J Ambient Intell Human Comput (2016) 7:567–578
DOI 10.1007/s12652-016-0346-7

ORIGINAL RESEARCH

Providing recommendations on location-based social networks

Pavlos Kosmides[1] · Konstantinos Demestichas[1] · Evgenia Adamopoulou[1] · Chara Remoundou[1] · Ioannis Loumiotis[1] · Michael Theologou[1] · Miltiades Anagnostou[1]

Received: 22 June 2015 / Accepted: 16 January 2016 / Published online: 8 February 2016

Abstract During the last decade, in parallel with the rapid growth of mobile communications and devices, location-based social networks have met a tremendous growth with the acceptance of the public being constantly increasing. Users have access to a plethora of venues and points of interest, while they are able to share their visits to various locations along with comments and ratings about their experience (a process which is often referred to as "check-ins"). Location recommendations based on users' needs have been a subject of interest for many researchers, while location prediction schemes have been developed in order to provide user's possible future locations. In this paper, we present a novel method for predicting a user's location based on machine learning techniques. In addition, following the incremental trend towards data accumulation in social networks, we introduce a clustering based prediction method in order to enhance the recommender system. For the prediction process we propose a probabilistic neural network and confirm its superior performance against two other types of neural networks, while for the clustering process we use a K-means clustering algorithm. The dataset we used was based on input from a well-known location-based social network. Prediction results can be used in order to make appropriate suggestions for venues or points of interests to users, based on their interests and social connections.

✉ Pavlos Kosmides
pkosmidis@cn.ntua.gr

[1] Institute of Communication and Computer Systems-ICCS, National Technical University of Athens, Iroon Polytechniou 9, Zografou, Athens, Greece

Keywords Location-based social networks · Neural networks · Probabilistic neural network · Clustering based prediction · Location recommendations

1 Introduction

The expansion of social media during the last decades has drastically increased the need for providing personalised suggestions to users about places, friends, activities or events. In parallel, there has been a rapid growth of location-based services in association with social media, resulting to the formulation of location-based social networks (LBSNs). In addition, it has become common for users of LBSNs to share their location, increasing in that way the demand for a more enhanced experience.

One of the most innovative features that has been introduced in LBSNs is location recommendations. Using this feature, LBSNs can provide recommendations to users about venues and places that are related to their interests. To achieve this, many researchers have employed learning algorithms in order to predict the users' future locations, based on their history (Lian et al. 2014). It is worth mentioning that this feature intrigued one of the most popular LBSNs, namely Foursquare, which recently announced that their new version will be centred around personalization and recommendations (Hamburger 2014).

In this paper, we follow the approach presented in Kosmides et al. (2014) and propose a system architecture for collecting and elaborating useful data from mobile users of LBSNs. Mobile users can be characterized as creatures of habits, tending to repeat their behaviours, something that is also attended in the places that they are visiting. This behaviour was also studied by Xiao et al. (2014) where the authors estimate the similarity between users according to their physical location histories, represented with social ties. This is why our proposed solution focuses on the use of neural networks. Specifically, we propose a probabilistic neural network (PNN) and compare its performance with two other algorithms that are also based on machine learning techniques. Attempting to further extend the research contracted in Kosmides et al. (2014), we also take into consideration that the number of people who use social networks increases drastically, resulting to an enormous amount of data produced every day (Feng et al. 2013). In order to deal with this problem, we propose a *clustering based prediction method*, enhancing the introduced location recommender system.

The rest of this paper is organized as follows. In the next section we present previous related research works. In Sect. 3, we describe the architecture of the proposed system for collecting data from LBSNs, and for processing them using machine learning mechanisms. In Sect. 4, we present the three different approaches that we use based on machine learning, namely radial basis function (RBF) neural network, support vector machine (SVM) and PNN. A clustering based prediction method is presented in Sect. 5. The dataset that was employed for location prediction is presented in Sect. 6, while in Sect. 7 we provide the results and discuss on the performance of the three proposed algorithms. Finally, the paper is concluded in Sect. 8.

2 Related work

There are several approaches that have been used in the literature for predicting a user's location. Specifically, in Fukano et al. (2013) the authors introduce a method for predicting a user's future location for context-aware systems on smartphones. This method employs contextual data that are extracted from sensor logs using a block-model. In this approach it is assumed that the future location can be estimated by "scoring" the current context, based on sensor output and without exploiting any information from social media. Another approach is proposed in McGee et al. (2013), where the authors suggest that some features of relationships are correlated with physical proximity. Specifically, they attempt to predict a user's location given the approximate location of his/her friends and followers, using a network-based approach for location estimation that integrates evidence of the social tie strength between users for improved location estimation. Similarly, in Gong et al. (2011) a location prediction scheme based on persons' social correlation is proposed. In this work, the authors present the social correlation as links between different terminals, weighted by statistic results of history information and they model topologies as sequences of snapshots to guide prediction. Nevertheless, although this approaches attempt to predict future users' location by the recent location of his closest friend, they do not consider the user's own location history.

The problem of recommending points of interest (POIs), and specifically restaurants, was presented in Wang and Yang (2009) and Yang and Wang (2009). The authors attempt to provide users with real-time location-based restaurants and recommend personalized navigation. However, although they equip their system with the ability for users to provide personalized tagging and recommendations, they do not take into consideration the input from well-known social networks. On the other hand, the authors in Wang et al. (2013) focus on providing recommendations about new venues and POIs to users, while they also study the properties of two real location-based social networks, namely Brightkite and Gowalla, analysing the relation between users and visited locations, in order to design two different recommendation algorithms. All these works manage to introduce effective techniques on poi recommendations, though without comprising user ratings on pois.

The authors in Ye et al. (2011) have developed a user-based collaborative filtering technique for recommending POIs to users by incorporating three complementary factors, namely user's historical preference, social influence and geographical influence. They manage to prove that geographical distance among POIs plays a significant role in users' possible future check-ins and should be taken into consideration. Following the same approach, the authors in Yuan et al. (2013) propose a similar technique adding time as one of the factors. They divide one day into time-slots, proposing a new method that utilizes temporal and spatial influence for POI recommendations and they prove that time-aware recommendations improve significantly the accuracy of POI recommendations. However, although time is considered as one of the factors, days are not treated differently in order to distinguish checkins during weekends from checkins during weekdays. Moreover, in both works, users' rating and poi categories (venues) are not taken into consideration.

