Incorporating Object and People Information to Improve Video Activity Recognition

A thesis submitted in partial fulfillment
of the requirements for the degree of

MASTER OF TECHNOLOGY

in

Computer Science & Engineering

by

Niranjan Viladkar

Entry No. 2012MCS2810

Under the guidance of

Dr. Parag Singla

Department of Computer Science and Engineering,
Indian Institute of Technology Delhi.

June 2014.

Certificate

This is to certify that the thesis titled **Incorporating Object and People Information to Improve Video Activity Recognition** being submitted by **Niranjan Viladkar** for the award of **Master of Technology** in **Computer Science & Engineering** is a record of bona fide work carried out by him under my guidance and supervision at the **Department of Computer Science & Engineering**. The work presented in this thesis has not been submitted elsewhere either in part or full, for the award of any other degree or diploma.

Dr. Parag Singla
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Acknowledgments

My sincere thanks to my project guide, Dr. Parag Singla. I am grateful to have his constant support and motivation during the difficulties that I faced at various stages of the project. He not only guided me towards the solutions to the problems but encouraged me to learn the concepts that gave me insights into the subject. I am thankful to Dr. Subhashis Banerjee, he gave me directions for the project in initial stages.

I thank Ruchin Kukreja and Arunim Samat for their help with acquiring data sets and initial project setup.

I would also like to thank Happy Mittal, a Ph.D scholar at the Department of Computer Science, IIT Delhi, who helped me understand markov logic networks and use of alchemy.

Niranjan Viladkar

Abstract

Human Activity Recognition is an important vision problem. In recent years, there have been various approaches towards the problem [8, 7, 9, 11]. Most existing approaches perform classification of video clips based on low level features like HoG-HoF. They don't consider the domain knowledge while classification. This methodology is vulnerable to noisy training data.

For example, presence of two persons in a clip increases chances of activity 'conversation', a partial occlusion in the scene will lead to different low level features. Thus low level features may not be able to capture the information conveyed by the presence of two persons effectively.

This project adds domain knowledge in the form of object and people information to the classification using Markov Logic. Markov Logic captures semantic relationship using weighted first order logic formulae. Thus the model learnt is less vulnerable to noisy training data and occlusions.

This project gives an end to end system for activity recognition. It takes input from a video classifier and an object detector and builds semantic model on top of this information. Activity predictions using this approach shows improvement over the existing frameworks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human activity recognition is an area of interest in the field of large scale surveillance. A highly accurate and precise system would obviate the need for human attention at each of the surveillance video output all the time. The system can recognize the actions being performed by the human in the video and can decide for itself if the actions are abnormal; in which case, it may raise an alarm for human attention.

This project tries to improve the existing activity recognition system by adding domain knowledge and support for uncertainty to it.

1.1 Motivation

Existing activity recognition systems (TODO - cite) do not take advantage of underlying domain. Domain may have significant influence over the activity types, the objects appearing in the activity etc. For example, presence of a dining table improves the chances of activity eating as compared to driving. Or if, the video is occluded partially, but objects like bottle and chair are visible, then also, chances of activity eating are more as compared to other activities. This relationship between objects and activities can be captured by First Order Logic (FOL).

But merely using FOL is not sufficient. In its basic form, FOL uses hard constraints. To explain this with example, consider a rule

$$\forall clip\ HasObject(clip, bottle) \implies HasActivity(clip, eating) \tag{1.1}$$

In real world scenario, it is perfectly possible that a video clip contains a bottle, but the activity might be different than eating. The rule gets violated in this scenario as it is universally quantified over all the clips. Rule may

also get violated in case of noisy training data. Thus, using hard constraints is neither practical nor robust for real world video clips.

Solution to this is to attach weights with rules like (1.1) with weights being proportional to the real world relevance. This idea can be captured using Markov Logic Networks (MLNs). Markov Logic allows us to capture the relationship between activities and objects using weighted first order logic formulae. Thus in case of noisy training data or in case of occlusion of part of the objects, such model performs better as compared to the models built only on low level features.

