Dirichlet Processes for Injury Prevention using Running Activity Categorization

Scott Sikorski
nqj5ak@virginia.edu

Abstract

Running is generally considered to be a highly individualistic sport where one can run at their preferences. This has been reinforced through wearable running watches that introduce a wide multitude of running related metrics to the user, during and after each run. This data contains valuable training information and trends that models can identify. Particularly, there exist overtraining trends which can lead to both acute and longer term stress injuries, sidelining runners for weeks up to months. Because everyone's running data is vastly different, an adaptable model is needed. This is presented in the form of a Dirichlet Process Mixture Model (DPMM) that can dynamically grow and shrink clusters with more data and training iterations. I present a DPMM implementation for running categorization with a novel hierarchical and standard evaluation metrics to assess its performance. 2 separate dataset partitions, random and recent, are used to check immediate and long-term predictive capabilities. DPMM achieves upward of 96% training and 94% hierarchical accuracy on the tiered results, a 5-23% increase compared to the SkLearn baseline.

1 Introduction

Long distance running has been an increasingly popular method to promote healthy lifestyles offering physical and mental benefits. It has cardiovascular and immune system benefits Ruta et al. [2024], can be a mental health outlet, or provide a source of competition and motivation. However, there can be high risks as 17% - 50% of people will suffer a running-related injury per year Taunton et al. [2002]. It was found that greatly increasing factors rapidly or over time, such as mileage, time running, and intensity, resulted in a higher injury incidence rate Van Gent et al. [2007]. This highlights the need for personalized approaches to managing training loads to prevent overtraining and reduce the risk of injuries. Nowadays, wearable fitness watches provide a multitude of data from each run. Heart rate, time, and distance can all be automatically computed and tracked to inform the effort level. These factors can define a person's baseline and progression over time. While there's some variability, average heart rate for a given pace and length of time should naturally lower over the course of becoming more used to running and aerobically able.

Accurately categorizing these runs and identifying deviations from the norm can provide valuable insights into an individual's fitness trajectory and potential risk factors. However, traditional clustering methods, k-means, require a predefined number of categories or assumptions about the data distribution. This does not account for running variability and severely limits a general model where we cannot gather this info about each person. As a result, Dirichlet Process Models Teh et al. [2010] offer a beneficial approach to labeling runs unique to each person. The growing nature of the clustering allows for training plans that include any number of run types. A person is not confined to what other runners are doing or designating universal terms. Additionally, a DPM approach can allow for natural fitness growth over time by the underlying Gaussian Mixture Model to alter with more data. Or when a run is distinctively different from recent data of what the model or person believes, there exists a possibility of overtraining.

This is especially useful for those who are new to running and cannot separate injury pain from soreness or fatigue. Introducing a model that updates dynamically with incoming data to reflect a runner's growth and changing fitness levels can be a game-changer for this demographic. In turn, it lowers the barriers to entry by offering a structured, adaptive, and evidence-based approach to training, mitigating the risks that often deter people from continuing their running journey. Preventing injuries that might otherwise derail a new runner's motivation ensures they can focus on achieving their health and fitness goals, fostering long-term engagement with running as a sustainable lifestyle choice. This proactive, data-driven approach not only promotes safety, but also aligns with the original intention of improving health and well-being, preventing setbacks that could lead to frustration or early quitting.

2 Relevant Work

Clustering has been found to be a beneficial model in health care to find underlying data patterns in unsupervised Marlin et al. [2012] and time series settings Aguiar et al. [2022]. These are constrained by the k-means nature Alsayat and El-Sayed [2016], Haraty et al. [2015] of most standard clustering algorithms. They are required to have prior knowledge of data distributions and know the cluster dimensionality. This is not always a viable solution to solving general individualistic health care problems.

Bayesian nonparametric models, like Dirichlet Process Mixture Models (DPMMs), have been used for their infinite flexibility in clustering. Teh Teh [2009] and Orbanz Orbanz and Teh [2010] Orbanz [2012] present introductions to building DPMMs. They focus on and present the Chinese Restaurant Process (CRP) and Stick Breaking Process as prior calculations in the clustering mechanisms. These approaches emphasize the dynamic nature of DPMMs to better handle variability in the number of clusters in a dataset. This has great opportunities for health care domains where it is required to be flexible. This includes running which is greatly explored to its highly individualistic nature.