Apart from the social relations, the authors in Hu et al. (2015) also take into consideration the reviews made by the users. To achieve that, they propose a data fusion technique combining the social matrix factorization with the topic matrix factorization techniques. Similarly, in Zhang and Chow (2015) the authors combine three kinds of correlations, namely social, geographical and categorical. Specifically, they attempt to utilize the check-in data to learn the preference of users on POIS by employing social links between users, exploiting geographical information of POIs and leveraging the popularity of each POI.

A ranking-based matrix factorization technique employing the Bayesian Personalized criterion, was introduced by the authors in Li et al. (2015). Their technique manages to alleviate the data scarcity problem as both visited and unvisited POIs contribute to the learning of a ranking function. They also take into consideration user preference, temporal popularity, geographical influence and temporal influence for providing POI recommendations to users. Nevertheless, in this paper, social bond between users is not considered as one of the factors. Authors in Cheng et al. (2012) also develop a recommendation technique based on matrix factorization. They model the probability of a user's check-in as a Multi-center Gaussian Model by clustering the POIs into sets with specific geographical centres. They also consider social influence and they utilize the inverse distance rule incorporating multi-center geographical influence into the fused matrix factorization framework, but they fail to incorporate time into their recommendations. A top-K location category based POI recommendation method is proposed in Chen et al. (2015), in which location categories have been taken into consideration for providing a wide range of flavours in POI recommendations. The authors prove that this is an NP-Hard problem and develop a greedy algorithm to solve it, though ratings are not taken into consideration for the final poi recommendations.

In Mathew et al. (2012) a hybrid method for predicting an individual's movement is presented. The proposed method groups human location histories according to their characteristics and then it employs a Hidden Markov Model for each group. A real-world location history dataset from the GeoLife project is used for the experiments of this method. Finally, the authors in Zhou et al. (2012) provide a study towards recommending locations to users of LBSNs. They categorize recommender systems into two categories, namely content-based filtering and collaborative filtering. The main difference of these categories is that in the content-based filtering, recommendations are made by analyzing the content of textual information, while in the collaborative filtering they analyze users' behaviours and activities. The authors also introduce a crawler to collect check-in data from Gowalla, and compare three techniques for utilizing them. However, in both works social influence and social connections among users are not exploited.

In this paper, we propose a collaborative filtering recommender system. As stated above, different approaches have been proposed by the research community. However, in this paper we form it as a classification problem (Basu et al. 1998; Wang et al. 2014) by examining about whether a specific location should be recommended to the user or not based on the recorded history. In addition, our work differs from previous works in that we take into consideration not only users' behaviour and habits, but also their friends by studying the friends' history too. Specifically, the recommendations are based on users' check-ins, the ratings that they left on a specific POI, their friendships and the time of the day they shared their location. In the proposed recommendation system, we also take into account the scalability problem that arises from the enormous amount of available data from social networks. This is why we propose a clustering based prediction method, in order to seek users for recommendation within smaller and highly similar clusters instead of the entire database and at the same time achieve better recommendations.

3 System description

In the proposed system, users will be equipped with mobile terminals connected to a location-based social network. One of the main concerns of the proposed LBSN is to provide its users with recommendations about points of interest. To achieve this, we take under consideration both users' preferences as well as their friends' preferences.

We take into consideration the architecture for mobile social networks that was proposed in Gebremeskel et al. (2013), depicted in Fig. 3. The architecture is composed of four core components, namely the *social network* (set of social structure of members or social object's actors), the *mobile technology* (making the social networks accessible anywhere and anytime adaptable), the *cloud data warehouse* (responsible for storing and sharing database stored on the social cloud servers) and finally the *intelligent cloud* (vitally significant to understand and make easy access of the mobile social networks' applications based on cloud contexts that considering data/information about people, objects, and surrounding, and so on). Since the *intelligent cloud* component is responsible for processing the available data that are gathered from the users and for executing the appropriate algorithms in order to enhance users' experience, we propose the introduction of a subcomponent responsible for applying machine learning algorithms in order to provide recommendations to users.

As observed in Fig. 1, the machine learning engine (MLE) process is considered as part of the *intelligent cloud* core component. The proposed MLE model is presented in Fig. 2 demonstrating the basic components and their interactions, using ArchiMate® notation (OpenGroup 2014). The proposed model consists of one product related to the mobile device and one product as part of the intelligent cloud. For the mobile device, the *Social Network Application* component is responsible for collecting users' information regarding the check-ins and the ratings that users leave for each place they visit.

On the side of the *intelligent cloud*, we recognize the following systems and components for the machine learning process:

– *Machine-learning engine training system* This system is responsible for the centralized training of the machine-learning engines that will be used by the Location Suggestions system. It is comprised of the *MLE Training Scheduler*, the *MLE Clustering* and the *MLE Training Execution components*.

  (i) *MLE training scheduler* The main purpose of this component is to initiate the generation of new MLEs (such as Multilayer Perceptron, SVMs, or PNNs).
  (ii) *MLE clustering* This component is responsible for retrieving all necessary data and group them in clusters based on clustering techniques (such as K-means), in order to enhance the classification process followed by the *MLE Training Execution* component.
  (iii) *MLE training execution* This component is responsible for retrieving the clustered training data and for executing the machine learning engine training algorithm.

– *Location suggestions system* This component is responsible for performing estimation of the locations that a user may be interested in, according to his/her own preferences, as well as the ones of his/her friends.

