Structural Semantic Models for Automatic Analysis of Urban Areas

Gianni Barlacchi[1,2], Alberto Rossi[3], Bruno Lepri[3], and Alessandro Moschitti[2,4]

[1] SKIL - Telecom Italia, Trento, Italy
gianni.barlacchi@gmail.com
[2] University of Trento, Trento, Italy
[3] Bruno Kessler Foundation (FBK), Trento, Italy
alrossi@fbk.eu, lepri@fbk.eu
[4] Qatar Computing Research Institute, HBKU, Doha, Qatar
amoschitti@gmail.com

Abstract. The growing availability of data from cities (e.g., traffic flow, human mobility and geographical data) open new opportunities for predicting and thus optimizing human activities. For example, the automatic analysis of land use enables the possibility of better administrating a city in terms of resources and provided services. However, such analysis requires specific information, which is often not available for privacy concerns. In this paper, we propose a novel machine learning representation based on the available public information to classify the most predominant land use of an urban area, which is a very common task in urban computing. In particular, in addition to standard feature vectors, we encode geo-social data from Location-Based Social Networks (LBSNs) into a conceptual tree structure that we call Geo-Tree. Then, we use such representation in kernel machines, which can thus perform accurate classification exploiting hierarchical substructure of concepts as features. Our extensive comparative study on the areas of New York and its boroughs shows that Tree Kernels applied to Geo-Trees are very effective improving the state of the art up to 18% in Macro-F1.

1 Introduction

The demographic trend clearly shows an increasing concentration of people in huge cities. By 2030, 9% of the world population is expected to live in just 41 *mega-cities*, each one with more than 10M inhabitants. Thus, the growing availability of data [2] makes it possible to discover new interesting aspects about cities and its life at a fine unprecedented granularity.

A fundamental challenge that policy makers and urban planners are dealing with is *land use classification*, which plays an important role for infrastructure planning and development, real-estate evaluations, and authorizations of business permits. More in detail, policy makers and urban planners need to associate different urban areas with specific human activities (e.g., residential, industrial, business, nightlife and others). However, traditional survey-based approaches to classify areas are time consuming and very costly to be applied to modern huge cities. Therefore, automatic approaches using novel sources of data (e.g., data from mobile phones, LBSNs, etc.) have been proposed. For example, [19] designed supervised and unsupervised approaches to infer New York City (NYC) land use from *check-in*. A check-in usually consists of latitude and longitude coordinates associated with additional metadata such as the venue where the user checked-in, comments and photos. Such data can be extracted from LBSNs like Foursquare [5], a social network application that provides the number and type of activities present in the target area (e.g., *Arts & Entertainment*, *Nightlife Spot*, etc.). The approach basically used feature vectors, mainly consisting of the number of check-in with the associated activity inferred from the Foursquare category of the place (e.g., *eating* if the check-in is done in a *restaurant*). As Gold Standard, the authors used data provided by the NYC Department of City Planning in 2013 mapped on a grid of 200×200 meters.

In this paper, we represent geographical areas in two different ways: (i) as a bag-of-concepts (BOC), e.g., *Arts and Entertainment*, *College and University*, *Event*, *Food* extracted from the Foursquare description of the area; and (ii) as the same concepts above organized in a tree, reflecting the hierarchical category structure of Foursquare activities. We designed kernels combining BOC vectors with Tree Kernels (TKs) [17, 6, 9, 10] applied to concept trees and used them in Support Vector Machines (SVMs). This way, our model (i) can learn complex structural and semantic patterns encoded in our hierarchical conceptualization of an area and (ii) highly improves the accuracy of standard classification methods based on BOC. Our GeoTK represents an interesting novelty as we show that TKs not only can capture semantic information from natural language text, e.g., as shown for semantic role labeling [12] and question answering [15, 3], but they can also convey conceptual features from the hierarchy above to perform semantic inference, such as deciding which is the major activity of a land. Our approach is largely applicable as (i) it can use any hierarchical category structure for POIs categories (e.g., OpenStreet Map POIs data); and (ii) many cities offer open access to their land use data.

Finally, we carry out a study with different granularities of the areas to be analyzed. This also enables to analyze the trade-off between the precision in targeting the area of interest and the accuracy with which we carry out the estimation. More in detail, we divide the NYC area in squares with edges of 50, 100, 200 and 250 meters and, for each cell, we classify its most predominant land use class (e.g., Residential, Commercial, Manufacturing, etc.). Our extensive experimentation, including a comparative study as well as the use of several machine learning models, shows that GeoTKs are very effective and improve the state of the art up to 18% in Macro-F1.