These applications become even more apparent when we consider traditional methods. Rogers et al. [2019], Richardson and Hartman [2018] suggest approaches for using non-parametric clustering for health care settings when regular regression models are insufficient. They highlight using these techniques for complex datasets with a variety of predictions. The data can be reduced without strong underlying distribution assumptions of both the variables and their corresponding clusters. By forgoing data assumptions, the model can evolve. This is incredibly necessary in tracking running metrics due to the dynamic nature of injury and fitness growth progression.

Current data science Python packages offer resources for fitting a DPMM. This comes in the form as Scikit-Learn's Bayesian Gaussian Mixture Xue and Guillemont [2024] class. The weight prior is set to that of Dirichlet Process with a given α value. This Bayesian Gaussian Mixture implementation offers a generic clustering that can be applied to any datasets. This exposes the need for developing a dedicated DPMM for health care applications to identify data patterns. I focus on identifying training patterns that illustrate rapid, higher stress.

3 Implementation Details

3.1 Preprocessing

The dataset that I used is a collection of my activities over the past year, December 2023 - December 2024, exported from Garmin Connect `https://connect.garmin.com`. These activities can consist of running activities, outside, track, or treadmill, and non-running activities, such as biking or swimming, which are not included. There are 557 rows of which 368 are eligible running activities. The initial Garmin dataset includes various columns, most of which are not necessary. Thus, 9 attributes are extracted which I find the most relevant and are fulfilled by all running activities. This includes average and best pace, power, and heart rate, moving time, distance, and calories. These are the main metrics given by most Garmin running watches that are raw data values (nothing computed by Garmin themselves).

These are then converted into standard units of meters, seconds, and meters per second. Any max values that were not given were replaced with the respective average. In conjunction, a row was dropped if any metric was not given or equaled 0. The data is then manually annotated by myself into one of seven categories: Training Run, Workout, Long Run, Warmup/Cooldown, Recovery, Shakeout, Race. See section 8.2 for extra info on these groups and distinctions.

Given the modified Garmin running activity dataset, the data frame is normalized using a MinMax scaler with a feature range of (0, 1). My data was fairly clean and did not include any activities that drastically corrupted the distribution of feature values. Generally, corrupt data would be filtered out by one of the attributes being invalid. However, more preprocessing and cleaning could be done to automatically detect corrupt values.

Lastly, Principal Component Analysis Abdi and Williams [2010] was conducted to reduce the dimensionality of the attributes to 2d. This was done to minimize the gaps within the data. Because DPMM extends to infinite clustering, at full dimensionality, points were reluctant to join an existing cluster unless forced by a set maximum number of clusters.

3.2 DPMM Fitting

α	ν	λ	μ_0	Ψ
3	d + 11	diag($1e^{-}3$)	$[.25]^d$	$1.5 \cdot cov$(data)

Table 1: DPMM Hyperparameters

The primary aspect of fitting DPMMs is the infinite number of clusters that are dynamically generated and destroyed. Algorithm 1 details the pseudocode for assigning data points to clusters. The assignments are initialized such that only the first data point s_0 is assigned to cluster 0.

```
def fit():
    while Not Converged do
        foreach data point $s_i$ in dataset do
            Remove $s_i$ from its current cluster if applicable
            Cluster = predict($s_i$)
            Assign $s_i$: Cluster
        if $s_{k,i} == s_{k,i-1} \; \forall s$ then
            Converged = True
    return Assignments
```

