according to:

$$\delta_{nk}^l = \frac{\lambda_{nk}^l + \gamma}{1 + LK\gamma}, \tag{2.7}$$

where λ_{nk}^{l} is defined as:

$$\lambda_{nk}^{l} = \begin{cases} 1 & \text{if} \quad x_n \in C_k \ \wedge \ y_n = l \\ 0 & \text{otherwise.} \end{cases}$$

We apply a Laplace smoothing (Nigam, McCallum, Thrun, & Mitchell, 1999) with a constant $\gamma = 0.001$. This small constant can be interpreted as the global uncertainty about the label of an element.

In terms of the **update-step**, we only need to find the optimal supervised means since each unsupervised mean u_k is a function of the corresponding supervised means u_k^l. Applying the corresponding partial derivatives to Equation (2.3), we have:

$$\frac{\partial}{\partial u_k^l} J \quad = \quad \sum_{n=1}^{N} -2\alpha\delta_{nk}^{l}\left(x_n - u_k^l\right)\rho_k^l + \sum_{n=1}^{N} -2(1-\alpha)\delta_{nk}\left(x_n - u_k\right)\rho_k^l \qquad (2.8)$$

By rearranging components in Equation (2.8) and setting the derivative to zero, we obtain:

$$\sum_{n=1}^{N} \alpha\delta_{nk}^{l} x_n - \sum_{n=1}^{N} \alpha\delta_{nk}^{l} u_k^l + \sum_{n=1}^{N} (1-\alpha)\delta_{nk} x_n - \sum_{n=1}^{N} (1-\alpha)\delta_{nk} u_k \quad = \quad 0 \qquad (2.9)$$

Assuming iteration t and that we are computing the optimization for the supervised mean of a given class label l', we use the previous supervised means $u_k^{l(t-1)}$ to approximate $u_k^{(t)}$ by updating only the maximized component $u_k^{l'}$ and fixing the rest. Then, $u_k^{(t)} = \sum_{l=1, l\neq l'}^{L} \rho_k^{l(t-1)} u_k^{l(t-1)} + \rho_k^{l'(t-1)} u_k^{l'(t)} = u_k^{(t-1)} - \rho_k^{l'(t-1)} u_k^{l'(t-1)} + \rho_k^{l'(t-1)} u_k^{l'(t)}$. Renaming the variables associated to previous iterations $u_k^{(t-1)}$, $u_k^{l(t-1)}$ and $\rho_k^{l(t-1)}$ as $\tilde{u}_k$, $\tilde{u}_k^l$ and $\tilde{\rho}_k^l$, respectively, and considering that the optimization is computed for l=l', we have:

$$\alpha \sum_{n=1}^{N} (\delta_{nk}^{l} x_n - \delta_{nk}^{l} u_k^l) + (1-\alpha) \sum_{n=1}^{N} (\delta_{nk} x_n - \delta_{nk}(\tilde{u}_k - \tilde{\rho}_k^l \tilde{u}_k^l + \tilde{\rho}_k^l u_k^l)) = 0 \qquad (2.10)$$

Then, by rearranging the components, we have:

$$u_k^l = \frac{\alpha \sum_{n=1}^{N} \delta_{nk}^l x_n + (1-\alpha) \sum_{n=1}^{N} \delta_{nk} \left(x_n - \tilde{u}_k + \tilde{\rho}_k^l \tilde{u}_k^l \right)}{\alpha \sum_{n=1}^{N} \delta_{nk}^l + (1-\alpha) \tilde{\rho}_k^l \sum_{n=1}^{N} \delta_{nk}} \tag{2.11}$$

Equation (2.11) has a straightforward interpretation. If we have $\alpha = 1$, then only supervised information is considered. On the other hand, if $\alpha = 0$ then the resulting clusters correspond to the unsupervised solution provided by the traditional K-Means algorithm. We summarize our procedure in Algorithm 3.

Algorithm 3 Labeled K-Means Algorithm

1: Initialize *K* initial means randomly.
2: Associate each data instance with its nearest mean and consider its class.
3: Compute supervised means u_k^l using Equation (2.11).
4: Compute unsupervised means u_k using Equation (2.5).
5: Compute indicatrices δ_{nk}^l considering Equation (2.6).
6: Compute the cost J using Equation (2.3).
7: Repeat 3 to 6 until there is no change in the cost evaluation (or cost change is below a threshold).

In terms of convergence, the assignment-step given by Equation (2.6) can only decrease the value of the relevant cost function in Equation (2.3). Similarly, the update step provides new parameter values that also decrease this cost function. Furthermore, given that set of possible assignments of training instances to clusters is finite, the procedure in Algorithm 3 can not decrease forever. As a consequence, it is possible to guarantee that the proposed algorithm will converge to a local or global optimum of the relevant cost function.

In our model, we do not consider specific strategies to deal with noisy or missing data. However, standard preprocessing strategies do exist to deal with these problems, and they can be used to complement our technique (Jiawei, 2005).

2.4. Experiments and Results

2.4.1. Experiments in general datasets

In this Section, we test the performance of LK-Means using diverse datasets. In particular, we use 8 real data sets from the UCI Machine Learning Repository (Asuncion &

Newman, 2007): Iris, Heart, Glass, Diabetes, Silhouttes, Segment, Ionosphere, and Sonar. Table 2.1 shows the main details for these datasets. We normalize all these datasets to the range $[0, 1]$. Following the regular implementation of K-means, we use Euclidean distance as the similarity metric. All experiments are performed on a PC with 2.0 Ghz Pentium IV processor with 2GB of RAM memory.

TABLE 2.1. Datasets details.

Dataset name	**# objects**	**# dimensions**	**# classes**
Iris	150	4	3
Heart	270	13	2
Glass	214	9	6
Diabetes	768	8	2
Silhouttes	846	18	4
Segment	2310	19	7
Ionosphere	351	33	2
Sonar	208	60	4

We compare our algorithm against classical K-Means and SRIDHCR. SRIDHCR is a K-Medoids algorithm based on a discriminative metric with random re-initialization if it detects a local minimum. We choose SRIDHCR because it shows good performance in relation to other supervised clustering methods (Eick et al., 2004). We compare these algorithms in terms of clustering quality and computational time. In particular, Meilă (Meilă., 2005) shows that there is not a single best metric to compare the outputs of clustering algorithms. There are alternative metrics for evaluating clustering quality, such as F-measure (Manning, Raghavan, & Schütze, 2008), Jaccard index (Rajaraman & Ullman, 2012), or Fowlkes Mallows index (Fowlkes & Mallows, 1983), however, we follow the metrics suggested in (Meilă., 2005). Consequently, we assess clustering quality using 4 different metrics commonly used to validate clustering results (Meilă., 2005): Adjusted Mutual Information (AMI) (Vinh, Epps, & Bailey, 2009), Adjusted Variation of Information (AVI) (Vinh et al., 2009), Mirkin distance (MD) (Mirkin., 1996), and Adjusted Rand Index (ARI) (Hubert & Arabie, 1985). AMI and AVI are variations of mutual information (MI), while MD and ARI are variations of Rand Index (RI). All these metrics do not make any assumption about

the form of the clusters. Also, they are in the range $[0, 1]$, where higher values indicate a better clustering, except in the case of MD where small values indicate better results.

In our experiments, we use cross-validation with ten folds (10-CV) to validate the results of each algorithm. In terms of selecting a suitable number of clusters K, it is possible to use previous strategies proposed in the context of the K-Means algorithm (Mardia, Kent, & Bibby, 1979). Also, it is possible to relate the selection of K to the number of known classes. Here, we do not focus in proposing new strategies to choose this value, and we run our experiments testing different numbers of clusters. For each dataset, we select values for K equally spaced according to 4 intervals beginning from the number of classes L to the upper bound $\left\lceil \sqrt{number\ of\ records/2} \right\rceil$. This upper bound is obtained from the t't'rule of thumb" of clustering (Mardia et al., 1979). For example, in the case of Heart dataset, as it has 2 classes and 270 instances, we test $K \in \{2, 5, 8, 11, 14\}$.

For each of the 4 clustering metrics considered here, we test the performance of LK-Means using parameter α with values $\{0.8, 0.9, 1.0\}$, and for each of the tests considered here, we report the average performance for these 3 values. In the case of SRIDHCR, we choose the best parameter β (see (Eick et al., 2004) for details) according to 3-CV in a grid with 11 values: 0 to 2.0 with a step of 0.2. K-Means does not require more parameters than the number of clusters. It is important to note that to be fair with K-Means, we do not optimize the value of parameter α in LK-Means. This is because when α approaches zero LK-Means behaves exactly like K-Means, therefore, by optimizing α, LK-Means can always at least match the performance of K-Means. Consequently, in all tests, we just consider high values of α to stress the relevance of the supervised information. To check if our results are statistically significant, in each case we use a paired Student's t-test (Behrens-Fisher problem (Rice, 1994)) to compare the results of LK-Means against the performance of each of the alternative techniques.

Table 2.2 shows our results using AMI metric. Considering the average AMI results for all values of K under test, our method outperforms K-Means and SRIDHCR in most of the cases, with the exceptions of the Silhouttes and Ionosphere datasets where K-Means shows

better performance. Considering only cases with confidence $\geq$ 75%, a paired t-Student test shows that for Iris, Heart, Glass, Segment, and Sonar datasets, LK-Means has better AMI than the nearest competitor with 84%, 100%, 88%, 91%, and 75% of confidence, respectively. On the other hand, K-Means shows the best performance in Ionosphere and Silhouttes datasets with 95% and 84% confidence, respectively.

Table 2.3 shows our results using AVI metric. By considering all datasets, we can observe that, on average, again LK-Means outperforms the other algorithms in most of the cases. Similarly to the results with AMI metric, the worst relative results for LK-Means is given for the case of the Ionosphere dataset. Considering only cases with confidence $\geq$ 75%, a paired t-Student test finds that LK-Means in Iris, Heat, Glass, Segment, and Sonar datasets has greater AVI than the nearest competitor with 87%, 100%, 94%, 75% and 80% of confidence, respectively. On the other hand, K-Means shows the best performance in Ionosphere and Silhouttes datasets with 96% and 95% confidence, respectively.

Table 2.4 shows results using MD metric. In terms of average results, in half of the 8 datasets LK-Means is the winner, while K-Means shows best performance in the rest of the datasets. In general, we notice that under MD metric there is not a clear winner between LK-Means and K-Means, and results depend on the type of dataset. Considering only cases with confidence $\geq$ 75%, a paired t-Student test shows that LK-Means in Heart, Diabetes, and Sonar datasets has lower MD than the nearest competitor with 100%, 92% and 87% of confidence, respectively. On the other hand, K-Means shows the best performance in Glass, Silhouttes, and Ionosphere datasets with 97%, 98%, and 95% of confidence, respectively.

Table 2.5 summarizes results using ARI metric. In average LK-Means is the winner in 5 of the 8 datasets. while K-Means shows best performance in 2 datasets, and SRIDCHR in one. Considering only cases with confidence $\geq$ 75%, a paired t-Student test finds that LK-Means in Iris, Heart, Glass, Segment, and Sonar datasets has better ARI than the nearest competitor with 75%, 100%, 95%, 98% and 93% of confidence, respectively. On the other hand, K-Means shows the best performance in Ionosphere dataset with 99% confidence.

TABLE 2.2. Adjusted Mutual Information results for real datasets using 10-CV. In average, LK-Means usually outperforms K-Means and SRIDHCR with variable confidence.

Method	**Number of clusters**					
Iris	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***Mean***
K-Means	**0.765**	0.515	0.359	0.385	0.242	0.453
SRIDHCR	0.196	0.260	0.204	0.236	0.241	0.227
LK-Means	0.655	**0.538**	**0.497**	**0.451**	**0.387**	**0.505**
Heart	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.273	0.145	0.087	0.079	0.045	0.126
SRIDHCR	0.011	0.082	0.078	0.104	0.097	0.074
LK-Means	**0.293**	**0.212**	**0.137**	**0.134**	**0.104**	**0.176**
Glass	***k=6***	***k=7***	***k=8***	***k=9***	***k=10***	***Mean***
K-Means	**0.168**	**0.166**	0.145	**0.136**	0.126	0.148
SRIDHCR	0.093	0.132	0.138	0.092	0.106	0.112
LK-Means	0.148	0.159	**0.156**	0.132	**0.149**	**0.156**
Diabetes	***k=2***	***k=7***	***k=12***	***k=17***	***k=22***	***Mean***
K-Means	0.050	0.059	**0.050**	**0.046**	**0.045**	0.049
SRIDHCR	**0.113**	0.049	0.044	0.043	0.041	0.058
LK-Means	0.086	**0.068**	0.047	0.040	0.043	**0.060**
Silhouttes	***k=4***	***k=8***	***k=12***	***k=16***	***k=20***	***Mean***
K-Means	**0.120**	0.125	**0.137**	0.124	0.114	**0.124**
SRIDHCR	0.076	0.107	0.132	**0.141**	0.116	0.114
LK-Means	0.112	**0.129**	0.128	0.118	**0.117**	0.120
Segment	***k=7***	***k=14***	***k=21***	***k=28***	***k=35***	***Mean***
K-Means	**0.578**	0.505	0.465	0.411	0.374	0.467
SRIDHCR	0.446	0.522	0.469	0.428	0.392	0.451
LK-Means	0.548	**0.551**	**0.492**	**0.439**	**0.411**	**0.488**
Ionosphere	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.169	**0.181**	**0.140**	**0.179**	**0.124**	**0.159**
SRIDHCR	0.053	0.112	0.069	0.082	0.075	0.078
LK-Means	**0.174**	0.177	0.125	0.156	0.108	0.148
Sonar	***k=2***	***k=4***	***k=6***	***k=8***	***k=10***	***Mean***
K-Means	0.001	0.022	**0.051**	0.032	0.019	0.025
SRIDHCR	0.012	0.001	0.050	0.019	0.020	0.020
LK-Means	**0.094**	**0.036**	0.017	**0.039**	**0.058**	**0.049**

Considering the different metrics and datasets used to evaluate clustering quality, the previous results indicate that in general LK-Means outperforms the alternative techniques under consideration. However, the superior performance of LK-Means depends on the type

TABLE 2.3. Adjusted Variation of Information results for real datasets using 10-CV. In average, LK-Means usually outperforms competitors with variable confidence.

Method	**Number of clusters**					
Iris	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***Mean***
K-Means	**0.781**	0.612	0.454	0.511	0.352	0.542
SRIDHCR	0.224	0.286	0.221	0.261	0.280	0.254
LK-Means	0.715	**0.613**	**0.591**	**0.560**	**0.485**	**0.592**
Heart	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.275	0.207	0.131	0.126	0.075	0.163
SRIDHCR	0.011	0.112	0.115	0.163	0.159	0.112
LK-Means	**0.300**	**0.271**	**0.188**	**0.201**	**0.164**	**0.225**
Glass	***k=6***	***k=7***	***k=8***	***k=9***	***k=10***	***Mean***
K-Means	0.193	**0.188**	0.168	**0.155**	0.150	0.170
SRIDHCR	0.100	0.149	0.150	0.109	0.128	0.127
LK-Means	**0.216**	0.183	**0.187**	0.154	**0.179**	**0.184**
Diabetes	***k=2***	***k=7***	***k=12***	***k=17***	***k=22***	***Mean***
K-Means	0.045	0.087	**0.080**	**0.077**	**0.078**	0.073
SRIDHCR	**0.117**	0.071	0.068	0.068	0.068	0.078
LK-Means	0.105	**0.092**	0.069	0.065	0.069	**0.080**
Silhouttes	***k=4***	***k=8***	***k=12***	***k=16***	***k=20***	***Mean***
K-Means	**0.122**	**0.151**	**0.182**	**0.174**	**0.166**	**0.159**
SRIDHCR	0.076	0.107	0.132	0.141	0.116	0.114
LK-Means	0.121	0.149	0.164	0.157	0.163	0.151
Segment	***k=7***	***k=14***	***k=21***	***k=28***	***k=35***	***Mean***
K-Means	**0.601**	0.575	0.578	0.538	0.511	0.561
SRIDHCR	0.561	0.583	0.568	0.544	0.513	0.554
LK-Means	0.570	**0.615**	**0.580**	**0.549**	**0.531**	**0.569**
Ionosphere	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.173	**0.238**	**0.201**	**0.258**	**0.195**	**0.213**
SRIDHCR	0.059	0.132	0.084	0.106	0.106	0.097
LK-Means	**0.182**	0.225	0.173	0.212	0.160	0.190
Sonar	***k=2***	***k=4***	***k=6***	***k=8***	***k=10***	***Mean***
K-Means	0.001	0.029	**0.076**	0.050	0.031	0.037
SRIDHCR	0.012	0.001	0.065	0.028	0.033	0.028
LK-Means	**0.099**	**0.048**	0.026	**0.056**	**0.075**	**0.061**

of dataset and the validation metric under consideration. In terms of the different datasets, in general LK-Means shows superior performance in most of them with the exception of Silhouttes and Ionosphere, where the unsupervised clustering strategy of K-Means leads to

TABLE 2.4. Mirkin distance (MD) results for real datasets using 10-CV. In half of cases, LK-Means usually outperforms K-Means and SRIDHCR with variable confidence.

Method	**Number of clusters**					
Iris	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***Mean***
K-Means	**0.098**	**0.126**	0.171	0.148	0.179	**0.144**
SRIDHCR	0.393	0.349	0.352	0.333	0.328	0.351
LK-Means	0.158	0.158	**0.148**	**0.147**	**0.158**	0.154
Heart	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.354	0.406	0.433	0.428	0.455	0.415
SRIDHCR	0.491	0.450	0.451	0.445	0.450	0.457
LK-Means	**0.347**	**0.361**	**0.402**	**0.403**	**0.419**	**0.386**
Glass	***k=6***	***k=7***	***k=8***	***k=9***	***k=10***	***Mean***
K-Means	**0.287**	0.303	**0.266**	**0.269**	0.268	**0.279**
SRIDHCR	0.325	**0.302**	0.309	0.298	0.286	0.304
LK-Means	0.324	0.314	0.317	0.291	**0.267**	0.302
Diabetes	***k=2***	***k=7***	***k=12***	***k=17***	***k=22***	***Mean***
K-Means	0.448	0.473	0.494	0.505	0.513	0.487
SRIDHCR	**0.406**	0.493	0.493	0.506	0.512	0.482
LK-Means	0.419	**0.471**	**0.492**	**0.504**	**0.503**	**0.478**
Silhouttes	***k=4***	***k=8***	***k=12***	***k=16***	***k=20***	***Mean***
K-Means	**0.341**	**0.274**	**0.250**	**0.242**	**0.237**	**0.269**
SRIDHCR	0.391	0.319	0.281	0.264	0.262	0.303
LK-Means	0.381	0.287	0.261	0.258	0.246	0.287
Segment	***k=7***	***k=14***	***k=21***	***k=28***	***k=35***	***Mean***
K-Means	**0.142**	0.115	**0.107**	0.110	0.114	0.118
SRIDHCR	0.149	0.111	0.112	0.120	0.124	0.123
LK-Means	0.154	**0.104**	**0.107**	**0.109**	**0.111**	**0.117**
Ionosphere	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.401	**0.393**	**0.432**	**0.408**	**0.438**	**0.414**
SRIDHCR	0.429	0.420	0.450	0.445	0.474	0.444
LK-Means	**0.398**	0.403	0.440	0.411	0.444	0.419
Sonar	***k=2***	***k=4***	***k=6***	***k=8***	***k=10***	***Mean***
K-Means	0.523	0.469	**0.447**	0.449	0.457	0.469
SRIDHCR	0.479	0.506	0.474	0.503	0.480	0.488
LK-Means	**0.455**	**0.460**	0.458	**0.444**	**0.440**	**0.451**

better clusters. We believe that, in general, the performance of LK-Means is closely related to the pertinence of our hypothesis that homogeneity in class information leads to more informative clusters. Clearly, the validity of this hypothesis depends of the application

TABLE 2.5. Adjusted Rand Index (ARI) results for real datasets using 10-CV. In average, LK-Means usually outperforms K-Means and SRIDHCR with variable confidence.