The reminder of this paper is organized as follows, Sec. 2 introduces the related work, Sec. 3 describes the task and the related data, Sec. 4 presents our hierarchical tree representation and our GeoTK. Then, Sec. 5 illustrates the evaluation of our approach, and finally Sec. 6 derives some conclusions.

[5] `https://foursquare.com`

2 Related Work

Several works have targeted land use inference by means of different sources of information. For example, [18] built a framework that, using human mobility patterns derived from taxicab trajectories and Point Of Interests (POIs), classifies the functionality of an area for the city of Beijing. The model is similar to the one used for topic discovery in a textual document, where the functionality of an area is the topic, the region is the document, and POIs and mobility patterns are metadata and words, respectively. Specifically, [18] have used an advanced model combining Latent Dirichlet Allocation (LDA) with Dirichlet Multinomial Regression (DMR), in order to insert also information coming from the POIs (metadata). Hence, for each region, after the parameter estimation with DMR, they have a vector representing the intensity of each topic. This vector is then used to aggregate formal regions having similar functions by k-means clustering.

Similarly, [1] proposed a spatio-temporal approach for the detection of functional regions. They exploited three different clustering algorithms by using different set of features extracted from Foursquare's POIs and check-in activities in Manhattan (New York). This task permits to better understand how the functionality of a city's region changes over time. Other works have used geo-tagged data from social networks: for example, [8] used tweets as input data to predict the land use of a certain area of Manhattan. Moreover, they try to infer POIs from tweets' patterns clustering the surface with Self-Organizing-Map, then characterizing each region with a specific tweet pattern and finally using k-means to infer land use. Again, [19] have used check-in data to compare unsupervised and supervised approaches to land use inference.

Finally, some works have also used Call Detail Records (CDRs) [8, 16, 13, 7], which are typically used by mobile phone operators for billing purposes. This data registers the time and type of the communication (e.g., incoming calls, Internet, outgoing SMS), and the radio base station handling the communication. For example, [16] have used CDRs jointly with a Random Forest classifier to build a time-varying land use classification for the city of Boston. The intuition behind this work is to mine a time-variant relation between movement patterns and land use. In particular, they perform a Random Forest prediction and then they compare it with the predictions obtained for the neighboring regions, applying a sort of consensus validation (e.g., they modify the prediction if a certain number of neighbors belong to a different uniform function). This way, they model different land uses for different temporal slots of the day.

Compared to the state of the art, the main novelties introduced by our work are the following: (i) we model the hierarchical semantic information of Foursquare using GeoTK, thus adding powerful structural features in our classification models; and (ii) we study how the size of the grid impacts on the accuracy of different models, thus investigating the trade-off between granularity of the analysis and accuracy. It should be also noted that, in contrast to previous work, GeoTK does not rely on external resources (e.g., mobile phone data) or heavy features engineering in addition to the structural kernel model.

3 Datasets

We use the shape file of New York provided by the NYC government [6]. This file is publicly available and contains the entire shape of New York divided in the 5 boroughs: Manhattan, Brooklyn, Staten Island, Bronx, and Queens. Then, we build a grid over the entire city in order to enable our classification task. The goal is to infer the land use of a region given a target label and a feature representation of the region. In the next subsections, we describe (i) the land use data and labels utilized by our approach, and (ii) the Foursquare's POIs used to obtain a feature representation of the land of a region.

3.1 Land Use

In our study, we use MapPluto, a freely available dataset provided by the NYC government, which contains precise geo-referenced information for each city's borough. For example, it provides the precise category and shape for each building in the city. More in detail, it contains the following land use categories: (i) *One and Two Family Buildings*, (ii) *Multi-Family Walk-Up Buildings*, (iii) *Multi-Family Elevator Buildings*, (iv) *Mixed Residential and Commercial Buildings*, (v) *Commercial and Office Buildings*, (vi) *Industrial and Manufacturing Buildings*, (vii) *Transportation and Utility*, (viii) *Public Facilities and Institutions*, (ix) *Open Space and Outdoor Recreation*, (x) *Parking Facilities*, and (xi) *Vacant Land*. Land use information is very fine-grained, and in most cases there is only one land use assigned to one building, thus making it very difficult to determine the land use with just POI information. A reasonable trade-off between classification accuracy and the desired area granularity consists in segmenting the regions in squared cells: each cell will refer to more than one land use but we consider the predominant class as its primary use.