Algorithm 1: Fitting DPMM

```
def predict(s):
    foreach cluster $c_k$ do
        Likelihood: $p(c_k|\theta) = f_{\mathcal{N}}(s, \mu_{c_k}, cov(c_k) + \lambda)$
        Prior: $p(\theta) = \frac{n_{c_k}}{n+\alpha}$
        Posterior: $p(\theta|c_k) = p(\theta) \cdot p(c_k|\theta)$
    // Find new cluster probability
    Draw from Inverse-Wishart: $\sigma^2 \sim \mathcal{W}(\Psi, \nu)$
    $\mu_{c_{k+1}} \sim \mathcal{N}(\mu_0, \sigma^2)$
    New Likelihood: $p(k+1|\theta) = f_{\mathcal{N}}(s, \mu_{c_{k+1}}, \sigma^2)$
    New Prior: $p(\theta) = \frac{\alpha}{n+\alpha}$
    New Posterior $p(\theta|c_{k+1}) = p(\theta) \cdot p(c_{k+1}|\theta)$
    Cluster = $\arg\max p(\theta|\cdot)$
    return Cluster
```

Algorithm 2: Predicting a data point's cluster

An iteration of fitting begins by taking the i-th data point $s_i, \; i \in [0, n)$ in the dataset and predicting the cluster to join. It calculates the posterior probability for each available cluster and to form a new cluster. For the k-th available cluster, the likelihood draws from the underlying multivariate-normal probability density function using the given data point. The mean and a regularized covariance of the cluster data is supplied to the PDF. The prior is calculated according to the existing cluster CRP that is standard to DPs. The covariance requires this regularization term as for some points it would cause the likelihood to be 0 for all clusters. This allowed the posteriors not to be all 0 even with a high prior or if one data point existed in the cluster.

The new cluster posterior probability differs in the likelihood and prior. The model finds new cluster covariance and mean by drawing from the multivariate normal conjugate prior distribution, the Inverse-Wishart. The degrees of freedom and scaling factor are determined by the dataset attribute dimension and the upwardly scaled data covariance. Now that we have a covariance from the Inverse-Wishart, the mean of a new cluster is simply drawn from a normal distribution of the original mean with that covariance. Likelihood is similarly calculated as before from the multivariate normal PDF. Then the prior takes on its new form from the CRP. With all the posterior probabilities, the newly assigned cluster is the cluster with the greatest probability.

Once we make a fitting pass through all the data, we make our convergence criterion check. It checks if any data point was assigned to a different cluster from the previous iteration. I view this as a strict criterion as with higher values of α, the points are more open to changing clusters which may result in a max iteration check hit. For my implementation, it was able to converge in less than 20 iterations in under a minute.

3.3 Using the Trained DPMM for Prediction

We fit the DPMM and have the underlying multivariate normal distributions defined for each cluster. The predict algorithm 2 is used during training and can be applied to find the cluster a data point should join. This point can either not be added to the assignments, used for testing, or be added to influence the next predictions. The latter is how an in system application would deploy the model as it receives daily data. The user would give us what they label the run as from the mentioned preset categories acting as the true label. From this prediction and user evaluation, insights can be provided given how difficult an effort seems compared to previous completions of the same run type. Predictions that vary enough from recent training trends to form new clusters or seem as a harder effort warrant notifying the user of a departure from they are used to handling. This may be early signs of overtraining or a positive reinforcement message that they should focus on extra recovery to be prepared and not risk injury.

4 Experimental Design

4.1 Scikit-Learn Baseline

To evaluate my DPMM implementation, I chose SkLearn's general package Bayesian Gaussian Mixture with the Dirichlet Process arguments. The maximum number of clusters was set to 50 for both implementations. A maximum is given to reduce saved memory requirements as it generally has no impact on the result unless it's close to true number of clusters. In both implementations, this maximum was not reached at any point during training or testing.

The main difference between implementations is the α concentration hyperparameter. This was set to 300 for all experiments which highly encourages forming new clusters. For $\alpha < 300$, it would usually only form 2 - 5 clusters for my 7 labels. As well, it would perceive clusters as one or two outliers which is not representative of the data. For $\alpha > 300$, there seemed to be no change in number of clusters formed. It would stay consistent at 5 clusters.