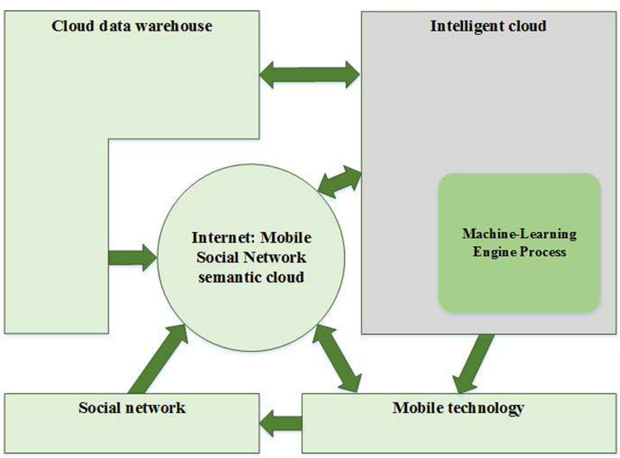


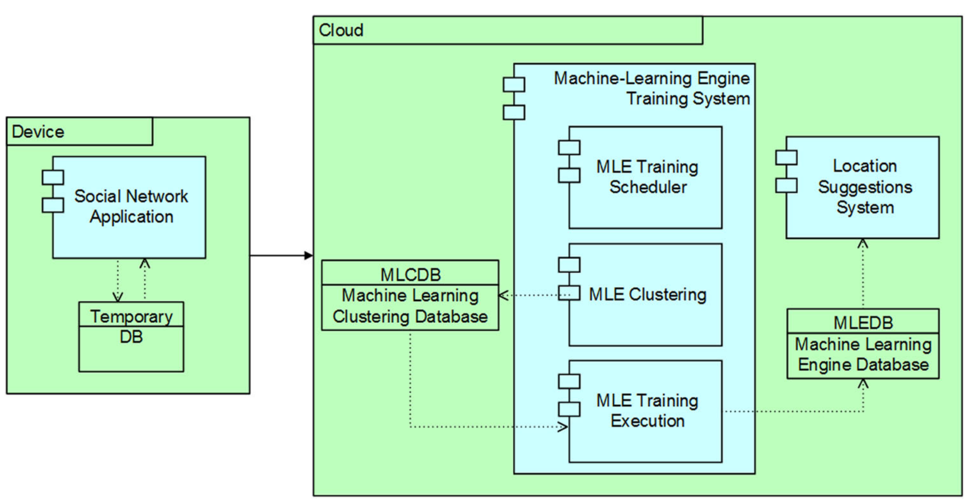
Fig. 1 Architecture based on (Gebremeskel et al. 2013)

In more detail, data related to users' use of the location based social network are gathered from their mobile devices. These data, in the proposed system, include details about each user's check-ins like the location, the rate the user has left to a place, the venue of the place, the time he/she checked-in and his/her friends. Upon being uploaded to the cloud, they are processed and accumulated to create training sets, which are used to train machine learning engines for location prediction. From this point on, the initial approach (Kosmides et al. 2014) was to create one machine learning engine (MLE) for all incoming data from the users. However, in order to deal with the scalability problem that arises from the enormous amount of available data from social networks, we introduce the MLE Clustering component in order to create highly similar clusters. In the next section, we describe in detail the prediction models that we use in order to train the created MLEs and the clustering algorithm that was used in order to create the clustered MLEs.

4 Prediction models

For the implementation of the proposed location recommendation system we take into account users' preferences from their history, which, according to (Pazzani and Billsus 2007) is a form of classification learning. Specifically, we use three different approaches based on machine learning techniques, namely RBF neural network, SVM and PNN.

Fig. 2 Machine learning engine process



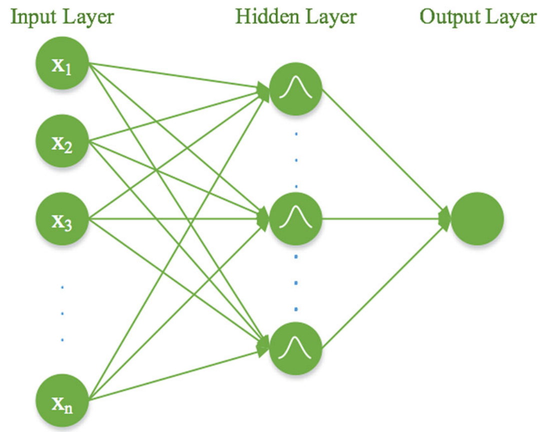


Fig. 3 Typical structure of radial basis function neural network

4.1 Radial basis function (RBF) neural network

RBF neural networks (NNs) provide an alternative to multilayer perceptrons. They share many features with ANNs that employ back propagation, and they are used for pattern recognition. However, RBF NNs have certain advantages which are not found in common ANNs. Specifically, the training procedures are faster than the multilayer perceptrons. In addition, RBF NNs are characterised by the absence of local minima, unlike common ANNs (Bianchini et al. 1995).

A typical RBF NN structure, in its most basic form (Fig. 3), involves three layers, namely an input, a hidden and an output layer. In the input layer, the space can be either normalized or be an actual representation of the input data. The data are then fed to the hidden layer nodes, which differ from other NNs in that each node represents a data cluster, centred at a particular point with a given radius. The hidden layer nodes are responsible for calculating the distance from the input vector to their own centre. The results are transformed using a basis function, and forwarded to the output layer. The output layer consists only one node, which sums the input from the hidden layer and produces the final result.

There are different possible choices of basis functions, but the most popular is based on the Gaussian (Fig. 4).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\beta(x-\mu)^2}{2\sigma^2}} \tag{1}$$

the basis function that is used in the neurons, in this paper, is of the following type:

$$\phi(x) = e^{-\beta||x-\mu||^2} \tag{2}$$

where μ refers to the center of each neuron and β coefficient controls the radius of each neuron.

The best predicted value for the new point is found by summing the output values of the RBF functions multiplied by weights computed for each neuron.

The number of the hidden layer nodes (neurons), the coordinated of the centre of each hidden-layer RBF function and the radius (spread) of each RBF function are determined in the training process. In our RBF we use an orthogonal forward selection (OFS) training algorithm based on the leave-one-out (LOO) criterion which is proposed in Chen et al. (2005). For the computation of the weights, a ridge regression method is implemented which was introduced in Orr et al. (1996).

4.2 Support vector machine (SVM)

SVM is a learning machine that is closely related to neural networks. It is based on statistical learning theory and uses linear, polynomial and radial basis kernels. An SVM, unlike common ANNs, is characterized by the absence of local minima. In addition, the computational complexity of SVMs does not depend on the dimensionality of the input space (Cristianini and Shawe-Taylor 2000).