This project captures the semantic relationship between activities and objects as well as makes the system more robust towards noisy training data. Aim of this project is to provide end to end recognition system to improve the existing human activity recognition systems by adding domain knowledge i.e. object and people information to them.

1.2 Thesis Outline

Chapter 2 gives the theoretical background. It explains video classification using only video features. It explains how to extract and use the video features. It further descirbes object detection in detail and also describes the theory of Markov Logic Networks. Chapter **??** throws light on previous approaches and points out potential improvement areas. Chapter 4 explains design decisions and approach of this project. Chapter 5 shows the result comparisons. Finally thesis concludes with chapter 6 which notes conclusion and future work.

Chapter 2

Background

Video clips representing human actions can be classified based on features extracted from the video. Object detection can add the object and people information to the recognition process.

This chapter explains three components in this project. Classifiers based on video features, object detectors and gives a brief introduction to Markov Logic Networks.

Classifiers based on video features use low level features like HoG-HoF features. Using clustering of these features, each clip is represented in bag-of-features representation. Finally, SVM is used to learn a model and classify the testing instances. Object detectors run on frames of videos and detect pre-defined objects along with a confidence value for the detections. Such detections are useful to add object and people information to the existing recognition system. Markov logic allows us to capture the semantic relationship using weighted first order formulae. Thus use of Markov Logic makes the overall system robust towards noisy training data.

2.1 Classification based on Video Features

2.1.1 Spatial Temporal Interest Points

As explained in [6] spatial temporal interest points (STIPs) are 3 dimensional points in space and time where the local video features show significant variations. In [8] the features extracted at STIPs are shown to perform satisfactorily for human activity recognition. At each STIP, histograms of oriented gradient (HoG) and histograms of optical flow (HoF) are evaluated. The number of HoG and HoF features at each STIP are 72 and 90 respectively.

Concatenating vectors of HoG and HoF features, we get a single 162 element descriptor for each STIP.

2.1.2 Clustering

After extraction of STIP features, each video clip roughly has $O(1000)$ such descriptors. 100,000 descriptors are sampled randomly from all the descriptors across all the clips. While doing random sampling, each clip is given equal chance - this avoids biased sampling towards (or against) any particular class of videos if they happen to be longer (or shorter) than other videos on an average. These 100,000 random descriptors are clustered into $k = 200$ clusters using k-means.

2.1.3 Bag of Features Representation

Each descriptor in each clip is clustered into one of the k clusters using least Euclidean distance. Thus each clip is represented by a k sized vector where i^{th} element in this vector represents number of descriptors of that clip nearest to the i^{th} cluster. This vector is called a Bag of Features (BoF) representation of the clip.

2.1.4 Support Vector Machines

The BoF representation of all training clips are used to train a support vector machine (SVM) and BoF representation of a disjoint set of testing clips is classified using learnt model. For training a SVM model, a radial basis function (RBF) kernel is used with parameters $C = 100$ and $\gamma = 0.01$. A one-versus-rest learning strategy is used to learn as many models as there are action classes. In case of this data set, there are 12 action classes and thus 12 models are learnt.

Classification using only video features are compared taking average average precision as a measure. As explained in [8], average precision(AP) approximates area under recall-precision curve. Thus, AP for each activity class is

calculated and then finally averaged over all the classes, average AP (AAP) is calculated.

2.2 Object Detection

Video clips of certain activity class have peculiar set of objects present in it. If one could find objects present in the video clip, the activity prediction can potentially be made more confident. Below subsections introduce the object detection method and its use in this project.

2.2.1 Object Detector

Object detector based on Discriminatively Trained Deformable Part Models [3] was used in the project. It can run on an image at a time. The detectors used in this project are trained on PASCAL 2007 data sets. There are 20 models corresponding to 20 different objects. These objects are : aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv monitor. For this project only 10 relevant models were used : bicycle, bottle, bus, car, chair, diningtable, motorbike, person, sofa, tvmonitor.

2.2.2 Detection in Video

Above described object detector models work on a single image at a time. To detect objects in video clips, 1 frame per second was extracted from the video clip and all the object detector models were run on the frame. In the data set used in this project, shortest clips is about 3-4 seconds long and longest clips are of the order of a minute. Frame rate is 24 frames per second.

