Recommending Running Routes: Framework and Demonstrator

Benedikt Loepp
University of Duisburg-Essen
Duisburg, Germany
benedikt.loepp@uni-due.de

Jürgen Ziegler
University of Duisburg-Essen
Duisburg, Germany
juergen.ziegler@uni-due.de

ABSTRACT

Recommending personalized running routes is a challenging task. For considering the runner's specific background as well as needs, preferences and goals, a recommender cannot only rely on a set of existing routes ran by others. Instead, each route must be generated individually, taking into account many different aspects that determine whether a suggestion will satisfy the runner in the end, e.g. height meters or areas passed. We describe a framework that summarizes these aspects and present a prototypical smartphone app that we implemented to actually demonstrate how personalized running routes can be recommended based on the different requirements a runner might have. A first preliminary study where users had to try this app and ran some of the recommended routes underlines the general effectiveness of our approach.

KEYWORDS

Recommender Systems; Running; Route Generation; Sports

1 INTRODUCTION AND RELATED WORK

In recent years, there has been a significant increase in research on interactive technologies that support users in performing sports activities [7]. While running is one of the most popular sports, contemporary applications such as *Runtastic*, *Strava* or *Endomondo*[1] support users primarily in the process of running a route or in keeping track of activities. More advanced tools such as *TrainingPeaks* or *SportTracks*[2] focus on structuring training and creating workout plans. Features that allow generating new running routes or receiving suggestions are, however, usually not available. Typically, the only possibility is searching for routes already recorded, either by the same user or by someone else in the platform's community, that can then be run again. Yet, in many cases, runners are not only looking for routes that start and end at some location, but also satisfy e.g. length constraints or pass through specific areas, while avoiding taking the same way or leading through the same area twice. Finding routes that fulfill such requirements can be cumbersome or even impossible without adequate support. *RouteLoops*[3], for instance, allows users to generate new routes automatically, but only takes start-/end point and length into account. To find closed cycles, random walks on the map graph are performed. Other tools require users to manually search for intermediate steps, or select via-vertices automatically and connect them by means of shortest path algorithms. Still, further adaptation towards current needs and personal preferences is not supported, neither is the consideration of the user's running history or any other route property that might be of relevance for running, such as amount of elevation or which areas are passed, e.g. forests or meadows.

Generating routes is a common task referring to the traditional route planning problem. Yet, most existing research has been attributed to finding shortest paths, although other aspects have also been found relevant for people who want to follow a route: For instance, in [9], an approach for recommending emotionally pleasant walking routes within a city is presented, which however requires availability of crowdsourced data regarding attractiveness of streets. In [11], itineraries between points of interest are created. Subsequently, users can customize the suggested routes, which appeared beneficial for learning about user preferences, and thus, further personalization of recommendations. While there have been similar attempts for automobile navigation [e.g. 8, 10], research on generating routes for cycling [14] and running [5] is limited to optimization with respect to length. Beyond that, there indeed exists research on recommender systems in this area, for instance, for helping runners to achieve new personal records or pace races [2, 12, 13]. Nevertheless, although it can be difficult for users to find running routes on their own without external assistance, especially in unknown environments or when trying to find routes with certain characteristics (e.g. specific length when practicing for a race or street lighting for evening runs), there is a lack of research on supporting runners with routes that are specifically tailored for them and take all such aspects into account.

In this paper, we propose a framework that may help to generate personalized running routes. We present a prototypical smartphone app which we developed to demonstrate the effectiveness of our approach, and describe a corresponding proof-of-concept study where participants had to use this app and ran recommended routes.

2 A FRAMEWORK FOR RECOMMENDING PERSONALIZED RUNNING ROUTES

As already outlined in the previous section, standard recommender algorithms are not sufficient for creating personalized running route recommendations: Suggesting routes ran by others might be difficult due to data sparsity at the current user's location. Moreover, each runner has a different background, ranging from beginners who want to change their lifestyle and start improving their fitness to experienced runners who train for the next marathon. Accordingly, runners have different needs, preferences and goals. Besides, some might follow a training plan, such that specific requirements have to be considered with respect to the route for the next workout. Consequently, ranking a list of existing alternatives as in typical recommender situations is not an option: Instead, recommendable

[1] https://www.runtastic.com, https://www.strava.com, https://www.endomondo.com
[2] https://www.trainingpeaks.com, https://www.sporttracks.mobi
[3] http://www.routeloops.com

ComplexRec 2018 Second Workshop on Recommendation in Complex Scenarios, October 7, 2018, Vancouver, Canada.

items, i.e. routes, first need to be generated especially for the current user. This, in turn, requires that map data is available and preprocessed. Moreover, as in multi-criteria recommendation [1], the user's individual preferences for specific item properties, but also context data, are much more relevant than in many other cases: While location is obviously the most important information for recommending an appropriate route, attributes like elevation or street lighting together with weather conditions and time of the day also need to be considered to satisfy the runner in the end.

