Capturing playing style with Weighted Gaussian Mixture Models

Tom Decroos and Jesse Davis

KU Leuven, Department of Computer Science

Abstract. bla

1 Introduction

Mining patterns in soccer data: difficult because mix of three different data types: categorical, continuous, angular

1.1 Sparsity vs expressibility

- discrete pattern mining algorithm, not expressive enough - in NMF, smoothing was essential in a fine-grained grid. Lots of useless cells (e.g., shots), NMF needed many examples, e.g., did not work for most types.

1.2 Soft clustering can solve this issue

1.3 Use the right clustering algorithm per data type

2 Data

2.1 The problem with results in actions

3 Methodology

3.1 Mixture models with categorical, Gaussian, and Von Mises distributions

3.2 Weighted mixture models

3.3 Hierarchical mixture models

The hierarchical mixture model takes as input another hierarchical mixture model to pre-split the data.

3.4 Tuning the number of components

4 Experiments

4.1 Illustration: In what way do Barcelona and Leicester differ?

4.2 Which teams have a consistent playing style and which teams have a variable playing style?

4.3 Can we characterize teams defensive style by how they make other teams play differently from their usual style?

4.4 Moving from actions to sequences

4.5 De-anonymizing teams

4.6 De-anonymizing players

Acknowledgements

Tom Decroos is supported by the Research Foundation-Flanders (FWO-Vlaanderen). Jesse Davis is partially supported by KU Leuven Research Fund (C14/17/07, C32/17/036), Research Foundation - Flanders (EOS No. 30992574, G0D8819N) and VLAIO-SBO grant HYMOP (150033).

References

Q: What is this paper about? A: it is about a novel soft clustering method for event stream data that enables a novel representation that makes it really easy to represent players and teams as the sum of their actions.

Q: What is novel about the representation? A: We represent each action as a probability vector over many different distributions that could have generated the action. These distributions can be viewed as "prototypical" actions. We can then easily represent playing style of players and teams by simply summing the probability vectors of their actions. The novelty is that (1) each action is a vector with continuous values rather than one discrete assignment to a distribution/cluster. This improves the sparsity issue. (2) Each distribution that represents a prototypical action is found in a data-driven way rather than manually created. (3) The method is able to represent ALL possible actions rather than just a subset of them. (4) Previous representations of soccer data were often at player or team-level. By having a representation at the action level, we can naturally represent and mine sequences of actions.

Q: What exactly is the sparsity issue? To detect playing style using a specific technique, you have to create a representation for an action that is compatible with the technique.

The purpose of the new representation for actions is to find a representation that allows the technique to aggregate and count over actions. Most often this is done by discretizing the continuous attributes of an action (e.g., location, direction) such that similar actions end up in the same category. We then aggregate actions by simply counting the support of each bin.

For example, when applying a pattern mining algorithm, you convert an action to an itemset and mine frequent combinations of specific items in sequences of itemsets. When applying a matrix decomposition technique, you have to convert an action to a vector of fixed length. The most common approach to convert an action to a vector is to overlay a grid on the field, represent the action as the cell in which it appears and one-hot-encode the cell. Both approaches involve somehow discretizing the continuous features of an action such that two actions that are similar are actually the same in their new representation and can thus be counted to find common or interesting patterns.

The problem with discretizing the continuous attributes of an action is the following. For each discretized representation, there is a trade-off between its expressiveness (i.e., the number of categories in the discretized continuous attribute) and mineability (i.e., are there enough examples of each category such that interesting patterns have enough support in the data). If the expressiveness of an action representation is too low, we can mine patterns but the found patterns are too general and not very interesting for a domain expert (See X). However, if the expressiveness is too high (i.e., an extremely fine-grained grid), no interesting patterns appear as each pattern's support is too low (in a fine-grained grid, many cells have a support of zero or 1).

What we found in practice from earlier research is that the mineability of a representation becomes too low before we can make the discretized representation expressive enough such that the resulting patterns are interesting for a domain expert. We call this the sparsity issue.

Q: So how do you fix the sparsity issue? A: The key insight is (1) we no longer represent each pattern by a discretized category, but represent it as a probability distribution over bins, and (2) We no longer perform the binning ourselves but construct the bins in a data-driven way using mixture models. By doing this using mixture models, we always find the best possible representation in terms of mineability for a requested level of expressivity. In addition, we find that the best possible representation in terms of mineability also often leads to the most human-interpretable bins for a requested expressiveness level.

