

# HOW TO GROUND A LANGUAGE FOR LEGAL DISCOURSE IN A PROTOTYPICAL PERCEPTUAL SEMANTICS

*L. Thorne McCarty\**

2016 MICH. ST. L. REV. 511

*An Edited Transcript of a Presentation at the Legal Quanta Symposium at Michigan State University College of Law on October 29, 2015.*

## TABLE OF CONTENTS

INTRODUCTION.....	511
I. PROTOTYPE CODING.....	514
A. Manifold Learning.....	515
1. <i>The Probabilistic Model</i> .....	516
2. <i>The Geometric Model</i> .....	518
3. <i>Prototypical Clusters</i> .....	520
B. Deep Learning .....	521
II. A LOGICAL LANGUAGE .....	526
III. DEFINING THE ONTOLOGY OF LLD.....	533
IV. TOWARD A THEORY OF COHERENCE.....	535

## INTRODUCTION

Thank you. It's really delightful to be here talking to this group.

I am going to present a paper that I presented previously at the Fifteenth International Conference on Artificial Intelligence and Law (ICAAIL 2015), which was held in San Diego this past June. Ted Sichelman, in fact, was the general chair of that conference, and I think he is still watching remotely. So I will just say: That was an outstanding conference, Ted. Thank you for helping to organize it.

You should have received a copy of my ICAAIL paper prior to the symposium. Here is the citation:

---

\* Professor Emeritus of Computer Science and Law, Rutgers, The State University of New Jersey.

L. Thorne McCarty, *How to Ground a Language for Legal Discourse in a Prototypical Perceptual Semantics*, in PROCEEDINGS OF THE FIFTEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 89-98 (2015).<sup>1</sup>

If you have a chance to look at this paper, you will see that it is very technical, especially Section 3, which is entitled “A Logical Language.” One of the things that I will try to do in my talk today is to present these ideas, especially Section 3, in a more intuitive and informal way. I will refer you to the paper itself for the more technical details.

You also should have received two background papers. I hope you have a chance to read these papers at some point, too, because they will explain the legal background of my ICAIL 2015 paper. The two papers are also from the Artificial Intelligence and Law conferences, in 1995 and 1997. Here is the 1995 paper:

L. Thorne McCarty, *An Implementation of Eisner v. Macomber*, in PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 276-86 (1995).<sup>2</sup>

As the title suggests, this paper presents a computational reconstruction of the arguments of Justice Pitney and Justice Brandeis in the case of *Eisner v. Macomber*, 252 U.S. 189 (1920), which I am sure you all know and love from your first introductory tax course. I don’t have time to say much about it, but it is based on a formal representation that I have referred to as the “prototype plus deformation” model of legal concepts. And that idea is going to recur in this talk, too. The second paper, from 1997, is:

L. Thorne McCarty, *Some Arguments About Legal Arguments*, in PROCEEDINGS OF THE SIXTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 215-24 (1997).<sup>3</sup>

This paper was primarily a critical view of the literature, that is, the literature in the field of Artificial Intelligence and Law on *legal argument*. But it also contained a section, towards the end, Section 5, entitled “The Correct Theory,” which was an (uncritical) summary of my own theory of legal argument. Part of the thesis is quoted here:

Legal reasoning is a form of *theory construction* . . . A judge rendering a decision in a case is constructing a *theory* of that case . . . If we are

- 
1. Available online at <http://bit.ly/1qCnLJq>.
  2. Available online at <http://bit.ly/1pfmtdD>.
  3. Available online at <http://bit.ly/1QU5CUm>.

looking for a computational analogue of this phenomenon, the first field that comes to mind is machine learning.<sup>4</sup>

In some sense, of course, machine learning *is* theory construction. However, if you were to look at the state of machine learning, *circa* 1997, you would see clearly that the available techniques were not sufficient to handle the complexity of the theories that are needed for legal reasoning. Part of the problem is discussed in the following passage, which is again from Section 5 of my 1997 paper:

Most machine learning algorithms assume that concepts have “classical” definitions, with necessary and sufficient conditions, but legal concepts tend to be defined by *prototypes*. When you first look at prototype models . . . , they seem to make the learning problem harder, rather than easier, since the space of possible concepts seems to be exponentially larger in these models than it is in the classical model. But empirically, this is not the case. Somehow, the requirement that the exemplar of a concept must be “similar” to a prototype (a kind of “horizontal” constraint) seems to reinforce the requirement that the exemplar must be placed at some determinate level of the concept hierarchy (a kind of “vertical” constraint). How is this possible? This is one of the great mysteries of cognitive science.

It is also one of the great mysteries of legal theory.<sup>5</sup>

The paper then proceeds to discuss Dworkin’s thesis in *Hard Cases* (1975) and *Law’s Empire* (1986),<sup>6</sup> and concludes that the mystery can only be solved by developing a computational theory of “coherence” in legal argument. But what do we mean by “coherence”? This is the key question.

If we now fast-forward to the present, and my ICAIL 2015 paper, here is a summary of my talk: What has happened, I claim, is that contemporary trends in machine learning have now shed new light on the subject. Remember, as I said, in 1997, machine learning was not adequate to the task. But I will describe in this talk some recent work, particularly my own recent work, on what is called “manifold learning,”<sup>7</sup> as well as some work in progress on “deep

---

4. *Id.* at 221.