Recommending personalized running routes can thus be seen as a complex and challenging task. Since route generation in general is well-explored, we propose to split the recommendation process into two steps: 1) finding candidate routes, and 2) creating recommendations. In the following, we detail on these steps, the challenges involved, and explain how we address them in our framework.

2.1 Generating Candidate Routes

First, we generate candidate routes by means of the graph model derived from the underlying map data. We follow one of the approaches proposed in [5], namely the *partial shortest paths algorithm*, which has been shown fast enough for practical application: The idea is to determine a number of via-vertices in a way that the intermediate paths between them are the shortest paths of equal length, summing up to the desired route length. Fig. 1 illustrates this procedure for two via-vertices, i.e. the route forms a "triangle".

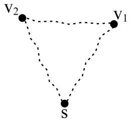

Figure 1: As in [5], we create closed routes by determining via-vertices v_1 and v_2 in a way that the length of the shortest paths between start-/end point s and v_1, v_1 and v_2, as well as v_2 and s, equals one third of the specified route length.

According to [5], this method guarantees to produce routes with a given length and only a maximum deviation, and can easily be extended to more than two via-vertices. Thus, we apply the same algorithm also with more via-vertices in order to get a more diverse set of candidates. Note that, since route generation is decoupled from creating actual recommendations, this method is interchangeable with any other algorithm that allows to find a number of cycles within a given graph, i.e. closed candidate routes in a map.

Besides constraining routes to a certain length, for later being able to recommend running routes several more aspects need to be taken into account. We have identified the following:

- Some graph edges might *not be suitable for running*, e.g. highways or closed parking lots. Consequently, these edges have to be identified and removed from the graph before applying the route generation algorithm. In case this leads to the starting vertex being in a small subgraph disconnected from the rest, a new starting point must be chosen.
- Shortest paths between vertices might *share edges*, i.e. route segments would be run twice. Also, segments of the shortest path towards a vertex x might be very *close to segments* of the path leaving x. Thus, users would, for instance, have to run on one side of a street, and return on the other. For these reasons, we introduce penalty values, which are assigned to edges already visited before. In addition, we calculate the area within the cycle that represents the route in the graph, and maximize this area to avoid long and narrow route shapes, but to create more rounded ones.
- Starting from the current location might lead to a set of candidate routes that later *do not allow to fulfill all requirements*. For instance, in case the distance to the nearest forest is more than $1/(n+1)$ of the desired route length, no route with n via-vertices will ever reach it from the original starting point. Fig. 2 illustrates two solutions: a) using a virtual starting point s_v as input for the route generation algorithm, and add the way towards and back from s_v, b) increasing the distance between s and v_1, and changing the other distances accordingly, so that the routes include more distant vertices.

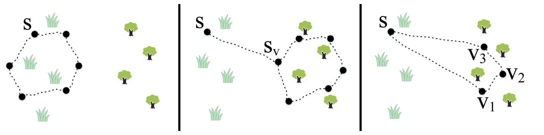

Figure 2: A route does not pass a desired area (left). As a solution, a virtual start point s_v can be introduced (center), or distances between vertices can be enlarged/reduced (right).

2.2 Creating Route Recommendations

The next step after having generated appropriate candidate routes is to rank them according to all properties that might be relevant for a runner with respect to his or her next workout. For this, we calculate scores for a number of criteria that we have identified to be important. Indeed, the following list is non-exhaustive and there might be more requirements some runners want to take into account. However, we in a first step aim at considering those that are, from our perspective, the most interesting ones, and in particular, can actually be implemented using available datasources.