Assuming that separable patterns in the context of pattern classification exist, the main idea of SVMs is to construct a hyperplane as the decision surface in such a way

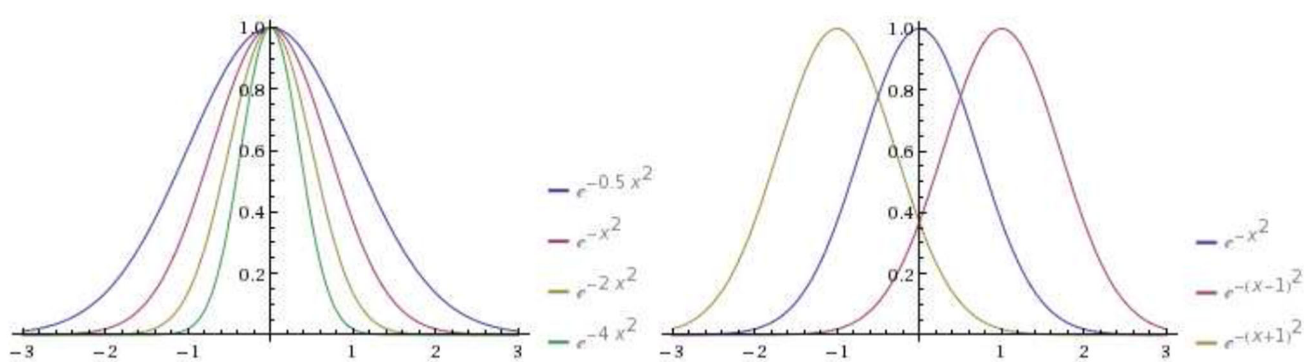


Fig. 4 Gaussian distribution used in RBF NN

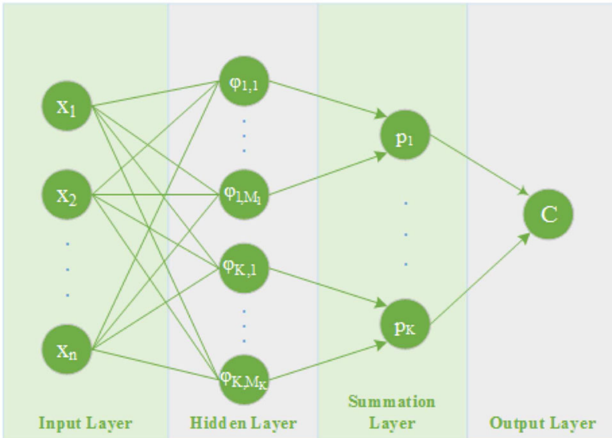


Fig. 5 Architecture of a probabilistic neural network

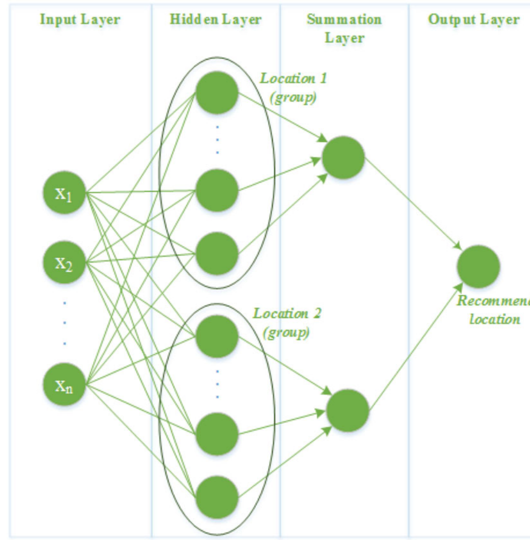


Fig. 6 Using probabilistic neural network to choose between two locations

that the margin of separation between positive and negative examples is minimized. A major advantage of SVMs is that they can provide a good generalization performance on pattern classification problems despite the fact that they do not incorporate problem-domain knowledge.

A central notion for the construction of the SVM learning algorithm is the inner-product kernel between a "support vector" $\mathbf{x_i}$ and the vector $\mathbf{x}$ drawn from the input space. The support vectors consist of a small subset of the training data extracted by the algorithm. Depending on how this inner-product kernel is generated, different learning machines characterized by nonlinear decision surfaces can be constructed.

4.3 Probabilistic neural network (PNN)

The PNN is a feedforward ANN, that was first introduced by Specht (1990). PNN is a supervised neural network and is used to perform classification where the target variable is categorical. Compared to the MLP networks, a PNN is usually much faster to train and more accurate. This is mainly due to the fact that the PNN is closely related to the Bayes classification rule (Specht 1990), and Parzen non-parametric probability density function estimation theory (Parzen 1962).

As it can be seen in Fig. 5, the architecture of a PNN consists of four layers; the input layer, the hidden layer, the summation layer, and the output layer. An input vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, is applied to the n input neurons. The input layer does not perform any computation and simply distributes the input to the neurons in the pattern layer dividing them into K groups, one for each class. On receiving a pattern x from the input layer, the neuron x_{ij} of the pattern layer computes its output using a Gaussian kernel of the form,

$$\phi_{k,i}(x) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(-\frac{||x - x_{k,i}||^2}{2\sigma^2}\right) \tag{3}$$

where $x_{k,i} \in \mathbb{R}^n$ is the centre of the kernel, and σ, also known as the spread (smoothing) parameter, determines the size of the receptive field of the kernel. The summation layer is responsible for summing the output of the hidden layer and produces a vector of probabilities that represent the probability of each feature that belongs to a specific class, through a combination of the previously computed densities

$$p_k(x) = \sum_{i=1}^{M_k} w_{ki}\phi_{k,i}(x), \qquad k \in \{1, \ldots, K\} \tag{4}$$

where M_k is the number of pattern neurons of class k, and w_{ki} are positive coefficients satisfying, $\sum_{i=1}^{M_k} = 1$. Finally, the output (or decision) layer unit classifies the pattern vector x in accordance with the Bayes decision rule based on the output of all the summation layer neurons

$$C(x) = \arg \max_{1 \leq k \leq K} (p_k) \tag{5}$$

It is worth mentioning that the smoothing parameter (σ) is the only parameter of the network that needs to be fixed at the beginning of the training process.