Method	**Number of clusters**					
Iris	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***Mean***
K-Means	**0.754**	**0.592**	0.424	0.477	0.322	0.513
SRIDHCR	0.190	0.259	0.196	0.221	0.233	0.220
LK-Means	0.644	0.568	**0.552**	**0.527**	**0.457**	**0.550**
Heart	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	0.293	0.155	0.093	0.097	0.038	0.135
SRIDHCR	0.019	0.110	0.099	0.111	0.099	0.088
LK-Means	**0.315**	**0.257**	**0.164**	**0.155**	**0.119**	**0.202**
Glass	***k=6***	***k=7***	***k=8***	***k=9***	***k=10***	***Mean***
K-Means	0.151	0.124	0.131	**0.124**	0.106	0.127
SRIDHCR	0.074	0.092	0.104	0.079	0.091	0.088
LK-Means	**0.168**	**0.134**	**0.143**	0.119	**0.137**	**0.140**
Diabetes	***k=2***	***k=7***	***k=12***	***k=17***	***k=22***	***Mean***
K-Means	0.094	**0.093**	0.062	0.045	0.033	0.654
SRIDHCR	**0.182**	0.059	**0.068**	**0.048**	0.041	**0.796**
LK-Means	0.150	0.089	0.060	0.043	**0.045**	0.774
Silhouttes	***k=4***	***k=8***	***k=12***	***k=16***	***k=20***	***Mean***
K-Means	**0.084**	0.098	**0.109**	**0.103**	**0.101**	**0.099**
SRIDHCR	0.051	0.082	**0.109**	0.110	0.088	0.088
LK-Means	0.082	**0.103**	0.108	0.098	0.100	0.098
Segment	***k=7***	***k=14***	***k=21***	***k=28***	***k=35***	***Mean***
K-Means	**0.460**	0.426	0.396	0.346	0.294	0.384
SRIDHCR	0.446	0.483	0.410	0.326	0.278	0.389
LK-Means	0.447	**0.502**	**0.444**	**0.388**	**0.357**	**0.428**
Ionosphere	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***	***Mean***
K-Means	**0.198**	**0.219**	**0.147**	**0.197**	**0.139**	**0.180**
SRIDHCR	0.115	0.163	0.112	0.120	0.080	0.118
LK-Means	0.196	0.199	0.130	0.189	0.124	0.168
Sonar	***k=2***	***k=4***	***k=6***	***k=8***	***k=10***	***Mean***
K-Means	0.001	0.016	**0.050**	0.032	0.016	0.023
SRIDHCR	0.042	0.001	0.048	0.018	0.025	0.027
LK-Means	**0.103**	**0.044**	0.034	**0.052**	**0.059**	**0.058**

under consideration, particularly, the semantic of the data labels under consideration. In terms of the 4 metrics used to evaluate clustering quality, LK-Means outperforms clearly the alternative algorithms in the case of AMI and AVI metrics, and to a lesser degree in the

case of ARI metric. In the case of MD metric, for the values of α under consideration, LK-Means is unable to improve the results of K-Means. Following the observations in (Vinh, Epps, & Bailey, 2010), MD metric and, to a lesser extend, ARI metric are affected by cluster size, therefore, they have a bias that affect their performance. As recommended in (Vinh et al., 2010), AMI and AVI produce more stable and suitable results. Coincidentally, in our case AMI and AVI produce similar results and they provide stronger support to the superiority of LK-Means with respect to the alternative techniques.

Additionally, we test the sensibility of performance respect to α by measuring adjusted mutual information (AMI). We consider values of α in the interval: 0.1 to 1.0 with a step of 0.1. In order to facilitate the analysis of results, we consider two representative datasets. Specifically, we choose Diabetes and Glass datasets because in our experiments they represent cases where, under the AMI metric, LK-Means and K-Means alternate the best performance for different values of K. In both cases, we use a fixed number of clusters. We choose the number of clusters using the classical silhouette method (Rousseeuw, 1987), where the cardinality of the set of clusters is selected to maximize the average silhouette of the clusters.

Figure 2.3(a) shows the relationship between α and AMI for Diabetes dataset. The best result for LK-Means is obtained when α=0.9 with a corresponding value of $AMI = 0.092$. The worst result is for alpha$= 0.1$ with a corresponding value of $AMI = 0.051$. For this dataset, K-Means obtains a value of $AMI = 0.050$, therefore, there is a big advantage in favor of LK-Means. On the other hand, Figure 2.3(b) shows the relationship between α and AMI for dataset Glass. In this case, the best results are obtained with low values of α (0.1 and 0.2). In particular, the best result for LK-Means is obtained when α=0.1 with a corresponding values of $AMI = 0.171$. For this dataset, K-Means obtains a value of $AMI = 0.168$. Consequently, both algorithms show a similar behavior. This is expected because, according to Equation (2.3), for values of α near zero LK-Means behaves like K-Means.

TABLE 2.6. Speed in seconds for real datasets using 10-CV. LK-Means is considerably faster than SRIDHCR. Even though K-Means is faster than LK-Means, the difference is not very high and is compensated by an increase in clusters quality.

Method	**Number of clusters**				
Iris	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***
K-Means	0.004	0.009	0.009	0.010	0.011
SRIDHCR	35.502	54.315	72.967	90.713	108.689
LK-Means	0.047	0.095	0.074	0.068	0.087
Heart	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***
K-Means	0.008	0.012	0.016	0.017	0.020
SRIDHCR	45.232	95.696	146.066	195.522	247.243
LK-Means	0.044	0.096	0.141	0.190	0.169
Glass	***k=6***	***k=7***	***k=8***	***k=9***	***k=10***
K-Means	0.010	0.018	0.014	0.015	0.030
SRIDHCR	88.823	102.429	115.432	128.478	154.635
LK-Means	0.400	0.356	0.516	0.565	0.548
Diabetes	***k=2***	***k=7***	***k=12***	***k=17***	***k=22***
K-Means	0.023	0.101	0.125	0.110	0.113
SRIDHCR	124.984	356.097	585.961	815.086	1063.223
LK-Means	0.230	1.556	2.686	3.997	4.339
Silhouttes	***k=4***	***k=8***	***k=12***	***k=16***	***k=20***
K-Means	0.123	0.098	0.093	0.106	0.137
SRIDHCR	251.755	464.094	715.121	887.593	1311.306
LK-Means	1.492	3.398	4.775	5.937	9.384
Segment	***k=7***	***k=14***	***k=21***	***k=28***	***k=35***
K-Means	0.245	0.307	0.394	0.494	0.622
SRIDHCR	1105.172	2108.405	3173.733	4086.154	4972.139
LK-Means	13.272	11.170	37.325	40.543	7.154
Ionosphere	***k=2***	***k=5***	***k=8***	***k=11***	***k=14***
K-Means	0.009	0.030	0.027	0.029	0.031
SRIDHCR	61.963	139.254	212.244	263.724	329.487
LK-Means	0.118	0.337	0.441	0.605	0.923
Sonar	***k=2***	***k=4***	***k=6***	***k=8***	***k=10***
K-Means	0.008	0.013	0.014	0.016	0.017
SRIDHCR	39.390	68.486	97.831	127.679	165.534
LK-Means	0.058	0.106	0.118	0.279	0.332

The processing time for the different algorithms is summarized in Table 2.6. As expected, K-Means is faster than the other methods, however, it is relevant to see that LK-Means is visibly faster than SRIDHCR. For example, for 12 clusters in the Silhouttes

dataset, K-Means, LK-Means and SRIDHCR use approximately 0.1, 5, and 715 seconds, respectively. The reason for the slowness of SRIDHCR is that K-Medoids requires a distance matrix to be calculated between all the records, while K-Means and LK-Means only require the distance measures from the means to all records. Considering the good results of the AMI score, we can see that LK-Means is capable of combining the speed of K-Means and the semantic gain to incorporate data labels during the clustering process.

2.4.2. Experiments in object datasets

In this Section we apply LK-Means to the task of codebook generation for a visual recognition task. Currently, the Bag-of-Visual-Words (BoVW) scheme is one of the most popular approaches for visual object recognition (Sivic & Zisserman, 2003b). Under this approach, the generation of a suitable codebook plays a key role. In general, most BoVW approaches build the codebook using a clustering algorithm, mainly K-Means. Interestingly, although class labels are usually available, these are not considered during the codebook generation. This suggests a suitable scenario to test the advantages that a supervised clustering technique, such as LK-Means, can offer to provide more discriminative codebooks.

Following the previous intuition, we compare the performance of LK-Means against K-Means for the task of codebook generation in object recognition applications. As a testbed, we select 4 object recognition datasets that are commonly used to benchmark object recognition techniques. These datasets are: UIUC, DARMSTADT, VEH-CALT, and OUTDOOR. UIUC contains 2 object classes: cars and backgorund. DARMSTADT contains 3 object classes: motorbike, cow, and cars (Leibe, Leonardis, & Schiele, 2004). VEH-CALTECH is a subset of CALTECH-101 (VEH-CALT) dataset (Fei-Fei, Fergus, & Perona, 2004), including 4 object classes: airplane, car, helicopter, and motorbike. Finally, OUTDOOR contains images of 8 types of outdoor scenes (Oliva & Torralba, 2001). Table 2.7 shows relevant details for all these datasets. Following a standard implementation of K-means, we use Euclidean distance as the main similarity metric for all our test.

TABLE 2.7. Details of real object datasets.

Dataset name	# objects	# classes
UIUC	1050	2
DARMSTADT	327	3
VEH-CALT	1801	4
OUTDOOR	2600	8

Following standard procedures for BoVW schemes (Sivic & Zisserman, 2003b), we use Histogram of Gradients (HoG) as a basic visual feature (Dalal & Triggs, 2005). In particular, we obtain the HoG descriptors using patches of 32x32 pixels. These patches are selected using a sliding window process over a fix grid on each input image. In particular, we use the variant UOCTTI of HoG proposed by Felzenswalb et al. (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010). UOCTTI considers a compressed representation of HoG given by 31 dimensions. For each dataset, we use the HoG descriptors of a set of training patches to build codebooks using K-Means and LK-Means. In the case of LK-Means, we assign to each patch the label of the object class that generates the patch. We evaluate the discriminative properties of the resulting codebooks using them to train a category-object classifier. As a classifier, we use the popular linear Support Vector Machine (SVM), as in (Dalal & Triggs, 2005). In relation to the training process, we use 15 random images for training and 15 images for testing. In order to evaluate the sensibility of our results in terms of the number of clusters, we consider the following number of codewords: $K = \{50, 100, 150, 200, 250\}$.

Table 2.8 shows the average accuracy achieved by the resulting classifier. These results are obtained using a 20-hold-out scheme and a fixed value of $\alpha = 0.8$. We select this value of α extrapolating the results of the previous Section, and as a good compromise between the supervised and unsupervised terms in Equation (2.3). In Table 2.8, we can observe that LK-Means outperforms K-Means in almost all cases; and in the few cases where K-Means shows superior performance the difference in accuracies is less than 1.0%. Furthermore, we observe that the positive difference in favor of LK-Means increases with the number of clusters, indicating that LK-Means benefits more that K-Means from a greater flexibility in the search for relevant patterns.

TABLE 2.8. Accuracy for real object datasets using 20-HoldOut. In average, LK-Means overcomes K-Means in almost all datasets and all configurations. The advantage of LK-Means is more clear for larger number of codewords (K>150).

Method	**Number of clusters**				
UIUC	***50***	***100***	***150***	***200***	***250***
K-Means	**79.33**	78.17	81.17	77.33	77.50
LK-Means	78.50	**80.83**	**82.17**	**81.33**	**83.00**
DARMSTADT	***50***	***100***	***150***	***200***	***250***
K-Means	86.33	86.78	86.44	86.89	89.33
LK-Means	86.33	**89.00**	**87.89**	**87.67**	**91.67**
VEH-CALT	***50***	***100***	***150***	***200***	***250***
K-Means	71.83	79.08	78.16	79.33	81.03
LK-Means	**74.75**	**80.83**	**79.75**	**81.00**	**82.75**
OUTDOOR	***50***	***100***	***150***	***200***	***250***
K-Means	50.79	**57.00**	55.50	57.88	57.46
LK-Means	**52.13**	55.71	**57.58**	**58.50**	**60.38**

Figure 2.4 shows the confusion matrix for each dataset by considering 250 codewords. The whiteness of each bin in grid represents the frecuency of each combination of classes. The improving in UIUC dataset is given in the case of car object. In case of DARMSTADT dataset, the best results is highlighted in car class. In case of VEH-CALT dataset, the improvement is notorious in airplane class. In OUTDOOR dataset the improving is variable, however, the diagonal of confusion matrix reveals that a better average accuracy is obtained by LK-Means. All these results confirm visually the improvements given by our approach.

Finally, Figure 2.5 shows some visual codewords resulting from the VEH-CALT dataset. We present the top-six most discriminative words according to the Fisher discriminant score (Bishop, 2007); and considering $K = 200$ for both algorithm. In Figure 2.5, each visual codeword is represented by its four nearest patches. In Figure 2.5, it is possible to observe in each row that, in general, LK-Means provides more discriminative codewords than K-Means.

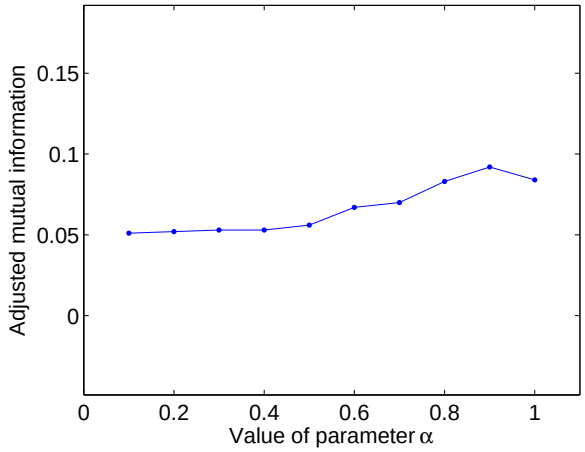
2.5. Conclusions

In this chapter we proposed LK-Means, an extension of the classical K-Means algorithm to the case of supervised clustering. As a main search strategy, LK-Means optimizes

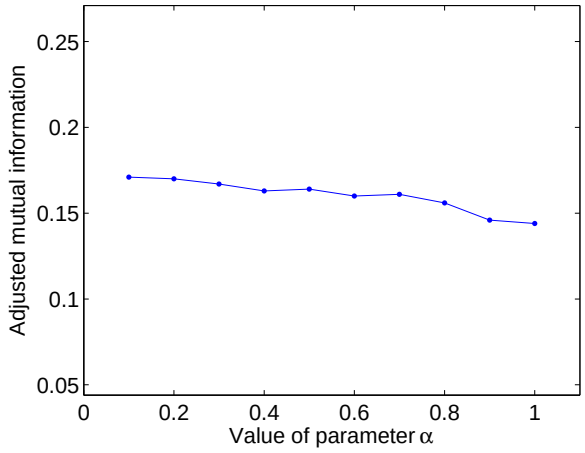
a convex combination of class dependent and non-class dependent cost functions. Experiments using a set of standard benchmark datasets and 4 different metrics to assess clustering quality, show that, on average, LK-Means outperforms classical K-Means and SRIDCHR algorithms. In the cases of AMI and AVI metrics, in most of our tests LK-Means outperforms the alternative algorithms by a large margin. In the case of ARI metric, on average, LK-Means also outperforms K-Means and SRIDCHR but by a narrower margin. In the case of MD metric, LK-Means and K-Means present mixed results. As it has been noticed in previous works, MD is negatively affected by clustering size, and it is in general less robust than metrics such as AMI and AVI. Additionally, we show an application of LK-Means as a codebook generator for object recognition applications. We consider several common benchmark datasets, and in all cases LK-Means outperforms K-Means, demonstrating the relevance of considering class information to find meaningful clusters.

Interestingly, our results indicate that the advantages of LK-Means over K-Means depends on the type of dataset. This is closely related to our hypothesis that homogeneity in class information leads to more informative clusters, which depend on the semantic of the data labels. For example, in the case of the object recognition application, where one expects a high correlation between clusters of visual features and object categories, the advantages of using LK-Means instead of regular K-Means are more clear. This observation offers a "rule of thumb" to set the value of the parameter α. For a dataset where it is expected a high correlation between class information and cluster composition α should be close to 1, increasing the relevance of class information.

In relation to time, our experiments show that LK-Means presents an attractive computational performance, being considerably faster than the alternative supervised clustering method considered in this work. In relation to future work, we plan to increase the reliability of the model by modifying the cost function of LK-Means to accommodate cluster shape. We also plan to extend this work to manage fuzzy labels inside of our model. Finally, we also plan to extend the idea behind LK-Means to the case of subspace clustering, which can provide a suitable extended search space to find relevant class-dependent clusters.