The object detector models give a bounding box and a confidence (also called decision value) for each detection on an absolute scale between $-\infty$ to ∞. Usually, a positive decision value represents true positive in all the models. A negative decision value represent lesser confident detection. It might be a true positive or a false positive. One can specify a threshold decision value

before running object detector model so that only the detections at least as confident as the threshold are considered. Usually, such thresholds are negative to allow detection of lesser confident objects also.

2.2.3 Output of Object Detector

The visual output of object detector looks as shown in figures 2.1 and 2.2. The red boxes are called as bounding boxes. For each box, a confidence value is also evaluated - also called as decision value.

Figure 2.1: Car detection in a frame.

Figure 2.2: Person detection in a frame.

2.3 Markov Logic Networks

As explained in [10], Markov Logic consists of set of First order logic (FOL) formulae associated with weights. The FOL formulae form the structure of Markov Logic Networks (MLNs). Background and theory of MLNs and calculation of weights of formula is explained in the coming sections.

2.3.1 Markov Networks

Markov Network is an undirected graph G with its nodes as set of variables $X = (X_1, X_2, \ldots, X_n) \in \chi$. Apart from graph, it also consists of potential functions ϕ_k for each clique k in the graph. Potential function ϕ_k is a real valued function of the state of k^{th} clique. The joint distribution is given by

$$P(X = x) = \frac{1}{Z}\prod_k \phi_k(x_{\{k\}}) \tag{2.1}$$

where $x_{\{k\}}$ is the state of k^{th} clique; Z, also known as partition function, is given by

$$Z = \sum_{x \in \chi}\prod_k \phi_k(x_{\{k\}}) \tag{2.2}$$

If clique potentials are replaced by exponentiation of weighted sum of features of state, joint distribution can be given as

$$P(X = x) = \frac{1}{Z} exp\left(\sum_i w_i f_i(x)\right) \tag{2.3}$$

where w_i is the weight of the i^{th} clique and $f_i(x) \in \{0, 1\}$ indicates state of the i^{th} clique.

2.3.2 Markov Logic

The domain knowledge is captured by First Order Logic (FOL) formulae. In its basic form, first order knowledge base is a set of hard constraints over a set of possible worlds. Because of these hard constraints, if a world does not satisfy even a single formula, the world is considered as false or impossible.

Markov Logic uses FOL with soft constrains instead of hard. In markov logic, if a world does not satisfy a formula F_i, it is considered as less probable world in comparison to a world which satisfies F_i assuming rest of the formulae have same state in both the worlds.

Definition 2.3.1 from [10] formally defines Markov Logic Network (MLN) as follows

Definition 2.3.1. *A Markov Logic Network(MLN) L is a set of pairs* (F_i, w_i)*, where* F_i *is a formula in first order logic and* w_i *is weight associated with the formula - a real number. Together with a finite set of constants* $C = \{c_1, c_2, \ldots, c_{|C|}\}$*, MLN L defines a markov network,* $M_{L,C}$ *as follows :*

1. $M_{L,C}$ *contains one binary node for each possible grounding of each atom appearing in L. The value of the node is 1 if ground atom is true and 0 otherwise.*

2. $M_{L,C}$ *contains one feature for each possible grounding of each formula* F_i *in L. The value of the feature is 1 if the ground formula is true and 0 otherwise. The weight of the feature is* w_i *associated with* F_i *in L.*

Probability distribution over possible world x specified by markov network $M_{L,C}$ is given by

$$P(X = x) = \frac{1}{Z} exp\left(\sum_{i=1}^{F} w_i n_i(x)\right) \tag{2.4}$$

where F is number of formulae in MLN and $n_i(x)$ is the number of true groundings of F_i in world x.

Chapter 3

Related Work

In [7], authors have come up with a method to annotate video clips from Hollywood movies using their text scripts. Authors present a method for classification using local space-time features, space-time pyramids and multi-channel non-linear SVM.

