Estimation of von Mises-Fisher Distribution Algorithm, with application to support vector classification

Adetunji David Ajimakin
Indian Institute of Science
Bangalore, India
ajimakind@iisc.ac.in

V. Susheela Devi
Indian Institute of Science
Bangalore, India
susheela@iisc.ac.in

ABSTRACT

The most common method in the Evolutionary Algorithm community to handle constraints is to use penalties. The simplest being the death penalty, which rejects solutions that violate constraints. However, its inefficiency in search spaces possessing small feasible regions spurred research into adaptive penalties and other competitive methods. A major criticism of these approaches is that they require the user to fine-tune parameters or design problem-dependent operators. We propose to do away with penalty functions for problems over the Euclidean space when the constraint is an equality concerning the Euclidean distance. This paper describes an evolutionary algorithm on the unit hypersphere based on representing the population with the von Mises-Fisher probability distribution from the field of Directional statistics. We demonstrate its utility by solving the support vector classification problem for a few datasets.

CCS CONCEPTS

• **Theory of computation** → **Bio-inspired optimization**; *Support vector machines*;

KEYWORDS

Estimation of Distribution Algorithms, Representation, Constraint handling, Pattern recognition and classification

ACM Reference Format:
Adetunji David Ajimakin and V. Susheela Devi. 2021. Estimation of von Mises-Fisher Distribution Algorithm, with application to support vector classification. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459470

1 INTRODUCTION

Evolutionary algorithms typically handle problems constrained in the set of acceptable solutions by imposing penalties on constraint violations[3]. However, there are significant challenges with using penalty functions, such as difficulty assigning priorities to the various constraints and the objective function and stagnation when the fitness function is undefined at infeasible points.

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459470

An alternative constraint handling method is to find transformations that make it easier to explore the feasible region. This paper presents an evolutionary algorithm used to search over the d-dimensional Euclidean space with an equality constraint on the Euclidean distance. We achieved this by noting that such constrained search in the Euclidean space is equivalent to an unconstrained search over the $(d-1)$-dimensional hypersphere. Hence, we use the von Mises-Fisher probability distribution as a mechanism to generate feasible solutions.

2 VON MISES-FISHER DISTRIBUTION

A von Mises-Fisher (vMF) distribution is a probability distribution over the points in a unit hypersphere. Its density, parameterized by a mean direction $\boldsymbol{\mu}$ and a concentration parameter κ, is given by

$$f(\boldsymbol{x} \mid \boldsymbol{\mu}, \kappa) = c_d(\kappa) \exp(\kappa \boldsymbol{\mu}^\top \boldsymbol{x})$$

where c_d is the normalization constant (see [4] for details). The vectors $\boldsymbol{x}$, $\boldsymbol{\mu} \in \mathbb{R}^d$ are both d-dimensional unit vectors, or equivalently, are elements of $\mathbb{S}^{d-1}$, the $(d-1)$-dimensional unit hypersphere. The concentration parameter determines how strongly the unit vectors generated according to f cluster around the mean direction.

Algorithm 1 provides the pseudocode to sample from a von Mises-Fisher distribution, and it incorporates Wood's algorithm [7] to efficiently sample from the marginal density of $w = \boldsymbol{\mu}^\top \boldsymbol{x}$. To estimate a vMF distribution, let $\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n)$ be a sequence of points in descending order of fitness, p_i be the selection probability of the point $\boldsymbol{x}_i$, $\boldsymbol{R} = \sum_{i=1}^n p_i \boldsymbol{x}_i$, $r = \|\boldsymbol{R}\|$, and $\sum_{i=1}^n p_i = 1$. From $\mathcal{X}$, we want to find the maximum likelihood estimates (MLE) of $\boldsymbol{\mu}$ and κ by maximizing $n \ln c_d(\kappa) + n\kappa \boldsymbol{\mu}^\top (\sum p_i \boldsymbol{x}_i)$ subject to the constraints $\boldsymbol{\mu}^\top \boldsymbol{\mu} = 1$ and $\kappa \geq 0$. Following the derivation in [2], we arrive at $\hat{\boldsymbol{\mu}} = \frac{\boldsymbol{R}}{r}$ as the MLE solution for $\boldsymbol{\mu}$, and $\hat{\kappa} \approx \frac{r(d-r^2)}{1-r^2}$ as an approximate solution for κ.

The proposed optimization algorithm samples and updates the vMF probability model in each iteration. We introduce a learning rate η and use an exponential learning schedule to control the pace at which we update κ. The goal is to encourage exploration at the early stages by diminishing the effects of the new κ estimates. We set the selection probabilities $p_i = \frac{max\left(0,\, \ln\left(1+\frac{\lambda}{2}\right) - \ln i\right)}{\sum_{i=1}^{\lambda} max\left(0,\, \ln\left(1+\frac{\lambda}{2}\right) - \ln i\right)}$, an adaptation of the weights used in the NES [6] algorithm. The population size λ is set to $4+\lfloor 3 \ln d \rfloor$, same as that of the CMA-ES [1] algorithm.