4.2 Datasets

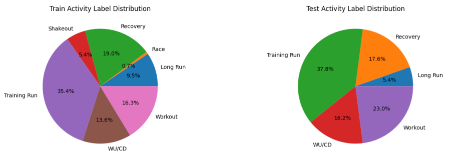


Figure 1: Random Shuffle Distribution

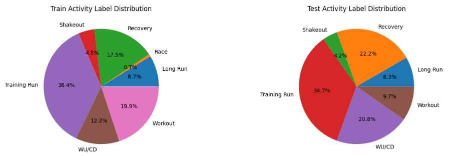


Figure 2: Recency Bias Distribution

I purpose two dataset partitioning methods to confirm the usage of DPMMs for general and continued use, random and recency biased. Both use a train/test split of 80%/20%. The dataset run type distribution of the partitions can be observed in Figure 1 and Figure 2. The random partition takes a typical approach of randomly splitting the dataset into the train and test partitions. This focus is to emphasize the raw prediction capability that the DPMM can take on after being trained. This prioritizes short-term performance gains while accurately fitting the model.

In contrast, the recency biased approach does not shuffle the dataset. It instead assigns the 20% most recent data points as the test data with the older data being the train data. This was chosen to analyze the effectiveness of classifying a stream of new data points given the trained DPMM. A recent stream of activities aligns with long term trends of the model performance. In turn, this process mimics how this model would be deployed in a larger system. It is generally not affordable to consistently retrain models with each new data point received so having a robust consistent model is essential.

4.3 Evaluation Metrics

I employ several standard metrics to evaluate my DPMM along with the baseline implementation. The first one of which is Adjusted Random Index (ARI) Santos and Embrechts [2009] which measures the clustering quality. ARI quantifies the similarity between the predicted and user labels while accounting for chance. This is beneficial for not needing a direct mapping as the number of clusters is not preset. ARI forms a good baseline for the dynamic nature of DPMMs.

Accuracy and F1-score are used to analyze the classification ability. These are used to assess that data points are being assigned to the correct category instead of similar clusters. However, clustering usually does not correctly map to the true label; thus, I utilized Hungarian Matching to align the predicted and true labels. Hungarian Matching is used to map the generated cluster labels, of which may not equal the number of true labels. These metrics are more generally used for evaluating the test data which is classified and not added into the DPMM.

Beyond these standard metrics, I introduce a domain-specific metric: Hierarchical Accuracy. This follows the hierarchical structure outlined in Figure 3. Effort increases as we move up in tiers which reflects a necessity to decrease frequency of those types of activities. Therefore, this metric only penalizes upward classification that overestimate the effort and intensity of the run (i.e., predicting a harder run than the user intended label). The model attempts to detect overtraining through a recent increase of intensity. If the run was harder than intended, insight is given to dial back some key run attributes.

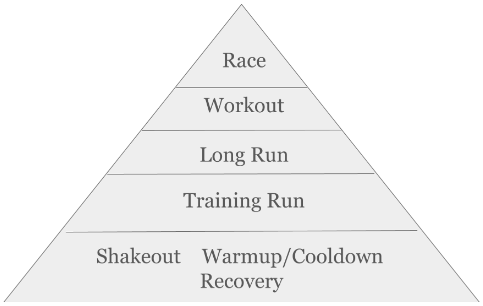


Figure 3: Run Type Hierarchy

In conjunction, this metric avoids penalizing downward classifications as to prevent negative feedback. This downward trend could indicate tapering for races, intentional rest periods, or natural fitness growth. The same run is easier as they have decreased overall intensity in preparation, or one can do more at the same effort level. This prioritizes finding overtraining trends and producing more guidance for training.

5 Results

5.1 On the Random Partitioning

On the random dataset, my DPMM implementation routinely outperforms the baseline version in both the train and test sets. This showcases DPMM's enhanced ability for fitting the run categorization clusters then predicting which cluster a given run should belong to. The overall results comparing DPMM with its baseline are highlighted in Figure 4. Specifically when training the base DPMM, the ARI, accuracy, and hierarchical accuracy are better by 6%, 0.5%, and 15.5%, respective to the baseline. The improved ARI score illustrates that DPM can more accurately group runs according to their true labels. DPMM showcases its main strength in identifying possible overtraining runs, seen through a greatly outperforming hierarchical accuracy.

When the trained clusters were mapped to the predefined tiers, SkLearn did outperform DPMM in ARI and accuracy by 4%. However, the DPMM continues to exhibit its strength in hierarchical accuracy achieving a score of 96% beating SkLearn by 7%. This is essential to establishing a strong hierarchical clustering model that can be used on a variety of data. Even though, it was outperformed by its tiered counterpart, tiered results became noticeably better from the full results. These increased values continue to exhibit the strong model that is useful for predicting trends in running activity.