Their approach is based on low level space time features such as HoG-HoF features. All video clips are represented in Bag-of-Features representation. These Bag-of-Features vectors are used to train a SVM model. Note that as this method uses low level features, it will be ineffective in case the video is partially occluded or the training data is noisy.

In [8], authors add context in the form of scene information to the classification problem. As quoted by authors, an example is, activity eating is more probable to happen in kitchen while activity running is likely to happen outdoor. Thus scene contexts of these two activities will help in classification.

There approach also consists of forming a Bag-Of-Features representation of video clips using low level features like HoG-HoF. Thus suffering similar problems as explained above. The scene information is retrieved using text scripts. Thus this information is limited by the text available and may not be able to capture the actual relationship between scene and activity as seen in the video.

In [9], authors have used object information present in the video clip. The correlation between activity and object has been found using analysis of large amount of text. Note that this correlation is not found directly via video but via text. Thus it has the limitation that only the correlations appearing in the text can be captured. Also, a naive Bayes assumption is made which assume that video features and object features are independent of each other given

the activity class. Finally an integrated classifier is formed which combines classification based on methods from section 2.1 and probability of activity deduced from object information.

In [11], Markov Logic Networks are used to capture the interaction between humans and vehicles, such as, open door, drive away etc. Although this method does capture the relationship irrespective of occlusions and noisy training data, it is limited to human - vehicle interaction. For real world videos, a more general system is required.

Thus, it can be seen that most of the previous approaches don't capture semantic relationship between activities and objects. Next chapter explains how this project combines object and people information with existing activity recognition to improve the recongnition results.

Chapter 4

Approach

This project aims at providing end to end system which takes inputs from video activity recognizer and object detector to enhance the video activity recognition.

Figure 4.1 explains the approach of this project.

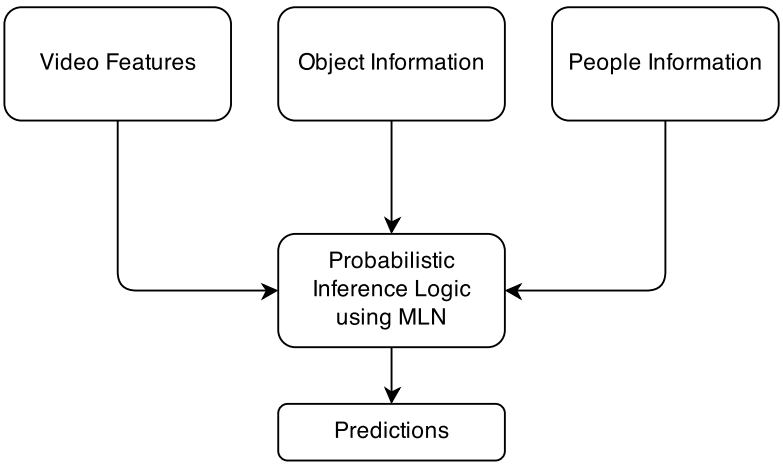

Figure 4.1: Approach of the Project

The video features are extracted as explained in section 2.1. The code used in this phase is publicly made available at [4]. Using one versus all (ovr) matlab interface of libsvm [2], model is learnt to extract the confidence values from video feature classification.

The object features are evaluated using [3]. Each object detection has a confidence value along with it. This value is used to determine whether to consider detection as a viable detection or spurious one. The thresholds for confidence values are found experimentally. Details are given in chapter 5.

Main contribution of this project is in the formulation of markov logic system. Alchemy [1] software is used to form markov logic network. The software

allows convenient representation of rules using first order logic and assigns weights to such rules using the evidence provided. Following sections explain in detail about the evidence creation, rules, etc.

4.1 MLN Evidence

The Hollywood2 data set [5] (described in section **??**) provides true activity labels for training data. These labels are used to create predicates like

$$HasActivity(\text{“}actioncliptrain00001\text{”}, \text{“}SitUp\text{”}) \tag{4.1}$$

The learnt SVM model is used to classify training data set itself to get confidence values of activities for all clips. These confidence values are partitioned into bins to create predicates like

$$\begin{aligned} &ActivityConf\_P1\_TO\_P15(\text{“}actioncliptrain00001\text{”}, \text{“}SitUp\text{”}) \\ &ActivityConf\_NI\_TO\_N2(\text{“}actioncliptrain00002\text{”}, \text{“}HandShake\text{”}) \end{aligned} \tag{4.2}$$

Here, P1_TO_P15 represents a bin with confidence value of +1 to +1.5. Similarly other bins are formed and predicates are generated to form the evidence. Each clip has 12 such predicates as there are 12 acitivity classes.