Algorithm 1: Sampling a von Mises-Fisher distribution

Input: $\boldsymbol{\mu}, \kappa, d$
Output: a realization of the vMF distribution

$b \leftarrow \dfrac{d-1}{2\kappa + \sqrt{4\kappa^2 + (d-1)^2}}$
$x \leftarrow \dfrac{1-b}{1+b}$
$c \leftarrow \kappa x + (d-1)\ln(1-x^2)$
repeat
  draw $z \sim Beta(\frac{d-1}{2}, \frac{d-1}{2})$
  draw $u \sim Uniform(0,1)$
  $w \leftarrow \dfrac{1-z(1+b)}{1-z(1-b)}$
until $\kappa w + (d-1)\ln(1-xw) - c \geq \ln u$
$\boldsymbol{y} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$
$\boldsymbol{t} \leftarrow \boldsymbol{y} - (\boldsymbol{\mu}^\top \boldsymbol{y})\boldsymbol{\mu}$
$\boldsymbol{v} \leftarrow \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|}$
return $w\boldsymbol{\mu} + \left(\sqrt{1-w^2}\right)\boldsymbol{v}$

3 APPLICATION TO LINEAR SUPPORT VECTOR CLASSIFICATION

A linear SVM finds a maximum-margin hyperplane, with normal vector $\boldsymbol{w}$ and intercept b, that separates two classes of training examples. We restrict the search for $\boldsymbol{w}$ to the set of unit vectors, i.e., $\|\boldsymbol{w}\|_2 = 1$, as only the direction matters.

The population in each iteration is a set of candidate solutions for $\boldsymbol{w}$. To find the intercept b_i for each individual $\boldsymbol{w}_i$, let $X^+ = \{x_1^+, x_2^+, \ldots, x_m^+\}$ and $X^- = \{x_1^-, x_2^-, \ldots, x_n^-\}$ be the set of projections on $\boldsymbol{w}_i$ for the positive and negative classes, respectively. We set b_i to $-t$ where t is the decision point that maximizes the number of correct classifications per class normalized by the size of the class, i.e.,

$$\frac{|\{x \mid x \leq t - \Delta^- \wedge x \in X^-\}|}{n} + \frac{\left|\left\{x \mid x > t + \Delta^+ \wedge x \in X^+\right\}\right|}{m}$$

Δ^- and Δ^+ are class margins introduced to improve generalization by considering only the points that are on the correct side of their class margins.

4 EXPERIMENT AND RESULTS

We experimented on five datasets from the subfield of multilabel classification, where each example is associated with a set of labels. The most widely used multilabel classification method, Binary Relevance, transforms the problem into learning an independent binary classifier for each label [5]. We reduced each dataset to include only the top five ranking labels. The labels were ranked by how close the number of positive examples is to the number of negative examples.

For each dataset, we compared the macro-averaged F_1-score (F_1-scores averaged over the labels) on a test set for two Binary Relevance schemes with different underlying classifiers. The first, termed the Standard SVM, used a linear SVM. For each label, the hyperparameter C is tuned through a 5-fold cross-validation grid search to maximize the F_1-score. The other classifier, termed EvMFA SVM, used a linear classifier trained by the proposed algorithm also to maximize the F_1-score in at most a thousand iterations. To prevent the model from overfitting to the training data, we evaluated the model on a validation dataset after every ten generations and saved the top five models. The prediction at test time was a simple majority vote of these five models.

The concentration parameter κ was initially set to 0, and class margins were set to 5% of the interquartile range of the projections. We used an initial learning rate of 0.01 and a strengthening factor of 1.004 so that the learning rate is at 0.5 after a thousand iterations. Table 1 details the outcome of the experiment. The two methods achieved similar results on the Yeast dataset and appeared to be on equal footing in the other four datasets.

Table 1: Comparison of base classifiers in a Binary Relevance multilabel classification strategy. The reported values are the mean and standard deviation (SD) of the macro-averaged F_1-scores averaged over ten repetitions.

Dataset	Standard SVM		EvMFA SVM	
	Mean	SD	Mean	SD
Yeast	0.6652	0	0.6628	0.007827
Delicious	0.6278	0	**0.6481**	0.001625
Enron	0.6724	0	**0.6854**	0.007047
Scene	**0.7184**	0	0.7065	0.007878
TMC2007	**0.7751**	0	0.7565	0.003562

5 CONCLUSIONS

This paper presented and empirically demonstrated the usefulness of an evolutionary algorithm for an unconstrained search over the unit hypersphere. The vMF distribution used is one of the simplest distributions in Directional statistics because it does not capture interactions between variables. This limitation suggests a promising path for future contributions.

REFERENCES

[1] Anne Auger and Nikolaus Hansen. 2012. Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. 827–848.
[2] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. 2005. Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. *Journal of Machine Learning Research* 6, 9 (2005).
[3] Carlos A Coello Coello. 2017. Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 675–701.
[4] Kanti V Mardia and Peter E Jupp. 2009. *Directional statistics*. Vol. 494. John Wiley & Sons.
[5] Grigorios Tsoumakas and Ioannis Vlahavas. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *European conference on machine learning*. Springer, 406–417.
[6] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014. Natural evolution strategies. *The Journal of Machine Learning Research* 15, 1 (2014), 949–980.
[7] Andrew TA Wood. 1994. Simulation of the von Mises Fisher distribution. *Communications in statistics-simulation and computation* 23, 1 (1994), 157–164.