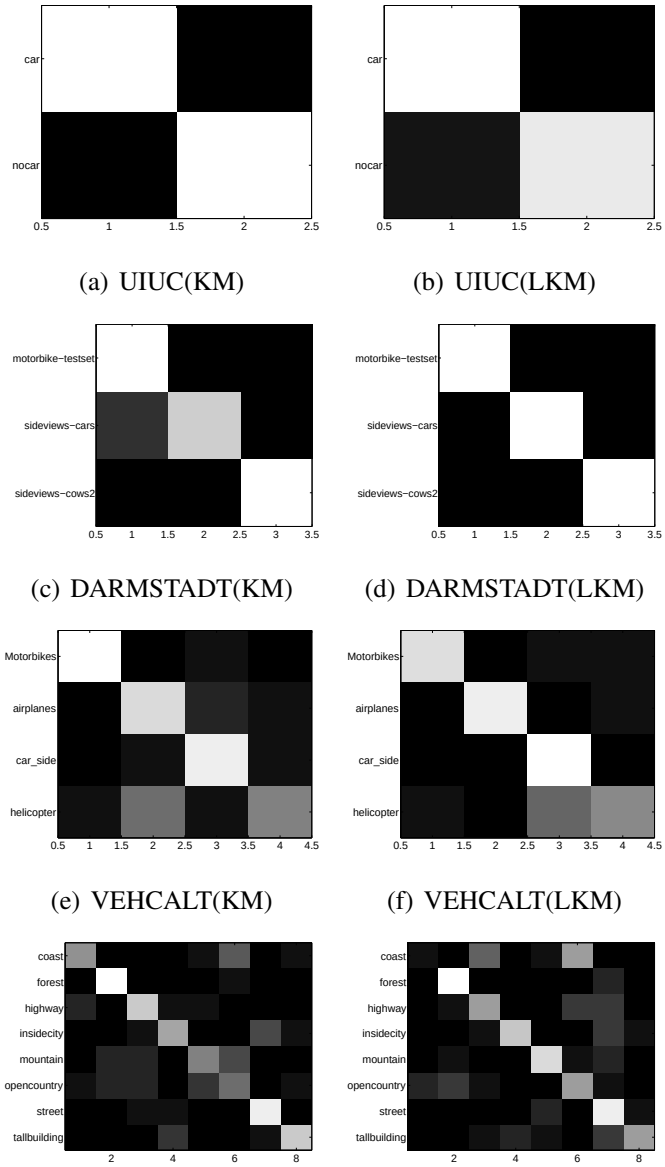


(a) **Diabetes dataset**(with 2 clusters)



(b) **Glass dataset** (with 11 clusters)

FIGURE 2.3. Comparison of sensibility of adjusted mutual information (AMI) respect to parameter α. Figure 2.3(a) shows that the best result for Diabetes dataset is obtained with α equal to 0.9, AMI=0.092, which is almost the double than the result with K-Means, AMI=0.050. On contrast, Figure 2.3(b) shows that in case of Glass dataset, the best result is obtained with α equal to 0.1, AMI=0.171, which is slightly greater than performance with K-Means, AMI=0.168. These results show that the discriminativity of clustering is dependant of data and parameter α.



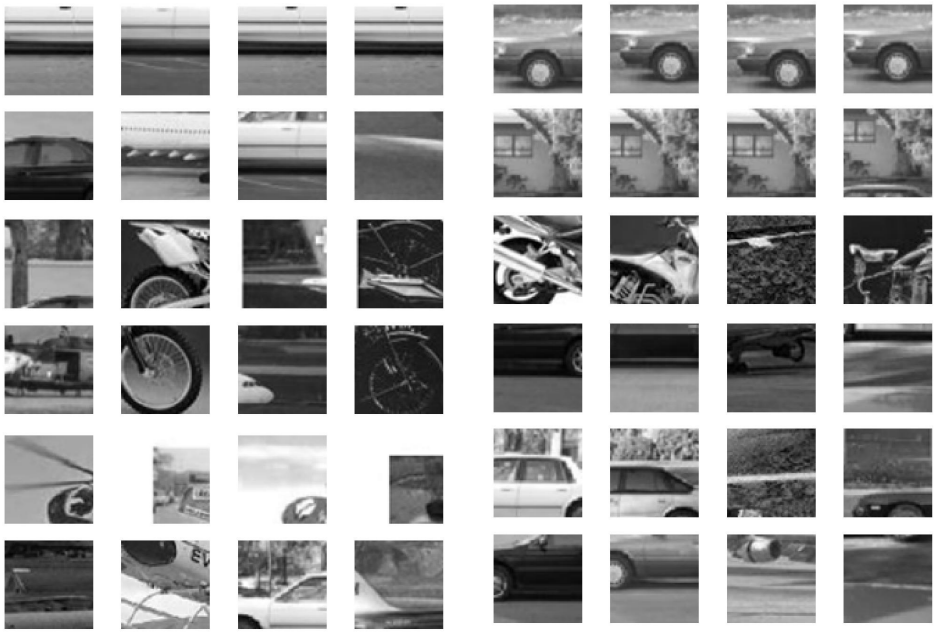
(a) UIUC(KM) (b) UIUC(LKM)

(c) DARMSTADT(KM) (d) DARMSTADT(LKM)

(e) VEHCALT(KM) (f) VEHCALT(LKM)

(g) OUTDOOR(KM) (h) OUTDOOR(LKM)

FIGURE 2.4. Confusion Matrices for object datasets. For space reasons, the algorithms K-Means and LK-Means are called in the figure as KM and LKM, respectively. The diagonal of matrices reveals the average improvement made by L-KMeans in comparison to K-Means.



(a) Top six codebooks obtained with K-Means.

(b) Top six codebooks obtained with LK-Means.

FIGURE 2.5. Top six more discriminative codewords for K-Means and LK-Means. Each codeword is represented by the four nearest patches to each one. In general, LK-Means appears to obtain visually more discriminative codewords than K-Means as we can see in the third codeword. Although a powerful clasiffier algorithm could be able to separate the classes, the quality of codeworks can help to the classifier.

References

Agarwal, S., Awan, A., & Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *Pattern Analysis and Machine Intelligence*, 1475 -1490.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html.*

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. In *International Conference of Machine Learning* (pp. 11–18).

Basu, S., Bilenko, M., & Mooney, R. (2003). Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In *Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining* (p. 29-42).

Bishop, C. (2007). *Pattern recognition and machine learning (information science and statistics)*. Springer.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Computational Learning Theory* (pp. 92–100).

Cohn, D., Caruana, R., & McCallum, A. (2003). *Semi-supervised clustering with user feedback* (Tech. Rep. No. TR2003-1892). Cornell University.

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision in European Conference on Computer Vision*, 1–22.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition* (pp. 886–893).

Deelers, S., & Auwatanamongkol, A. (2007). Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. *International Journal of Electrical and Computer Engineering*, 247–252.

Demiriz, A., Benett, K., & Embrechts, M. (1999). Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering Conference* (p. 809-814).

Dorkó, G., & Schmid, C. (2005). *Object class recognition using discriminative local features* (Tech. Rep. No. RR-5497). INRIA - Rhone-Alpes.

Eick, C., Zeidat, N., & Zhao, Z. (2004). Supervised clustering - algorithms and benefits. In *International Conference on Tools with Artificial Intelligence* (p. 774-776).

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Workshop of Generative Model Based Vision*.

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis Machine Intelligence*, 1627–1645.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Conference on Computer Vision and Pattern Recognition* (pp. 264–271).

Fowlkes, E., & Mallows, C. (1983). A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, *78*(383), 553–569.

Hubert, L., & Arabie, P. (1985). Comparing Partitions. *Journal of Classification*, 193–218.

Jiawei, H. (2005). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers Inc.

Jurie, F., & Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision* (pp. 604–610).

Kumar, R., Puran, R., & Dhar, J. (2011). Enhanced k-means clustering algorithm using red black tree and min-heap. *International Journal of Innovation, Management and Technology*, 49–54.

Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., ... Dietterich, T. (2007). Automated insect identification through concatenated histograms of local appearance features. In *Applications of Computer Vision.*

Lasserre, J., Bishop, C., & Minka, T. (2006). Principled hybrids of generative and discriminative models. In (p. 87-94).

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision.*

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Mardia, K., Kent, J., & Bibby, J. (1979). *Multivariate analysis*. Academic Press, London.

Markov, Z., & Larose, D. (2007). *Data mining the web : Uncovering patterns in web content, structure, and usage*. John Wiley and Sons.

Marszaek, M., & Schmid, C. (2006). Spatial weighting for bag-of-features. In *Conference on Computer Vision and Pattern Recognition* (p. 2118 - 2125).

Meilă., M. (2005). Comparing clusterings: An axiomatic view. In *International Conference on Machine Learning* (pp. 577–584).

Mirkin., B. (1996). *Mathematical classification and clustering*. Kluwer Academic Press.

Moosmann, F., Triggs, B., & Jurie, F. (2007). Fast discriminative visual codebooks using randomized clustering forests. , 985–992.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1999). Text classification from labeled and unlabeled documents using em. , 103–134.

Nowak, E., Jurie, F., & Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision.*

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal on Computer Vision*, 145–175.

Perronnin, F. (2008). Universal and adapted vocabularies for generic visual categorization. *Pattern Analysis and Machine Intelligence*, 1243–1256.

Rajaraman, A., & Ullman, J. (2012). *Mining of massive datasets*. Cambridge University Press.

Rice, J. (1994). *Mathematical statistics and data analysis, 2nd ed.* Duxbury Press.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(1), 53–65.

Shanmugasundaram, R., & Sukumaran, S. (2010). Enhancing k-means algorithm with semi-unsupervised centroid selection method. *International Journal of Computer Science and Information Security*, 337–343.

Sinnkkonen, J., Kaski, S., & Nikkila, J. (2002). Discriminative clustering: Optimal contingency tables by learning metrics. In *European Conference on Machine Learning* (pp. 418–430).

Sivic, J., & Zisserman, A. (2003a). Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision* (pp. 1470–1477).

Sivic, J., & Zisserman, A. (2003b). Video google: A text retrieval approach to object matching in videos. In *Ieee international conference on computer vision (iccv).*

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. In *Allerton Conference on Communications and Computation* (p. 368Ű-377).

Tishby., N., & Slonim, N. (2000). Document clustering using word clusters via information bottleneck method. In *International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 208–215).

Vinh, N., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *International Conference on Machine Learning* (pp. 1073–1080).

Vinh, N., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, *11*(3), 2837–2854.

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision* (pp. 1800 – 1807).

Wu, X., Kumar, V., Ross, J., Ghosh, J., Yang, Q., Motoda, H., ... Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 1–37.

Xing, E., Ng., A., Jordan, M., & Russell, S. (2003). Distance metric learning with applications to clustering with side information. *Advances in Neural Information Processing Systems*, 505–512.

Yang, L., Jin, R., Sukthankar, R., & Jurie, F. (2008). Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Conference on Computer Vision and Pattern Recognition.*

Ye, J., Z.Zhao, & Wu, M. (2008). Discriminative k-means for clustering. In *Advances in neural information processing systems* (pp. 1649–1656). Retrieved from `http://dblp.uni-trier.de/db/conf/nips/nips2007.htmlYeZW07`

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 213–238.

Zhang, W., Surve, A., Fern, X., & Dietterich, T. (2009). Learning non-redundant codebooks for classifying complex objects. In *International Conference on Machine Learning* (pp. 1241–1248).

3. EMBEDDED LOCAL FEATURE SELECTION WITHIN MIXTURE OF EXPERTS

3.1. Introduction

Performing classification in scenarios with large and complex intra and inter-class variation is a challenging task for most classification methods. In these cases, different subsets of instances might respond to different patterns and, even more, these patterns might arise in different subsets of dimensions. As an example, in visual recognition, changes in illumination or pose conditions usually produce drastic variations in the visual appearance of relevant objects, affecting the discriminative properties of different visual features (Pinto, Cox, & DiCarlo, 2008). As a further example, in gene function prediction, the expression level of particular genes can change substantially under different experimental conditions, affecting the discriminative properties of different co-expression patterns that usually arise on subsets of experiments (Guyon, Weston, Barnhill, & Vapnik, 2002).

A useful strategy to deal with complex classification scenarios is the "divide and conquer" approach. Under this strategy, a complex problem is divided into multiple simpler problems. Decision trees (DTs) are one of the oldest and most widely used classification techniques based on this strategy (Quinlan, 1993). This technique consists of building a tree using a partitioning scheme that recursively divides the input space and adjusts local classifiers within each partition. Interestingly, each branch of the resulting tree is in charge of classifying a different subset of instances. Furthermore, classification in each branch is performed using a particular subset of dimensions. We believe that this double "divide and conquer" strategy, that adaptively adjusts each branch of the tree to deal with a selected subsets of instances and dimensions, is one of the main reasons to explain the good performance shown by DTs and their later extensions based on ensemble strategies (Breiman, 2001). Unfortunately, the representational space and usual learning strategies used by DTs impose relevant limitations that affect their abilities to deal with complex classification scenarios. In particular, a DT embeds a hypothesis space given by a disjunction of conjunctions of constraints. These constraints are usually based on single (Quinlan, 1993) or

low dimensional (Murthy, Kasif, & Salzberg, 1994) partitions of the input space. Furthermore, common training strategies are based on greedy schemes that can lead to suboptimal solutions. As an example, the greedy decision at the root node of the tree constrains the conjunctions embedded by all the branches of the tree.

A probabilistic approach to the "divide and conquer" strategy is the mixture of experts (MoE) technique (Jacobs, Jordan, Nowlan, & Hinton, 1991). In contrast to DTs, this technique uses a probabilistic framework that is advantageous in managing the intrinsic uncertainty in the data. MoE divides the data into multiple regions where each region has its own classifier or expert (Jacobs et al., 1991). Each expert is specified by a probability distribution that is conditioned on class values. In the mixture, predictions of experts are weighed using a global model known as gate function. This function adaptively estimates the relevance or weight assigned to each expert for the classification of each input instance.

Both, DTs and MoE, use a "divide and conquer" strategy that divides the input space to perform classification, using a hard partitioning in the case of a DT and a probabilistic, or soft, partitioning in the case of MoE. A relevant difference arises in terms of how each technique handle the dimensionality of each instance: while a DT incorporates an embedded feature selection scheme, MoE does not. We believe that a suitable embedded feature selection scheme can be a useful tool to boost the performance of the MoE technique. In particular, in our experiments for the case of high dimensional datasets we notice that the traditional MoE technique has serious difficulties to learn adequate models. Also, as the number of parameters increases with the number of dimensions, the resulting MoE models become complex usually leading to overfitting problems.

This work contributes with a MoE model that incorporates embedded local feature selection using L_1 regularization. Our main intuition is that particular subsets of dimensions, or subspaces, are usually more appropriate to classify certain input instances. Consequently, we expect to improve the accuracy of traditional MoE models by introducing a technique that adaptively selects subsets of dimensions to train each expert in the mixture.

This chapter is organized as follows. Section 3.2 presents background information about feature selection methods, in particular $L1$ regularization. Section 3.3 describes relevant previous works. Section 3.4 presents the proposed approach. Section 3.5 presents and discusses the results of our experiments. Finally, Section 3.6 presents our main conclusions and future avenues of research.

3.2. Background

3.2.1. Feature selection

In classification problems, the goal corresponds to learn a mapping from an input vector x to an output value y, where x is a vector with D dimensions and y takes categorical values. If vector x is high-dimensional, one can usually improve classification accuracy by discarding irrelevant and redundant features (Kohavi & John, 1997; Guyon & Elisseeff, 2003). This process is known as feature or variable selection. In general, there are three main methods for feature selection:

- *Filter methods* rank each input feature x_j in relation to predicting y using a metric of goodness, such as mutual information (Battiti, 1994), Pearson correlation (Hall, 1999), Fisher score (Duda, Hart, & Stork, 2001), and chi-square statistic (Liu & Setiono, 1995), among others (Guyon & Elisseeff, 2003). Next, the features are selected according to ranking results. These methods can be incorporated in a sequential forward selection in order to find a subset of discriminant dimensions (Hall, 1999). Generally, the chosen metric is independent of the final classification model (Guyon & Elisseeff, 2003). Filter methods are usually fast and simple, in comparison to alternative techniques.
- *Wrapper methods* search the feature space looking for possible subsets that improve performance. For each subset, these methods execute the classification model and evaluate its resulting predictive power (Kohavi & John, 1997), usually using accuracy or F-measure (Van-Rijsbergen, 1979). Then, the subset of features with greatest predictive power is chosen. Main issues associated with these

methods are difficulties defining the best metric to measure predictive power, as well as the computational complexity associated to the evaluation of a large number of subsets of features, $2^D - 1$ subsets in the worst case (exhaustive search).

- *Embedded methods* combine feature selection and model fitting into a single optimization problem. DT (Quinlan, 1993) and Adaboost (Freund & Schapire, 1995) can be considered embedded techniques, although an explicit criterium for feature minimization is not included during the training process. Two popular techniques to embed feature selection inside a classification algorithm are L_1-regularization (Tibshirani, 1996) and automatic relevancy determination (MacKay, 1995).

This chapter concentrates on embedded models, specifically L_1-regularization, due to their computational tractability and formal soundness. Although filter methods are faster than alternative techniques, they are usually less effective, as they use a metric that is independent of the final classification scheme. On the other hand, wrapper methods are generally more reliable, as they can take advantage of robust classification algorithms. Nevertheless, these methods are slow due to the usually large number of subsets to explore, and the complexity associated to repeatedly training a robust classifier (Kohavi & John, 1997). Embedded models are attractive because they use a reliable measure of goodness, similar to wrapper methods, but they avoid retraining a predictor for each feature subset explored.

3.2.2. L_1 regularization

Consider the context of linear models given by the expression $y = w^T x + b$, where $x \in \Re^D$ is the input vector, $y \in \Re$ is the output value, $w \in \Re^D$ is the vector of coefficients, and $b \in \Re$ is the bias (Bishop, 2007). Selecting features by means of regularization fits a vector w of parameters, and at the same time maximizes the number of coefficients w_i of w that takes value equal to zero. . As a consequence, for each coefficient $w_i = 0$, the associated dimension x_i of x can be ignored.

The details of L_1 regularization are derived in the context of probabilistic models (Tibshirani, 1996). Specifically, the optimization for simultaneous fitting and regularization of the likelihood function of a probabilistic model can be expressed as:

$$\hat{L_0} \;=\; L(w) \;+\; \lambda \left\| w \right\|_0 , \tag{3.1}$$

where L is the likelihood function that depends on the parameter w. In this case, $\|w\|_0 = \sum_{j=1}^{D} I(|w_j| > 0)$ counts the number of nonzero elements of vector w. $\lambda > 0$ is a trade-off constant

that balances between model fitting and regularization. The direct maximization of Equation (3.1) is in general unfeasible due to the discrete nature of $I(|w_j|)$. Considering a relaxation of the previous objective function, one has:

$$\hat{L_1} \;=\; L(w) \;+\; \lambda \left\| w \right\|_1 , \tag{3.2}$$

where $\|w\|_1 = \sum_{j=1}^{d} |w_j|$. This results in a sparse weight vector w, which means that many of its elements are zero. The nonzero elements correspond to relevant features. This method is known in the statistics community as Lasso (least absolute shrinkage and selection operator), or L_1 regularization (Tibshirani, 1996). Equation (3.2) can be rewritten as:

$$\begin{aligned} \hat{w} \;&=\; \arg\max_{w} L(w) \\ \text{subject to} \quad & \quad ||w||_1 \leq t, \end{aligned}$$

where t is an upper bound on the L_1 norm of weights. More precisely, a tight bound t is equivalent to a heavy regularization λ, whereas a loose bound t corresponds to a small value of λ. Lasso can be interpreted similarly to a quadratic cost function with linear constraints

and is thus a convex quadratic problem, which has efficient algorithms to solve it (Boyd & Vandenberghe, 2004).

3.3. Related work

In a seminal work, Jacobs et al. (Jacobs et al., 1991) introduce the MoE technique. They divide the space of data into several separate models, where each model has its own supervised classifier. Then, they use a gradient approach to learn parameters which, in this case, is a vector of weights. Finally, they apply this model to multi-speaker vocal recognition. They base their work on Hampshire et al. (Hampshire & Waibel, 1992), who combine the outputs of local experts but without considering localization.