3.2 Foursquare's Point of Interests

We extracted 206,602 POIs from the entire NYC. As for the land use data, we have several sources of information, but we focused on the ten macro-categories of the POIs, each one specialized in maximum four levels of detail. These levels follow a hierarchical structure[7], where each level of a category has a finite number of subcategories as node children. For instance, the first level of POIs main categories is constituted by: (i) *Arts and Entertainment*, (ii) *College and University*, (iii) *Event*, (iv) *Food*, (v) *Nightlife Spot*, (vi) *Outdoors and Recreation*, (vii) *Professional and Other Places*, (viii) *Residence*, (ix) *Shop and Service*, and (x) *Travel and Transport*. The second level includes 437 categories whereas the third level contains a smaller number of categories, 345.

[6] http://www1.nyc.gov/site/planning/data-maps/open-data/districts-download-metadata.page

[7] https://developer.foursquare.com/categorytree

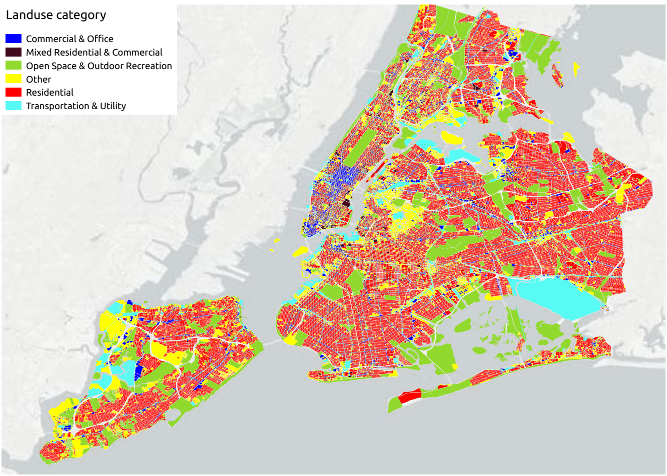

Fig. 1. Example of land use distribution in New York City.

4 Semantic Structural Models for Land Use Analysis

Previous works [14, 13, 4] have mainly used features extracted from LBSNs (e.g., Foursquare's POIs) in the XGboost algorithm [5]. However, these feature vectors have several limitations such as (i) the small amount of information available for the target area and (ii) their inherent scalar nature, which does not capture the existence and the type of relations between different POIs. Here, we propose a much powerful approach based on TKs applied to a semantic structure based on the hierarchical organization of the Foursquare categories.

4.1 Bag-Of-Concepts

The most straightforward way to represent an area by means of Foursquare data is to use its POIs. Every venue is hierarchically categorized (e.g., *Professional and Other Places* → *Medical Center* → *Doctor's office*) and the categories are used to produce an aggregated representation of the area. We use this feature representation by aggregating all the venues together, namely we count the macro-level category (e.g., *Food*) in all the POIs that we found in any cell grid. This way, we generate the Bag-Of-Concepts (BOC) feature vectors, counting the number of each activity under each macro-category.

4.2 Hierarchical Tree Representation of Foursquare POIs

Every LBSN (e.g., Foursquare) has its own hierarchy of categories, which is used to characterize each location and activity (e.g., restaurants or shops) in the database. Thus, each POI in Foursquare is associated with a hierarchical path, which semantically describes the type of location/activity (e.g., for *Chinese*

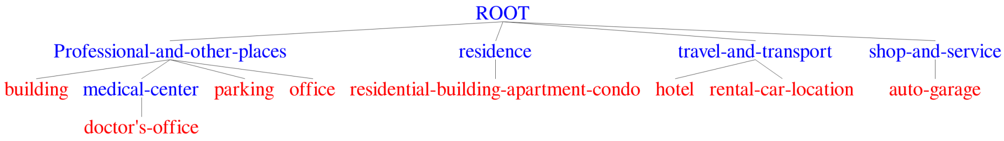

Fig. 2. Example of Geo-Tree built according to the hierarchical categorization of Foursquare venues.