Specifically, for the recommendations on LBSNs we assume that the K instances represent the K available locations in the nearby area of the user. For example, if there are two available locations for recommendation, the architecture of the PNN is formed as depicted in Fig. 6. The input layer contains n nodes, one for each of the input variables that are discussed in Sect. 6. The hidden nodes are collected into K groups, each one representing one of the available locations, which in our example is two. Each input node branch to all nodes in the hidden layer so that each hidden node receives the complete input feature vector **x**. Each hidden node corresponds to a Gaussian function centred on its associated feature vector in the class that the node belongs. Finally, the Bayes decision rule is applied in order to select the location that is going to be recommended to the user.

5 Clustering based prediction

Users of LBSNs, apart from the fact that they are constantly increasing, they tend to use the provided services more regularly. This habit results to a high number of check-in records occuring during a single day. In order to enhance the classification process and improve the recommendations that are given to the users, we propose a clustering based prediction mechanism, which we presented in the proposed location recommender system in Sect. 3.

The clustering based prediction consists of the following two steps:

(i) *Clustering* As a first step, we cluster the stored dataset using a clustering algorithm, described in the next subsection.
(ii) *Classification* The clustered dataset is then used for classification using one of the prediction algorithms that are analyzed in Sect. 4.

5.1 K-means clustering

K-Means is one of the most popular clustering methods, which can be defined as the partitioning of a finite amount of data into a number of clusters by understanding the underlying structure (Cordeiro 2008). It is noted, that this problem belongs to the general class of NP-hard problems, and as a result, several heuristic algorithms are commonly employed for convergence to an optimum solution.

Two important issues in using the K-means clustering algorithm is the determination of the optimal number of clusters and the centre of each cluster. However, given the number of clusters, the problem is reduced in finding the centre of the clusters. As a result, assuming that there are n observations $(x_1, x_2, \ldots, x_n)$, and k sets $S = (S_1, S_2, \ldots, S_k)$, the objective function is

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 \tag{6}$$

where μ_i is the mean of S_i.

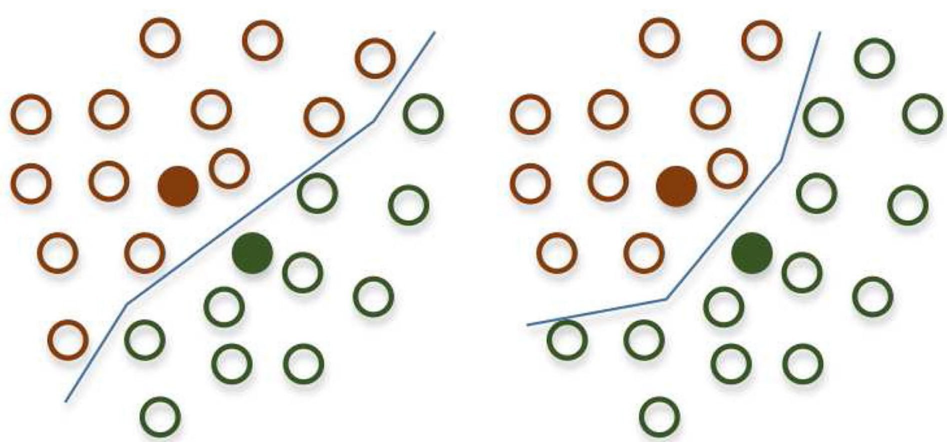
A simple example of using K-means with two clusters is given in Fig. 7. At the initial stage, the data are divided into two non-optimal clusters, while at the final stage the clusters have reached their ending form.

5.2 EM clustering

Expectation maximization (EM) algorithm (Dempster et al. 1977; Celeux and Govaert 1992) is a well-established algorithm which is used for data clustering in machine learning techniques. It is a distance based algorithm and its strong point is that it can handle the absence of data in a multidimensional dataset.

EM assumes that your data is composed of multiple multivariate normal distributions and in order to create clusters, determines a mixture of Gaussians that match a dataset. Each Gaussian has a specific mean and a covariance matrix.

$$p(X) = N(X|\mu, \Sigma) \tag{7}$$



(a) Initial clusters **(b)** Final clusters

Fig. 7 K-means clustering example with 2 clusters

Table 1 Variables and values used

Variable	Description
user	User's unique id {total 187 users}
friend	Unique id of user's friend
venue	Unique id of venue {total 118 venues}
day	Values 1–7 representing each day of the week {Monday Sunday}
hour	The hour that the user has checked in the specific location.
rate	The rate that user has left for the specific location scaling 1–5
location	Each location is represented by a unique id created from the corresponding coordinates [(latitude, longitude)] {total 219 locations}

where N is the Gaussian Normal distribution, μ is the mean and Σ the covariance matrix.

So the Gaussian Mixture distribution is of the form:

$$p(X) = \sum_{k=1}^{K} \pi_k N(X|\mu_k, \Sigma_k) \tag{8}$$

After the random initialization of the parameters, the algorithm converges on a local optima by repeatedly updating values for means and covariances.

EM algorithm follows two basic steps:

- *E-step* Computes probabilities for assignments of each datapoint to each cluster using the model equation (Expectation of the likelihood function).

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)} \tag{9}$$

- *M-step* Updates the cluster means and covariances, that maximize the expectation expression, based on the set of datapoints predominantly belonging to that cluster (Maximization of the likelihood function).

EM algorithm alternates this steps until convergence has been achieved.

6 Data analysis

The dataset used in this paper is based on measurements that were taken for research purposes in Levandoski et al. (2012). The data were collected from a well-known social network (Foursquare), which allows location-based check-ins, as well as ratings.

The available files were divided according to the location that the user has visited (check-ins file), the social connection between two users (socialgraph file), the venue of a specific location, e.g. restaurants (venues file), and the rating that each user has left on each visited location (ratings file).

