

# Mirror mirror on the wall, what is the fairest linear parameter space representation of all?

## On Representations of Linear Parameter Space in Context of Clustering

Daniyal Kazempour<sup>1</sup>, Andrian Mörtlbauer, Peer Kröger<sup>1</sup>, and Thomas Seidl<sup>1</sup>

Ludwig-Maximilians-Universität München, Munich, Germany  
{kazempour,kroeger,seidl}@dbs.ifi.lmu.de

**Abstract.** The parameter space transform has been utilized over decades in context of edge detection in the computer vision domain. However the usage of the parameter space transform in context of clustering is a more recent application with the purpose of detecting (hyper)linear correlated clusters. There are various representations of the linear parameter space. Yet there is one representation which has not been used yet - the Hessian Normal Form (HNF) representation. This work reveals the properties of the HNF representation in comparison to its competitors.

**Keywords:** Parameter Space Representation · Hessian Normal Form · Clustering.

## 1 Introduction

With Paul V.C. Hough introducing the first application of parameter space transformation for detecting lines on images, he used the slope-intercept representation of the linear parameter space. This representation has its drawbacks. Within the following decades more representations were developed. However all of the representations have been designed with the domain of edge detection on images in mind. The purpose of this work is to introduce and elaborate on the advantages and disadvantages of each linear parameter space representation in context of clustering. In the clustering context (hyper)linear correlated clusters are detected by using parameter space transformation. Using this type of transformation is especially for higher dimensional data sets useful. By scanning the parameter space for intersections of lines or (hyper)planes we avoid the computation of distances which become useless in the high-dimensional setting. Regarding related work, the first parameter space transform used a slope-intercept representation of the parameter space. One of the inherent drawbacks of this representation is its unboundedness regarding its dimensions in parameter space and with vertical lines or in higher dimensions, hyperplanes. For overcoming the issues of the slope-intercept form, Duda and Hart provided in [2] a polar normal-form. On the slope-intercept, polar and the other representation, which are not suitable in the context of clustering, the authors in [4] have elaborated on their advantages and disadvantages in context of edge detection on images.

## 2 Linear Parameter Space Representations

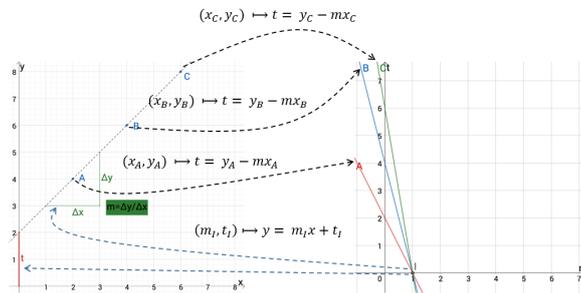
In this section we describe four representations of the linear parameter space and elaborate on their properties and advantages as well as disadvantages in context of data mining.

### 2.1 Slope intercept Representation

The very first representation of the linear parameter space is the slope intercept form which has been introduced first by the work of Paul V.C. Hough. Given a data point  $p := (x, y)$  The core intuition of this representation is that a line can be expressed through its slope  $m$  and its intercept  $t$ . The slope intercept equation is therefore:

$$y = mx + b \tag{1}$$

The slope intercept variant provides so called point-to-line mappings (PTLM) which means that a point in data space corresponds to a line in parameter space and a point in parameter space corresponds to a line in data space. This duality can be seen in figure1. This property is favorable as it facilitates to map the actions made in data space to parameter space and vice versa.



**Fig. 1.** data-parameter-space duality. left: data space, right: parameter space

This representation of a linear parameter space comes with the advantage of being easy to understand and providing the afore mentioned PTLM property. It is quite comprehensive to see the duality between data and parameter space. However the slope intercept form comes with several limitations. First, given the case that we have an infinite slope, we can not represent vertical lines. Second, if we extend this form to higher dimensions, let's say to three dimensional dataset, the equation extends to  $z = m_1x + m_2y + t$ . Here the question emerges, which of the axis to select as intercept, and as more slopes are introduced, there are more axis to which a plane can be vertical to, limiting the method to detect several axis parallel linear correlations. And finally the method fails by its unboundedness

in parameter space. We do not have any limitation of the slope and intercept axis. As a consequence one can set an arbitrarily high initial boundary, which may result in many additional computations. On the other side of the coin, one may set the initial boundaries quite small, which can lead to a loss of regions of interest and thus to losing potential (hyper)linear correlated clusters.

### 2.2 Polar Representation

As a solution to the issues of unboundedness and the case of axis parallel lines Duda and Hart introduced in [2] a polar representation of the parameter space. The parameter space is defined by an inclination angle  $\alpha$  which describes the angle between a coordinate axis and the line  $\mathcal{L}$  which is orthogonal to the line we aim to detect. The length  $d$  of  $\mathcal{L}$  is the second parameter of our parameter space. We thus obtain the following equation:

$$d = y \cdot \sin(\alpha) + x \cdot \cos(\alpha) \tag{2}$$

The consequence of using polar coordinates regarding the unboundedness can be seen in equation 2 as well as in figure 2. The angle is limited between  $[0, 2\pi]$ . However the distance axis  $d$  is unbounded. The gain of the boundedness comes however at a cost. First, the functions in parameter space are no longer lines but sinusoidal curves, this implies that the PTLM property is no longer maintained, as a point in data space corresponds to a sinusoidal curve in parameter space, but a point in parameter space does not correspond to a curve in data space. Second, it is difficult to get the semantics of the curves in parameter space. The meaning of the curves extrema is not convertible to a semantics which leads to the third issue. It is difficult to compute the intersection of a sinusoidal curve with an bounding box. On the challenges of detecting intersections with a bounding box, the authors in [3] have elaborated on.