Restaurant, we have the path *Food* → *Asian Restaurant* → *Chinese Restaurant*). The path is much more informative than just the target POI name, as it provides feature combinations following the structure and the node proximity information, e.g., *Food* & *Asian Restaurant* or *Asian Restaurant* & *Chinese Restaurant* are valid features whereas *Food* & *Chinese Restaurant* is not.

In this work, we propose, a tree structure, Geo-Tree (GT), where its nodes are Foursquare categories and the edges among them are the same provided in the hierarchical category tree of Foursquare. Our structure is basically composed of all paths associated with the POIs that we find in the target grid cell. Precisely, we connect all these paths in a new root node. This way, the first level of root children corresponds to the most general category in the list (e.g., *Arts & Entertainment*, *Event*, *Food*, etc.), the second level of our tree corresponds to the second level of the hierarchical tree of Foursquare, and so on. The terminal nodes are the finest-grained descriptions in terms of category about the area (e.g., *College Baseball Diamond* or *Southwestern French Restaurant*). For example, Fig. 2 illustrates the semantic structure of a grid cell obtained by combining all the categories' chains of each venue. Given such representation, we can encode all its substructures in kernel machines using TKs as described in the next section.

4.3 Geographical Tree Kernels (GeoTK)

Structural kernels are very effective means for automatic feature engineering [11]. In kernel machines both learning and classification algorithms only depend on the evaluation of inner products between instances, which correspond to compute similarity scores. In several cases, the similarity scores can be efficiently and implicitly handled by kernel functions by exploiting the following dual formulation of the classification function: $\sum_{i=1..l} y_i \alpha_i K(o_i, o) + b$, where o_i are the training objects, o is the classification example, $K(o_i, o)$ is a kernel function that implicitly defines the mapping from the objects to feature vectors $\boldsymbol{x_i}$. In case of tree kernels, K determines the shape of the substructures describing trees.

4.4 Tree Kernels

In the majority of machine learning approaches, data examples are transformed in feature vectors, which in turn are used in dot products for carrying out both learning and classification steps. Kernel Machines (KMs) allow for replacing the dot product with kernel functions, which compute the dot product directly from examples (i.e., they avoid the transformation of examples in vectors).

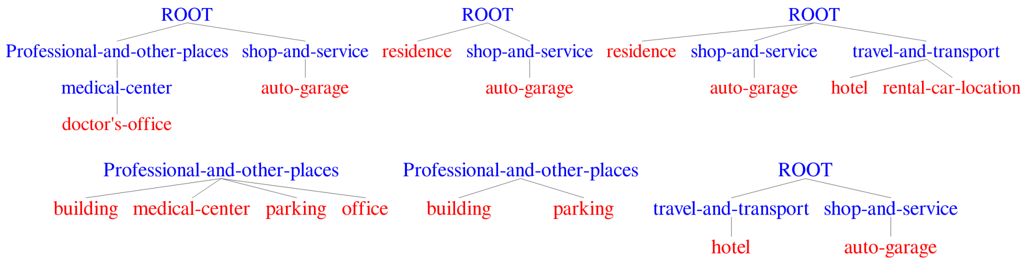

Fig. 3. Some of the exponential fragment features from the tree of Figure 2

Given two input trees, TKs evaluate the number of substructures, also called fragments, that they have in common. More formally, let $\mathcal{F} = \{f_1, f_2, \ldots ..f_{\mathcal{F}}\}$ be the space of all possible tree fragments and $\chi_i(n)$ an indicator function such that it is equal to 1 if the target f_1 is rooted in n, equal to 0 otherwise. TKs over T_1 and T_2 are defined by $TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$, where N_{T_1} e N_{T_2} are the set of nodes of T_1 and T_2 and

$$\Delta(n_1, n_2) = \sum_{i=1}^{\mathcal{F}} \chi_i(n_1)\chi_i(n_2) \qquad (1)$$

represents the number of common fragments rooted at nodes n_1 and n_2. The number and the type of fragments generated depends on the type of the used tree kernel functions, which, in turn, depends on $\Delta(n_1, n_2)$.