Jordan and Jacobs (Jordan & Jacobs, 1994) extend the MoE formulation to a hierarchical case. They treat this model as a conditional mixture model where the distribution of outputs is given by a mixture of component distributions referred as experts. These experts and mixing coefficients are conditioned on input features. Also, the mixing coefficients are controlled by gating distributions. They use the EM algorithm (Dempster, Laird, & Rubin, 1977) to learn model parameters through a maximum likelihood scheme. They experiment with a robot dynamic problem where they obtain results that are comparable to a neural network trained with the Backpropagation algorithm, but with greater speed. A comparison of MoE with other ensemble methods is given by Vogdrup (J.Vogdrup, 2010), where a variant of MoE outperforms Adaboost, Bagging, and hierarchical MoE techniques in terms of classification performance.

Titsias and Likas (Titsias & Likas, 2002) present a three-level hierarchical mixture model for classification where each mixture component has an independent class-conditional mixture. The model estimates the posterior probability of class membership using a similar scheme to the MoE classifier. Model parameters are learned using maximum likelihood. Their results indicate that the final model improves classification with respect to the case where class-conditional information is not considered in each mixture component.

Bishop and Svensén (Bishop & Svensén, 2003) propose a full Bayesian treatment of the hierarchical mixture of experts combining local and global variational methods. For doing this, they establish a lower bound on the marginal probability of data under the model. The greatest difficulty with this Bayesian approach comes from the resulting gating distribution that do not admit a conjugate prior. They use a variational approximation for the logistic function and approximate the joint distribution of the model parameters by a factorized distribution. They apply this method to a kinematics problem, outperforming the results of a hierarchical MoE.

In the context of nonparametric Bayesian models, Rasmussen and Ghahramani (Rasmussen & Ghahramani, 2001) present a nonparametric extension to MoE models, where they use Gaussian processes to model the experts. Also, they use an input-dependent adaptation of the Dirichlet Process to implement a gating network for an infinite number of experts. Inference is performed using Gibbs sampling. This model adjusts the covariance function according to inputs. Simulations show the viability of their approach, however, this model is complex, as it depends on many hyperparameters where interpretability is not natural.

Meeds and Osindero propose an alternative infinite mixture of experts where each expert comprises a multivariate Gaussian distribution to model its inputs, and a Gaussian Process to model its outputs (Meeds & Osindero, 2005). They use a full generative model over input and output spaces. This approach presents some advantages related to conditional models due to its capability to deal with incomplete data, however, as in (Rasmussen & Ghahramani, 2001), this work requires fitting a large number of hyperparameters.

Additionally, some variations in the form of gating and expert functions have been proposed. Xu et al. (Xu, Jordan, & Hinton, 1994) suggest to replace the usual multinomial logit model with Gaussian basis functions, where each expert is modeled by a Gaussian function. This idea adds flexibility to model the local covariance of the data. Nguyen et al. (Nguyen, Abbass, & McKay, 2006) propose a variation to the classical MoE by using an evolutionary algorithm to learn the model. The overall model is an ensemble, where each component is a mixture of experts. In the context of regression, Lima et al. (Lima, Coelho,

& Zuben, 2007) combine MoE with support vector machines in a probabilistic framework, where the gate functions are represented by a normalized kernel function and the experts correspond to support vector machines.

On the other hand, there have been recently some interesting applications of MoE. Saragih et al. (Saragih, Lucey, & Cohn, 2009) apply MoE to a deformable model fitting problem. Ebrahimpour and Jafarlou (Ebrahimpour & Jafarlou, 2010) apply a hierarchical MoE to view-independent face recognition. They use principal component analysis to find a suitable representation of the data and neural networks to model experts and the gate function.

Closely related to our approach, in terms of embedded local feature selection in a mixture model, Pan and Shen (Pan & Shen, 2007) propose the use of L_1 regularization for selecting features in the unsupervised case of model-based clustering. In experiments with real datasets, as they state, they do not attain good results for the case of classification, most likely due to a mismatch between true labels and resulting clusters. Wang and Zhu (S. Wang & Zhu, 2008) also apply regularization over clustering but using L_∞ norm. They propose to use quadratic programming to solve equations related to a constrained optimization.

As it can be seen from the our review, a common issue among previous works on MoE for classification is the fact that they do not consider an explicit feature selection scheme. Notable exceptions are (Pan & Shen, 2007) and (S. Wang & Zhu, 2008) where an embedded feature selection step is applied, but in the context of clustering and not of classification. Also, in the context of regression and closely related to our work, Khalili (Khalili, 2010) presents a MoE model that includes an embedded feature selection approach based on a regularization scheme and Gaussian models. Similarly, we propose a regularization scheme to add feature selection to the MoE model, however, we formulate our model in the context of classification using multinomial logit functions. As a consequence, our domain of applications, mathematical formulation, and optimization solution are highly different from the one proposed in (Khalili, 2010). In particular, we use an iterative optimization

scheme similar to the one used in (Lee, Lee, Abbeel, & Ng, 2006), while (Khalili, 2010) uses a local quadratic approximation for the regularization term.

In the context of alternative techniques to MoE, there are also interesting works on approaches to combine classifiers, and to perform embedded feature selection. Xiao et al. (Xiao, He, Jiang, & Liu, 2010) propose to combine classifiers by jointly maximizing accuracy and ensemble diversity using a neural network architecture. Ulas et al. (Ulas, Taner, & Alpaydin, 2012) combine classifiers by employing the most informative components of the eigenvectors corresponding to the correlation matrix among classifier outputs. Ñanculef et al. (Nanculef, Valle, Allende, & Moraga, 2012) propose to learn an ensemble of regressors using a sequential scheme and a score minimizing classification error and ensemble diversity. All these works do not include an embedded feature selection mechanism.

In terms of classification schemes that include an embedded feature selection process. Wu et al. (Wu, Yuan, & Zhuang, 2010) propose to select groups of features for image classification arguing that, usually, subsets of visual features are related to specific group of instances. Consequently, they incorporate a group Lasso regularizer inside a logistic regression classifier, solving the resulting optimization problem using a co-ordinate descent method. In the context of object recognition, Yang et al. (Yang, Wang, Hua, Yan, & Zhang, 2011) boost a standard object classifier based on parts, by adding supplementary parts. These additional parts are selected using a classification scheme that includes Lasso regularization over the selected parts. Maldonado et al. (Maldonado, Weber, & Basak, 2011) perform feature selection inside a support vector machine considering a penalization score over the features used by the kernel functions. In contrast to our approach, (Wu et al., 2010) and (Maldonado et al., 2011) consider a global feature selection and (Yang et al., 2011) performs a global part selection scheme, while our work considers local feature selection for each expert classifier.

The idea of adding a regularizer over weights that are used to integrate multiple models has been explored in previous works. Hua et al. (M. Wang et al., 2009) propose a framework to annotate videos considering different aspects such as low level features and

temporal consistency, where each aspect is represented by multiple instance graphs. In particular, they develop a procedure to find a weight for each graph by jointly optimizing all graphs and a regularization score over the weights. Geng et al. (Geng, Tao, T.Xu, Yang, & Hua, 2012) develop a method to learn the intrinsic ensemble of manifolds for unlabeled data in a semi-supervised scenario. Their method begins with a guess for initial manifolds, iterating then to find suitable weights that increase the smoothing and discriminative power of the manifolds. Wang et al. (M. Wang, Li, Tao, Lu, & Wu, 2012) also present a model that learns the weights of a graph ensemble. In particular, their method re-ranks web images, constrained to be near the outputs of a textual search. A relevance score for each graph is learned jointly with the weight of each graph by constraining that visually similar images should have similar relevance scores. In this way, these previous works regularize the parameters of the respective models in order to smooth the labels of a graph (M. Wang et al., 2009)(Geng et al., 2012), increase discrimination (Geng et al., 2012)(M. Wang et al., 2012), or construct a suitable similarity metric between pairs of instances (M. Wang et al., 2012). Our work can be considered complementary to these previous works, in the sense that it focuses on sparsely selecting the features used by the local models of a MoE scheme.

3.4. Proposed Approach

In this section we present our main contribution, RMoE, a regularized version of MoE technique that incorporates a local feature selection scheme inside each expert and gate function. Our main intuition is that, in the context of classification, different partitions of the input data can be best represented by specific subsets of features. This is particularly relevant in the case of high dimensional spaces, where the common presence of noisy or irrelevant features might obscure the detection of particular class patterns. Specifically, our approach takes advantage of the linear nature of each local expert and gate function in the classical MoE formulation (Jacobs et al., 1991), meaning that L_1 regularization can be directly applied. Below, we first briefly describe the classical MoE formulation for classification. Afterwards, we discuss the proposed modification to the MoE model that provides embedded feature selection.

3.4.1. Mixture of Experts

In the context of supervised classification, there is available a set of N training examples, or instance-label pairs (x_n, y_n), representative of the domain data (x, y), where $x_n \in \Re^D$ and $y_n \in C$. Here C is a discrete set of Q class labels $\{c_1, ..., c_Q\}$. The goal is to use training data to find a function f that minimizes a loss function which scores the capacity of f to predict the true underlying relation between x and y. From a probabilistic point of view (Bishop, 2007), a useful approach to find f is to use a conditional probability formulation:

$$f(x) \;=\; \arg\max_{c_i \in C} \; p(y = c_i|x).$$

In the general case of complex relations between x and y, a useful strategy consists of approximating f through a mixture of local functions. This is similar to the case of modeling a mixture distribution (Scott & Sain, 2004) and it leads to the MoE model.

We decompose the conditional likelihood $p(y|x)$ as:

$$p(y|x) \;=\; \sum_{i=1}^{K} p(y, m_i|x) \;=\; \sum_{i=1}^{K} p(y|m_i, x)\, p(m_i|x), \tag{3.3}$$

where Equation (3.3) represents a MoE model with K experts m_i. Figure 3.1 shows a schematic diagram of the MoE approach. The main idea is to obtain local models in such a way that they are specialized in a particular region of the data. In Figure 3.1, x corresponds to the input instance, $p(y|m_i, x)$ is the expert function, $p(m_i|x)$ is the gating function, and $p(y|x)$ is the weighted sum of experts. Note that the output of each expert model is weighed by the gating function. This weight can be interpreted as the *relevance* of expert m_i for the classification of input instance x. Also note that the gate function has K outputs, one for each expert. There are K expert functions that have Q components, one for each class.

The traditional MoE technique uses multinomial logit models, also known as softmax functions (Bishop, 2007), to represent the gate and expert functions. An important

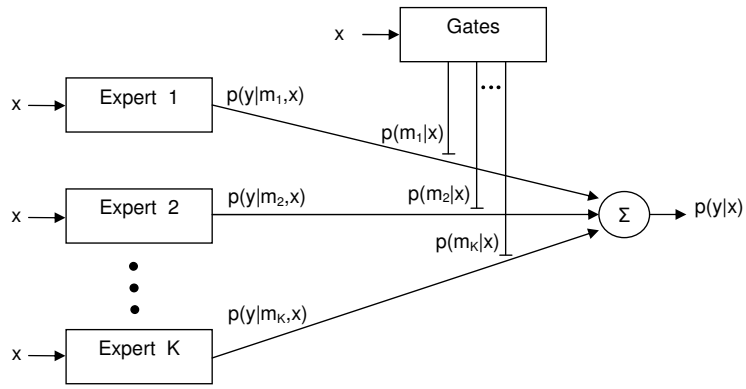


FIGURE 3.1. Mixture of experts scheme.

characteristic of this model is that it forces competition among its components. In MoE, such components are expert functions for the gates and class-conditional functions for the experts. The competition in soft-max functions enforces the especialization of experts in different areas of the input space (Yuille & Geiger, 1998).

Using multinomial logit models, a gate function is defined as:

$$p(m_i|x) \quad = \quad \frac{exp(\nu_i^T x)}{\sum_{j=1}^{K} exp(\nu_j^T x)} \tag{3.4}$$

where $i \in \{1, \ldots, K\}$ indexes expert i and $\nu_i \in \Re^D$ is a vector of model parameters. Component ν_{ij} of vector ν_i models the relation between the gate and dimension j of input instance x.

Similarly, an expert function is defined as:

$$p(y = c_l|x, m_i) \quad = \quad \frac{exp(\omega_{li}^T x)}{\sum_{j=1}^{M} exp(\omega_{ji}^T x)} \tag{3.5}$$

where ω_{li} depends on class label c_l and expert i. In this way, there are a total of $Q \times K$ vectors ω_{li}. Component ω_{lij} of vector ω_{li} models the relation between expert function i and dimension j of input instance x.

There are several methods to find the value of the hidden parameters ν_{ij} and ω_{lij} (Moerland, 1997). An attractive alternative is to use the EM algorithm. In the case of MoE, the EM formulation augments the model by introducing a set of latent variables, or *responsibilities*, indicating the expert that generates each instance. Accordingly, EM iterations consider an expectation step that estimates expected values for *responsibilities*, and a maximization step that updates the values of parameters ν_{ij} and ω_{lij}. Specifically, the posterior probability of the *responsibility* R_{in} assigned by the gate function to expert m_i for an instance x_n is given by (Moerland, 1997):

$$\begin{aligned} R_{in} &= p(m_i|x_n, y_n) \\ &= \frac{p(y_n|x_n, m_i)\, p(m_i|x_n)}{\sum_{j=1}^{K} p(y_n|x_n, m_j)\, p(m_j|x_n)} \end{aligned} \tag{3.6}$$

Considering these responsibilities and Equation (3.3), the expected complete log-likelihood $\langle L_c \rangle$ used in EM iterations is (Moerland, 1997):

$$\langle L_c \rangle = \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in} \; [log\; p(y_n|x_n, m_i) \; + log\; p(m_i|x_n)] \tag{3.7}$$

3.4.2. Regularized Mixture of Experts (RMoE)

To embed a feature selection process in the MoE approach, we use the fact that in Equations (3.4) and (3.5), the multinomial logit models for gate and experts functions contain linear relations in the relevant parameters. This linearity can be straightforwardly used in feature selection by considering that a null parameter component ν_{ij} or ω_{lij} implies that dimension j is irrelevant for gate function $p(m_i|x)$, or expert model $p(y|m_i, x)$, respectively. Consequently, we propose to penalize complex models using L_1 regularization. A similar consideration is used in (Pan & Shen, 2007), but in the context of unsupervised learning. The idea is to maximize the likelihood of the data while simultaneously minimizing the number of non-null parameter components ν_{ij} and ω_{lij}. Considering that there are

Q classes, K experts, and D dimensions, the expected L_1 regularized log-likelihood $\langle L_c^R \rangle$ is given by:

$$\langle L_c^R \rangle \;=\; \langle L_c \rangle - \lambda_\nu \sum_{i=1}^{K} \sum_{j=1}^{D} |\nu_{ij}| - \lambda_\omega \sum_{l=1}^{Q} \sum_{i=1}^{K} \sum_{j=1}^{D} |\omega_{lij}| \; . \tag{3.8}$$

To maximize Equation (3.8) with respect to model parameters, we first use the standard fact that the likelihood function in Equation (3.7) can be decomposed in terms of independent expressions for gate and expert models (Moerland, 1997). In this way, the maximization step of the EM based solution can be performed independently for gate and expert parameters (Moerland, 1997). In our problem, each of these optimizations has an additional term given by the respective regularization term in Equation (3.8). To handle this case, we observe that each of these optimizations is equivalent to a regularized logistic regression (Lee et al., 2006). As shown in (Lee et al., 2006), this problem can be solved by using a coordinate ascent optimization strategy (Tseng, 2001) given by a sequential two-step approach that first models the problem as an unregularized logistic regression and afterwards incorporates the regularization constraints.

In summary, we handle Equation (3.8) by using an EM based strategy that, at each step, solves the maximization with respect to model parameters by decomposing the problem in terms of gate and expert parameters. Each of these problems is, in turn, solved using the strategy proposed in (Lee et al., 2006). Next, we provide details of this procedure.

Optimization of the unregularized log-likelihood

In this case, we solve the unconstrained maximization of the log-likelihood given by Equation (3.7). First, we optimize the log-likelihood with respect to vector ω_{li}. The maximization of the expected log-likelihood $\langle L_c \rangle$ implies differentiating Equation (3.7) with respect to ω_{li}:

$$\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}\ [log\ p(y_n|x_n, m_i)\ ]\omega_{li} \quad = \quad 0 \tag{3.9}$$

which is equivalent to:

$$-\sum_{n=1}^{N} R_{in}\left(p(y_n|x_n, m_i) - y_n\right) x_n \quad = \quad 0. \tag{3.10}$$

In this case, the classical technique of least-squares cannot be directly applied because of the soft-max function in $p(y_n|x_n, m_i)$. Fortunately, as described in (Jordan & Jacobs, 1994) and later in (Moerland, 1997), Equation (3.10) can be approximated by using a transformation that implies inverting the soft-max function. Using this transformation, Equation (3.10) is equivalent to an optimization problem that can be solved using a weighted least squares technique (Bishop, 2007):

$$\min_{\omega_{li}} \quad \textstyle\sum_{n=1}^{N} R_{in}\left(\omega_{li}^T x_n - log\ y_n\right)^2 \tag{3.11}$$

A similar derivation can be performed with respect to vectors ν_i. Again differentiating Equation (3.7), in this case with respect to parameters ν_{ij} and applying the transformation suggested in (Jordan & Jacobs, 1994), we obtain:

$$\min_{\nu_i} \quad \textstyle\sum_{n=1}^{N} \left(\nu_i^T x_n - logR_{in}\right)^2 \tag{3.12}$$

Optimization of the regularized likelihood

Following the procedure in (Lee et al., 2006), we add the regularization term to the optimization problem given by Equation (3.11), obtaining an expression that can be solved by any standard algorithm for Lasso resolution (Tibshirani, 1996):

$$\min_{\omega_{li}} \quad \sum_{n=1}^{N} R_{in} \left(log\, y_n - \omega_{li}^T x_n \right)^2$$
$$\text{subject to:} \qquad ||\omega_{li}||_1 \leq \lambda_\omega \tag{3.13}$$

Similarly, we can also obtain a standard Lasso optimization problem to find parameters ν_{ij} :

$$\min_{\nu_i} \quad \sum_{n=1}^{N} \left(logR_{in} - \nu_i^T x_n \right)^2$$
$$\text{subject: to} \qquad ||\nu_i||_1 \leq \lambda_\nu \tag{3.14}$$

Specifically, in the case of T iterations, there are a total of $T * K * (Q + 1)$ Lasso optimization problems related to the maximization step of the EM algorithm. To further reduce this computational load, we slightly modify this maximization by applying the following two-steps scheme:

- Step-1: Solve K Lasso optimization problems to find gate parameters ν_{ij} assuming that each expert uses all the available dimensions. In this case, there are $T-1$ iterations.
- Step-2: Solve $K*(Q+1)$ Lasso optimization problems to find expert parameters ω_{lij} applying the feature selection process. In this case, there is a single iteration.

Using the previous scheme we reduce from $T*K*(Q+1)$ to $K*(T-1)+K*(Q+1)$ the number of Lasso optimization problems that we need to solve in the maximization step of the EM algorithm. In our experiments, we do not notice a drop in performance by using this simplification, but we are able to increase processing speed in one order of magnitude.