5. *Id.*

6. Ronald Dworkin, *Hard Cases*, 88 HARV. L. REV. 1057 (1975); RONALD DWORKIN, *LAW’S EMPIRE* (1986).

7. The three main historical approaches to manifold learning were introduced in the following papers: Joshua B. Tenenbaum, Vin de Silva & John C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, 290 SCI. 2319 (2000); Sam T. Roweis & Lawrence K. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, 290 SCI. 2323 (2000); Mikhail Belkin & Partha Niyogi, *Laplacian Eigenmaps for Dimensionality*

learning,”<sup>8</sup> and I will explain the implications of this work. Of course, I will also have to explain what I mean by these terms. If you want to look more deeply into the subject, I have several papers, either available now or available soon, where this material is presented in some detail. The mathematical foundations of the work are presented in:

L. Thorne McCarty, *Clustering, Coding, and the Concept of Similarity*, PREPRINT, arXiv:1401.2411 [cs.LG] (2014).<sup>9</sup>

(This paper has been submitted for journal publication, and it is currently under review.) I also have two follow-up technical papers in preparation: (i) *Differential Similarity in Higher Dimensional Spaces: Theory and Applications*; and (ii) *Deep Learning with a Riemannian Dissimilarity Metric*. (I originally cited these papers as “Forthcoming, 2015,” but that turned out to be slightly optimistic. Nevertheless, I do expect to have a draft of the first paper available sometime during the first half of 2016.) Finally, Section 3 of my ICAIL 2015 paper, which I mentioned earlier, extends all of this work to construct “A Logical Language.” In fact, the main claim of my talk today, which was also the main claim of my ICAIL 2015 paper, is the following:

Taken together, this work leads to a logical language grounded in a prototypical perceptual semantics, with implications for legal theory.<sup>10</sup>

Obviously, I have to explain what I mean by a “prototypical perceptual semantics.”

## I. PROTOTYPE CODING

But first, let’s look at the concept of *prototype coding*. The basic idea is to represent a point in an  $n$ -dimensional space by

---

*Reduction and Data Representation*, 15 NEURAL COMPUTATION 1373 (2003). My own work on manifold learning is most closely related to the *Laplacian Eigenmaps* of Belkin and Niyogi.

8. The literature on deep learning is now very extensive, but the field was initiated by three important papers in 2006: Yoshua Bengio et al., *Greedy Layer-Wise Training of Deep Networks*, 19 ADVANCES IN NEURAL INFO. PROCESSING SYSTEMS 153 (2006); Geoffrey E. Hinton, Simon Osindero & Yee-Whye Teh, *A Fast Learning Algorithm for Deep Belief Nets*, 18 NEURAL COMPUTATION 1527 (2006); Marc’Aurelio Ranzato et al., *Efficient Learning of Sparse Representations with an Energy-Based Model*, 19 ADVANCES IN NEURAL INFO. PROCESSING SYSTEMS 1137 (2006).

9. Available online at <http://bit.ly/1phP8q4>.

10. McCarty, *supra* note 1, at 89.

measuring its *distance* from a *prototype* in several specified *directions*. Furthermore, assuming that our initial space is Euclidean, we want to select a prototype that lies at the *origin* of an *embedded, low-dimensional, nonlinear subspace*, which is in some sense “optimal”. This second point leads us to the field of *manifold learning*.

### A. Manifold Learning

What is manifold learning? Here is a quotation from a relatively recent paper by a prominent research group at the University of Montreal, which was published in the proceedings of the Conference on Neural Information Processing Systems in 2011 (NIPS 2011). The authors identify three hypotheses motivating their work, two of which are as follows:

1. . . .

2. The **(unsupervised) manifold hypothesis**, according to which real world data presented in high dimensional spaces is likely to concentrate in the vicinity of non-linear sub-manifolds of much lower dimensionality. . . .

3. The **manifold hypothesis for classification**, according to which points of different classes are likely to concentrate along different sub-manifolds, separated by low density regions of the input space.<sup>11</sup>

Notice that these hypotheses combine geometric concepts (e.g., “non-linear sub-manifolds of much lower dimensionality”) with probabilistic concepts (e.g., “low density regions of the input space”). Similarly, my foundational paper from 2014, cited above,<sup>12</sup> combines a geometric model with a probabilistic model, in a principled way, or so I claim.

So let’s look at that these two models, first the probabilistic model, and then the geometric model.

---

11. Salah Rifai et al., *The Manifold Tangent Classifier*, 24 ADVANCES IN NEURAL INFO. PROCESSING SYSTEMS 2294 (2011) (citations omitted).

12. McCarty, *supra* note 9.

## 1. The Probabilistic Model

**Figure 1**  
**THE PROBABILISTIC MODEL**

Brownian motion with a drift term. More precisely, a *diffusion process* generated by the following differential operator:

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}\Delta + \nabla U(\mathbf{x}) \cdot \nabla \\ &= \frac{1}{2} \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2} + \sum_{i=1}^n \frac{\partial U(\mathbf{x})}{\partial x_i} \frac{\partial}{\partial x_i}\end{aligned}$$