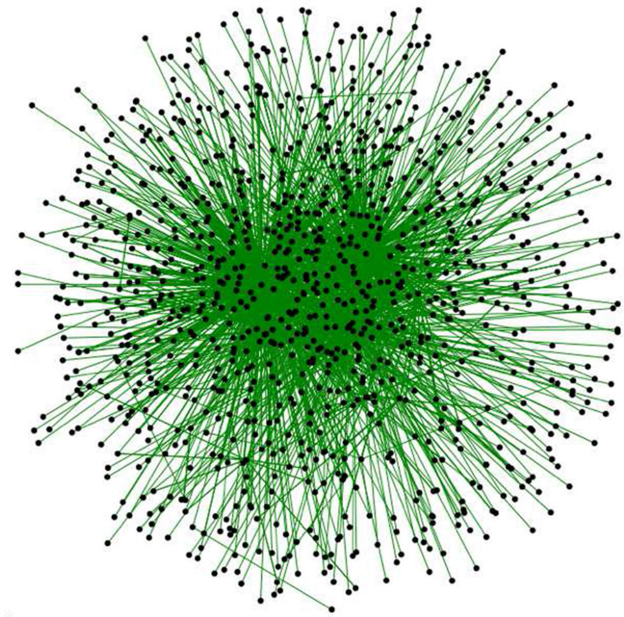


Fig. 8 Social graph using Fruchterman-Reingold lay out algorithm

The above mentioned files were merged, using users' id to create a seven-tuple with the variables described in Table 1.

Using the described variables, the input set for the proposed algorithms can be defined as:

$$\mathbf{x} = (user, friend, venue, day, hour, rate, location) \tag{10}$$

The restructured dataset takes into consideration not only the place that a user has visited, but also the type of each specific location (venue), as well as the rating of the user, which reflects his/her satisfaction. We also consider users' friendships with social connections between two users, resulting to a social graph depicted in Fig. 8.

In order to provide location recommendations, we use the methods described in Sect. 4, forming a classification problem, where each class represents a specific location. It is worth mentioning, that in contrast to other state of the art solutions (Ye et al. 2011, Zhang and Chow 2015, Yuan et al. 2013) our proposed method is a learning-based mechanism and instead of rating corresponding places, it is focusing on discovering which locations may be of interest to the user, taking also into consideration contextual data and specifically the timestamp of each user's checkin.

7 Results

7.1 Direct classification

To demonstrate the appropriateness of the PNN, we compared the results of the validation process of the PNN (σ =

Table 2 Results of validation process

Learning method	Misclassification percentage (%)
RBF NN	29.403
SVM	23.435
PNN ($\sigma = 0.1$)	19.369

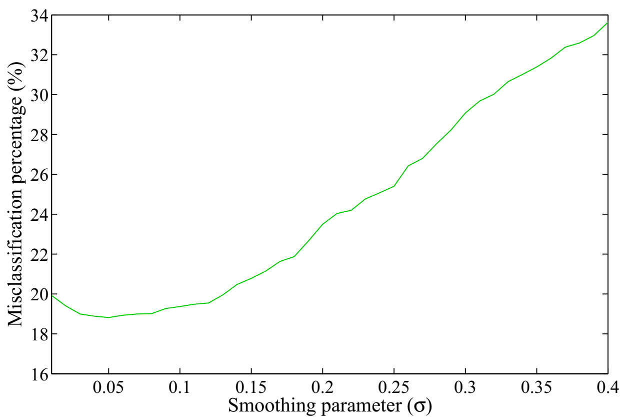


Fig. 9 Misclassification percentage for PNN with respect to σ values

0.1), with two other types belonging to the family of neural networks, as described in Sect. 4. The proposed learning algorithms were used in order to provide location recommendations based on users' preferences that are inferred from both their own and their's friends history. In Table 2, we present the misclassification percentage that was derived from each learning method, using 10-fold cross-validation method. Specifically, the RBF is formulated with 100 neurons in the hidden layer, while the SVM is constructed with an RBF kernel function.

It is clear that the PNN network outperforms both the RBF NN and the SVM, making it highly suitable for location recommendations. However, the smoothing parameter (σ) affects the performance of the PNN and needs to be appropriately fixed in order to give better prediction results. In Fig. 9, we present the effect of the smoothing parameter with values that range from 0.01 to 0.4. It can be observed that for all possible σ values, the PNN network in most cases performs better compared to the other two prediction algorithms, whereas for $\sigma \approx 0.05$ it provides the minimum misclassification percentage.

7.2 Clustering

The location recommender system that we propose in this paper, is based on the predictions provided by the introduced classifications algorithms. As we observed in the previous subsection, the misclassification percentage varies according to the used prediction method and its parameters.

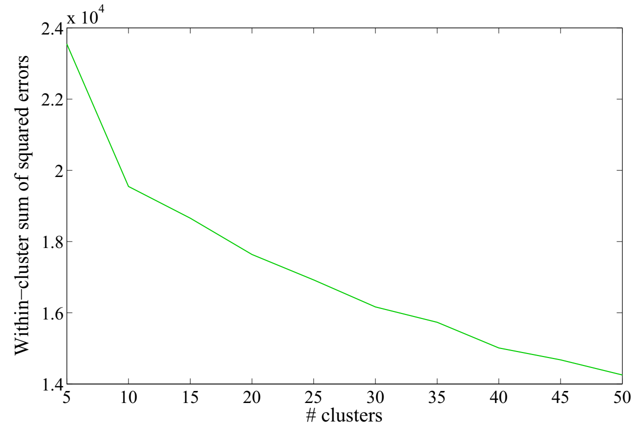


Fig. 10 Within-cluster sum of squared errors

However, an equally important factor that affects the algorithms' behaviour is the size of the provided dataset. This is why, in this paper we propose a clustering based prediction model, presented in Sect. 5.

7.2.1 K-means clustering

We first apply the K-means algorithm in order to cluster the dataset, which is then used for the prediction process. One important parameter that needs to be set in the K-means algorithm, is the number of clusters that the dataset is going to be divided into. In order to choose the appropriate number of clusters, we present in Fig. 10 the within-cluster sum of squared errors. As we can see, when the number of clusters is close to 35, the marginal returns of adding more clusters is less than was the marginal return for adding the clusters prior to that.

In order to further clarify the appropriate number of clusters, we also depict in Fig. 11 the misclassification percentage that we get, using the PNN with smoothing parameter $\sigma = 0.05$, with respect to the increasing number of clusters. It is clear, that the better performance of the classification algorithm, is achieved for 35 clusters with 16.13 % misclassification error.

7.2.2 EM clustering

Apart from the K-means, we also use the EM algorithm to cluster the dataset. Similarly to the previous subsection, we depict in Fig. 12 the misclassification percentage that we get, using the PNN with smoothing parameter $\sigma = 0.05$, with respect to the increasing number of clusters.