In summary, starting by assigning random values to the relevant parameters ν_{ij} and ω_{lij}, our EM implementation consists of iterating the following two steps:

- Expectation: estimating responsabilities for each expert using Equation (3.6), and then estimating the outputs of gate and experts using Equations (3.4) and (3.5).
- Maximization: updating the values of parameters ν_{ij} and ω_{lij} in Equations (3.13) and (3.14) by solving $K * (T-1) + K * (Q+1)$ Lasso optimization problems according to the approximation described above in Step-1 and Step-2.

3.5. Experiments

In this section, we use synthetic and real datasets to analyze the performance of RMoE. In particular, we compare its performance against the traditional MoE technique. Furthermore, we analyze RMoE in terms of classification accuracy and dimensionality reduction under different parameter configurations. RMoE is oriented to classification tasks, therefore, it requires categorical class variables. Finally, we compare the performance of RMoE against three popular classification algorithms that also consider embedded feature selection: Random Subspace (*RS*) (Kam, 1998), Decision Trees (*DT*) (Quinlan, 1993), and Adaboost (*AB*) (Freund & Schapire, 1995). In the case of Adaboost, we use decision stumps as weak classifiers.

All results reported in this section are obtained by averaging performances on 30 hold-out folds. In each case, classification performance is measured by using an independent test set that is not used in the determination of any of the parameters associated to each method. For each case, the estimation of parameters is performed by using training sets corresponding to 50% of the available data. In particular, in the case of RMoE, we first use the training set to apply a 3-fold cross-validation procedure to obtain suitable values for the regularization parameters λ_ν and λ_ω. Specifically, we test a total of 24 combinations of parameter values for λ_ν and λ_ω. For parameter λ_ω we test the set of values: $\{20, 10, 5, 2, 1, 0.5, 0.2, 0.1\}$ and, for each of these cases, we select the corresponding λ_ν by multiplying λ_ω by the factors in: $\{2, 1, 0.5\}$. We choose the combination of λ_ν and λ_ω with highest accuracy, according to the 3-fold cross validation. After the values of λ_ν and λ_ω are selected, we use the complete training set to estimate parameters for experts and gate

function. For the cases of Random Subspace, Decision Trees, and Adaboost, main parameters are the number of trees in the forest, the minimum number of records in leaf nodes, and the number of decision stumps (weak classifiers), respectively. These parameters are obtained from the best model according to 2-fold cross-validation inside the training set of each hold-out fold. For Random Subspace, we experiment using from 5 to 50 trees in the forest with a step size of 5. For Decision Trees, we test using from 1 to 10 records in leaf nodes with a step size of 1. For Adaboost, we test using from 10 to 100 decision stumps with a step size of 10.

3.5.1. Synthetic datasets

We generate 6 synthetic datasets, each consisting of two equiprobable classes. We define relevant patterns for each class using a subset of the total number of dimensions. Specifically, relevant dimensions for each class are represented by a multivariate Gaussian distribution using a randomly selected subset consisting of 4 to 6 dimensions. For each training instance the remaining dimensions are filled using samples from an Uniform distribution. As shown in Table 3.1, we vary the total number of dimensions in the datasets from 200 to 1200 dimensions. In terms of parameters, Gaussian distributions are selected in such a way that their central parts do not overlap. Means vectors are randomly selected within the range $[0, 20]$, while covariance matrices are diagonal with non-zero values randomly selected within the range $[0, 1]$. In case of Uniform distributions, they are defined within the range $[0, 20]$.

TABLE 3.1. Synthetic datasets used for experiments.

Dataset name	#Instances	#Dimensions	#Relevant dimensions {class-1,class-2}
Dataset 1	400	200	$\{4, 4\}$
Dataset 2	400	400	$\{4, 4\}$
Dataset 3	500	600	$\{5, 5\}$
Dataset 4	500	800	$\{5, 5\}$
Dataset 5	600	1000	$\{6, 6\}$
Dataset 6	600	1200	$\{6, 6\}$

TABLE 3.2. Accuracy on synthetic datasets using 30 hold-out partitions. RMoE(λ_ν^*,λ_ω^*) indicates average (std. deviation) classification accuracy for best parameters configuration. The pair ($\lambda_\nu^{'}$,$\lambda_\omega^{'}$) show the median of the best parameters obtained by 3-fold cross-validation inside the training set of each hold-out partition (see main text for details).

Dataset name	MoE	RMoE($\lambda_\nu^{'}$,$\lambda_\omega^{'}$)	(λ_ν^*,λ_ω^*)
Dataset 1	52.9(4.6)	**93.1(2.0)**	(5,10)
Dataset 2	54.4(4.0)	**98.3(1.3)**	(10,10)
Dataset 3	53.4(2.9)	**97.0(1.2)**	(5,5)
Dataset 4	51.7(3.0)	**96.4(1.1)**	(10,10)
Dataset 5	52.6(4.0)	**97.4(1.0)**	(7.5,10)
Dataset 6	51.9(2.8)	**98.2(0.9)**	(10,5)

Table 3.2 shows that in terms of classification accuracy, RMoE outperforms MoE technique in all the tested datasets. We can observe that the level of improvement of RMoE with respect to MoE fluctuates among the datasets. For example, in Dataset 1 RMoE improves the performance of MoE by 40%, while in Dataset 8 the improvement is 46%.

Table 3.3 compares RMoE and MoE in terms of parameter dimensionality for the classification results shown in Table 3.2. The main observation is that, as expected, for these types of high dimensional datasets RMoE provides sparse models.

TABLE 3.3. Average parameter dimensionality for results shown in Table 3.2.

Dataset name	MoE	RMoE(λ_ν^*,λ_ω^*)	Feature reduction
Dataset 1	200	33	83.5%
Dataset 2	400	23	94.2%
Dataset 3	600	38	93.7%
Dataset 4	800	22	97.2%
Dataset 5	1000	30	97.0%
Dataset 6	1200	39	96.7%

Finally, we evaluate the performance of the feature selection step included in RMoE. In this case, we use the fact that for the synthetic datasets we know the true dimensions used to generate the class pattern behind each instance. Specifically, we define the following score that quantifies the relevance assigned by RMoE to dimension j for the classification of input instance x:

$$\varphi(j;x) \quad = \quad \sum_{i=1}^{K} p(m_i|x) * |\omega_{y^*ij}|, \tag{3.15}$$

where K is the number of experts, y^* is the class label provided by RMoE to input instance x, ω_{y^*ij} is the parameter associated to dimension j when expert i is applied to class y^*, and $p(m_i|x)$ is the posterior probability assigned by the gate to expert i given input x. This score evaluates the relevance of each data dimension considering both: the weight assigned to the data dimension by each expert and the weight assigned by the gate to the respective expert. Equation (3.15) considers absolute values for parameters ω_{y^*ij} because, according to the regularization, only values near zero imply that the corresponding feature is irrelevant for the mixture.

Given scores $\varphi(j;x)$ for each instance x, we construct a feature relevance ranking by sorting these scores in descending order. Afterwards, we analyze the positions reached in the ranking by the true dimensions used to build each data instance. Table 3.4 indicates the position in the ranking under which, in average, it is possible to find a given percentage of the relevant dimensions. We use average percentage because each data instance has a different ranking of relevance. As an example, Table 3.4 shows that in Dataset 3, according to the ranking, on average 90% of the relevant features are among the first 10 dimensions with highest score.

In general results are variable, for example, for Dataset 2 all relevant dimensions appear among the first 18 positions of the ranking, while for Dataset 1 all relevant dimensions appear among the first 75 dimensions of the ranking. These positions correspond to approximately 4.5% and 37.5% of the total number of dimensions, respectively. The presence of irrelevant features at the top of rankings can be related to the work of (Guyon & Elisseeff, 2003) that unexpectedly shows how redundant features can improve classification.

TABLE 3.4. Relative relevance assigned by RMoE to features used to generate class patterns in synthetic datasets using the score in Equation (3.15). Each cell indicates the position in the ranking where, in average, it is possible to find a given percentage of the relevant dimensions (header of the respective column).

Dataset name	60%	70%	80%	90%	100%	Total Dimensions
Dataset 1	15	72	73	74	75	200
Dataset 2	12	13	16	17	18	400
Dataset 3	7	8	9	10	46	600
Dataset 4	10	25	30	31	33	800
Dataset 5	8	9	15	16	17	1000
Dataset 6	6	8	50	51	52	1200

3.5.2. Real datasets

We test the performance of RMoE using 13 real datasets. Table 3.5 describes the main characteristics of each of these datasets. Arrhythmia, Ionosphere, Musk-1, Secom, Semeion, Spectf, and Sonar datasets are taken from UCI Machine Learning Repository (Asuncion & Newman, 2007). Leukemia, Lymphoma, Colon, and Dataset-C are biological datasets taken from (Aguilar, 2008). BrainTumor is a biological dataset taken from (Statnikov, Tsamardinos, Dosbayev, & Aliferis, 2011). PIE10P is a face recognition dataset taken from (Liu, 2012). In the case of PIE10P and Leukemia datasets, we select the top 1000 and 1500 features, respectively, according to the Fisher score filter (Duda et al., 2001). We reduce the number of features in these two datasets to obtain a pool of datasets with a highly diverse number of features, as shown in Figure 3.2. Finally, in all cases we removed the variables with variance equal to zero.

We test RMoE using the same combinations of parameter values for λ_ν and λ_ω considered in the case of the synthetic datasets. Table 3.6 shows the accuracy of the best RMoE model obtained for each dataset (RMoE(λ_ν^*,λ_ω^*)).

Regarding results in Table 3.6, RMoE outperforms the traditional MoE technique in all the tested datasets. The increase in performance is variable and depends on each particular dataset. As expected, the advantages of RMoE with respect to MoE increases with the dimensionality of the dataset. This is the case of datasets such as Secom, PIE10P, Leukemia, Lymphoma, Colon, BrainTumor, and Dataset-C.

TABLE 3.5. Real datasets used for experiments.

Dataset name	#Objects	#Dimensions	#Classes
Ionosphere	351	33	2
Spectf	267	44	2
Sonar	208	61	2
Musk-1	486	168	2
Semeion	1593	256	10
Arrhythmia	452	279	16
Secom	1567	471	2
PIE10P	210	1000	10
Leukemia	75	1500	2
Colon	62	2001	2
Lymphoma	45	4027	2
BrainTumor	90	5921	5
Dataset-C	60	7130	2

TABLE 3.6. Accuracy on real datasets using 30 hold-out partitions. RMoE(λ^*_ν,λ^*_ω) indicates average (std. deviation) classification accuracy for best parameters configuration. The pair (λ'_ν,λ'_ω) show the median of the best parameters obtained by 3-fold cross-validation inside the training set of each hold-out partition (see main text for details).

Dataset name	MoE	RMoE(λ^*_ν,λ^*_ω)	(λ'_ν,λ'_ω)	RS	DT	AB
Ionosphere	82.7(3.0)	84.1(2.6)	(0.25,0.5)	**93.0 (1.6)**	87.8 (2.2)	91.3 (1.7)
Spectf	70.1(3.7)	76.6(3.4)	(10,20)	**80.2(1.5)**	74.7(3.9)	79.5(2.6)
Sonar	64.1(5.6)	74.1(4.2)	(2.25,2)	**79.1(4.0)**	70.8(4.9)	**79.1(3.6)**
Musk-1	67.2(4.9)	80.0(2.0)	(1,0.75)	**85.5(2.4)**	75.3(3.4)	82.1(2.7)
Semeion	66.1(2.3)	85.1(1.5)	(2.5,2)	**91.7(0.9)**	66.6(1.9)	60.9(2.4)
Arrhythmia	45.0(10.9)	66.0(2.2)	(1,2)	**69.8(1.9)**	65.2(3.2)	82.1(2.7)
Secom	59.4(7.3)	73.1(1.6)	(10,10)	**74.6(1.3)**	66.3(4.0)	71.9(2.4)
PIE10P	32.9(11.0)	**99.4(1.1)**	(4,2)	95.2(1.8)	78.8(5.1)	76.5(6.5)
Leukemia	59.0(13.1)	93.1(5.6)	(0.5,1)	**95.9(3.1)**	90.5(4.3)	80.8(16.3)
Colon	54.7(11.8)	**82.0(5.2)**	(1.5,1.5)	59.7(4.9)	57.7(8.2)	60.0(9.3)
Lymphoma	53.3(10.0)	**88.8(4.7)**	(3.25,3.5)	85.1(7.1)	72.2(9.5)	68.3(19.9)
BrainTumor	36.9(7.1)	**83.8(4.1)**	(1.5,2)	79.3(3.3)	65.7(5.2)	74.5(6.1)
Dataset-C	51.2(8.3)	**62.7(9.0)**	(1,1)	59.7(4.9)	57.7(8.2)	60.0(9.3)

To check if these results are statistically significant, we run a paired Student's t-test (Behrens-Fisher problem (Rice, 1994)) to compare the results of RMoE against the results of each of the alternative techniques. When comparing RMoE with respect to MoE, RMoE has greater accuracy than MoE with over 95% of confidence in all but one of the

datasets where the confidence is just 93% (Ionosphere). In terms of the rest of the alternative techniques and the six high-dimensional datasets under evaluation (over 500 dimensions), RMoE also shows superior performance with over 95% of confidence. Exceptions are for the case of Random Subspace (RS) technique where in the cases of Lymphoma and Dataset-C datasets, RMoE has better accuracy with a 83% and 82% of confidence, respectively, and in the case of Leukemia dataset, where RMoE is defeated by RS with 96% of confidence.

Figure 3.2 shows the accuracy achieved by the different methods considered in this work in function of the dimensionality of the dataset. We can observe that for datasets with low dimensionality ($<$ 500 dimensions) the performance of RMoE is slightly lower than classifiers such as Random Forest and Adaboost. However, in the case of datasets with high dimensionality, RMoE shows comparable and in most cases superior performance that the alternative techniques under evaluation. This confirms our intuition about the relevance of a suitable embedded feature selection scheme for the case of high dimensional data. A complementary advantage of our method is that it provides a sound probabilistic framework for classification.

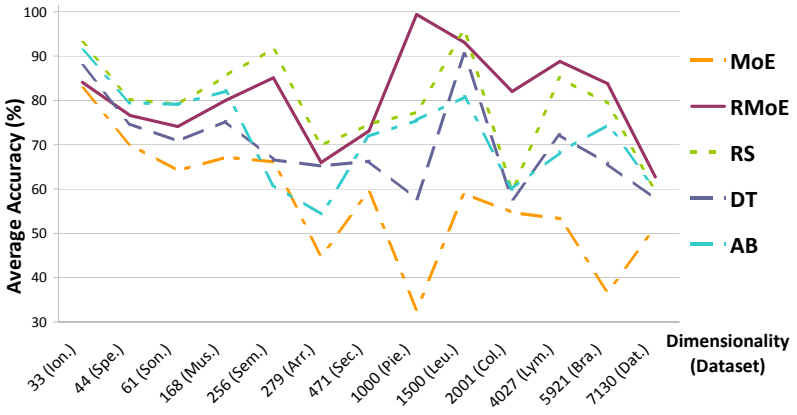


FIGURE 3.2. Average accuracy on real datasets for all tested algorithms with respect to the number of dimensions of the datasets.

Table 3.7 shows average parameter dimensionality as well as the percentage of feature reduction provided by RMoE with respect to MoE. As in the case of synthetic datasets, Table 3.7 shows that RMoE favors sparse solutions with a competitive or superior accuracy

than traditional MoE technique. In general, results are variable in terms of sparsity. For example, for datasets Colon, Lymphoma, BrainTumor, and Dataset-C, the best models provided by RMoE use less than 1% of the available dimensions. On the other hand, for the dataset Ionosphere, RMoE uses 78.1% of all dimensions. In general, when the dataset has few dimensions, the difference with MoE is less noticeable. Therefore, as expected, feature selection tends to be more useful when datasets have more dimensions.

TABLE 3.7. Average dimensionality of parameters in real datasets.

Dataset name	MoE	RMoE(λ_ν^*,λ_ω^*)	Features reduction
Ionosphere	32	25	21.9%
Spectf	43	5	88.4%
Sonar	60	17	71.7%
Musk-1	167	34	79.6%
Semeion	256	77	70.0%
Arrhythmia	279	18	93.5%
Secom	471	12	97.5%
PIE10P	1000	20	98.0%
Leukemia	1500	23	98.5%
Colon	2000	19	99.1%
Lymphoma	4026	13	99.7%
BrainTumor	5921	24	99.6%
Dataset-C	7129	14	99.8%

Now, we analyze execution times of MoE and RMoE. By considering n records and d dimensions with $d > n$, in the case of MoE each parameter (ν, ω) are found by weighted least squares optimization that usually is dominated by complexity $O(d^3)$, while in the case of RMoE such step depends of the method for solving Lasso optimization problem. In the case of RMoE, we use the iterative solution proposed by (Bradley, Kyrola, Bickson, & Guestrin, 2011). Using $\lambda_\nu = \lambda_\omega = 1$, Table 3.8 shows that RMoE is usually slower than MoE in the case of few dimensions, however, in high dimensional cases RMoE is able to run faster than MoE by taking advantage of the sparsity given by the Lasso optimization.

Finally we apply our technique to an object recognition problem. In particular, we use PASCAL dataset with 10 random categories. We apply dense SIFT method for obtaining a suitable representation of images which is used in the same way for all algorithms. We

TABLE 3.8. Average execution time (in miliseconds) of mixture-of-experts and regularized mixture-of-experts for different datasets using in each case 100 independent executions.

Dataset name	MoE	RMoE
Ionosphere	80	110
Spectf	40	310
Sonar	40	450
Musk-1	170	850
Semeion	2270	5130
Arrhythmia	920	2110
Secom	430	1550
PIE10P	1220	2700
Leukemia	3160	420
Colon	5160	2260
Lymphoma	2.6E4	0.1E4
BrainTumor	43.2E4	1.8E4
Dataset-C	13.5E4	0.4E4

compare our method against Support Machine Vector with linear kernel (SVM), Random Forest (RF), Nearest Neighbor (NN) and MoE. The parameters for RMoE were selected using cross-validation over training set. We obtain the results given in Table 3.9. In this experiment, RMoE shows better performance than its competitors. This confirms our intuition about the relevance of a suitable embedded feature selection scheme for the case of complex data, which is the case of object dataset.

TABLE 3.9. Average accuracy in subset of Pascal-2007 dataset. The methods compared with RMoE are standard classification algorithms: MoE, SVM, Random Forest and 1NN

Method name	Average accuracy (std)
RMoE	**83.7**%
MoE	82.9%
SVM	78.0%
RF	77.9%
1NN	75.0%

3.6. Conclusions

This chapter proposed RMoE, a regularized variant of mixture of experts, where local feature selection is performed on experts and gate function using $L1$ regularization. Our experiments provide evidence that the proposed technique improves classical mixture of experts in terms of accuracy and sparseness of the solution. In particular, using a diverse set of synthetic and real datasets, RMoE is able to find classification models that provide not only greater accuracy but also use less than 5% of the available features. In this respect, as expected, the proposed technique has demonstrated greater utility when the datasets have a large number of dimensions. In the case of data with few dimensions, there is no significant difference in comparison to classical MoE models. In terms of goodness of feature selection, in the case of synthetic datasets, where there is ground truth information about the process used to generate the data, the proposed method is able to recover most of the true relevant dimensions. In terms of the performance of RMoE with respect to popular alternative techniques that also uses embedded feature selection, we also observe that RMoE shows a superior performance for the case of datasets with a high number of dimensions. As future work, we believe that an important constraint of RMoE is the assumption that the conditional distributions by each expert has to be modeled by a logistic regressor. We plan to explore alternative and more flexible expressions to model gate and expert functions. Another avenue of future research is to explore the incorporation of an embedded feature selection scheme for the case of hierarchical mixture of experts.