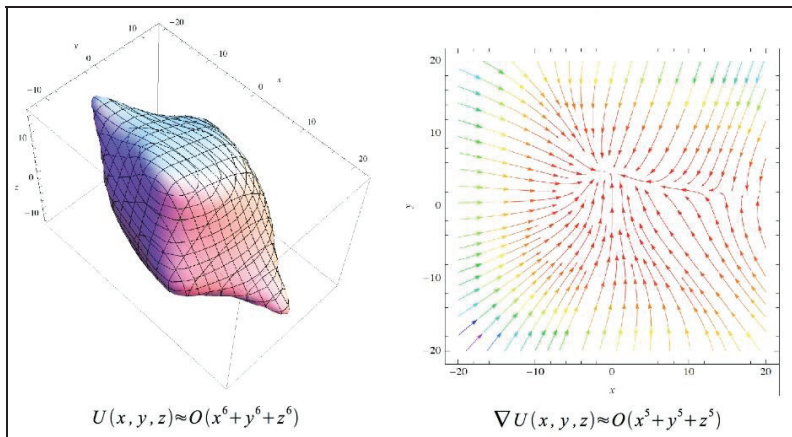
- The invariant probability measure is proportional to  $e^{2U(x)}$ .
- Thus  $\nabla U(x)$  is the *gradient* of the *log* of the probability density.

The probabilistic model is defined in Figure 1. It is *Brownian motion with a drift term*. You may be familiar, intuitively, with Brownian motion, and I will explain shortly what I mean by a “drift term.” More precisely, the probabilistic model is a *diffusion process* generated by the differential operator,  $\mathcal{L}$ , as shown in Figure 1. (There is no way to avoid using some mathematical notation here, but I will try to make it as simple as possible.) The first term in the differential operator is one-half of the *Laplacian* operator,  $\Delta$ , expressed in Cartesian coordinates, which is written out in standard calculus notation in the second equation. The second term in the differential operator is also written out in standard calculus notation in the second equation. The symbol that looks like a triangle printed upside-down is called the *gradient*, and  $U(\mathbf{x})$  represents a *scalar potential function*, i.e., a function that outputs a single real-valued number at each point,  $\mathbf{x}$ , of our  $n$ -dimensional space. Now, if we had a diffusion process generated by just the first term here, that is, a diffusion process generated by one-half of the Laplacian, we would have pure Brownian motion. But pure Brownian motion *dissipates*. By that I mean: It has no invariant probability measure, no “steady-state” distribution, other than zero. However, if we add this second term—which is called the “drift term”—it turns out that we have a finite non-zero probability measure as an invariant. In fact, there is a theorem which states that the invariant probability measure is proportional to  $e$  raised to the power  $2U(\mathbf{x})$ . This means that the gradient of  $U(\mathbf{x})$  is proportional to the *gradient* of the *log* of the probability density. And this is an important result: It means, among

other things, that we can estimate the gradient of  $U(\mathbf{x})$  from sample data, as we will see later.

Figure 2 shows an example. (Thanks to *Mathematica* for these graphics.) It's a three-dimensional example, so that we can visualize it easily. The first figure illustrates a potential function,  $U(\mathbf{x}) = U(x, y, z)$ , which is defined here as a sixth-degree polynomial. What we're seeing is a contour plot of a *surface* in three-dimensional space, namely, the surface on which  $U(\mathbf{x})$  is equal to a constant. The second figure illustrates the gradient of  $U(\mathbf{x})$ , which is a fifth-degree polynomial. What we're seeing is a plot of the *gradient vector*, which is also called the *drift vector*, in the  $xy$  plane at the value  $z = -10$ . Notice that all the arrows in the second figure are pointing towards a point somewhere in the center of the plot. This illustrates how the drift term works in the diffusion equation in Figure 1. Remember: Brownian motion by itself would dissipate completely, but the drift vector is counteracting the dissipative effects of the Laplacian term. Intuitively, we can think of the drift vector as "transporting probability mass towards the origin." Furthermore, if the two terms are in perfect balance, the diffusion process would be maintaining an *invariant probability measure*. In an important sense, then, all of the critical information about the invariant probability measure, and the diffusion process generating it, is captured by this drift vector.

Figure 2



One more observation about the second figure in Figure 2: If you look at the pattern of arrows pointing towards the origin in this plot, you might realize that you could use the drift vector to define a



*nonlinear coordinate system* in what was initially a *linear* Euclidean space. This leads us to a discussion of the geometric model.

## 2. The Geometric Model

The geometric model is defined in Figure 3. To implement the idea of prototype coding, we choose a *radial* coordinate,  $\rho$ , and the *directional* coordinates,  $\theta_1, \theta_2, \dots, \theta_{n-1}$ , where  $n$  is the dimensionality of the initial Euclidean space. The radial coordinate will follow the gradient of  $U(\mathbf{x})$ , as suggested in Figure 2, and the directional coordinates will be defined in such a way as to be *orthogonal* to the gradient of  $U(\mathbf{x})$ . But what we really want is a lower-dimensional subspace, a  $k$ -dimensional subspace, say, where  $k < n$ . Somehow, we must choose  $k-1$  out of the  $n-1$  directional coordinates, and project our diffusion process onto this  $k-1$  dimensional subspace, which can then be combined with our one-dimensional radial coordinate to give us a  $k$ -dimensional space. The device we need is a *Riemannian metric*, which we interpret as a measure of *dissimilarity*. (Again, we cannot avoid using some mathematical terminology and notation here: A Riemannian metric is a generalization of the concept of *distance* to a nonlinear space, and it specifies how a nonlinear space is stretched and curved.) Crucially, the dissimilarity metric should depend on the probability measure. Roughly speaking, the dissimilarity should be small in a region in which the probability density is high, and large in a region in which the probability density is low. This will give us the properties that we need for manifold learning.