DPMM continues to beat the full result baseline on the testing set exhibiting a 13% to 23% increase in performance. This trend continues with the tiered testing of 8% to 17%. These trends demonstrate the effectiveness for both establishing a correct model and reliably classifying unseen data in both the full and tiered settings.

In contrast, the baseline model exhibited significant performance drops from training to testing, with decreases ranging from 7% to as much as 22%. This suggests that the baseline model struggles to generalize, likely due to overfitting or poor representation of true underlying cluster structures. However, DPMM showcased consistency with discrepancies between training and testing results rarely exceeding 2%. This stability highlights DPMM's robustness and ability to handle new, unseen data effectively. Moreover, the model's capacity to predict more runs as belonging to easier tiers further reflects its nuanced understanding of the dataset's hierarchical structure.

All Types

	Train	Test	Sk Train	Sk Test
ARI	.412	.391	.354	.249
Acc.	.645	.634	.641	.500
F1	.592	.592	.628	.415
HA	.930	.931	.774	.708

Tiered

	Train	Test	Sk Train	Sk Test
ARI	.474	.423	.513	.282
Acc.	.728	.736	.767	.653
F1	.679	.692	.764	.642
HA	.962	.944	.892	.778

Figure 4: Random Dataset Shuffle Results

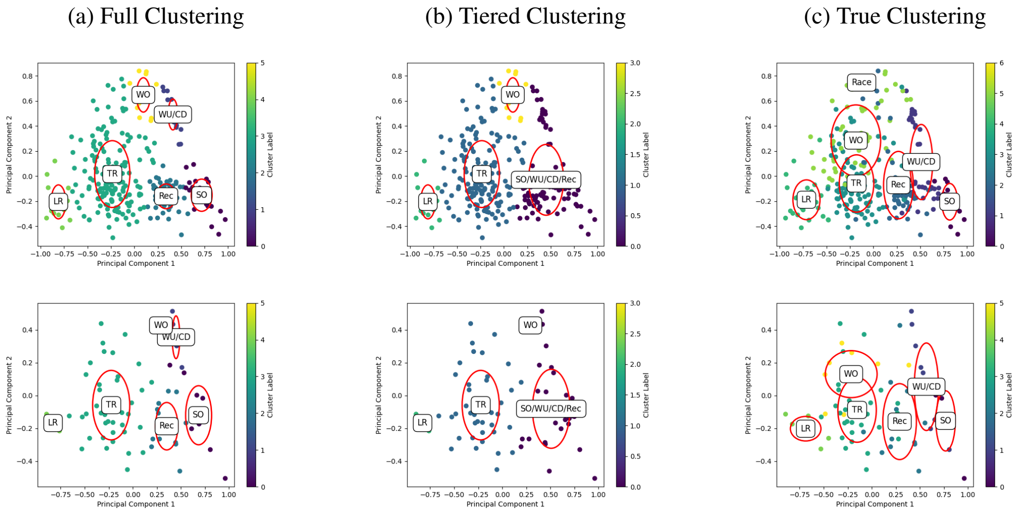
These results are strengthened by the shown clustering in Figure 5. The components are correlated to the gradient of length (x) and intensity (y). Each label was effectively separated similarly to the true clustering. However, in the full clustering, the training run was predicted more than its data distribution. This can be seen as a result of the CRP prior calculation favoring more populated distributions when likelihoods could be close or even favor a different cluster. This would lead the posteriors to be close but by choosing the maximum probability, it eliminates the random chance. The increased hierarchical accuracy can be from the training run taking the majority of overlapping workouts. These are on the same effort level as a regular run which favors the hierarchy.

Overall, these results demonstrate that DPMM is a reliable and robust clustering approach for predicting any given run's category and tier. It is particularly well-suited for datasets with hierarchical underpinnings. In this random dataset approach, the consistent improved performance emphasizes the raw prediction capabilities that a strongly trained DPMM has.