As we may observe, using the EM clustering the classification algorithm performs worst compared to the K-means, while the minimum misclassification percentage that can be achieved is 17.73 %. For this reason, in the next section we use the K-means for clustering the available data.

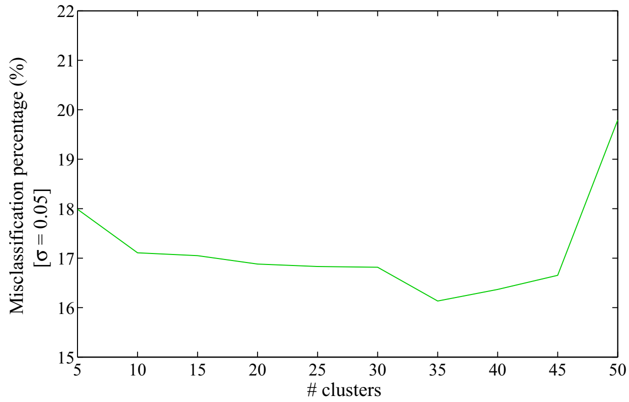


Fig. 11 Misclassification percentage using PNN with $\sigma = 0.05$ for increasing number of clusters using K-means

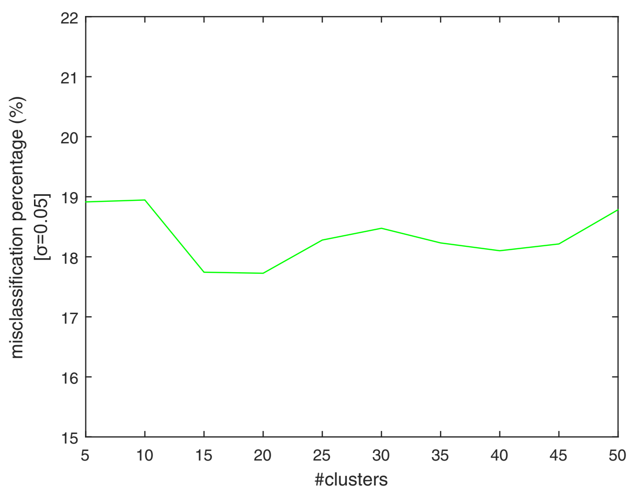


Fig. 12 Misclassification percentage using PNN with $\sigma = 0.05$ for increasing number of clusters using EM algorithm

Table 3 Results of validation process (clustered dataset)

Learning method	Misclassification percentage (%)
RBF NN	23.305
SVM	19.889
PNN ($\sigma = 0.1$)	15.368

7.3 Classifiaction of clustered data

As described in Sect. 5, the clustering based model is comprised of two stages. In the first stage, a K-means clustering algorithm is applied to the dataset with a target of 35 clusters. Then, the clustered dataset is feeded to the MLE training execution component, in order to provide location recommendations. We first compare the results of the validation process of the PNN ($\sigma = 0.1$), with the SVM and the RBF NN, presented in Table 3. The

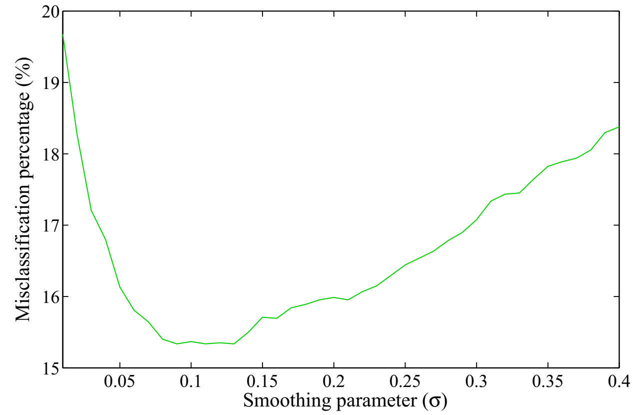


Fig. 13 Misclassification percentage for PNN with respect to σ values for dataset with 35 clusters

Table 4 Results of validation process (clustered dataset)

Locations	5	10	20
Precision			
USG Ye et al. (2011)	0.020	0.012	0.007
PNN (σ = 0.09)	0.046	0.032	0.011
Recall			
USG Ye et al. (2011)	0.088	0.011	0.14
PNN (σ = 0.09)	0.093	0.010	0.14

misclassification percentage was derived from each learning method using 10-fold cross-validation method, while the RBF was formulated with 100 neurons in the hidden layer and the SVM was conducted with an RBF kernel function.

As we may see, the PNN network outperfoms again the two other methods, while it is clear that the misclassification percentage from all three algorithms has reduced due to the clustering of the available data. Since the smoothing parameter (σ) affects the performance of the PNN, in Fig. 13 we present the effect of the smoothing parameter with values that range from 0.01 to 0.4.

It can be observed that for all possible σ values, the PNN network always performs better compared to the other two prediction algorithms, whereas for $\sigma \approx 0.09$ it provides the minimum misclassification error compared to the other σ values. In addition, it is worth noticing that the overall misclassification percentage for the clustering based prediction model has declined significantly compared to the direct classification.

In order to further examine the effectiveness of the proposed clustering based prediction model we used the performance metrics that were used in Ye et al. (2011). Specifically we used the precision and the recall metrics while scoring for N locations as recommendations (where

N is set to 5, 10 and 20). In Table 4 we compare the results of the PNN using the clustering based prediction model, with the precision and recall values reported in Ye et al. (2011) for users that do not have many check-in records. As we can see, our proposed method generally demonstrates better performance, especially in terms of precision, giving an average of 48 % better results, which is comparable to the solution introduced in Yuan et al. (2013) and Zhang and Chow (2015)) that give an average of 45 and 37 % better results in precision respectively compared, to the USG (Ye et al. 2011).

8 Conclusions

In this paper, we have presented our work on providing location recommendations using data from location-based social networks. Specifically, we have introduced a system architecture for collecting necessary data from location-based social networks, and sending them to a cloud platform. The cloud platform is responsible for running the machine-learning algorithms in order to provide recommendations to users about possible locations that they would be interested in. Three different learning algorithms based on neural networks, namely RBF NN, SVM and PNN, were used in order to predict locations according to users preferences and their social connections. Recognizing the increasing tendensy of the available data from social networks, we proposed a clustering based prediction model where the available data were clustered using a K-means clustering algorithm. The dataset used includes information collected from a well-known social network (Foursquare), depicting users friendships and ratings on specific locations that they have visited. From the results we have concluded that the PNN outperforms the other two proposed algorithms, giving predictions with high accuracy, while the clustering of the available dataset further enhances this behaviour.