**Figure 3**  
**THE GEOMETRIC MODEL**

To implement the idea of prototype coding, we choose:

- A *radial* coordinate,  $\rho$ , which follows  $\nabla U(\mathbf{x})$ .
- The *directional* coordinates,  $\theta_1, \theta_2, \dots, \theta_{n-1}$ , orthogonal to  $\nabla U(\mathbf{x})$ .

But we actually want a *lower-dimensional* subspace, obtained by projecting our diffusion process onto a  $k-1$  dimensional subset of the directional coordinates. The device we need is a Riemannian metric,  $\mathbf{g}_{ij}(\mathbf{x})$ , which we interpret as a measure of *dissimilarity*. Crucially, *the dissimilarity metric should depend on the probability measure*.

For the precise definition of my Riemannian dissimilarity metric, and for the proof that it possesses the desired properties, you



will have to consult my foundational paper.<sup>13</sup> Once these details are established, however, we can compute the directional coordinates in three steps:

1. We find a *principal axis* for the  $\rho$  coordinate, by minimizing the Riemannian distance along the drift vector.
2. We choose  $k-1$  *principal directions*, from a point somewhere along the principal axis, by computing the minimal *eigenvectors* of the Riemannian dissimilarity matrix,  $g_{ij}(x)$ . (Some familiarity with *linear algebra* is necessary to understand this step.)
3. To compute the  $k-1$  *coordinate curves*, we follow the *geodesics* of the Riemannian metric in each of the  $k-1$  principal directions. (A geodesic is a curve in the Riemannian manifold with minimal length, as measured by the Riemannian metric.)

Notice, in all three steps, that we are minimizing dissimilarity and maximizing probability. Figure 4 shows the results of these computations in the example that was illustrated in Figure 2. This is a three-dimensional example, so we are looking for a two-dimensional subspace. You can see that the principal axis in Figure 4 follows the axis of maximal probability in Figure 2. The other radial coordinate curves have a shorter Euclidean distance to the origin, although the Riemannian distance to the origin along these curves is the same. There is only one principal direction at each point on the principal axis, as indicated by the blue arrows, and you can see that the geodesic coordinate curves follow the contours of the potential function,  $U(x)$ , in Figure 2. It is obvious that the coordinate system is nonlinear.

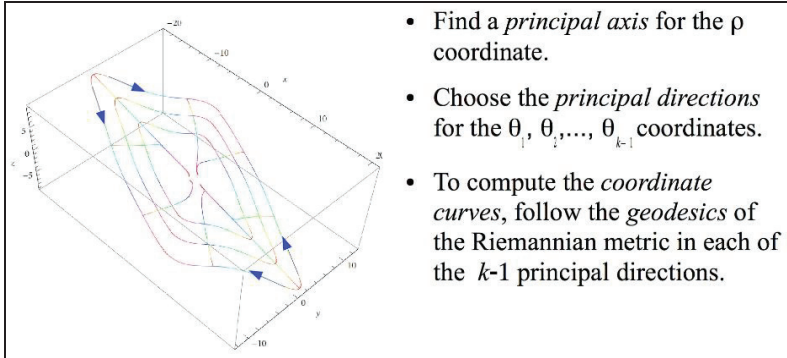
So that's my theory of manifold learning. Notice that it follows and exploits the “**(unsupervised) manifold hypothesis**, according to which real world data presented in high dimensional spaces is likely to concentrate in the vicinity of non-linear sub-manifolds of much lower dimensionality.”<sup>14</sup>

---

13. *Id.*

14. Rifai et al., *supra* note 11.

Figure 4



### 3. Prototypical Clusters

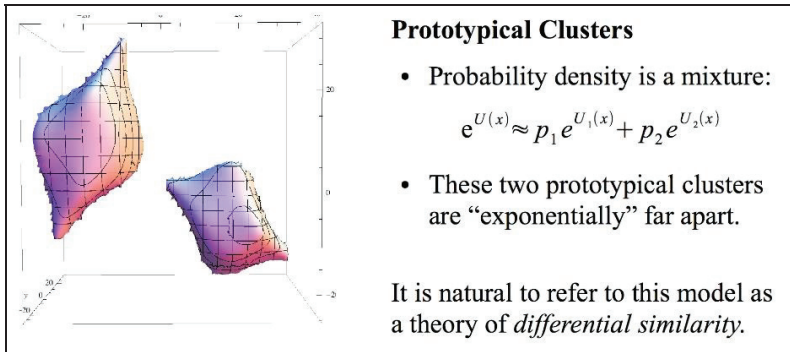
In many real-world cases, there are multiple prototypes, as illustrated by the toy example in Figure 5. Here, I have taken two copies of the potential function in Figure 2, and I have rotated them and translated them to new positions. I call these *prototypical clusters*, and I represent their probability density as a *mixture*, with two potential functions,  $U_1(\mathbf{x})$  and  $U_2(\mathbf{x})$ . If you focus on  $U(\mathbf{x})$  in the formula in Figure 5, which is equal to the *log* of the probability density, you can see that the gradient of  $U(\mathbf{x})$  in a neighborhood of the first prototype would be approximately the same as the gradient of  $U_1(\mathbf{x})$ , and the gradient of  $U(\mathbf{x})$  in a neighborhood of the second prototype would be approximately the same as the gradient of  $U_2(\mathbf{x})$ . Thus, intuitively, using our Riemannian dissimilarity metric, we could say that these two prototypical clusters are “exponentially” far apart. Notice, too, that our model satisfies the “**manifold hypothesis for classification**, according to which points of different classes are likely to concentrate along different sub-manifolds, separated by low density regions of the input space.”<sup>15</sup>

Because of the prominent role of the Riemannian dissimilarity metric in this theory, I will often refer to it as a theory of *differential similarity*.