5.2 On the Recency Biased

Unlike the random dataset, my implementation on the recency biased generally slightly lacked behind the baseline when training the model. This was only a drop of 2% and 5% for ARI and accuracy, respectively, found in Figure 6. However, the superior hierarchy score continues to reflect DPMM's effectiveness towards a more wholistic hierarchical tier prediction. Proportionally similar increases occurred after mapping the clusters to their tiers, a range of 1% to 10% for training and 6% to 12% for testing. The same can be said for the baseline that increased with tiers.

However, the baseline continued to show regression on testing after training. It was not met with the proportionally same decreases, showcasing that the sequential data stream possibly did not suffer from overfitting as much as previously. As a result, DPMM made up the training difference and more to perform better in all metrics, increasing its dominance in the hierarchical structure. DPMM bettered the baseline by 13% in ARI, 6% in accuracy, and 17% in HA.


(a) Full Clustering

(b) Tiered Clustering

(c) True Clustering

Figure 5: Random Dataset Clustering for Train (top) and Test (bottom) Sets

Although, the real advantages of DPMM are displayed in the consistent 10% performance increase from the training to the test set. This enhancement ratifies the overwhelming benefit of DPMM when applied to identifying overtraining. DPMM consistently found that the previous training was reflective of healthy increases in volume, reducing the chance of acute or stress injuries. While the data has an increase in efforts and intensities, previously experienced efforts became easier and the clusters can grow and adapt without a drop in predictive ability.

All Types

	Train	Test	Sk Train	Sk Test
ARI	.309	.429	.329	.294
Acc.	.565	.634	.615	.577
F1	.551	.588	.602	.485
HA	.837	.887	.778	.718

Tiered

	Train	Test	Sk Train	Sk Test
ARI	.348	.538	.468	.413
Acc.	.649	.746	.747	.704
F1	.638	.701	.741	.667
HA	.847	.944	.903	.845

Figure 6: Recency Biased Results

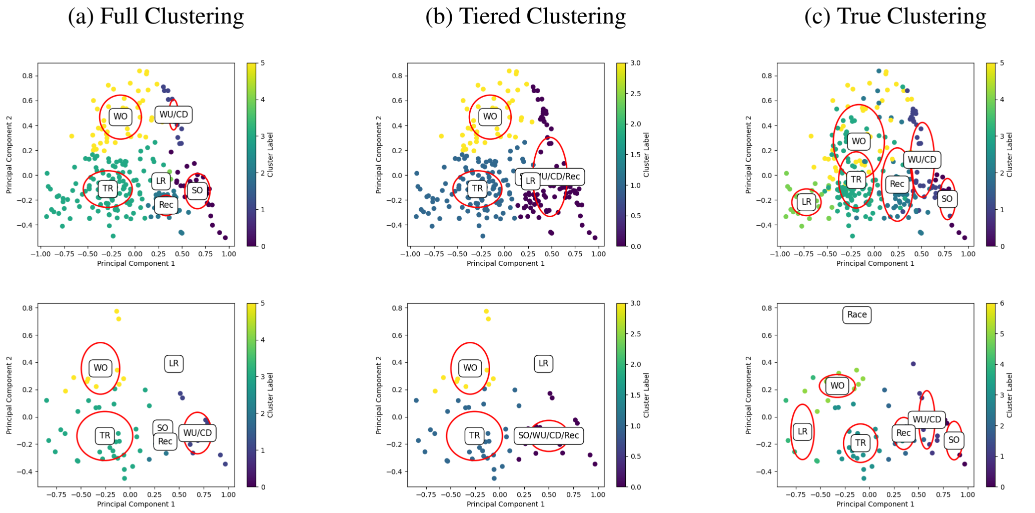
The true clustering closely followed the random dataset clustering but with a lower clustered intensity. This is explained by a growth of fitness for my running. The train data encapsulated the past 2 months after an intentional break of 2 weeks and starting to run again. Previous workouts started to become efforts closer to training runs and were effectively grouped in that cluster and tier.

These results continue to strongly illustrate DPMM's heightened ability to understand the run tier hierarchy and how growth affects the clustering. This function is essentially to building a model and system that should grow and adapt with the data. Dirichlet Process Mixture Models have demonstrated a greatly enhanced potential to become useful in creating a robust predictive and data stream model.