Syntactic Tree Kernels (STK) Its computation is carried out by using $\Delta_{STK}(n_1, n_2)$ in Eq. 1 defined as follows (in a syntactic tree, each node can be associated with a production rule):
(i) `if the productions at` n_1 `and` n_2 `are different` $\Delta_{STK}(n_1, n_2) = 0$;
(ii) `if the productions at` n_1 `and` n_2 `are the same`, `and` n_1 `and` n_2 `have only leaf children then` $\Delta_{STK}(n_1, n_2) = \lambda$; and
(iii) `if the productions at` n_1 `and` n_2 `are the same, and` n_1 `and` n_2 `are not pre-terminals` then $\Delta_{STK}(n_1, n_2) = \lambda \prod_{j=1}^{l(n_1)} (1 + \Delta_{STK}(c_{n_1}^j, c_{n_2}^j))$,
where $l(n_1)$ is the number of children of n_1 and c_n^j is the j-th child of the node n. Note that, since the productions are the same, $l(n_1) = l(n_2)$ and the computational complexity of STK is $O(|N_{T_1}||N_{T_2}|)$ but the average running time tends to be linear, i.e., $O(|N_{T_1}| + |N_{T_2}|)$, for natural language syntactic trees [10].

Finally, by adding the following step:
(0) `if the nodes` n_1 `and` n_2 `are the same then` $\Delta_{STK}(n_1, n_2) = \lambda$,
also the individual nodes will be counted by Δ_{STK}. We call this kernel STK$_b$.

The Partial Tree Kernel (PTK) [10] generalizes a large class of tree kernels as it computes one of the most general tree substructure spaces. Given two trees, PTK considers any connected subset of nodes as possible features of the

Size	Commercial	Mixed	Open Space	Other	Residential	Transportation
Train	394	225	1220	1622	6248	538
Test	175	85	534	615	2330	214

Table 1. Distribution of land use classes in the training and test set for NYC.

substructure space. Its computation is carried out by Eq. 1 using the following Δ_{PTK} function:

`if the labels of` n_1 `and` n_2 `are different` $\Delta_{PTK}(n_1, n_2) = 0$;

$$\texttt{else } \Delta_{PTK}(n_1, n_2) = \mu\Big(\lambda^2 + \sum_{\boldsymbol{I}_1, \boldsymbol{I}_2, l(\boldsymbol{I}_1)=l(\boldsymbol{I}_2)} \lambda^{d(\boldsymbol{I}_1)+d(\boldsymbol{I}_2)} \prod_{j=1}^{l(\boldsymbol{I}_1)} \Delta_{PTK}(c_{n_1}(\boldsymbol{I}_{1j}), c_{n_2}(\boldsymbol{I}_{2j}))\Big),$$

where $\mu, \lambda \in [0, 1]$ are two decay factors, $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are two sequences of indices, which index subsequences of children u, $\boldsymbol{I} = (i_1, ..., i_{|u|})$, in sequences of children s, $1 \le i_1 < ... < i_{|u|} \le |s|$, i.e., such that $u = s_{i_1}..s_{i_{|u|}}$, and $d(\boldsymbol{I}) = i_{|u|} - i_1 + 1$ is the distance between the first and last child.

When the PTK is applied to the semantic Geo-Tree of Fig. 2, it can generate effective fragments, e.g., those in Fig. 3.

Combination of TKs and feature vectors Our TKs do not consider the frequency[8] of the POIs present in a given grid cell. Thus, it may be useful to enrich the feature space with further information that can be encoded in the model using a feature vector. To this end, we need to use a kernel that combines tree structures and feature vectors. More specifically, given two geographical areas, x^a and x^b, we define a combination as: $K(x^a, x^b) = TK(\mathbf{t}^a, \mathbf{t}^b) + KV(\mathbf{v}^a, \mathbf{v}^b)$, where TK is any structural kernel function applied to tree representations, $\mathbf{t}^a$ and $\mathbf{t}^b$ of the geographical areas and KV is a kernel applied to the feature vectors, $\mathbf{v}^a$ and $\mathbf{v}^b$, extracted from x^a and x^b using any data source available (e.g., text, social media, mobile phone and census data).

5 Experiments

We test the effectiveness of our approach on the *land use classification* task, where the goal is to assign to each area the predominant land use class as performed in previous work by [19, 16]. We first test several models on Manhattan using several grid sizes, then we focus on evaluating the best models on all NYC boroughs and finally, we use the best models on the entire NYC, also enabling comparisons with previous work.