Fixing this sparsity issue also allows us to do something new: hierarchical binning over different feature sets wherein we perform a new binning within a bin, but on a different feature. (concretely, we first bin actions per type, next we construct location bins within each type bin, and finally construct angular bins within each location within each type bin. This allows us to construct bins that adequately describe an action's type, location, and direction without the representation being too sparse to aggregate.

Q: Why were previous clustering methods not good enough? A: In our first attempt we manually clustered actions using fixed sections of the field. We found that the pattern-algorithms broke down when we went too fine-grained (e.g., basically just finding small variations of the same boring patterns over and over again, while the interesting patterns got lost in the big forest). However, the representation for actions that worked well enough with the pattern mining algorithm to actually produce unique enough patterns was too broad and hence the resulting patterns were not descriptive or specific enough to be useful.

In our second attempt, we divided the field into a grid, represented players with heatmaps per action type and used NMF to find the building blocks of these heatmaps. Downsides of this are (1) that there is no representation for individual actions, (2) a lot of grid space goes wasted when actions only occur on a small part of the pitch (e.g., goals), (3) this heatmap representation was too sparse, we had to apply a blur, however, we don't like that we are modifying the data this way and that the results we get also greatly depend on the parameter for this blur. There is also no data-driven way to determine what the best value for this blur parameter should be. (4) Individual characteristics of actions go lost through this aggregate representation of actions. E.g., goalkicks on the left tend to end up on the left of the field while goalkicks on the right tend to end up on the right of the field. However, this connection between the goalkick's start and end location is lost when we only consider the total heatmap of goalkeepers. (5) The NMF breaks down if the data is too sparse and it is thus not a valid representation for the complete playing style of a team or player.

Q: How does your new method fix these issues? A: 1) one action = one vector, 2) representation is based on the raw event data, no prior discretizing is needed 3) Because it is soft clustering, we can avoid the sparsity issue for a lot longer, since one data point can serve as an example for many clusters, hence you need fewer actions to get an accurate picture of a team or players' playing style. There is no chance involved with being slightly to the left or to the right of a discretized feature. (4) Our new representation obtained through this soft clustering is respectful of an action's individual characteristics, e.g., its type, its start-location and the direction in which it sends the ball. 5) Our soft clustering method does not break down, even when dealing with very sparse and little data.

Q: Great, what exactly do you mean when you say that no prior discretizing is needed? A: In both the pattern-based and grid-based approaches, you take a continuous variable ((x,y)-location) and discretize it to either a zone on the field (midfield, left-flank) or a cell in a grid (e.g., a 50x100 grid). Now the variable is no longer continuous and this leads to loss of information. In addition, now that the variable has been discretized, it can no longer serve as an example of a zone or cell it was really close to, but just not close enough. The NMf-based approach can somewhat get around this by applying a blur on the heatmaps, but this is not a very elegant solution.

Q: How does your method solve this issue? You need to aggregate data somehow in order to count data and use them as examples for patterns, right? A: The trick is that we no longer map one example/action to one discretized category, but probabilistically map an example to multiple clusters. A cluster is thus no longer a fixed zone, but is a probability distribution over the feature space. We search for the best clusters that fit the best with the data points using the Expectation Maximalization algorithm. In addition, our clusters are multi-dimensional and use the right distribution per feature.

Q: What do you mean by ”the right distribution” per feature space? A: Different features live in different spaces. For example, we represent an actions by three features: 1. its type, 2. its start-location 3. its direction. Its type is categorical, hence you need a categorical distribution, for start-location you need something spatial such as a binomial gaussian distribution and for direction you need an angular distribution such as the Von Mises Distribution.

Q: Wait, what is a Von Mises Distribution and why do you need it? A: Gaussian distributions live in infinite space, a variable can take on any value in the interval [-inf,+inf] and it will receive a probability of being generated by a distribution. However, variables in the radian space live in the interval [-pi,+pi]. In addition, two points -pi+sigma and pi+sigma with sigma really small are actually really close together and probably belong to the same distribution. However, this is impossible to represent with a gaussian distribution! Hence, this is why Von Mises distributions exist, they are basically the counterpart of Gaussians in angular space.

Q: why is it a big deal that your clusters are multi-dimensional, isn't that always the case with any clustering algorithm? By clustering our data per small feature-set, we ensure (1) that our clusters always remain interpretable for humans, and (2) that we can use the right distribution per feature. Although on second thought that last one is not really true... We could perfectly make up one cluster made of three different distributions per feature set where these three distributions per cluste are independent and just do one big gigantic EM-optimalization for like 200 clusters.

Q: What do you call the collection of clusters that represent the data? A: A Mixture Model (MM). A mixture model is a generative model for the data set. It uses a collection of distributions D_i with a prior distribution over the collection of distribution D_i.