References

Aguilar, J. (2008). *Dataset repository in arff.* http://www.upo.es/eps/aguilar/datasets.html.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository.* http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *Transactions on Neural Networks*, 537–550.

Bishop, C. (2007). *Pattern recognition and machine learning (information science and statistics)*. Springer.

Bishop, C., & Svensén, M. (2003). Bayesian hierarchical mixtures of experts. In *Conference on Uncertainty in Artificial Intelligence* (pp. 57–64).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Bradley, J., Kyrola, A., Bickson, D., & Guestrin, C. (2011). Parallel coordinate descent for l1-regularized loss minimization. In *International conference on machine learning* (pp. 321–328).

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

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1-38.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. Wiley-Interscience.

Ebrahimpour, R., & Jafarlou, F. M. (2010). View-independent face recognition with hierarchical mixture of experts using global eigenspaces. *Journal of Communication and Computer*, 1103–1107.

Freund, Y., & Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory* (p. 23-37).

Geng, B., Tao, D., T.Xu, Yang, L., & Hua, X. (2012). Ensemble manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 1227–1233.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Journal of Machine Learning*, 389–422.

Hall, M. (1999). *Correlation-based feature selection for machine learning*. Unpublished doctoral dissertation, University of Waikato.

Hampshire, J., & Waibel, A. (1992). The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *Pattern Analysis and Machine Intelligence*, 751–769.

Jacobs, R., Jordan, M., Nowlan, S., & Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 79–87.

Jordan, M., & Jacobs, R. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 181–214.

J.Vogdrup. (2010). Combining predictors:comparison of five meta machine learning methods. *Information Sciences*, *119*, 91–105.

Kam, T. (1998). The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence*, 832–844.

Khalili, A. (2010). New estimation and feature selection methods in mixture-of-experts models. *Canadian Journal of Statistics*, *38*, 519–539.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 273–324.

Lee, S., Lee, H., Abbeel, P., & Ng, A. (2006). Efficient L1 regularized logistic regression. In *National Conference on Artificial Intelligence (AAAI).*

Lima, C., Coelho, A., & Zuben, F. V. (2007). Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences*, *177*, 2049–2074.

Liu, H. (2012). *Arizona state university: Feature selection datasets.* http://featureselection.asu.edu/datasets.php.

Liu, H., & Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. In *International conference on tools with artificial intelligence* (pp. 388–391).

MacKay, D. (1995). Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 469–505.

Maldonado, S., Weber, R., & Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences*, *181*, 115–128.

Meeds, E., & Osindero, S. (2005). An alternative infinite mixture of gaussian process experts. In *Advances in neural information processing systems* (pp. 883–890).

Moerland, P. (1997). *Some methods for training mixtures of experts* (Tech. Rep.). IDIAP Research Institute.

Murthy, S., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 1–32.

Nanculef, R. ., Valle, C., Allende, H., & Moraga, C. (2012). Training regression ensembles by sequential target correction and resampling. *Information Sciences*, *195*, 154–174.

Nguyen, M., Abbass, H., & McKay, R. (2006). A novel mixture of experts model based on cooperative coevolution. *Neurocomputing*, 155–163.

Pan, W., & Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 1145–1164.

Pinto, N., Cox, D. D., & DiCarlo, J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology*, 151-156.

Quinlan, J. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

Rasmussen, C., & Ghahramani, Z. (2001). Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems* (pp. 881–888).

Rice, J. (1994). *Mathematical Statistics and Data Analysis*. Duxbury Press.

Saragih, J., Lucey, S., & Cohn, J. (2009). Deformable model fitting with a mixture of local experts. *International Conference on Computer Vision*, 2248–2255.

Scott, D., & Sain, S. (2004). Multidimensional density estimation. In *Handbook of statistics* (pp. 229–263). Elsevier.

Statnikov, A., Tsamardinos, I., Dosbayev, Y., & Aliferis, C. (2011). Gems: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics*, 491–503.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society (Series B)*, 267–288.

Titsias, M., & Likas, A. (2002). Mixture of experts classification using a hierarchical mixture model. *Neural Computation*, 2221–2244.

Tseng, P. (2001). Convergence of block coordinate descent method for nondifferentiable maximization. *Journal of Optimization Theory and Applications*, 475–494.

Ulas, A., Taner, O., & Alpaydin, E. (2012). Eigenclassifiers for combining correlated classifiers. *Information Sciences*, *187*, 109–120.

Van-Rijsbergen, C. (1979). *Information retrieval*. Butterworth-Heinemann.

Wang, M., Hua, X., Hong, R., Tang, J., Guo-Jung, J., & Song, Y. (2009). Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*, 733–746.

Wang, M., Li, H., Tao, D., Lu, K., & Wu, X. (2012). Multimodal graph-based reranking for web image search. *IEEE Transactions on Image Processing*, *21*, 4649–4661.

Wang, S., & Zhu, J. (2008). Variable selection for model-based high dimensional clustering and its application to microarray data. *Biometrics*, 440–448.

Wu, F., Yuan, Y., & Zhuang, Y. (2010). Heterogeneous feature selection by group lasso with logistic regression. *International Conference on Multimedia*, 983–986.

Xiao, J., He, C., Jiang, X., & Liu, D. (2010). A dynamic classifier ensemble selection approach for noise data. *Information Sciences*, *180*, 3402–3221.

Xu, L., Jordan, M., & Hinton, G. (1994). An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems* (pp. 633–640).

Yang, K., Wang, M., Hua, X., Yan, S., & Zhang, H. (2011). Assemble new object detector with few examples. *IEEE Transactions on Image Processing*, *20*, 3341–3349.

Yuille, A., & Geiger, D. (1998). Winner-take-all mechanisms. In *The Handbook of Brain Theory and Neural Networks.*

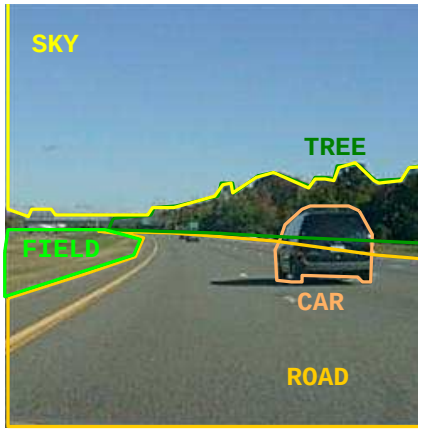
4. ADAPTIVE HIERARCHICAL CONTEXTS FOR OBJECT RECOGNITION WITH CONDITIONAL MIXTURE OF TREES

4.1. Introduction

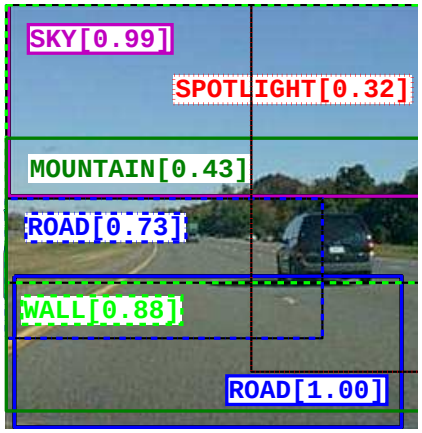
Lately, the synergistic combination of computer vision and machine learning techniques have been successfully applied to the problem of automatic visual recognition (Viola & Jones, 2001) (Fergus, Perona, & Zisserman, 2003) (Fei-Fei, 2005) (Felzenszwalb, McAllester, & Ramanan, 2008). In particular, contextual information has emerged as an attractive option to boost the performance of single object detectors (Galleguillos & Belongie, 2010)(Choi, Lim, Torralba, & Willsky, 2010)(Desai, Ramanan, & Fowlkes, 2011).

Context based methods can be divided into two groups: global and local context methods (Galleguillos & Belongie, 2010). Regarding global or holistic context methods, most works exploit whole scene statistics to perform recognition. In (Ulrich & Nourbakhsh, 2000), Ulrich and Nourbakhsh introduce color histograms as the holistic representation of an image that is used by a K-nearest neighbors scheme to classify scenes. In (Torralba, 2003), Torralba proposes an image representation based on global features that represent dimensions in a space that they call spatial envelope. In (Chang, Goh, Sychay, & Wu, 2003), Chang et al. use low-level global features that are used to estimate a belief or confidence function over scene labels.

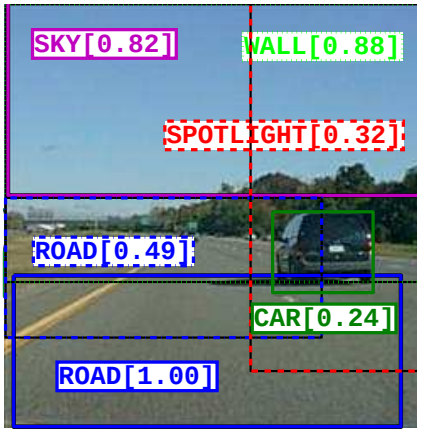
Regarding local context techniques, contextual information is derived from specific blocks or localized areas around object positions. Sinha and Torralba (Sinha & Torralba, 2002) improve face detection using local contextual regions. Torralba et al. (Torralba, Murphy, & Freeman, 2005) introduce a Boosting approach in combination with a Conditional Random Field (CRF) to recognize objects. They apply their method to recognize objects and structures in office and street scenes. Shotton et al. (Shotton, Winn, Rother, & Criminisi, 2007) combine layouts of textures and context to recognize objects. They use a CRF to learn a model of objects and a boosting algorithm to combine the texture information and the object model. Galleguillos et al. (Galleguillos & Belongie, 2010) present a critical review of different contextual cues and machine learning models commonly used to improve



(a) Ground-Truth Image



(b) Results with Single Tree (Choi et al.,2010)



(c) Results with Mixture of trees

FIGURE 4.1. An example of the results of our method with respect to a state-of-the-art model (Choi et al.,2010). We show the top six most confident detections for each model using the same underlying single object detector (Felzenszwalb et al, 2008). In the case of 4.1(b), the model uses fixed contextual relations that do not detect the car object and produce a wrong detection of a mountain object. In contrast, our model 4.1(c) uses global scene information to adaptively select a suitable component of a mixture of trees that embeds particular contextual relations that provide a correct detection of the car object and do not detect a phantom mountain object.

object categorization. Rabinovich et al. (Rabinovich, Vedaldi, Galleguillos, Wiewiora, & Belongie, 2007) show that textual data from the web is a useful source to estimate co-ocurrence between objects. Choi et al. (Choi et al., 2010) presents an efficient scheme to model inter-object relations using a tree-structured Bayesian network. Recently, (Desai et al., 2011) shows a technique that is able to model contextual cueing, spatial co-ocurrence,

and inhibitory intra-class constraints among objects using a max-margin approach. In all these cases, contextual relations among objects are fixed and do not depend of the type of scene being analyzed.

We believe that using an adaptive scheme to model contextual relations among objects can boost the performance of current object recognition techniques. We can illustrate this idea by the following example. Consider the case of the contextual relation between the presence of a person and a dog objects. Under a park scene, person and dog objects co-occur frequently, but in an office scene, they hardly co-occur, therefore modeling such relation with a fixed contextual constraint limits the flexibility of the model to fit real data. Moreover, in terms of a probabilistic graphical model (PGM) representation for object relations, such as (Choi et al., 2010), the relevance of the information provided by each type of object can change dramatically for different types of scenes. For example, in an office scene a computer monitor is commonly a highly informative object, therefore under a suitable PGM representation it should have strongly related children objects. In contrast, in situations like a living room scene, a monitor is usually not very informative, therefore under a suitable PGM representation its related children structure is not very relevant.

In the previous cases, the flexibility of a mixture model able to provide adaptive contextual relations among objects can be a useful tool to boost object recognition performance. Consequently, in this work, we present AH-MoT, a method that learns adaptive conditional relationships among objects according to the scene information. In particular, we achieve this by introducing a PGM based on a conditional mixture of trees (Meila & Jordan, 2001). Next, we provide first background information relevant to our model. Afterwards, we describe the details of our approach. Finally, we present the results of our experiments showing the advantages of our approach.

4.2. Hierarchical context

In this section, we summarize the work by Choi et al. (Choi et al., 2010) that is the baseline algorithm considered in our work. The model in (Choi et al., 2010) is composed

of a prior and a measurement model. Next we provide details about prior and measurement models. Note that we use b_i to refer to a specific object class i, while we use B to refer to a generic object class.

4.2.1. Prior Model

The prior model uses a binary tree structured PGM to represent co-ocurrence and spatial relationships among object categories. Nodes in this tree are given by variables indicating object presence, as well as, its location and scale (see (Choi et al., 2010) for details). Specifically:

(i) **Object presence**: $b_i \in \{0, 1\}$ corresponds to the presence of an object of class i.

(ii) **Object location and scale**: L_i corresponds to the location and scale of object instance b_i. $L = \{L_i, \ldots, L_N\}$ resumes all object classes. L is modelled as dependant of the presence of objects b: $p(L|b) = p(L_{root}|b_{root}) \prod_i p(L_i|L_{pa(i)}, b_i, b_{pa(i)})$, where L_i is the median of the location and scale for all instances of object i and is composed by $(L_y^i, \log L_z^i)$. L_y^i is the median of vertical positions for object i and L_z^i is the median of scales for object i. Medians are computed using all training images. The use of a logarithm for scales and the omission of horizontal positions is justified in (Choi et al., 2010).

4.2.2. Measurement Model

The measurement model predicts the presence of an object category b_i in an image by using global gist features and outputs of object detectors. Figure 4.2(a) shows the PGM that relates the variables considered in the measurement model. Specifically:

(i) **Correct detections**: $c_{ik} \in \{0, 1\}$ represents the k-th detection of instances of object category i, being 1 if the detection is a true positive and 0 otherwise. Correct detections depend on object presence, where $p(c_{ik} = 1|b_i = 1)$ corresponds to the frequency of correct detections in the training set and $p(c_{ik} = 1|b_i = 0) = 0$.

(ii) **Classifier scores**: $s_{ik} \in \Re$ represents classifiers scores, which according to Figure 1 depends on correct detections c_{ik}. Using Bayes rule, $p(s_{ik}|c_{ik}) = p(c_{ik}|s_{ik})p(s_{ik})/p(c_{ik})$. Here a logistic regression is used to model $p(c_{ik}|s_{ik})$.

(iii) **Detection window location**: $w_{ik} = (L_y^{ik}, \log L_z^{ik})$ represents the location of a detection window, where L_y^{ik} and L_z^{ik} are vertical location and scale of the window corresponding to the k-th detection of an instance of object category i. Location is modeled as a Gaussian distribution and it depends on c_{ik} and the median location L_i of instances of object class i. If a window is a correct detection then w_{ik} is modeled as $p(w_{ik}|c_{ik} = 1, L_i) = Gaussian(w_{ik}; L_i, \Lambda_i)$, where Λ_i is the covariance around the predicted location. If a window is a false positive then w_{ik} does not depend on L_i and it is modeled with a uniform distribution.

(iv) **Gist**: Gist features g_L (Torralba, 2003) are used to related global image features to object presence by estimating $(g_L|b_i$. To deal with the high dimensionality of the gist vector g_L, a logistic regression is used to estimate $p(b_i|g)$, then likelihoods $p(g_L|b_i)$ are estimated indirectly using $p(g|b_i) = p(b_i|g)p(g)/p(b_i)$.

Following the notation in (Choi et al., 2010), from here on we use variables without subindexes to denote the set of variables related to individual object class detections in a image. For example, $b = \{b_1, \ldots, b_N\}$ denotes the binary values of all variables related to the detection of the D possible object classes. In the same way, W resumes all candidate detection windows variables w_{ik} in a given image.

4.3. Adaptive Hierarchical Mixture of Trees (AH-MoT)

As mentioned earlier, an important limitation of the method by Choi et al. (Choi et al., 2010) is that it assumes a fixed contextual relationship among objects. In this work, we avoid this limitation by incorporating in the model adaptive contextual relationships between objects that depend on a estimation of the current scene type. Our main intuition is that contextual object-to-object co-occurrences strongly depend on the underlying scene, as shown in the person and dog example mentioned before. In particular, we propose to

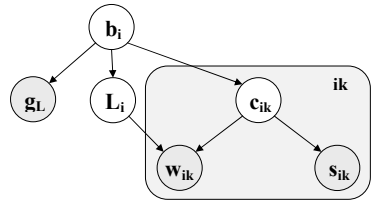
modify the fixed single tree co-occurrence model by Choi et al., using a model based on a mixture of trees that we call AH-MoT. This mixture of trees incorporates scene information to adaptively represent different possible contextual relations between objects. In terms of the original PGM considered (Choi et al., 2010), our main modification is the incorporation of a latent variable representing the underlying scene type. This latent variable depends on the output of a Gist feature (Torralba, 2003). Figure 4.2(b) shows the resulting modified PGM after adding the new latent variable. Also global features x_G are added to the PGM, as observation to infer the scene type.

Our mixture of trees context model, AH-MoT, is built by a conditional mixture of tree-structured Bayesian networks, each of which is an expert in some partition of the set of images. These networks have a weight that depends on the global scene information (given by the Gist feature). The model can be seen as a mixture of experts where the gate function is given by a function of the global representation, and each expert function is given by an individual Bayesian networks. We stress that our main contribution is the joint modeling of the dependence between the ensemble of trees and a global representation of the image data. Next, we provide details of the proposed conditional mixture of trees model and how we conduct estimation and inference with this model.

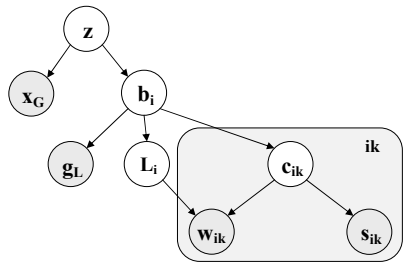
4.3.1. Conditional mixture of trees

In order to incorporate global scene information in AH-MoT, we model object presence as dependent on scene type. In our current implementation, we infer scene type using a Gist feature vector x_G. In this sense, in our PGM $x_G = g_L$, however, other global features can be used to estimate scene type, therefore we consider x_G and g_L as separate variables in the PGM shown in Figure 4.2(b).