5.1 Experimental Setup

We performed our experiments on the data from NYC boroughs, evaluating grids of various dimensions: 50 × 50, 100 × 100, 200 × 200 and 250 × 250 me-

[8] It is possible to add the frequency in the kernel computation but for our study we preferred to have a completely different representation from previous typical frequency-based approaches.

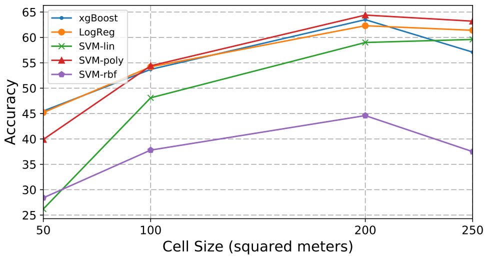

Fig. 4. Accuracy of common machine learning models on different cell sizes in Manhattan.

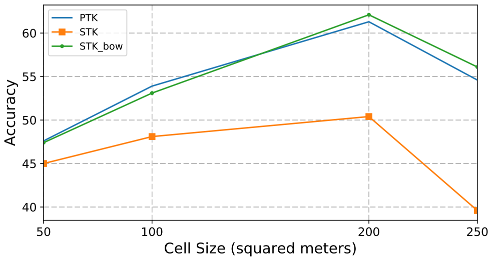

Fig. 5. Accuracy of GeoTKs according to different cell sizes of Manhattan.

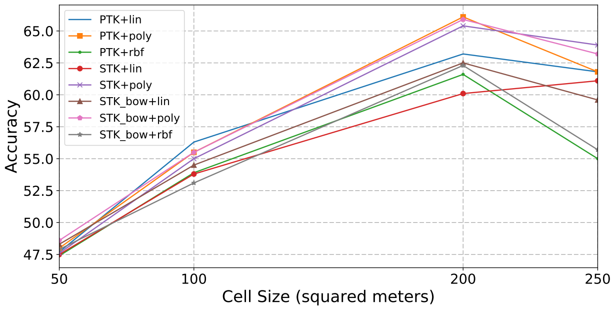

Fig. 6. Accuracy of kernel combinations using BOC vectors and GeoTKs according to different cell sizes of Manhattan.

ters. We applied a pre-processing step in order to filter out cells for which it is not possible to perform land use classification. In particular, from each grid, we removed the cells (i) that cover areas without a specified land use (e.g., cell in the sea) and (ii) for which we do not have POIs (e.g., cells from Central Park). For each grid, we created training and test sets, randomly sampling 80% vs. 20% of the cells, respectively. We labelled the dataset following the same category aggregation strategy proposed by [19], who assigned the predominant land use class to each grid cell. Note that given the categories described in Sec. 3.1, we merged (i) *One & Two Family Buildings*, (ii) *Multi-Family Walk-Up Buildings* and (iii) *Multi-Family Elevator Buildings* into a single general *Residential* category. Then, we also aggregated (i) *Industrial & Manufacturing*, (ii) *Public Facilities & Institutions*, (iii) *Parking Facilities* and (iv) *Vacant Land* into a new category called *Other*. Thus, the aggregated dataset contains six different classes: (i) *Residential*, (ii) *Commercial and Office Buildings*, (iii) *Mixed Residential and Commercial Buildings*, (iv) *Open Space and Outdoor Recreation*, (v) *Transportation and Utility*, (vi) *Other*. The names and distribution of examples in training and test sets (for the grid of 200 × 200) are shown in Tab. 1. Compared to the original categorization, this new taxonomy has a lower granularity, thus facilitating the identification of the predominant class in each cell.

