Assessing the Credibility of User-Generated Content For Real-Time Refinement of Forecast and Nowcast

Daniil Shitov
Faculty of Computer Science
National Research University Higher School of Economics
Moscow, Russia
dsshitov@edu.hse.ru

***Abstract*—Weather forecasting and nowcasting have always been really tough challenges for scientists, meteorologists, and programmers due to complex weather patterns, large amounts of data, and the limited capabilities of modern servers. However, algorithms are constantly being improved, and more and more parameters are taken into account when calculating the forecast, such as user-generated content. On the one hand, this is very useful information because web and mobile applications are made for people, and it acts like feedback. It can also be taken into account in subsequent calculations, so other users will see more accurate data. On the other hand, there are users who specifically send incorrect information, known as fraudsters, and users whose opinions are just very different from the rest, too. The paper will focus on how to take user-generated content into account most effectively.**

***Keywords*—User-generated content, forecast, nowcast, ground truth**

I. Introduction

In our time, there are many data sources through which weather conditions can be determined: satellites, radars, stations, camera recordings, among others. All sources have their own advantages and disadvantages. For instance, stations provide good accuracy but are not found everywhere. Conversely, satellites offer wide coverage but can be inaccurate, also, regularly requesting data from them is expensive. Thus, it is impossible to choose a single specific source to rely on when making a weather forecast. For this reason, data is collected from different sources, combined and aggregated using mathematical statistics and machine learning methods. Historical data can also be used for constructing weather forecasts to identify patterns in them and extrapolate to the present. As can be seen, there is a vast amount of data available for forecast construction, and they are all very diverse. Additionally, it must be considered that Earth is a giant planet with a large atmosphere. Calculating an absolutely accurate forecast for the Earth is impossible, even on the most powerful modern computers, partly because there simply isn't enough processing power, and partly because weather change patterns are not fully understood - currently, there are hundreds, or even thousands of meteorological parameters that are interconnected and not so easy to calculate. Therefore, constructing a weather forecast is not only a computational task but also an approximative one. It is necessary to understand how the Earth can be divided into a grid and how to calculate the forecast in each pixel, this article will not touch on a comprehensive description of these details. Further, the article will discuss user-generated content and how it helps improve the forecast and validate the current nowcast.

The remainder of this article is organized as follows. In Section 2, related literature will be reviewed and some crowdsourcing methods will be briefly described. In Section 3, the methodology is introduced. Section 4 discusses the expected results. Section 5 concludes the paper.

II. Background on Crowdsourcing

User-generated content helps to improve nowcasting, one could even say that it constitutes essential data that should be relied upon when calculating nowcasts. However, there are several issues. The first is that there are not users everywhere willing to provide information, which is related to the fact that users do not often need to check the weather throughout the day, and even fewer visit an app or website for this purpose, with even fewer ready to provide information about the current weather situation in their location. Nevertheless, it is precisely in such places that user data plays a significant role, as less populated areas usually have fewer sources of data. The second problem is frauders, in other words, people who intentionally provide maliscious information.

To simplify, the task will first be solved only for precipitation. Users can report in the interface whether there is precipitation in their location at the current moment. Thus, our task is reduced to the well-known crowdsourcing problem: decision-making task with single right choice, which is inherently unknown.

There are studies [1] that compare algorithms for solving such a task. It is planned to select the algorithms that have performed the best according to *F1-score* and *Accuracy*: D&S [2], BCC [3], LFC [4]. These methods are based on an EM-algorithm [5], that allows to iteratively calculate the weight of the voice for the user. The article also suggests some optimizations for improving the detection of ground truth, which could be applicable to our task as well.

III. Methodology

A. Ground truth

The question of how to determine which algorithm ultimately provides the best result remains open. To do this, the algorithm's performance is usually checked on data where the correct answer is known in advance. In the case of precipitation forecasting, such data do not exist, because all sources can be inaccurate. In that case, there is no choice but to select the sources in which we have the most confidence. Meteorologists have reported that data from stations around the world and radars, but only within the territory of Russia, can be considered the ground truth. The pipeline for selecting an algorithm is as follows: run calculations of the chosen methods on historical log of reports, then compare the final results with the ground truth data and calculate the *F1-score* and *Accuracy*.