Lastly, the two implementations differed in their abilities to correctly identify all 7 true labels. SkLearn routinely only found 5 clusters during training for both datasets. These 5 clusters generally embodied traditional clustering shapes and was unable to detect possible nuance, even with varying α values. In contrast, DPMM detected 6 distinct clusters, having the two race data points of 368 being close to workouts as the race length and intensity matched closely to the majority workouts.

6 Limitations and Future Work

The major limitation of DPMMs lies in the dataset distribution. The data that I used is my own runs, which contain a lot of overlap and is representative of one style of long distance running training. I'm also a very experienced runner that knows how my body reacts to training and can more easily



(a) Full Clustering

(b) Tiered Clustering

(c) True Clustering

Figure 7: Recency Bias Dataset Clustering for Train (top) and Test (bottom) Sets

detect overtraining fatigue or just soreness. I concur with a coach and build dedicated plans from his experience. This causes a lot of borderline efforts where there are either mixed and nuanced intentions going into each run or sessions are intended to be easier to prepare for a race or supplement training. Perhaps a continuous model could give more insights into overtraining efforts. Companies such as Garmin and Strava provide continuous "training efforts" that build up over week. I found that these calculated efforts can also be easily confused but offer a more nuanced option.

The main analysis is only on a singular run basis as of now. I attempt to try to view the long-term success that this model may have in a slightly upwards trajectory. However, I find that it would be significant to propose more experimentation in dissecting these trends on a more granule level. The focus can help acute injuries but more so adapted to offer guidance on more chronically built injuries, such as stress fractures. Finding those weekly trends compared to just the daily ones provides more wholistic insight into training. As well, someone most likely won't get an overuse related injury from one harder session, in fact those are the sessions that make you better. However, we also cannot trust that one will make an automatic overuse trend by seemingly random overuse comments.

In practice, the α concentration hyperparameter can be and should be tuned over timed when retraining and making predictions. Dynamically adjusting this automatically or manually can take place in observing other running trends, such as breaks, injury, or continued trends in running daily. In addition, experienced runners would benefit from a higher α as there exists a lot more nuance and less drastic diversion over the same period of time compared to their beginner counterparts. Even with normalized data, the 6-month improvement of a beginner will proportionally be larger than the experienced runner. This is crucial to providing an individualistic system to guide everyone.

7 Conclusion

This work implemented a Dirichlet Process Mixture Model to predict running activity categories and tiers, with a focus on identifying overtraining trend. I also propose a new domain specific metric: Hierarchical Accuracy. This metric does not penalize downward classification, instead focusing on overtraining from more intense and stressful activities. This aligns with monitoring athletic performance and condition to fuel future sessions.

The DPMM was fitted using two different dataset partitions to analyze the immediate and long-term prediction ability. For the random dataset, the built DPMM outperformed the baseline in all metrics when testing and most training metrics. These results highlight the ability to effectively handle a complex, overlapping dataset with a strong classification performance. The long-term predictive strength is explored through the recency biased dataset which uses the 20% most recent runs as the test set. This approach better simulates real-world deployment where the main aim is to identify recent trends. DPMM maintained its prediction strength, improving in all metrics from the train to test achieving a 94% hierarchical accuracy in the tiered test set.

I present findings that illustrate the effectiveness of combining non-parametric Bayesian modeling with tailored evaluation metrics to measure overtraining. The results affirm the potential of DPMM as a powerful tool for analyzing running performance. It provides beginners, experienced, and runners of all levels with advice to succeed in their training preparing for their next big race or their daily peaceful runs.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Henrique Aguiar, Mauro Santos, Peter Watkinson, and Tingting Zhu. Learning of cluster-based feature importance for electronic health record time-series. In *International conference on machine learning*, pages 161–179. PMLR, 2022.

Ahmed Alsayat and Hoda El-Sayed. Efficient genetic k-means clustering for health care knowledge discovery. In *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 45–52. IEEE, 2016.