Area	Cell	XGBoost	SVM-poly	PTK	PTK+poly	STK	STK+poly	STK_b	STK_b+poly
Manhattan	50	45.0	39.9	47.6	48.0	45.0	47.6	47.4	48.6
	100	54.0	54.4	53.9	55.5	48.1	55.0	53.1	55.5
	200	63.0	64.4	61.3	66.1	50.4	65.4	62.1	65.9
	250	57.0	63.2	54.6	61.8	39.6	63.9	56.1	63.2
Bronx	50	43.0	30.9	44.9	44.9	42.2	43.4	42.4	43.2
	100	50.0	43.7	53.2	54.1	51.2	53.2	54.7	54.0
	200	59.0	56.4	62.6	60.6	56.4	60.4	61.8	61.8
	250	59.0	58.6	63.5	64.9	59.3	59.6	63.0	65.2
Brooklyn	50	49.0	44.2	51.3	51.6	48.7	51.3	51.4	52.2
	100	61.0	61.0	63.1	63.5	62.4	62.9	63.1	63.2
	200	71.0	71.5	72.9	73.6	70.1	73.2	73.3	73.8
	250	70.0	68.9	71.3	72.6	67.9	70.3	70.6	71.4
Queens	50	48.0	32.4	51.5	51.5	50.2	51.0	50.5	50.3
	100	58.0	57.2	61.4	61.3	59.8	60.6	61.6	61.7
	200	67.0	66.5	70.5	71.3	69.3	69.9	70.4	71.0
	250	68.0	68.3	72.9	73.1	70.1	72.2	72.4	73.6
StatenIsland	50	51.0	38.63	54.4	55.2	52.8	54.6	53.8	54.9
	100	57.0	56.73	58.1	58.7	53.6	57.4	56.0	58.1
	200	60.0	60.0	61.8	61.1	60.2	60.0	61.3	60.9
	250	66.0	64.87	67.4	66.3	66.0	67.2	67.9	67.4

Table 2. Accuracy of the best models for each New York borough and cell size.

To train our models, we adopted SVM-Light-TK[9], which allow us to use structural kernels ([10]) in SVM-light[10]. We experimented with linear, polynomial and radial basis function kernels applied to standard feature vectors. We measured the performance of our classifier with Accuracy, Macro-Precision, Macro-Recall and Macro-F1 (Macro indicates the average over all categories).

5.2 Results for Land Use Classification

We trained multi-class classifiers using common learning algorithm such as Logistic Regression (LogReg), XGboost [5], and SVM using linear, polynomial and radial basis function kernel, named SVM-{Lin, Poly, Rbf}, respectively, and our structural semantic models, indicated with STK, STK$_b$ and PTK. We also combined kernels with a simple summation, e.g., PTK+Poly indicates an SVM using such kernel combination.

We first tested our models individually just on Manhattan using different grid sizes. Figures 4 and 5 show the accuracy of the multi-classifier for different models according to different granularity of the sampling grid. We note that SVM-Poly, XGboost and LogReg show comparable accuracy. PTK and STK$_b$ perform a little bit less than the feature vector models. Interestingly, the kernel combinations in Fig. 6 provide the best results. This is an important finding as XGboost is acknowledged to be the state of the art for land use classification. Additionally, when the size of the grid cell becomes larger, the accuracy of TKs degrades faster than the one of kernels based on feature vectors, mainly because the conceptual tree becomes too large. After the preliminary experiments above, we selected the most accurate models on Manhattan and tested them on the other boroughs of NYC. Tab. 2 shows that TKs are more accurate than vectors-based models and the combinations further improve both models.

In the final experiments, we tested our best models on the entire NYC with a grid of 200×200. We first tuned the following parameters on a validation set:

[9] http://disi.unitn.it/moschitti/Tree-Kernel.htm

[10] http://svmlight.joachims.org/

Model	**Acc.**	**F1**	**Prec.**	**Rec.**
baseline	58.9	12.4	0.98	16.6
XGBoost	63.2	36.1	57.9	31.9
SVM-poly	62.1	27.4	51.3	25.9
STK_*b*+poly	**67.4**	**42.6**	**63.9**	**37.4**
PTK+poly	66.9	41.4	63.8	36.2
STK_*b*	66.6	38.1	52.8	33.9
PTK	65.9	37.2	58.7	33.0
STK+poly	65.5	37.3	54.5	33.3
STK	62.7	25.9	41.5	24.7
Zhan et al.	65.6	-	-	-

Table 3. Classification results on New York City.

(i) the decay factors μ and λ for TK, (ii) C value for all the SVM approaches, and the specific parameters, i.e., degree in *poly* and γ in RBF kernels, (iii) the important and the parameters of XGBoost such as the maximum depth of the tree and the minimum sum of weights of all observations in a child node.

Table 3 shows the results in terms of Accuracy, Macro F1, Macro-Precision and Macro-Recall. The model *baseline* is obtained by always classifying an example with the label *Residential*, which is the most frequent. We note that: (i) all the feature vector and TK combinations show high accuracy, demonstrating the superiority of GeoTK over all the other models. (ii) STK$_b$+poly (polynomial kernel of degree 2) achieved the highest accuracy, improving over XGBoost up to 4.2 and 6.5 absolute percent points in accuracy and F1, respectively: these correspond to an improvement up to 18% over the state of the art.