---

15. *Id.*

**Figure 5**  
**PROTOTYPICAL CLUSTERS**



## B. Deep Learning

Let’s now return to the paper in the Conference on Neural Information Processing Systems (NIPS 2011), and look at the authors’ first hypothesis, which was omitted in our previous discussion:

1. The **semi-supervised learning hypothesis**, according to which learning aspects of the input distribution  $p(x)$  can improve models of the conditional distribution of the supervised target  $p(y|x)$  . . . This hypothesis underlies not only the strict semi-supervised setting where one has many more unlabeled examples at his disposal than labeled ones, but also the successful unsupervised pretraining approach for learning deep architectures . . . .<sup>16</sup>

The key phrase here is “learning deep architectures.” This phrase refers to a multi-layered classifier, usually for a vision system, which (i) learns a set of features bottom-up, in an unsupervised manner, and then (ii) applies a supervised learning algorithm at the top level. The architecture can be traced back to three important papers from 2006,<sup>17</sup> which initiated the modern field of “deep learning.”<sup>18</sup>

For a standard example, let’s consider the MNIST dataset of handwritten digits, 0 through 9,<sup>19</sup> which is the kind of image that the

16. *Id.* at 2294 (citations omitted).

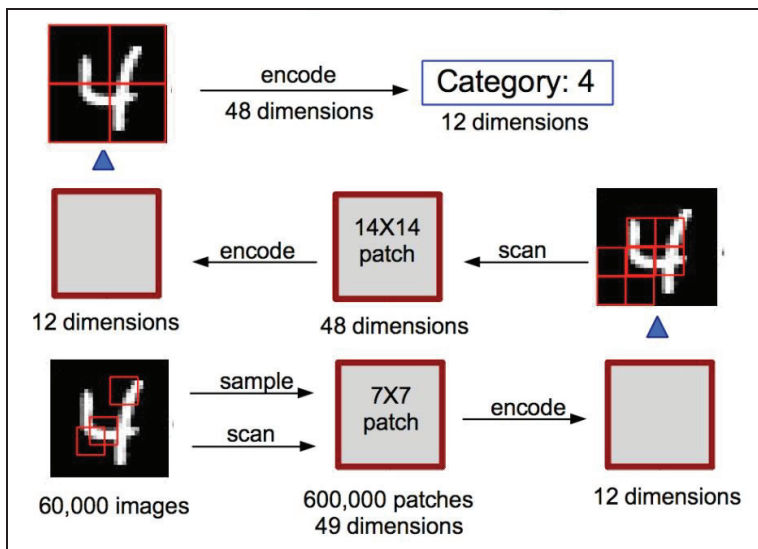
17. See Bengio et al., *supra* note 8; Hinton, Osindero & Teh, *supra* note 8; Ranzato et al., *supra* note 8.

18. Most commercial deep learning systems today are based on fully supervised learning, or “learning with a teacher,” but the original model was semi-supervised, which is much more interesting, theoretically.

19. Yann LeCun et al., *Gradient-Based Learning Applied to Document Recognition*, 86 PROC. IEEE 2278, 2287 (1998).

Post Office needs to recognize in order to process ZIP codes automatically. Each image consists of  $28 \times 28$  pixels with values in the range  $[0, 255]$ .<sup>20</sup> (Such images are sometimes called “quasi-binary.” The original NIST dataset was binary, but the edges of the digits in Modified NIST, or MNIST, have been blurred slightly by preprocessing.<sup>21</sup>) The full dataset consists of 60,000 training set images and 10,000 test set images.<sup>22</sup> Historically, it has been used as a benchmark for supervised pattern recognition,<sup>23</sup> but we are interested in viewing it as a problem in unsupervised feature learning.

Figure 6



An architecture for deep learning on the MNIST dataset is shown in Figure 6, based on several examples in the recent literature.<sup>24</sup> The process starts in the lower-left corner and follows the

20. *Id.*

21. *Id.*

22. *Id.*

23. *Id.*

24. See, e.g., Marc’Aurelio Ranzato, Unsupervised Learning of Feature Hierarchies (May 2009) (unpublished Ph.D. dissertation, New York University), [https://www.cs.nyu.edu/media/publications/ranzato\\_marcaurelio.pdf](https://www.cs.nyu.edu/media/publications/ranzato_marcaurelio.pdf) [<https://perma.cc/TTC8-3VHB>]; Adam Coates, Demystifying Unsupervised Feature Learning (Sept. 2012) (unpublished Ph.D. dissertation, Stanford University), [https://cs.stanford.edu/~acoates/papers/acoates\\_thesis.pdf](https://cs.stanford.edu/~acoates/papers/acoates_thesis.pdf) [<https://perma.cc/7UEU-D5X6>].