For a training set of N images, we can construct N instance-label pairs (x_G, b), where, as stated before, $b \in \{0, 1\}^D$ represents the potential presence of the D possible object categories in a given image. Our goal is to use instances (x_G, b) to include in our model different types of contextual relations between object classes. We achieve this goal by introducing latent variable z which simplifies the analysis of the model. We refer to this



(a) A single tree model



(b) Adaptive hierachical mixture of trees (AH-MoT)

FIGURE 4.2. Modification of contextual objects relationships: (a) The model by Choi et al. (b) Proposed model, AH-MoT, that incorporates global scene information as a root element that influences object-to-object relationships.

latent variable as the *context variable*. We assume that there are K possible values for z, i.e., we assume the existence of K *contexts* for object classes. This is similar to a mixture of experts model with the exception that in our case b is conditionally independent of x_G given z. The context variable is assumed as a winner-take-all variable, i.e., each object class detection occurs under a specific contextual scenario.

Considering K contexts and using a Gaussian Kernel for the weighting of experts, we can model the conditional density $p(b|x_G)$ (Xu, Jordan, & Hinton, 1994) as:

$$p(b|x_G) \;=\; \sum_{i=1}^{K} p(b, z_i|x_G) = \sum_{i=1}^{K} p(b|z_i, x_G)\, p(z_i|x_G) = \sum_{i=1}^{K} p(b|z_i)\, p(z_i|x_G) \quad (4.1)$$

Here, we have K contexts represented by z_i, with $i = \{1, ..., K\}$, where each context has its own class-conditional probability function.

We specify the two components of the mixture model given by Equation 4.1 as follows:

- ***Context gate:*** given by $p(z_i|x_G)$, represents the influence of each local context. It represents an estimate of the likelihood of selecting each of the K experts for the input x_G. The gate function has K components, one for each expert.
- ***Tree experts:*** given by $p(b|z_i)$ represents the class-conditional local models. It represents an estimate of the probability of appearances b given the expert z_i for input x_G. There are K context functions. In this case, we use a Bayesian Network model following Choi et al (Choi et al., 2010).

The proposed model is similar to the mixture of trees model presented by Meila and Jordan (Meila & Jordan, 2001). A mixture of trees model represents the distribution of a convex sum of K tree components over a random variable x as: $Q(x) = \sum_{k=1}^{K} \lambda_k T^k(x)$ with $\lambda_k \geq 0$ and $\sum_{k=1}^{K} \lambda_k = 1$. Tree distributions $T^k(x)$ are the mixture components and coefficients λ_k are the mixture proportions. This model can be viewed as containing a latent variable z that with probability λ_k selects mixture component k. Therefore, conditioned on the value of z, the distribution of mixture Q is represented by a single tree. An advantage of this model is its flexibility because the trees may have different structures and parameters. Nonetheless, the main difference with our work is that while in (Meila & Jordan, 2001) they assume the weight of each component as fixed, we model these weights as variable. In particular, in the proposed model these weights depends on the global representation of a given image using the context gates.

Regarding context gate function, we use normalized Gaussian Kernels (Xu et al., 1994). This function can be interpreted as a simple mixture model. In this case, $p(z_i|x) = \frac{\alpha_i P_i(x)}{\sum_j \alpha_j P_j(x)}$, where each P_i is a Gaussian probability density functions with weights α_i, $\sum_j \alpha_j = 1$ and $\alpha_i \geq 0$.

In the case of the tree expert function, we use a Bayesian Network function similar to Meila and Jordan (Meila & Jordan, 2001). Following their work, we parameterize a tree with a graph $G = (V; E)$, where V is the vertex set and E is the set of edges. Assuming a set of K trees, V_i represents the vertex set of tree i, where $i \in \{1, \ldots, K\}$. In the case of our adaptive contextual model, the probability distribution of variable b conditioned on

the context variable z_i, $p(b|z_i)$, is represented by $T^i(b)$. Each tree models this component as $T^i(b) = \prod_{v \in V^i} T_{v|pa(v)}(b_v|b_{pa(v)})$, where $T_{v|pa(v)}(b_v|b_{pa(v)})$ is an arbitrary conditional distribution. Variable $pa(v)$ represents the parents of variable v inside the tree.

In order to find optimal parameter values for tree experts and the gate function, we use the Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). To obtain a one-pass solution for the gating function, instead of the conditional log-likelihood for the complete data, we use the joint log-likelihood (Xu et al., 1994). Assuming that the posterior probabilities of context gates or *responsabilities* R_{in} for each expert i and training instance n are known, we can apply the EM algorithm over the expected log-likelihood:

$$\langle \boldsymbol{L}_c \rangle = \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in} \, log \, (\, p(b_n|z_i) \, P_i(x_n) \, \alpha_i) \tag{4.2}$$

The expectation step of EM is given by the calculation of the posterior probability of context gate i, which is given by:

$$R_{in} = p(z_i|x_n, b_n) = \frac{p(b_n|z_i) \, p(z_i|x_n)}{\sum_{j=1}^{K} p(b_n|z_j) \, p(z_j|x_n)} \tag{4.3}$$

The maximization step of EM is given by the maximization with respect to each parameter. We can observe two decoupled components in the expected log-likelihood:

$$\boldsymbol{E}_{expert} = \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in} \, \left[log \, T^i(b_n) \right], \; \boldsymbol{E}_{gate} = \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in} \, \left[log \, (\alpha_i P_i(x_n)) \right] \tag{4.4}$$

In the case of the tree expert component, we must minimize the negative cross-entropy between R and T. Following the work of Meila and Jordan (Meila & Jordan, 2001), this problem is solved using a weighted version of the Chow-Liu algorithm. This component requires K runs of the Chow-Liu algorithm, where R_{in} is the normalized posterior probability obtained in the E-step.

In order to find the parameters of the gating function, α_i, μ_i and σ_i (being μ_i and σ_i the parameters of the Gaussian representation), we use Equation 4.2, obtaining:

$$\alpha_i = \frac{1}{N}\sum_{n=1}^{N} R_{in}, \quad \mu_i = \frac{\sum_{n=1}^{N} R_{in}\, x_n}{\sum_{n=1}^{N} R_{in}}, \quad \sigma_i = \frac{1}{d}\frac{\sum_{n=1}^{N} R_{in}\, \|x_n - \mu_i\|^2}{\sum_{n=1}^{N} R_{in}} \tag{4.5}$$

Algorithm 4 Conditional Mixture of Context Trees

```
while Not convergence do
    Compute responsabilities R according to Equation 4.3
    for i = 1 → K do
        Estimate α,μ and σ according to Equation 4.5 and T with a weighted Chow Liu
        algorithm according to (Meila & Jordan, 2001).
    end for
end while
```

The operation of the EM algorithm for a conditional mixture of trees is summarized in Algorithm 4. We initialize the gates using a variant of the K-means algorithm that clusters variables b over the training set using Hamming distance. The resulting conditional mixture of context model follows the intuition that general context is naturally divided into many component contexts, thus, we can make inference on each tree and then combine the outputs using the gating function.

4.3.2. Inference

Inference in AH-MoT is straightforward, as we separate each tree in its own partition. Similarly to (Choi et al., 2010), we make inference using message passing algorithms for each tree ($p(b, c, L/g, W, s, z)$) (Pearl, 1982). Afterwards, similarly to (Meila & Jordan, 2001), we obtain the final score by combining the scores of each component with its respective parameters.

$$\hat{b}, \hat{c}, \hat{L} \;=\; argmax_{b,c,L} \sum_{z} p(z) * p(b, c, L/g, W, s, z) \tag{4.6}$$

Following Choi et al. (Choi et al., 2010), we use an iterative procedure. First, we make inference without considering the locations ($\hat{b_0}, \hat{c_0} \propto p(b, c|g, s)$), then we infer the locations ($\hat{L} \propto argmax_L p(L|\hat{b_0}, \hat{c_0}, W)$), and finally we infer the object presence ($\hat{b}, \hat{c} \propto p(b, c|s, g, \hat{L}, W)$) considering the previous inferred location. The last step is equivalent to sampling from a binary tree with node and edge potentials modified by $p(\hat{L}, W/b, c)$.

4.4. Experiments

In this section, we perform an empirical evaluation of the proposed approach considering two real datasets: (i) OUTDOOR dataset created by Oliva and Torralba (Oliva & Torralba, 2001), and (ii) SUN09 dataset created by Choi (Choi et al., 2010). OUTDOOR dataset has 2600 images and includes 8 outdoor scene categories, such as coast, mountain, forest, etc. We randomly divide the dataset into two sets of approximately equal size, one for training and one for testing. Similarly to Choi et al. (Choi et al., 2010), we prune object categories by considering only those that have at least 3 true detections in the training set. As a result, for OUTDOOR dataset we have 21 object categories with equal-sized training and test sets. In the case of SUN09 dataset, as in (Choi et al., 2010), we prune the dataset considering only object categories with at least 4 true detections in the training dataset. As a result, for SUN09 dataset we have 111 object categories, 4367 training images, and 4317 test images.

In general, in both datasets object detections are highly challenging, including a variety of poses, scales, rotations, and scene types. We use the object detector proposed by Felzenszwalb et al. (Felzenszwalb et al., 2008), which is based on the mixture of multi-scale deformable parts model and a latent SVM approach. We use the same object detector models for both datasets. In average, this detector outputs approximately 5 detections per category in each image. In both datasets, for each image we consider the top 10 detections. We use the average precision-recall (APR) (Davis & M.Goadrich, 2006) as a performance metric for our model. This metric corresponds to the area under the precision-recall curve.

Table 4.1 shows APR for both datasets: OUTDOOR and SUN09. We show the resulting APR for: i) Direct object detections provided by the underlying object class detector (Felzenszwalb et al., 2008) (*Object detector*), ii) Choi et al. (Choi et al., 2010) method based on hierarchical context (*Single Tree*), iii) AH-MoT using different number of trees (*AH-MoT*(X), where X is the number of used trees). Relative improvement in APR with respect to Choi et al. is shown in parenthesis.

Method	**OUTDOOR**	**SUN09**
Object detector	14.02 (-6.5%)	6.82 (-13.2%)
Single Tree	15.00 (0.0%)	7.87 (0.0%)
AH-MoT(2)	15.07 (0.5%)	7.98 (1.5%)
AH-MoT(3)	14.87 (-0.9%)	8.09 (2.9%)
AH-MoT(4)	15.12 (0.8%)	8.06 (2.5%)
AH-MoT(5)	15.25 (1.7%)	8.03 (2.2%)
AH-MoT(6)	**15.83 (5.5%)**	**8.31 (5.7%)**
AH-MoT(7)	14.84 (-1.1%)	7.88 (0.3%)

TABLE 4.1. APR for OUTDOOR and SUN09 databases provided by the tested methods. Relative improvement with respect to Choi et al. is shown in parenthesis.

Analyzing Table 4.1, in OUTDOOR database we note that performance of AH-MoT increases as the number of trees grows up to 6. After that, APR decays. In general, results improve the performance of Choi et al.(Choi et al., 2010). Considering the best number of trees in this dataset (six), relative improvement is 5.5%. In the case of SUN09 database, the improvement with respect to Choi et al. in terms of APR is 5.7% for the best number of trees (also six). It is important to note that in this case adding additional trees does not necessarily improve performance. In terms of individual classes, we found for the case of six trees that in OUTDOOR and SUN09 dataset, the APR increases for 10 and 53 object classes and decreases for 6 and 34 classes objects, respectively.

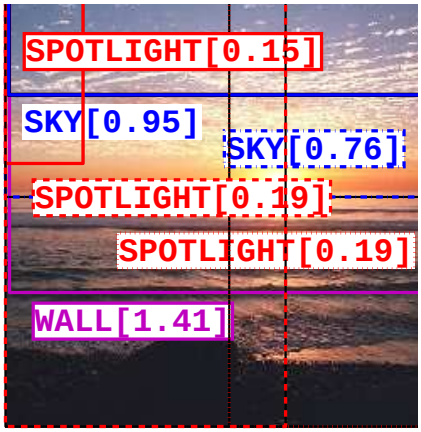
Figure 4.3 shows example detections of the top six most confident detectors from our model and Choi et al.(Choi et al., 2010). For example in Figure 4.3(b) AH-MoT correctly detect a sea object that is not detected by the single tree model in Figure 4.3(a). In Figure 4.3(d) a car object is recovered and a streetlight object is discarded in relation to Figure 4.3(c).

Figure 4.4 presents examples of resulting trees of AH-MoT for OUTDOOR database. The value over each edge represents the strength of dependency relation between each pair of object classes. Following (Choi et al., 2010), these dependencies are calculated using the magnitude of the mutual information between each pair of object classes, while the sign is positive if $p(b_i = 1, b_j = 1 > p(b_i = 1)p(b_j = 1)$. This scheme is used in (Choi et al., 2010). Tree A shows relationships mainly for mountain and rural highways scenes, while tree B shows relationships associated to street scenes. As an example of variable relationships between objects, we observe the correlation between the objects road and sign. Both objects are connected in both trees A and B, however, in tree B the dependence of both variables (0.71) is considerably higher than in tree A (0.01). This reflects the fact that these two objects relationship are more important in streets scenes than in other cases.

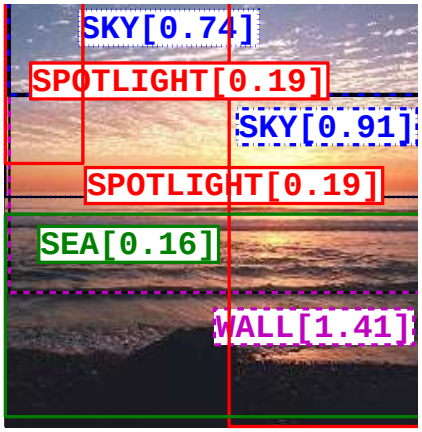
Figure 4.5 shows in a grid the dependences between objects for the single tree model and for two trees of AH-MoT with six trees. In order to facilitate the analysis of dependence between objects, we only show the values related to the eleven most frequent objects. For example, we can evaluate an image of rural highway scene. If we inspect the tree of the single tree model 4.5(a), we see that the objects tree and road are weakly correlated. On contrast, in tree A of the conditional mixture tree model 4.5(b), both objects appear as strongly correlated.

4.5. Conclusions

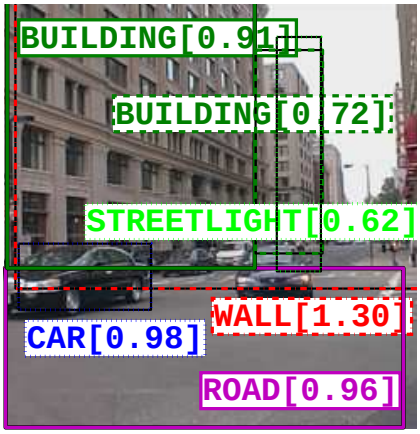
In this chapter, we proposed AH-MoT, an adaptive context model that uses a conditional mixture of trees to overcome relevant limitations of a fixed tree context model. Our experiments using standard object datasets indicate that the proposed model improves object recognition performance with respect to a single tree model, as it considers underlying scene information that influences object-to-object relationships. As future work, we plan to enhance our model using more powerful features for the gating function. Finally, we also plan to include adaptive policies to control the execution of object classifiers, similar to the method proposed in (Espinace, Kollar, Soto, & Roy, 2010).



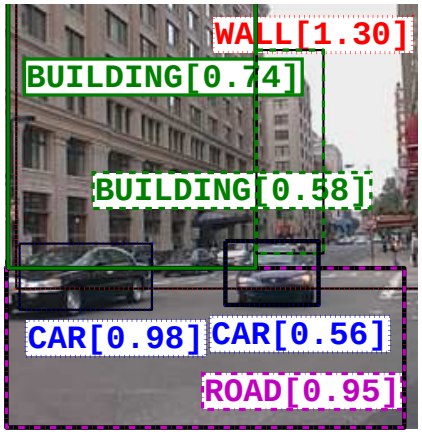
(a) Image **A**:S. Tree



(b) Image **A**:AH-MoT(6)

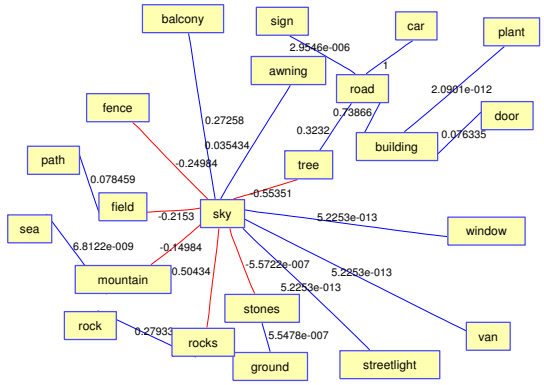


(c) Image **B**:S. Tree

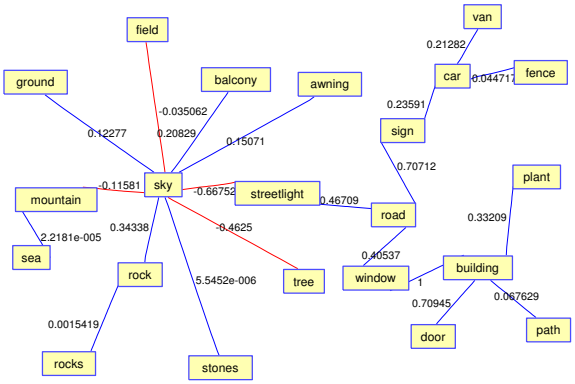


(d) Image **B**:AH-MoT(6)

FIGURE 4.3. Some detections considering a single tree model (Choi et al., 2010) and AH-MoT. Our model, AH-MoT, usually provides better detections than a single tree model.

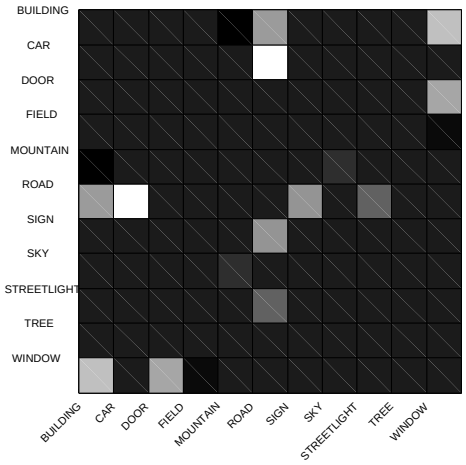


(a) Tree **A**: related to mountain scenes.

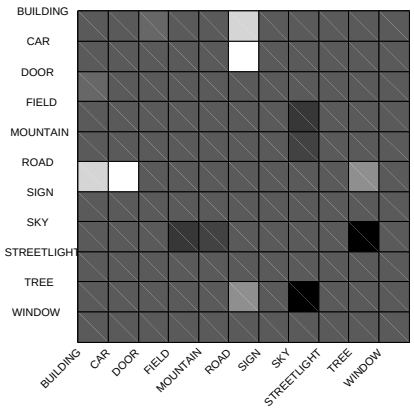


(b) Tree **B**: related to street scenes.

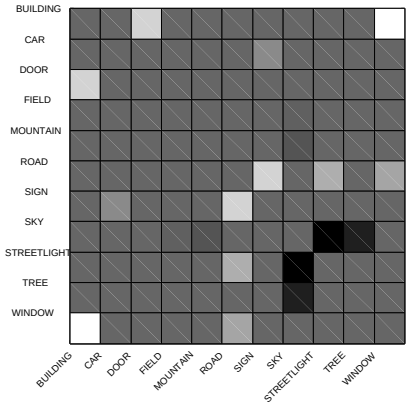
FIGURE 4.4. Examples of component trees for AH-MoT with six trees in the OUTDOOR dataset. Positive and negative correlations are indicated respectively with blue and red lines.