Finally, Zhan et al. [19] is the result obtained on the same dataset using check-in data from Foursquare. Although an exact comparison cannot be carried out for possible differences in the experiment setting (e.g., Foursquare data changing over time), we note that our model is 1.8 absolute percentage points better.

6 Conclusions

In this paper, we have introduced a novel semantic representation of POIs to better exploit geo-social data in order to deal with the *primary land use classification* of an urban area. This gives the urban planners and policy makers the possibility to better administrate and renew a city in terms of infrastructures, resources and services. Specifically, we encode data from LBSNs into a tree structure, the Geo-Tree and we used such representations in kernel machines. The latter can thus perform accurate classification exploiting hierarchical substructure of concepts as features. Our extensive comparative study on the areas of New York and its boroughs shows that TKs applied to Geo-Trees are very effective, improving the state of the art up to 18% in Macro-F1.

Acknowledgments

This work has been partially supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

References

1. Assem, H., Xu, L., Buda, T.S., O'Sullivan, D.: Spatio-temporal clustering approach for detecting functional regions in cities. In: ICTAI. pp. 370–377. IEEE (2016)
2. Barlacchi, G., De Nadai, M., Larcher, R., Casella, A., Chitic, C., Torrisi, G., Antonelli, F., Vespignani, A., Pentland, A., Lepri, B.: A multi-source dataset of urban life in the city of milan and the province of trentino. Scientific data 2, 150055 (2015)
3. Barlacchi, G., Nicosia, M., Moschitti, A.: Sacry: Syntax-based automatic crossword puzzle resolution system. ACL-IJCNLP 2015 p. 79 (2015)
4. Calabrese, F., Di Lorenzo, G., Ratti, C.: Human mobility prediction based on individual and collective geographical preferences. In: ITSC. pp. 312–317. IEEE (2010)
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: KDD. pp. 785–794. ACM, New York, NY, USA (2016)
6. Collins, M., Duffy, N.: New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In: ACL (2002)
7. De Nadai, M., Staiano, J., Larcher, R., Sebe, N., Quercia, D., Lepri, B.: The death and life of great italian cities: a mobile phone data perspective. In: Proceedings of the 25th International Conference on World Wide Web. pp. 413–423. International World Wide Web Conferences Steering Committee (2016)
8. Frias-Martinez, V., Soto, V., Hohwald, H., Frias-Martinez, E.: Characterizing urban landscapes using geolocated tweets. In: SocialCom. pp. 239–248. IEEE (2012)
9. Gärtner, T.: A survey of kernels for structured data. ACM SIGKDD Explorations Newsletter 5(1), 49–58 (2003)
10. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: ECML. pp. 318–329. Springer (2006)
11. Moschitti, A.: Making tree kernels practical for natural language learning. In: EACL. vol. 113, p. 24 (2006)
12. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Computational Linguistics 34(2), 193–224 (2008)
13. Noulas, A., Mascolo, C., Frias-Martinez, E.: Exploiting foursquare and cellular data to infer user activity in urban environments. In: MDM. vol. 1, pp. 167–176. IEEE (2013)
14. Noulas, A., Scellato, S., Mascolo, C., Pontil, M.: Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. The Social Mobile Web 11(2) (2011)
15. Severyn, A., Moschitti, A.: Automatic feature engineering for answer selection and extraction. In: EMNLP. vol. 13, pp. 458–467 (2013)
16. Toole, J.L., Ulm, M., González, M.C., Bauer, D.: Inferring land use from mobile phone activity. In: SIGKDD International Workshop on Urban Computing. pp. 1–8. ACM (2012)
17. Vishwanathan, S.V.N., Smola, A.J.: Fast kernels for string and tree matching. In: Becker, S., Thrun, S., Obermayer, K. (eds.) NIPS. pp. 569–576. MIT Press (2002)
18. Yuan, J., Zheng, Y., Xie, X.: Discovering regions of different functions in a city using human mobility and pois. In: KDD. pp. 186–194. ACM (2012)
19. Zhan, X., Ukkusuri, S.V., Zhu, F.: Inferring urban land use using large-scale social media check-in data. Networks and Spatial Economics 14(3-4), 647–667 (2014)