1. *Length:* Especially for experienced runners constraining the route to a specific length is very important. We use the deviation of candidate routes (derived as explained in Sec. 2.1) from the desired length to determine a score.
2. *Uniqueness:* Maximum uniqueness of a route is reached when each edge is different from each other, i.e. runners do not have to run a certain path twice. Under the assumption that there is no meaningful reason not to maximize this value, we always try to reach a high score in this respect.
3. *Shape:* This score is defined by the area within a route's cycle (as explained above), and should be as high as possible.
4. *Lighting:* Runners who prefer routes with street lights might want this criterion to be considered after sunset, which is defined by the proportion of a route that is lit. This score can automatically be ignored at day time.
5. *Elevation:* The elevation score is defined as the amount of incline and decline on a route. Having a lot of height meters largely influences a route's difficulty, which can either be seen challenging (i.e. as a special kind of training) or as an undesired property of the route.
6. *Pedestrian friendliness:* Some ways or paths are more suitable for running than others, e.g. large streets or bikeways. The corresponding score describes the proportion of a route that is designated for pedestrians, e.g. small paths or tracks. As

special pedestrian zones raise this score as well, this may also help runners who prefer well-traveled routes.

(7) *Turns:* The number of turns a runner has to take refers to the complexity of a route. While some might see a high number to be an interesting feature, others might not desire this because it makes navigation more difficult or is inappropriate for a specific form of training (e.g. intervals). This score is calculated by means of the angles of adjacent edges.

(8) *Nature:* Running in cities can be exhausting and dangerous. Also, some runners might find it less attractive. Thus, the amount of nature is an important factor as more scenic routes may positively influence the running experience. Moreover, runners might enjoy, for instance, quietness and less air pollution. Accordingly, we introduce four different scores:
- *Trees:* Represents the proportion of a route leading through forests or segments being surrounded by trees.
- *Grass:* Represents the proportion a route goes through grass, meadows or farmland.
- *Sand:* Represents the proportion of a route crossing beaches or segments being surrounded by sand.
- *Water:* Represents whether lakes or oceans are visible, taking the distance to water into account.

(9) *History:* This score is related to the possibility of providing an opinion on routes ran in the past. Beyond the consideration of preferences expressed with respect to the aforementioned criteria, this allows us to automatically refine the suggestions to better reflect the current runner's taste. For this personalization step, similarities between items, i.e. routes, are calculated as in multi-criteria recommender systems based on all relevant item properties [1]. Then, the more similar a route, the larger the influence of the corresponding user rating (if available) on this score. For instance, if a user rated a route with lots of trees and few height meters very positive, a similar candidate route will receive a higher score.

Finally, we calculate an *overall score* for each candidate route. For this, we take the mean of the differences between the *individual scores* (as introduced above) and *desired values* for all criteria. These values can be either predefined, e.g. high for *Shape* and low for *Elevation*, set by the user initially (e.g. *Length*), or later during an interactive preference elicitation phase (e.g. *Nature*). Independent of the actual implementation of individual scores (see Sec. 3 for details on how we calculate them in our prototypical smartphone app), the overall score thus allows to rank the candidate routes.

3 THE *RUNNERFUL* APP

Runnerful is a prototypical Android app that implements our framework. We use the *OpenStreetMap API* to collect map data. Using edge annotations contained in this data (e.g. to ignore highways), we then create a graph to apply the route generation algorithm as described in Sec. 2.1. To find shortest paths between via-vertices (we use 2–4, after initial pretests), we use a modified *A^* search algorithm* that penalizes nodes already visited (we vary distances and set different starting points, as described in Sec. 2.1). Scores are calculated for all criteria described in Sec. 2.2: For some criteria, such as *Length* or *Shape*, we calculate scores based on the graph data itself. For others, such as *Lighting* or *Pedestrian friendliness*, we rely on edge annotations provided by *OpenStreetMap*. For *Elevation*, we additionally send requests to the *Google Maps Elevation API*. Regarding the amount of *Nature*, we take surrounding areas and their *OpenStreetMap* annotations into account: Using a ray-casting algorithm, we determine whether segments cross forests, farmland or beaches. The proportion of edges for which this applies then defines the respective score. Moreover, we calculate the distance of every route point to areas that represent water.

As user input, the app initially only requires the desired route length. Then, taking the current GPS position, recommendations are generated. Fig. 3 shows a screenshot with two suggested routes: The user has requested routes of 4 km length. When looking at the actual values, both routes have high accuracy in this regard, which is reflected accordingly in the net diagram (dimension depicted by yellow ruler). As also shown in the net diagram, both routes go through some forest, which can easily be seen in the map (route 1 through the Zoo in the north, route 2 through the community garden in the south). Furthermore, the routes strongly differ in shape (depicted by the oval in the net diagram): While route 1 fills a large area and avoids visiting streets twice, this is very different for route 2, which has a more narrow shape with route segments close to each other or even ran multiple times. Using the arrows left and right to the net diagram, the user can scroll through the results. The buttons below allow to request a new set of recommendations, run the recommended route (which leads to a new screen for route navigation, showing workout duration and progress of the run), and critique the current recommendation.