B. Implementation

Choosing an algorithm is one side of the task. The other side is no less challenging - its implementation. There are many sources for calculating the forecast, so new data often appears, and old data may be updated, meaning the forecast and nowcast needs to be recalculated regularly. The forecast calculation pipeline involves a lot of data, sources, stages, and models. Currently, the trend is to create such complicated services on a microservice architecture. This approach simplifies development and release, while in our company, there are all the tools for configuring routing between microservices, for tracing requests, and for conducting experiments. A separate service is responsible for user reports, with functionality to:

1) Save user reports.
2) Send aggregated results for a given time and pixel.
3) Be able to handle tens of millions of users and tens of millions of pixels. This sets the performance requirements for the algorithm to be implemented in the microservice.

There are several approaches to working with user credibility:

1) Work directly with the database.
2) Save a snapshot of the database in RAM, perform aggregations on data from it. Update the credibility in real-time in the database.
3) Perform aggregations on the snapshot loaded into memory, and make all credibility updates offline.

The first approach can be implemented with modern databases, such as PostgreSQL or MySQL, for instance, if they are configured with sufficient resources. However, in our case, all microservices use a single database, which already experiences significant load. We consider it a flawed approach when the database load depends on the service traffic, making the behavior unpredictable and dangerous. Also, it is worth noting that recalculating credibility online is not essential, it doesn't change significantly over a few nowcasting steps. Therefore, the third approach seems the most valid. If credibility is updated offline, additional data not available at runtime can be used. For example, in calculating the ground truth for a nowcasting step, the confidence of our model in the presence of precipitation can be taken into account.

C. Experimaents

When the logic within a large and production-critical microservice changes, it is essential to ensure the modifications work and actually improve the service before release. The main difficulty in evaluating changes in nowcasting calculations stems from the fact that all possible data are already being used for calculation. There is no absolutely correct ground truth result, and validating the model's work on the data it used for the calculation is obviously the wrong approach. So how should one check? There are two approaches:

1) **Online check.** It is possible to conduct A/B testing. A/B testing is a technology that allows the comparison of two versions of a product. In short, a certain sample of users will see and work with one version (A), while all others interact with version B. In both cases, certain metrics are measured, which can be compared at the end of the testing to decide which version performs better. In our task, different samples of users can be shown different results of the nowcast calculation. Naturally, this will change the number and nature of user reports. Then, all the data can be analyzed, trying to find the ground truth and compare it with that.
2) **Offline check.** For the current model, it is known which data were fed to it in the past. The model also calculates a forecast for some time in the future. That is, the model computes an array of results $[r_{t_0}, r_{t_1}, r_{t_2}, \dots, r_{t_n}]$. In such a case, the new model can be given the same data and asked to calculate the result, but only considering its own report credibility, resulting in $[r_{t_0}^{new}, r_{t_1}^{new}, r_{t_2}^{new}, \dots, r_{t_n}^{new}]$. In this case, it will be possible to compare the results for future predictions, i.e. $[r_{t_k}^{new}, \dots, r_{t_n}^{new}]$ and $[r_{t_k}, \dots, r_{t_n}]$. At the same time, these moments also eventually become part of the past, meaning the more accurate forecast known about them can be used as ground truth.

IV. Expected results

V. Conclusion

In conclusion, assessing the credibility of user-generated content for real-time refinement of forecast and nowcast is a comprehensive task that includes researches upon various areas. Given the uncertainty in selecting methods for determining credibility and ground truth, coupled with the challenges in conducting experiments, it can be stated that this problem possesses both a programming and a research dimension. Nevertheless, it is clear that the task is useful and should bring some improvements to the calculation of weather forecasts and nowcasts, because it makes the data more accurate. If this does not happen, it would more likely be a reason to doubt the correctness of the experiments and data processing, rather than the usefulness of the task.

References

[1] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" in

Proceedings of the VLDB Endowment, vol. 10, 2017, pp. 541–552.

[2] A. Dawid and A. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," in *Applied statistics*, 1979, pp. 20–28.

[3] C. Kim, H and Z. Ghahramani, "Bayesian classifier combination," in *AISTATS*, 2012, pp. 619–627.

[4] V. Raykar, S. Yu, L. Zhao, G. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," in *JMLR*, 2010, pp. 1297–1322.

[5] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," in *Journal of the royal statistical society*, ser. B (methodological), 1977, pp. 1–38.

Word count: 1545