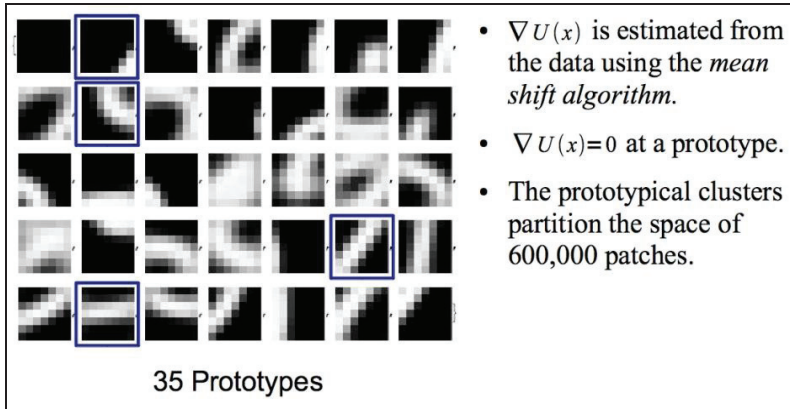
arrows to the upper-right corner. The first step is to *scan* and *randomly sample* the images to extract a collection of  $7 \times 7$  “patches” from each one. If we choose a sampling rate of 10 scans per image, which is approximately 2%, we will end up with *600,000* patches, each one represented as a point in a 49-dimensional space. Our immediate task is to reduce the dimensionality of this space, and one obvious way to do this is to apply the theory of manifold learning that we examined previously.

Figure 7 shows how to get started. First, as I suggested earlier, we need to estimate the gradient of  $U(\mathbf{x})$  from the sample data, namely, our sample of  $7 \times 7$  patches. The estimation procedure is explained in my forthcoming paper on *Differential Similarity in Higher Dimensional Spaces: Theory and Applications*, but it uses a well-known technique in the literature based on the *mean shift algorithm*.<sup>25</sup> Second, we need to identify a set of prototypes. The main criteria are listed in Figure 7: (i) We want the gradient of  $U(\mathbf{x})$  to equal zero at a prototype, which means that the prototype will be located at a *mode* of the probability distribution; and (ii) We want the prototypical clusters to partition the space of *600,000* patches. Figure 7 shows 35 prototypes that satisfy these criteria (it is not a uniquely defined set), and marks four prototypes that I have chosen, subjectively, for further detailed study. I think you would agree with me that these are reasonable features for the digits 0 through 9.

---

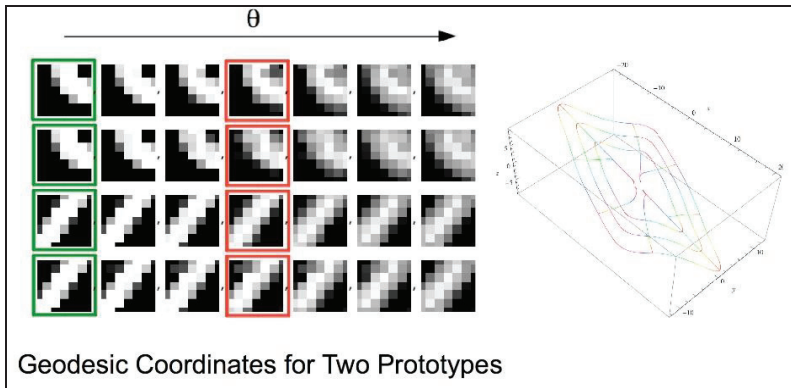
25. See Keinosuke Fukunaga & Larry D. Hostetler, *The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition*, 21 IEEE TRANSACTIONS ON INFO. THEORY 32 (1975); Yizong Cheng, *Mean Shift, Mode Seeking, and Clustering*, 17 IEEE TRANSACTIONS ON PATTERN ANALYSIS & MACHINE INTELLIGENCE 790 (1995); Dorin Comaniciu & Peter Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, 24 IEEE TRANSACTIONS ON PATTERN ANALYSIS & MACHINE INTELLIGENCE 603 (2002).

Figure 7



We now follow the three steps listed in Figure 4, placing each of our prototypes at the origin of one of the coordinate systems that we are trying to compute. In Figure 4, the computation was confined to just three dimensions, and we are now working in 49 dimensions, but the procedure is exactly the same. Figure 8 shows a small sample of the results, along with a comparison to the results in Figure 4. (Note: There are several parameters that must be set when we run this computation, and I have chosen a variant for Figure 8 that leads to a particularly simple illustration. In my forthcoming paper on *Differential Similarity in Higher Dimensional Spaces: Theory and Applications*, I will explore a range of parameter settings that lead to more complex and more realistic coordinate curves.) In Figure 8, we are looking at two geodesic coordinate curves through the 49-dimensional space, for each of two prototypes, corresponding to the  $\theta$  coordinate curves in Figure 4. The patches on the left (outlined in green) are points on the principal axes; the patches in the middle (outlined in red) are points at a location approximately  $90^\circ$  along the coordinate curves; and the patches on the right are points at approximately  $180^\circ$ . Notice that the coordinate curves converge to essentially the same patch at  $180^\circ$ , but they are distinct at  $90^\circ$ . This is one of the properties that we would expect in our particular nonlinear coordinate system.

Figure 8



Let's now return to the architecture for deep learning shown in Figure 6. Suppose we choose the 12 most informative coordinate curves for each prototype, and use these to *encode* the 7 X 7 patches. This step—the *dimensionality reduction* step—is shown in the lower-right corner of Figure 6. We can then assemble four adjacent 7 X 7 patches into a 2 X 2 matrix, and *resample* the image using the larger 14 X 14 patch. If we use the encoded values of the 7 X 7 patches in the resampling procedure, each 14 X 14 patch would be represented as a point in a 48-dimensional space. And since there are only 9 distinct scans of our 2 X 2 matrix across a 28 X 28 image, we can actually sample at a rate of 100% and generate 540,000 data points in the new space. We can then apply the techniques of manifold learning again, and reduce the dimensionality back to 12. In summary, our general procedure is:

- construct the *product manifold* (in this case, the 2 X 2 matrix) from the encoded values of the smaller patches, and then
- construct a *submanifold* using the Riemannian dissimilarity metric.