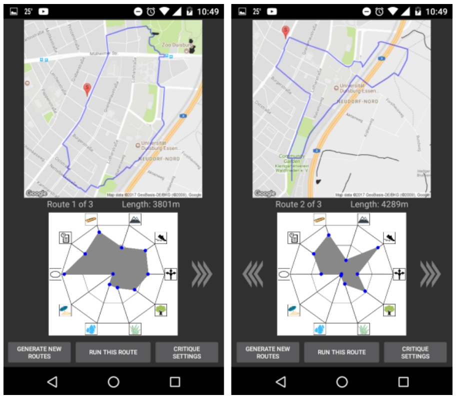

Figure 3: Two routes recommended by *Runnerful*: The user is presented with a map view as well as a net diagram showing the scores for the different criteria.

Fig. 4 shows a part of the screen for critiquing: The user can drag criteria he or she wants to be considered less or more into the respective areas. This decreases or increases the *desired values* used to calculate the *overall scores* of the candidate routes.

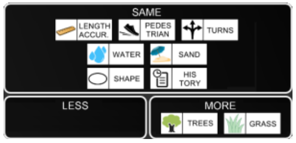

Figure 4: The user is critiquing the recommended route.

After finishing a run, the user can express his or her opinion by rating the route. This rating then influences the *History* score as described in Sec. 2.2 to give more personalized recommendations.

4 EVALUATION

We conducted a first user study as a proof-of-concept for the application of our framework. We recruited 11 participants (6 female) with an average age of 28.18 (*SD* = 11.42), 64 % students and 36 % employees. They had to use their own Android smartphone to test *Runnerful*. Apart from a short introductory video, no further help was provided, nor were participants controlled in any way. The study took place over two weeks, with the only task to run at least two recommended routes. Before the experiment, participants had to fill in a questionnaire we used to elicit demographics, fitness using IPAQ [4], running route preferences and previous experience with running apps. Afterwards, we assessed usability by means of SUS [3], and used items from [6] to assess recommendation quality and related aspects. Items were assessed on a positive 5-point Likert scale. We also recorded finished routes and corresponding ratings.

Participants reported that they performed vigorous physical activities for $M = 46.73$ min ($SD = 28.68$) on $M = 3.63$ days ($SD = 2.06$) in the week prior to the experiment. Most of them reported that nature is an important route property (9). Length (2) and elevation (2) were mentioned less frequently, which could however be due to our sample, without any competitive runners. Only 3 stated to have never used a running app before. Nevertheless, none of the participants ever tried a route recorded by another community member. Most of them reported to spontaneously decide for a route (82 %), but 6 stated to sometimes use a map or ask friends for advice.

We recorded 17 workouts from 9 participants (recording failed in two cases). Routes received average ratings ($M = 3.05$, $SD = 0.80$), and slightly higher ones when critiques were applied ($M = 3.17$, $SD = 0.69$). However, participants did not use critiquing very often, possibly because it was not displayed prominent enough. They stated that effort for receiving recommendations was low ($M = 2.00$, $SD = 0.98$), while perceived recommendation quality was above average ($M = 3.32$, $SD = 1.09$). When asked whether routes had the expected amount of desired properties, results were broadly average (e.g. for trees and forest: $M = 3.11$, $SD = 1.29$). Yet, this could be due to parametrization, favoring criteria such as length and elevation, together with our sample. Nevertheless, participants stated that they almost always found a suitable route ($M = 3.55$, $SD = 1.37$), which was most often novel ($M = 3.45$, $SD = 1.30$). Usability was rated as "good" (SUS-score of 80). Overall, it thus seems that our approach is in principle valid and appreciated by users. Fine-tuning, e.g. of the calculation of individual and overall scores, is, however, clearly needed. Still, most qualitative comments were concerned with aspects of our preliminary implementation (e.g. issues with the navigation function) rather than of our approach in general.