Ramzi A Haraty, Mohamad Dimishkieh, and Mehedi Masud. An enhanced k-means clustering algorithm for pattern discovery in healthcare data. *International Journal of distributed sensor networks*, 11(6):615740, 2015.

Benjamin M Marlin, David C Kale, Robinder G Khemani, and Randall C Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT international health informatics symposium*, pages 389–398, 2012.

Peter Orbanz. Lecture notes on bayesian nonparametrics. *Journal of Mathematical Psychology*, 56: 1–12, 2012.

Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. *Encyclopedia of machine learning*, 1:81–89, 2010.

Robert Richardson and Brian Hartman. Bayesian nonparametric regression models for modeling and predicting healthcare claims. *Insurance: Mathematics and Economics*, 83:1–8, 2018. ISSN 0167-6687. doi: https://doi.org/10.1016/j.insmatheco.2018.06.002. URL `https://www.sciencedirect.com/science/article/pii/S0167668717305437`.

T.J. Rogers, K. Worden, R. Fuentes, N. Dervilis, U.T. Tygesen, and E.J. Cross. A bayesian non-parametric clustering approach for semi-supervised structural health monitoring. *Mechanical Systems and Signal Processing*, 119:100–119, 2019. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2018.09.013. URL `https://www.sciencedirect.com/science/article/pii/S088832701830623X`.

Damian Ruta, Bogumił Bocianiak, Anna Kajka, Julia Hamerska, Joanna Antczak, Laura Hamerska, Urszula Fenrych, Karolina Wojtczak, Olga Skupińska, and Julia Lipska. Health aspects of amateur long-distance running. *Quality in Sport*, 20:53342–53342, 2024.

Jorge M Santos and Mark Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks*, pages 175–184. Springer, 2009.

Jack E Taunton, Michael B Ryan, DB Clement, Donald C McKenzie, D Robert Lloyd-Smith, and Bruno D Zumbo. A retrospective case-control analysis of 2002 running injuries. *British journal of sports medicine*, 36(2):95–101, 2002.

Yee Whye Teh. An introduction to bayesian nonparametric modelling. *Machine Learning Summer School*, 2009.

Yee Whye Teh et al. Dirichlet process. *Encyclopedia of machine learning*, 1063:280–287, 2010.

RN Van Gent, Danny Siem, Marienke van Middelkoop, AG Van Os, SMA Bierma-Zeinstra, and BW Koes. Incidence and determinants of lower extremity running injuries in long distance runners: a systematic review. *British journal of sports medicine*, 41(8):469–480, 2007.

W. Xue and T. Guillemont. Scikit-learn: Bayesian gaussian mixture, 2024. URL `https://scikit-learn.org/1.5/modules/generated/sklearn.mixture.BayesianGaussianMixture.html`.

8 Supplementary Material

8.1 Source Code

Source code can be found at `https://github.com/sgsikorski/STAT6020/tree/main`. The src folder contains the effective source code with my written DPMM in src/DPMM.py. Any output lies in res. The cluster plots in 2d for my implementation, SkLearn, and true clustering are found. This includes the results for both train and test dataset partitions.

8.2 Run Type Distinctions

These are the main descriptions for types of runs that I group all of my runs on which I base this work off of. This does not represent how others categorize their training or use a specific label for each group. And in fact, within some categories like workouts, there exists a lot of nuances between those subtypes. However, that may require more assumptions about the data and how people record them.

- Race: The most extreme case where average heart pace, pace, and power are close to their max counterparts. These averages and maxes will also be higher than any other run that is done.
- Workout (WU): Dedicated sessions where average and max pace, heart rate, and power are elevated but not spending as much time in those stats as a race, so averages are generally lower.
- Long Run (LR): Generally, but not always, about 1.5-2x longer than the TR with a slightly elevated pace.
- Training Run (TR): The most common run and general running. Variable feature values but are not high or low as the other runs with emphasis on being the middle of all features.
- Warmup/Cooldown (WU/CD): These are short 15-20 minutes jogs done at TR pace prior to workouts or races with low or high heart rates depending on intensity intention.
- Shakeout (SO): Less than 2 miles at a slow pace just to wake up the body.
- Recovery: Usually shorter time/distance but does not depend on it. Intention on low heart rate and low power output