Object detector [3] gives object detections in frames along with the corresponding confidence values. Experimentally determined thresholds are used to decide whether to consider object detection and predicates like following are formed

$$\begin{aligned} &ObjPresent(\text{“}actioncliptrain00001\text{”}, \text{“}person\text{”}) \\ &ObjPresent(\text{“}actioncliptrain00002\text{”}, \text{“}car\text{”}) \end{aligned} \tag{4.3}$$

Finally, to generate evidence for people information, average number of “person” objects are calculated in each clip. These averages are partioned into bins and each clip is assigned one bin. Predicates like following are formed

using this method

$$\begin{aligned} &NumPersons\_1\_TO\_15(\text{“}actioncliptrain00001\text{”}) \\ &NumPersons\_0\_TO\_1(\text{“}actioncliptrain00002\text{”}) \end{aligned} \tag{4.4}$$

4.2 MLN Rules

For learning MLN model, rules corresponding to bins of confidence from **video classifier** are written. For example,

$$\begin{aligned} ActivityConf\_N1\_TO\_N05(clip, activity) &\implies HasActivity(clip, activity) \\ ActivityConf\_P1\_TO\_P15(clip, activity) &\implies HasActivity(clip, activity) \end{aligned} \tag{4.5}$$

Here, the 1^{st} rule captures relation between low confidence (-1 to -0.5) of presence of activity and actual evidence. And $2nd$ rule captures relation between higher confidence ($+1$ to $+1.5$) of presence of activity and actual evidence. Thus intuitively, 1^{st} rule will get lower weight as compared to 2^{nd} rule. Exact weights are given in chapter 5.

For adding **object information**, both positive and negative rules are written. Positive rules are intuitive rules which enhance the probability of likely inference. Negative rules are counter intuitive rules which suppress the probability of unlikely rules. In both the types, the confidence of classification improves. Below is example of few positive rules

$$\begin{aligned} ObjPresent(c, \text{“}chair\text{”}) &\implies HasActivity(c, \text{“}Eat\text{”}) \\ ObjPresent(c, \text{“}car\text{”}) &\implies HasActivity(c, \text{“}DriveCar\text{”}) \end{aligned} \tag{4.6}$$

and few negative rules

$$\begin{aligned} ObjPresent(c, \text{“}bus\text{”}) &\implies HasActivity(c, \text{“}StandUp\text{”}) \\ ObjPresent(c, \text{“}car\text{”}) &\implies HasActivity(c, \text{“}HandShake\text{”}) \end{aligned} \tag{4.7}$$

For learning relationship between **people information** and activities, rules consisting of bins of number of people and activities are written. For example,

$$\begin{aligned} NumPersons\_0\_TO\_1(clip) &\implies HasActivity(clip, +a) \\ NumPersons\_1\_TO\_15(clip) &\implies HasActivity(clip, +a) \end{aligned} \tag{4.8}$$

where, 1^{st} rule has a bin that corresponds to average number of people per frame between 0 to 1. "+a" means rule will be expanded to all possible activites. Thus for each bin, rules corresponding to all 12 activites will be used in MLN learning.

4.3 MLN Inference

For inference, the learnt SVM model is used to classify testing data set clips. This step provides with the confidence values of activities for each clip. These confidence values are partioned into bins in same manner as done for generating evidence. The object and people information is added in similar fashion as that of evidence generation process.

Inference is made on predicate "HasActivity". Thus only difference between evidence and inference is absense of "HasActivity" predicates.