5 CONCLUSIONS AND OUTLOOK

In this paper, we discussed the challenges arising when recommending running routes, such as availability of data that is rich enough to adequately personalize these routes, and presented a framework describing aspects that have to be taken into account for this personalization to be effective. *Runnerful*, our proposed app, exploits easily accessible map datasources to generate routes of user-specified length. Then, by further processing the map data, we rank these candidate routes according to individual requirements. Critiquing allows the runner to interactively refine the results.

While the implementation of our framework together with the study shows the potential of the underlying approach, there is still room left for improvement. Also, more comprehensive evaluation with a larger number of users performing more workouts is required. For instance, the influence of the history criterion could not yet be adequately investigated. On the other hand, exploiting the user's running history more extensively might help to reduce interaction effort even further by letting the system learn which criteria are most important for him or her. Moreover, there exist contextual factors that could additionally be considered, such as current weather for recommending runs through the forest in midday heat or avoiding steep climbs in case of icy roads. Also, current fitness state as well as training fatigue could automatically be integrated when calculating the scores. Beyond that, practical issues such as scalability will be subject of future work: Our prototype lacks efficiency when generating longer routes which is necessary for experienced runners, but also in case it is adapted to e.g. cycling. Generating 5 km routes took up to 1 min, but in a dense city environment and with a rather average VM running the algorithm in the background. Thus, this is not a principle limitation, but requires additional preprocessing of map data, a more advanced selection mechanism of candidate routes, and highly depends on server infrastructure and used datasources. In general, the app needs technical improvements: For example, to make it more useful outside the study context, users should receive more support for following a route, e.g. by spoken navigation instructions. Nevertheless, and despite small sample size and the fact that some study results are still rather average, we think that the prototypical implementation of our framework successfully demonstrates our approach, and can thus be seen as a promising starting point for further research.

REFERENCES

[1] G. Adomavicius and Y. Kwon. 2015. *Recommender Systems Handbook*. Springer US, Chapter Multi-Criteria Recommender Systems, 847–880.

[2] J. Berndsen, A. Lawlor, and B. Smyth. 2017. Running with recommendation. In *HealthRecSys '17*.

[3] John Brooke. 1996. SUS – A quick and dirty usability scale. In *Usability Evaluation in Industry*. Taylor & Francis, 189–194.

[4] C. L. Craig, A. L. Marshall, M. Sjorstrom, A. E. Bauman, M. L. Booth, B. E. Ainsworth, M. Pratt, U. Ekelund, A. Yngve, J. F. Sallis, et al. 2003. International physical activity questionnaire: 12-country reliability and validity. *Med. Sci. Sport. Exer.* 35, 8 (2003), 1381–1395.

[5] A. Gemsa, T. Pajor, D. Wagner, and T. Zündorf. 2013. Efficient computation of jogging routes. In *SEA '13*. Springer, 272–283.

[6] B. P. Knijnenburg, M. C. Willemsen, and A. Kobsa. 2011. A pragmatic procedure to support the user-centric evaluation of recommender systems. In *RecSys '11*. ACM, 321–324.

[7] F. Mueller, J. Marshall, R. A. Khot, S. Nylander, and J. Tholander. 2014. Jogging with technology: Interaction design supporting sport activities. In *CHI '14*. ACM, 1131–1134.

[8] D. Münter, A. Kötteritzsch, T. Islinger, T. Köhler, C. Wolff, and J. Ziegler. 2012. Improving navigation support by taking care of drivers' situational needs. In *AutoUI '12*. ACM, 131–138.

[9] D. Quercia, R. Schifanella, and L. M. Aiello. 2014. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *HT '14*. ACM, 116–125.

[10] N. Runge, P. Samsonov, D. Degraen, and J. Schöning. 2016. No more autobahn! Scenic route generation using Google's street view. In *IUI '16*. ACM, 147–151.

[11] R. Schaller and D. Elsweiler. 2014. Itinerary recommenders: How do users customize their routes and what can we learn from them?. In *IIiX '14*. ACM, 185–194.

[12] B. Smyth and P. Cunningham. 2017. A novel recommender system for helping marathoners to achieve a new personal-best. In *RecSys' 17*. ACM, 116–120.

[13] B. Smyth and P. Cunningham. 2017. Running with cases: A CBR approach to running your best marathon. In *ICCBR '17*. Springer, 360–374.

[14] P. Stroobant, P. Audenaert, D. Colle, and M. Pickavet. 2018. Generating constrained length personalized bicycle tours. *4OR* (2018), 1–29.