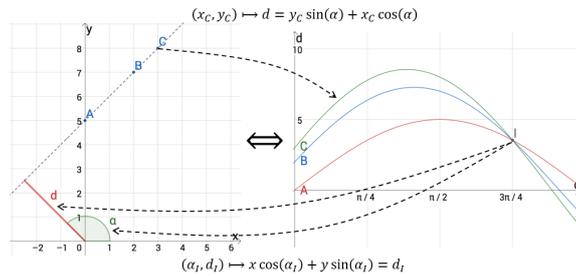


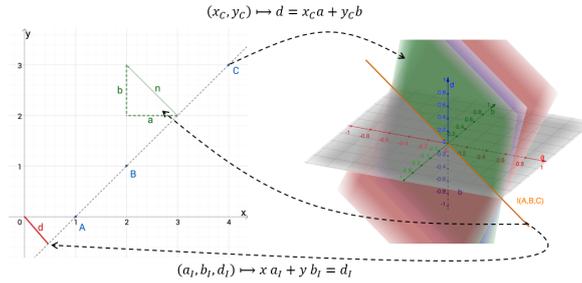
Fig. 2. left: data space, right: polar parameter space

### 2.3 Hessian Normal Representation

The disadvantages of the polar representation can be overcome by our third representation which, to our surprise, hasn't been used yet in the context of parameter space transformation. We take again a closer look at equation 2 and represent it as follows:

$$\begin{aligned}
 x \cdot \cos(\alpha) + y \cdot \sin(\alpha) &= d \\
 x \cdot a + y \cdot b &= d \\
 \begin{pmatrix} x \\ y \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} &= d \\
 \vec{p} \cdot \vec{n} &= d
 \end{aligned} \tag{3}$$

In equation 3 it can be seen that the polar representation is just the polar coordinate form of the so called hessian normal form (HNF). Just like in the polar form,  $d$  describes the distance between the origin to the line to which it is orthogonal. The vector  $\vec{n}$  is the normal vector which provides the direction to the orthogonal line as it can be seen in figure 3.



**Fig. 3.** left: data space, right: polar parameter space

This representation however seems to be unbounded. For this purpose the normal vector is normalized as  $\vec{n}_0 = \frac{\vec{n}}{|\vec{n}|}$ . As by now the length of the vector is normed to unit length. By this each of the dimensions of  $\vec{n}_0$  are bounded to the interval  $[0, 1]$ . Also the issue with axis parallel slope is not a problem in the HNF. While the vector  $\vec{n}_0$  would point to the direction of the  $x$ -axis, in case we have an  $y$ -axis parallel line, we would not have any undefined normal vector, as we have in the case of the slope intercept form, where the slope is undefined as we would divide by zero in the denominator. The only difference and probably disadvantage of the HNF is that our parameter space has one dimension more than our data space as seen in figure 3. Also the PTLM property is given, as a data point in data space is mapped to a linear data point function in parameter space and a normal point in parameter space represents a line in data

space. Further, the check if a plane intersects with a bounding box in parameter space is computationally less intensive. Regarding higher dimensional data and detecting hyperlinear correlated clusters, this representation is simple to extend, as the normal vector gets one feature more as the data points have an increased dimensionality.

## 2.4 Overview

Taking the information from the previous subsections we can distill the following table with the properties:

Representation	PTLM property	bounded	intersection cost	vertical lines
Slope-Intercept	✓	✗	low	✗
Polar Normal Form	✗	✓(except $d$ )	high	✓
Hessian Normal Form	✓	✓(except $d$ )	low	✓

## 3 Conclusion

In this work we have elaborated on two representations of linear parameter space, the slope-intercept and the polar normal form. Both of them come with their own strengths but also weaknesses. However there is a third representation which, as it can be seen in the table in subsection 3.4, has all of the strengths of the previously mentioned two forms but none of their weaknesses. It has the PTLM property, is bounded among all dimensions except the distance  $d$ , is low in its costs for checking if a bounding box is intersected by such a data point function and has the capability of detecting vertical lines. It is surprising why the HNF has not been utilized so far as the representation of choice for linear parameter spaces. We'd like to conclude this work with an answer to the question posed by our title: It is the HNF representation, bringing advantages to the parameter space transformation.

## References

1. Paul V.C. Hough: Method and means for recognizing complex patterns, US Patent US3069654A, 1960
2. Richard O. Duda, Peter E. Hart: Use of the Hough Transformation to Detect Lines and Curves in Pictures, Comm. ACM, Vol 15, No.1, pp. 11-15, 1972
3. Elke Aichert, Christian Böhm, Jörn David, Peer Kröger, Arthur Zimek: Global Correlation Clustering Based on the Hough Transform, Inc. Statistical Analysis and Data Mining Vol.1, pp.111-127, 2008
4. Adam Herout et. al.: Real-Time Detection of Lines and Grids, Springer Briefs in Computer Science, Chpt.2, 2013
5. Priyanka Mukhopadhyay, Bidyut B. Chaudhuri: A survey of Hough Transform, Pattern Recognition, 2014