Sample inference instance looks like

$$\begin{aligned} &ActivityConf\_P1\_TO\_P15(\text{“}actioncliptest00001\text{”}, \text{“}Kiss\text{”}) \\ &ActivityConf\_N2\_TO\_N15(\text{“}actioncliptest00002\text{”}, \text{“}Eat\text{”}) \\ &ObjPresent(\text{“}actioncliptest00001\text{”}, \text{“}person\text{”}) \\ &ObjPresent(\text{“}actioncliptest00002\text{”}, \text{“}person\text{”}) \\ &NumPersons\_2\_TO\_I(\text{“}actioncliptest00001\text{”}) \\ &NumPersons\_1\_TO\_15(\text{“}actioncliptest00002\text{”}) \\ &\vdots \end{aligned} \tag{4.9}$$

AAP Calculations - Using the probabilities given by MLN inference, inferred activity for each clip is determined corresponding to maximum prob-

ability. The data set provides with true activity labels for test clips. These true activities and MLN inferred activities are used to calculate average average precision of MLN system. Next chapter shows the results.

Chapter 5

Results

In order to compare results, as explained in section 2.1.4, average precision (AP) is calculated for each activity class. AP is taken as a measure of performance in the work of [8].

5.1 Markov Logic Networks

Using alchemy 2.0 software, various experiments were performed to add the object and people information.

TODO - table showing comparisons of APs for MLNs.

5.2 Results of experiments using tf-idf features

Table 5.1 shows APs for all classes as per SVM setup of this project.

These AP values are obtained with using only the video features. Object and People information is not added to the feature vector. When term frequency and inverse document frequency features of object and/or people are also added to the basic SVM feature vector, it shows significant improvemnt as described in table 5.2.

Activity Class	AP as per Paper	AP as per project
AnswerPhone	8.80%	11.36%
DriveCar	74.90%	66.96%
Eat	26.30%	45.45%
FightPerson	67.50%	57.63%
GetOutCar	9.00%	17.86%
HandShake	11.60%	25.93%
HugPerson	13.50%	15.15%
Kiss	49.60%	18.18%
Run	53.70%	38.78%
SitDown	31.60%	40.96%
SitUp	7.20%	5.26%
StandUp	35%	35.20%
Average AP	32.39%	31.56%

Table 5.1: Average precision for Basic SVM Setup - without using tf-idf features

Activity Class	AP - Basic SVM	AP - Object and People
AnswerPhone	11.36%	16.67%
DriveCar	66.96%	70.09%
Eat	45.45%	50.00%
FightPerson	57.63%	66.04%
GetOutCar	17.86%	12.12%
HandShake	25.93%	31.82%
HugPerson	15.15%	17.86%
Kiss	18.18%	20.69%
Run	38.78%	39.35%
SitDown	40.96%	42.05%
SitUp	5.26%	8.70%
StandUp	35.20%	37.88%
Average AP	31.56%	34.44%

Table 5.2: Average precision after adding tf-idf features

Chapter 6

Conclusion

Various experiments aimed at improving video activity recognition were performed during this project. It was found that adding object and people information to the existing recognition framework improves the precision results.

Moreover, wherever there is a relationship beetween action and object, that can be represented by first order logic, markov logic networks effectively capture the relationship.

Bibliography

[1] Alchemy - 2 : Inference and learning in markov logic. http://code.google.com/p/alchemy-2/downloads/list.

[2] Chih-Chung Chang and Chih-Jen Lin. Libsvm – a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[3] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://cs.brown.edu/~pff/latent-release4/.

[4] Ivan Laptev. Hog-hof feature extraction at space time interest points code. http://www.di.ens.fr/~laptev/download/stip-2.0-linux.zip.

[5] Ivan Laptev. Hollywood 2 data set (action). http://www.di.ens.fr/~laptev/actions/hollywood2/.

[6] Ivan Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.

[7] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*, Jun 2008.

[8] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[9] Tanvi S. Motwani and Raymond J. Mooney. Improving video activity recognition using object recognition and text mining. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI-2012)*, pages 600–605, August 2012.

[10] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, February 2006.

[11] Son D. Tran and Larry S. Davis. Event modeling and recognition using markov logic networks. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 610–623, Berlin, Heidelberg, 2008. Springer-Verlag.

Index