(a) Single Tree



(b) Tree **A** of Mixture of 6 Trees



(c) Tree **B** of Mixture of 6 Trees

FIGURE 4.5. Dependence of the top 11 more frequent objects. Figure 4.5(a) shows the relationships for a single tree. Figures 4.5(b) and 4.5(c) show two relationships in AH-MoT with 6 trees.

References

Chang, E., Goh, K., Sychay, G., & Wu, G. (2003). Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *Transactions on Circuits and Systems for Video Technology*, 26–38.

Choi, M., Lim, J., Torralba, A., & Willsky, A. (2010). Exploiting hierarchical context on a large database of object categories. In *Conference on Computer Vision and Pattern Recognition (cvpr)* (pp. 129–136).

Davis, J., & M.Goadrich. (2006). The relationship between precision-recall and roc curves. In *International Conference on Machine Learning* (pp. 233–240).

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 1-38.

Desai, C., Ramanan, D., & Fowlkes, C. (2011). Discriminative models for multi-class object layout. *International Journal of Computer Vision*.

Espinace, P., Kollar, T., Soto, A., & Roy, N. (2010). Indoor scene recognition through object detection using adaptive objects search. In *European Conference on Computer Vision (ECCV), Workshop on Robotics for Cognitive Tasks.*

Fei-Fei, L. (2005). A bayesian hierarchical model for learning natural scene categories. In *Conference on Computer Vision and Pattern Recognition (cvpr)* (pp. 524–531).

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Conference on Computer Vision and Pattern Recognition (cvpr).*

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Proceedings of ieee Conference on Computer Vision and Pattern Recognition (cvpr)* (pp. 264–271).

Galleguillos, C., & Belongie, S. (2010). Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, *114*(6), 712 - 722.

Meila, M., & Jordan, M. (2001). Learning with mixtures of trees. *Journal of Machine Learning.*, 1–48.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 145–175.

Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *American Association of Artificial Intelligence National Conference on AI* (pp. 133–136).

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *International Conference on Computer Vision* (p. 1-8).

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2007). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 2–23.

Sinha, P., & Torralba, A. (2002). Detecting faces in impoverished images. *Journal of Vision*.

Torralba, A. (2003). Contextual priming for object detection. *Proceedings of International Journal of Computer Vision (IJCV)*, *53*, 169–191.

Torralba, A., Murphy, K., & Freeman, W. (2005). Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems* (pp. 1401–1408).

Ulrich, I., & Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *International Conference on Robotics and Automation* (pp. 1023–1029).

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Conference on Computer Vision and Pattern Recognition*, 511–518.

Xu, L., Jordan, M., & Hinton, G. (1994). An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems* (pp. 633–640).

5. CONCLUSIONS AND FUTURE RESEARCH

5.1. Conclusions

This thesis presents three different methods oriented to improve specific stages of object recognition process. These methods exploit information of labeled data in codebook generation, selecting relevant features for local classifiers, and adapting contextual cues according to scene type.

First, a supervised clustering algorithm, LK-Means, is presented. In this thesis is mainly used to build codebooks for category-based object recognition tasks, but it can also be used in other situation as a generic supervised clustering technique. LK-Means is a variant of the classical K-Means clustering algorithm but it considers the information of labels using a convex combination of a class dependent and a non-class dependent cost function. By jointly using unsupervised and supervised information, LK-Means is able to outperform standard methods for codebook generation in the majority of tested real datasets considering adjusted mutual information. Assuming the number of classes as constant, the complexity of LK-Means is similar to K-Means.

Afterwards, a variant of mixture-of-experts with embedded local feature selection, RMoE, is introduced and it is applied to several datasets. RMoE is based on the classical mixture-of-experts technique, but it also includes a $L1$ norm penalty term in order to select relevant features. The training is based on an Expectation-Maximization scheme. By selecting local relevant features, our method is able to outperform standard machine learning algorithms in terms of accuracy when the datasets have high dimensionality. It also has the advantage of proving sparse solution which can facilitate the understanding of the classifiers. We also show that by embedding a local feature selection scheme we can obtain a notable increment in performance with respect to classical mixture-of-experts. Assuming a constant number of iterations in Lasso optimization, the computational complexity of RMoE is similar to MoE.

Finally, AH-MoT, a context-based object recognition method based on the exploitation of latent dependence between scene and inter-object co-occurrence is presented. In particular, AH-MoT uses a conditional mixture of trees for modeling contextual relations among objects according to scene type. The training of this model is performed using an Expectation-Maximization algorithm. Tests using real object datasets show that AH-MoT is able to outperform a state-of-the-art technique for context-based object recognition. Assuming a constant number of trees, the computational complexity of AH-Mot is similar to the method based on a fixed tree.

The three methods presented in this thesis provide useful insights about key steps of the typical object recognition pipeline. The first insight is that the use of supervised information can lead to the generation of more suitable codebooks to support object recognition tasks, as we observe in our experiments using LK-Means. Another relevant observation is that adaptability is useful to deal with complex object recognition cases. In particular, RMoE adaptively selects features according to available data using a gate function and a set of conditional mixture of trees. We also show that adaptability is also a key element when using contextual information, as it is shown by AH-MoT which is able to outperform current technique for object recognition based on fixed contextual schemes.

In summary, we have shown several advantages of the proposed methods with respect to alternative state-of-the-art approaches. In particular, we have shown that flexible adaptive models can improve key stages of the typical object recognition pipeline. Each of the proposed methods can be improved and, therefore, they open several avenues of further research.

5.2. Future Research Topics

Each chapter includes some avenues of future research for the techniques presented in this thesis. Most relevant future work is focused on increasing the quality, robustness, and efficiency of each particular method.

In the case of the first method, we plan to explore the incorporation of feature selection schemes, similarly to subspace clustering techniques. Also, we plan to add a covariance term for modeling more complex codeword representations.

In the case of the second method, we will investigate another functions inside the local expert functions. The logistic regressor used in our current implementation is a suitable general classifier, however, it is also worth to test alternative techniques such a maximum margin classifiers. We pretend to experiment with this type of technique inside our model.

Finally, in terms of the third method, we plan to face computational issues related to the number of object classes available for inference. Currently, our method has the drawback that it needs the score of all object detectors. Unfortunately, this strategy does not scale properly in terms of the number of possible object classes. An alternative strategy is to use our mixture of trees framework to adaptively select only the most informative object detectors. We plan to explore this issue as part of our future work.

References

Agarwal, S., Awan, A., & Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *Pattern Analysis and Machine Intelligence*, 1475 -1490.

Aguilar, J. (2008). *Dataset repository in arff.* http://www.upo.es/eps/aguilar/datasets.html.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html.*

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. In *International Conference of Machine Learning* (pp. 11–18).

Basu, S., Bilenko, M., & Mooney, R. (2003). Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In *Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining* (p. 29-42).

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *Transactions on Neural Networks*, 537–550.

Bishop, C. (2007). *Pattern recognition and machine learning (information science and statistics)*. Springer.

Bishop, C., & Svensén, M. (2003). Bayesian hierarchical mixtures of experts. In *Conference on Uncertainty in Artificial Intelligence* (pp. 57–64).

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Computational Learning Theory* (pp. 92–100).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Bradley, J., Kyrola, A., Bickson, D., & Guestrin, C. (2011). Parallel coordinate descent for l1-regularized loss minimization. In *International conference on machine learning* (pp. 321–328).

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

Chang, E., Goh, K., Sychay, G., & Wu, G. (2003). Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines. *Transactions on Circuits and Systems for Video Technology*, 26–38.

Choi, M., Lim, J., Torralba, A., & Willsky, A. (2010). Exploiting hierarchical context on a large database of object categories. In *Conference on Computer Vision and Pattern Recognition (cvpr)* (pp. 129–136).

Cohn, D., Caruana, R., & McCallum, A. (2003). *Semi-supervised clustering with user feedback* (Tech. Rep. No. TR2003-1892). Cornell University.

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, 1–22.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition* (pp. 886–893).

Davis, J., & M.Goadrich. (2006). The relationship between precision-recall and roc curves. In *International Conference on Machine Learning* (pp. 233–240).

Deelers, S., & Auwatanamongkol, A. (2007). Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. *International Journal of Electrical and Computer Engineering*, 247–252.

Demiriz, A., Benett, K., & Embrechts, M. (1999). Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering Conference* (p. 809-814).

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1-38.

Desai, C., Ramanan, D., & Fowlkes, C. (2011). Discriminative models for multi-class object layout. *International Journal of Computer Vision*.

Dorkó, G., & Schmid, C. (n.d.). *Object class recognition using discriminative local features* (Tech. Rep. No. RR-5497). INRIA - Rhone-Alpes.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. Wiley-Interscience.

Ebrahimpour, R., & Jafarlou, F. M. (2010). View-independent face recognition with hierarchical mixture of experts using global eigenspaces. *Journal of Communication and Computer*, 1103–1107.

Eick, C., Zeidat, N., & Zhao, Z. (2004). Supervised clustering - algorithms and benefits. In *International Conference on Tools with Artificial Intelligence* (p. 774-776).

Espinace, P., Kollar, T., Soto, A., & Roy, N. (2010). Indoor scene recognition through object detection using adaptive objects search. In *European Conference on Computer Vision (ECCV), Workshop on Robotics for Cognitive Tasks.*

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Workshop of Generative Model Based Vision*.

Fei-Fei, L. (2005). A bayesian hierarchical model for learning natural scene categories. In *Conference on Computer Vision and Pattern Recognition* (pp. 524–531).

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Conference on Computer Vision and Pattern Recognition.*

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis Machine Intelligence*, 1627–1645.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Conference on Computer Vision and Pattern Recognition* (pp. 264–271).

Fowlkes, E., & Mallows, C. (1983). A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, *78*(383), 553–569.

Freund, Y., & Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory* (p. 23-37).

Galleguillos, C., & Belongie, S. (2010). Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, *114*(6), 712 - 722.

Geng, B., Tao, D., T.Xu, Yang, L., & Hua, X. (2012). Ensemble manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 1227–1233.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Journal of Machine Learning*, 389–422.

Hall, M. (1999). *Correlation-based feature selection for machine learning*. Unpublished doctoral dissertation, University of Waikato.

Hampshire, J., & Waibel, A. (1992). The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *Pattern Analysis and Machine Intelligence*, 751–769.

Hubert, L., & Arabie, P. (1985). Comparing Partitions. *Journal of Classification*, 193–218.

Jacobs, R., Jordan, M., Nowlan, S., & Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 79-87.

Jiawei, H. (2005). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers Inc.

Jordan, M., & Jacobs, R. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 181–214.

Jurie, F., & Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision* (pp. 604–610).

Kam, T. (1998). The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence*, 832–844.

Khalili, A. (2010). New estimation and feature selection methods in mixture-of-experts models. *Canadian Journal of Statistics*, *38*, 519–539.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 273–324.

Kumar, R., Puran, R., & Dhar, J. (2011). Enhanced k-means clustering algorithm using red black tree and min-heap. *International Journal of Innovation, Management and Technology*, 49–54.

Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., ... Dietterich, T. (2007). Automated insect identification through concatenated histograms of local appearance features. In *Workshop on Applications of Computer Vision.*

Lasserre, J., Bishop, C., & Minka, T. (2006). Principled hybrids of generative and discriminative models. In (p. 87-94).

Lee, S., Lee, H., Abbeel, P., & Ng, A. (2006). Efficient L1 regularized logistic regression. In *National Conference on Artificial Intelligence (AAAI).*

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision.*

Lima, C., Coelho, A., & Zuben, F. V. (2007). Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences*, *177*, 2049–2074.

Liu, H. (2012). *Arizona state university: Feature selection datasets.* http://featureselection.asu.edu/datasets.php.

Liu, H., & Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. In *International conference on tools with artificial intelligence* (pp. 388–391).

MacKay, D. (1995). Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 469–505.

Maldonado, S., Weber, R., & Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences*, *181*, 115–128.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Mardia, K., Kent, J., & Bibby, J. (1979). *Multivariate analysis*. Academic Press, London.

Markov, Z., & Larose, D. (2007). *Data mining the web : Uncovering patterns in web content, structure, and usage*. John Wiley and Sons.

Marszaek, M., & Schmid, C. (2006). Spatial weighting for bag-of-features. In *Conference on Computer Vision and Pattern Recognition* (p. 2118 - 2125).

Meeds, E., & Osindero, S. (2005). An alternative infinite mixture of gaussian process experts. In *Advances in neural information processing systems* (pp. 883–890).

Meila, M., & Jordan, M. (2001). Learning with mixtures of trees. *Journal of Machine Learning*, 1–48.

Meilă., M. (2005). Comparing clusterings: An axiomatic view. In *International Conference on Machine Learning* (pp. 577–584).

Mirkin., B. (1996). *Mathematical classification and clustering*. Kluwer Academic Press.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Moerland, P. (1997). *Some methods for training mixtures of experts* (Tech. Rep.). IDIAP Research Institute.

Moosmann, F., Triggs, B., & Jurie, F. (2007). Fast discriminative visual codebooks using randomized clustering forests. , 985–992.

Murthy, S., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 1–32.

Nanculef, R. ., Valle, C., Allende, H., & Moraga, C. (2012). Training regression ensembles by sequential target correction and resampling. *Information Sciences*, *195*, 154–174.

Nguyen, M., Abbass, H., & McKay, R. (2006). A novel mixture of experts model based on cooperative coevolution. *Neurocomputing*, 155–163.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1999). Text classification from labeled and unlabeled documents using em. , 103–134.

Nowak, E., Jurie, F., & Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision.*

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal on Computer Vision*, 145–175.

Pan, W., & Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 1145–1164.

Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *American Association of Artificial Intelligence National Conference on AI* (pp. 133–136).

Perronnin, F. (2008). Universal and adapted vocabularies for generic visual categorization. *Pattern Analysis and Machine Intelligence*, 1243–1256.

Pinto, N., Cox, D. D., & DiCarlo, J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology*, 151-156.

Quinlan, J. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *International Conference on Computer Vision* (p. 1-8).

Rajaraman, A., & Ullman, J. (2012). *Mining of massive datasets*. Cambridge University Press.

Rasmussen, C., & Ghahramani, Z. (2001). Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems* (pp. 881–888).

Rice, J. (1994). *Mathematical statistics and data analysis, 2nd ed.* Duxbury Press.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(1), 53–65.

Saragih, J., Lucey, S., & Cohn, J. (2009). Deformable model fitting with a mixture of local experts. *International Conference on Computer Vision*, 2248–2255.

Scott, D., & Sain, S. (2004). Multidimensional density estimation. In *Handbook of statistics* (pp. 229–263). Elsevier.

Shanmugasundaram, R., & Sukumaran, S. (2010). Enhancing k-means algorithm with semi-unsupervised centroid selection method. *International Journal of Computer Science and Information Security*, 337–343.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2007). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 2–23.

Sinha, P., & Torralba, A. (2002). Detecting faces in impoverished images. *Journal of Vision*.

Sinnkkonen, J., Kaski, S., & Nikkila, J. (2002). Discriminative clustering: Optimal contingency tables by learning metrics. In *European Conference on Machine Learning* (pp. 418–430).

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision* (pp. 1470–1477).

Statnikov, A., Tsamardinos, I., Dosbayev, Y., & Aliferis, C. (2011). Gems: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics*, 491–503.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society (Series B)*, 267–288.

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. In *Allerton Conference on Communications and Computation* (p. 368Ű-377).

Tishby., N., & Slonim, N. (2000). Document clustering using word clusters via information bottleneck method. In *International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 208–215).

Titsias, M., & Likas, A. (2002). Mixture of experts classification using a hierarchical mixture model. *Neural Computation*, 2221–2244.

Torralba, A. (2003). Contextual priming for object detection. *Proceedings of International Journal of Computer Vision (IJCV)*, *53*, 169–191.

Torralba, A., Murphy, K., & Freeman, W. (2005). Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems* (pp. 1401–1408).

Tseng, P. (2001). Convergence of block coordinate descent method for nondifferentiable maximization. *Journal of Optimization Theory and Applications*, 475–494.

Ulas, A., Taner, O., & Alpaydin, E. (2012). Eigenclassifiers for combining correlated classifiers. *Information Sciences*, *187*, 109–120.

Ulrich, I., & Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *International Conference on Robotics and Automation* (pp. 1023–1029).

Van-Rijsbergen, C. (1979). *Information retrieval*. Butterworth-Heinemann.

Vinh, N., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *International Conference on Machine Learning* (pp. 1073–1080).

Vinh, N., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, *11*(3), 2837–2854.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Conference on Computer Vision and Pattern Recognition*, 511–518.

Viola, P., & Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 137–154.

J.Vogdrup. (2010). Combining predictors:comparison of five meta machine learning methods. *Information Sciences*, *119*, 91–105.

Wang, M., Hua, X., Hong, R., Tang, J., Guo-Jung, J., & Song, Y. (2009). Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*, 733–746.

Wang, M., Li, H., Tao, D., Lu, K., & Wu, X. (2012). Multimodal graph-based reranking for web image search. *IEEE Transactions on Image Processing*, *21*, 4649–4661.

Wang, S., & Zhu, J. (2008). Variable selection for model-based high dimensional clustering and its application to microarray data. *Biometrics*, 440–448.

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *International conference on computer vision* (pp. 1800–1807).

Wu, X., Kumar, V., Ross, J., Ghosh, J., Yang, Q., Motoda, H., ... Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 1–37.

Wu, F., Yuan, Y., & Zhuang, Y. (2010). Heterogeneous feature selection by group lasso with logistic regression. *International Conference on Multimedia*, 983–986.

Xiao, J., He, C., Jiang, X., & Liu, D. (2010). A dynamic classifier ensemble selection approach for noise data. *Information Sciences*, *180*, 3402–3221.

Xing, E., Ng., A., Jordan, M., & Russell, S. (2003). Distance metric learning with applications to clustering with side information. *Advances in Neural Information Processing Systems*, 505–512.

Xu, L., Jordan, M., & Hinton, G. (1994). An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems* (pp. 633–640).

Yang, L., Jin, R., Sukthankar, R., & Jurie, F. (2008). Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Conference on Computer Vision and Pattern Recognition.*

Yang, K., Wang, M., Hua, X., Yan, S., & Zhang, H. (2011). Assemble new object detector with few examples. *IEEE Transactions on Image Processing*, *20*, 3341–3349.

Ye, J., Z.Zhao, & Wu, M. (2008). Discriminative k-means for clustering. In *Advances in neural information processing systems* (pp. 1649–1656). Retrieved from

http://dblp.uni-trier.de/db/conf/nips/nips2007.htmlYeZW07

Yuille, A., & Geiger, D. (1998). Winner-take-all mechanisms. In *The Handbook of Brain Theory and Neural Networks.*

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal on Computer Vision*, 213–238.

Zhang, W., Surve, A., Fern, X., & Dietterich, T. (2009). Learning non-redundant codebooks for classifying complex objects. In *International Conference on Machine Learning* (pp. 1241–1248).