References

Basu C, Hirsh H, Cohen W, et al. (1998) Recommendation as classification: Using social and content-based information in recommendation. In: AAAI/IAAI. pp 714–720

Bianchini M, Frasconi P, Gori M (1995) Learning without local minima in radial basis function networks. Neural Netw IEEE Trans 6(3):749–756

Celeux G, Govaert G (1992) A classification em algorithm for clustering and two stochastic versions. Comput Stat Data Anal 14(3):315–332

Chen S, Hong X, Harris CJ (2005) Orthogonal forward selection for constructing the radial basis function network with tunable nodes. In: Advances in Intelligent Computing. Springer, pp 777–786

Chen X, Zeng Y, Cong G, Qin S, Xiang Y, Dai Y (2015) On information coverage for location category based point-of-interest recommendation. In: Twenty-Ninth AAAI Conference on Artificial Intelligence. pp 37–43

Cheng C, Yang H, King I, Lyu M R (2012) Fused matrix factorization with geographical and social influence in location-based social networks. In: Twenty-Sixth AAAI Conference on Artificial Intelligence. pp 17–23

Cordeiro R (2008) Constrained intelligent k-means: improving results with limited previous knowledge. In: Second international conference on advanced engineering computing and applications in sciences. IEEE, pp 176–180

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, Cambridge

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc Ser B 39(1):1–38

Feng Y, Yu Z, Lu X, Tian J (2013) Understanding human dynamics of check-in behavior in lbsns. In: International conference on green computing and communications (GreenCom). IEEE, pp 923–930

Fukano J, Mashita T, Hara T, Kiyokawa K, Takemura H, Nishio S (2013) A next location prediction method for smartphones using blockmodels. In: Virtual Reality (VR). IEEE, pp 1–4

Gebremeskel GB, He Z, Jing X (2013) Semantic integrating for intelligent cloud data mining platform and cloud based business intelligence for optimization of mobile social networks. In: Bhatnagar V (ed), Data Mining in Dynamic Social Networks and Fuzzy Systems. IGI Global, pp 173–211

Gong Y, Li Y, Jin D, Su L, Zeng L (2011) A location prediction scheme based on social correlation. In: Vehicular Technology Conference (VTC Spring). IEEE, pp 1–5

Hamburger E (2014) The next age of foursquare begins today. The Verge, http://www.theverge.com/2014/7/23/5926843/this-is-the-new-foursquare. Accessed 12 Jan 2016

Hu G N, Dai XY, Song Y, Huang SJ, Chen JJ (2015) A synthetic approach for recommendation: combining ratings, social relations, and reviews. In: 24th international conference on artificial intelligence. AAAI Press, pp 1756–1762

Kosmides P, Remoundou C, Demestichas K, Loumiotis I, Adamopoulou E, Theologou M (2014) A location recommender system for location-based social networks. In: International conference on mathematics and computers in sciences and in industry (MCSI). IEEE, pp 277–280

Levandoski JJ, Sarwat M, Eldawy A, Mokbel MF (2012) Lars: a location-aware recommender system. In: 28th International Conference on Data Engineering (ICDE). IEEE, pp 450–461

Li X, Cong G, Li XL, Pham TAN, Krishnaswamy S (2015) Rank-geofm: a ranking based geographical factorization method for point of interest recommendation. In: 38th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 433–442

Lian D, Zhu Y, Xie X, Chen E (2014) Analyzing location predictability on location-based social networks. In: Advances in Knowledge Discovery and Data Mining. Springer, pp 102–113

Mathew W, Raposo R, Martins B (2012) Predicting future locations with hidden markov models. In: Conference on Ubiquitous Computing. ACM, pp 911–918

McGee J, Caverlee J, Cheng Z (2013) Location prediction in social media based on tie strength. In: 22nd international conference on information and knowledge management. ACM, pp 459–468

OpenGroup (2014) ArchiMate 2.1 Specification. The Open Group Specifications.http://pubs.opengroup.org/architecture/archimate2-doc. Accessed 12 Jan 2016

Orr M J, et al. (1996) Introduction to radial basis function networks. Technical Report, Center for Cognitive Science, University of Edinburgh

Parzen E (1962) On estimation of a probability density function and mode. Ann Math Stat 33(3):1065–1076

Pazzani MJ, Billsus D (2007) Content-based recommendation systems. In: The adaptive web. Springer, pp 325–341

Specht DF (1990) Probabilistic neural networks. Neural Netw 3(1):109–118

Wang H, Terrovitis M, Mamoulis N (2013) Location recommendation in location-based social networks using user check-in data. In: 21st SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 374–383

Wang R, Chow CY, Nutanong S, Lyu Y, Li Y, Yuan M, Lee V (2014) Exploring cell tower data dumps for supervised learning-based point-of-interest prediction. In: 22nd SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 457–460

Wang ZM, Yang F (2009) An optimized location-based mobile restaurant recommend and navigation system. WSEAS Trans Inf Sci Appl 6(5):809–818

Xiao X, Zheng Y, Luo Q, Xie X (2014) Inferring social ties between users with human location history. J Ambient Intel Human Comput 5(1):3–19

Yang F, Wang Z (2009) A mobile location-based information recommendation system based on gps and web2.0 services. WSEAS Trans Comput 8(4):725–734

Ye M, Yin P, Lee WC, Lee DL (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: 34th international SIGIR conference on research and development in information retrieval. ACM, pp 325–334

Yuan Q, Cong G, Ma Z, Sun A, Thalmann N M (2013) Time-aware point-of-interest recommendation. In: 36th International SIGIR conference on research and development in information retrieval. ACM, pp 363–372

Zhang JD, Chow CY (2015) Geosoca: exploiting geographical, social and categorical correlations for point-of-interest recommendations. In: 38th international SIGIR conference on research and development in information retrieval. ACM, pp 443–452

Zhou D, Wang B, Rahimi SM, Wang X (2012) A study of recommending locations on location-based social network by collaborative filtering. In: Advances in Artificial Intelligence. Springer, pp 255–266