We can now repeat this procedure, as shown in the top line of Figure 6. The submanifolds from the prior step are assembled into a new 2 X 2 product manifold, and the dimensionality of the space is reduced again, using the Riemannian dissimilarity metric. Notice that we have designed the architecture to maintain a roughly constant dimensionality as we proceed from the bottom to the top of Figure 6.

Finally, at the top level of the architecture in Figure 6, we would like to output a classification, for example: *This particular 12-dimensional manifold belongs to the category "4."* In the original




model of deep learning,<sup>26</sup> this would be a supervised learning step, but there is another possibility. Could the classification at the top level also be an *unsupervised* step? I do not know the answer to this question, because I am still crunching the data, but when I have the final results I will publish them in my paper on *Deep Learning with a Riemannian Dissimilarity Metric*.

## II. A LOGICAL LANGUAGE

We can now address the title of this talk: *How to Ground a Language for Legal Discourse in a Prototypical Perceptual Semantics*. Specifically, we will see how to use the machinery of manifold learning and deep learning to define a *semantics* for a logical language. Why do I call this a “prototypical perceptual semantics”? Well, it’s a *prototypical* semantics because it is based on my model of prototypical clusters, as we will see. Why is it a prototypical *perceptual* semantics? Well, notice that our primary examples are drawn from the field of image processing, and therefore, if we can build a logic on these foundations, we will have a plausible account of how human cognition could be *grounded* in human perception.

Figure 9

Rewrite the top four patches as a *logical product*:



Use the syntax of my *Language for Legal Discourse (LLD)*:

```

(ImageFOUR ?i (Patch23 ?p23)
                (Patch14 ?p14)
                (Patch37 ?p37)
                (Patch08 ?p08))

```

Recall our general procedure for deep learning in the image processing example: (i) construct the *product manifold* from the encoded values of the smaller patches, and then (ii) construct a *submanifold* using the Riemannian dissimilarity metric. In the top

26. See *supra* note 18.

line of Figure 6, for example, four “patches” were combined into an “image” of the digit “four” and the dimensionality of the image was reduced from 48 to 12. The four patches are shown again in Figure 9, arranged as a *logical product*, or what we normally call a *conjunction*. Intuitively, an image is the conjunction of four patches. How can we turn this observation into a logic? The expression at the bottom of Figure 9 shows how we might encode the MNIST example as an *atomic formula* in a logical language that I developed a number of years ago for the representation of legal discourse. I called this a *Language for Legal Discourse (LLD)*,<sup>27</sup> and I have used it for various applications over the years.<sup>28</sup>

Let’s see how our image processing example maps into the syntax of *LLD*. The expression in Figure 9 says that  $?p23$  is a variable that can be instantiated to a point on the manifold *Patch23*,  $?p14$  is a variable that can be instantiated to a point on the manifold *Patch14*, ... , and  $?i$  is a variable that can be instantiated to a point on **ImageFOUR**, which is a *submanifold* of the *product manifold* constructed from the four patches. To assist your intuition, let’s also review how this atomic formula would be interpreted in classical logic. This would be a *sorted* logic, classically, in which *Patch23*, *Patch14*, etc., are syntactic “sorts,” and **ImageFOUR** is a syntactic “predicate.” Thus, semantically, the expression in Figure 9 would be saying that  $?p23$  is a variable that can be instantiated to an element of the set *Patch23*,  $?p14$  is a variable that can be instantiated to an element of the set *Patch14*, and so on. Now, what is the interpretation of a predicate in classical logic? A predicate is interpreted semantically as a *relation*, which is a subset of the *Cartesian product* of those four sets which, in a sorted logic, provide the interpretation of the four sorts. Furthermore, any subset will do, so the set of all subsets is the same as the set of all relations. Finally, although the variable  $?i$  is not commonly used in classical logic, I have always used it in my *Language for Legal Discourse (LLD)*.<sup>29</sup> Here, interpreted classically,  $?i$  would be a variable that can be

---

27. L. Thorne McCarty, *A Language for Legal Discourse: I. Basic Features*, in ICAIL ‘89 PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 180 (1989).

28. See, e.g., L. Thorne McCarty, *Ownership: A Case Study in the Representation of Legal Concepts*, 10 ARTIFICIAL INTELLIGENCE & L. 135 (2002); L. Thorne McCarty, *Deep Semantic Interpretations of Legal Texts*, in ICAIL ‘07 PROCEEDINGS OF THE 11TH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 217 (2007).

29. See McCarty, *supra* note 27, at 181.

instantiated to an instance of the relation, i.e., to an element of the Cartesian product, or what we might intuitively think of as a *relationship*.

Figure 10

For this interpretation, we need a logical language based on *category theory*:

	<i>objects</i>	<i>morphisms</i>	<i>logic</i>
<b>Set</b>	abstract sets	arbitrary mappings	classical
<b>Top</b>	topological spaces	continuous mappings	intuitionistic
<b>Man</b>	differential manifolds	smooth mappings	???

Define: Categorical Product

- In **Man**, this is the product manifold.

Define: Categorical Subobject

- In **Man**, this is a submanifold.

So what I am proposing, in general, is to replace the standard semantics of classical logic, based on sets and their elements, with a semantics based on manifolds and their points. And the way to do this, systematically, I claim, is to work with a logical language based on *category theory*, or what is known as a *categorical logic*. (It's impossible to explain category theory in three minutes, but I can give you some references if you would like to explore the subject further.<sup>30</sup>) Put very simply, category theory studies two kinds of mathematical structures, *objects* and *morphisms*. See Figure 10. Depending on which category we are working with, the objects and the morphisms will be different. In the category called **Set**, for example, the objects are abstract sets and the morphisms are arbitrary mappings. In the category called **Top**, the objects are topological spaces and the morphisms are continuous mappings. And, in the category called **Man**, the objects are differential manifolds and the morphisms are smooth mappings. What do I mean by a *smooth*

30. See, e.g., F. WILLIAM LAWVERE & STEPHEN H. SCHANUEL, *CONCEPTUAL MATHEMATICS: A FIRST INTRODUCTION TO CATEGORIES* (2d ed. 2009); F. WILLIAM LAWVERE & ROBERT ROSEBRUGH, *SETS FOR MATHEMATICS* (2003); STEVE AWODEY, *49 OXFORD LOGIC GUIDES: CATEGORY THEORY* (2006); S. Abramsky & N. Tzevelekos, *Introduction to Categories and Categorical Logic*, in *LECTURE NOTES IN PHYSICS 813: NEW STRUCTURES FOR PHYSICS 3* (Bob Coecke ed., 2011); BART JACOBS, *CATEGORICAL LOGIC AND TYPE THEORY* (2001). These references are arranged in order from the more elementary to the more advanced, so that the reader can access the literature at whatever level would be the most comprehensible.

*mapping?* A smooth mapping is a continuously differentiable function, up to some finite order  $k$ , or, alternatively, for all  $k$ , which would be an infinitely differentiable function. It should be clear that I have been using the category **Man** all along in my work on the theory of differential similarity.

The remainder of Figure 10 lists two standard concepts from category theory, which are both defined in Section 3 of my ICAIL 2015 paper.<sup>31</sup> Although these definitions are universal, they designate different mathematical objects in different categories. For example, the *categorical product* in **Set** is the ordinary Cartesian product, and the *categorical subobject* is an ordinary subset. Recall: That's what we used for the semantics of classical logic. But in **Man**, the categorical product is the product manifold, and the categorical subobject is a submanifold. That's what we used in our image processing example. In general, we can construct a logic using any of these categories, but we get a different logic in each case. Using the category **Set**, we obtain classical logic. Using the category **Top**, we obtain intuitionistic logic, which is strictly weaker than classical logic. As far as I know, a categorical logic based on the category **Man** does not have a standard name in the literature, but I show in Section 3 of my ICAIL 2015 paper that it has the proof theory of a syntactically restricted version of intuitionistic logic,<sup>32</sup> with several novel properties.

Figure 11

**Sequent Calculus:**

$$\mathbf{1} \vdash (\text{Control } ?r \text{ (Actor macomber) (Corporation so)})$$

$$(\text{Control } ?r \text{ (Actor macomber) (Corporation so)}) \vdash \mathbf{0}$$

- **Actor** and **Corporation** are interpreted as differential manifolds.
- **macomber** and **so** are interpreted as points on these manifolds.
- **Control** is interpreted as a submanifold of the product manifold.

$$(\mathbf{Q} \ ?q \text{ (Actor } ?a) \text{ (Corporation } ?c))$$

$$\vdash_c (\text{Control } ?r \text{ (Actor } ?a) \text{ (Corporation } ?c))$$

- A sequent is interpreted as a morphism.

31. See McCarty, *supra* note 1, at 93-94.

32. See *id.* at 94-96.

A fragment of the proof theory is illustrated in Figure 11. It uses what logicians call a *sequent calculus*. The two expressions at the top of the figure are called *sequents*, and they represent the concept of *provability*. The odd symbol that looks like a turnstile is called a “turnstile,” and the sequent is interpreted as saying: “From the formula on the left of the turnstile you can prove the formula on the right of the turnstile.” What are the formulas in these examples? You will see that I have switched from the image processing example in Figure 9 to a toy legal example suggested by the case of *Eisner v. Macomber* in my ICAIL 1995 paper.<sup>33</sup> The individual *macomber* (Myrtle H. Macomber in the real case) is an *Actor*, the individual *so* (Standard Oil in the real case) is a *Corporation*, and the predicate is *Control*. So the formula expresses the proposition that “the actor Myrtle H. Macomber controls the corporation Standard Oil.” Let’s first compare the semantic interpretation of this formula in classical logic with its interpretation in a categorical logic based on the category of differential manifolds. In classical logic, *Actor* and *Corporation* would be sets, and *macomber* and *so* would be elements in those sets. *Control* would be a relation, that is, a subset of the Cartesian product of *Actor* and *Corporation*. But in the category **Man**, *Actor* and *Corporation* would be manifolds, and *macomber* and *so* would be points on those manifolds. Likewise, *Control* would be a manifold, specifically, a submanifold of the product manifold of *Actor* and *Corporation*. If you now transfer your intuitions about the image processing example back to this simple legal example, you should be able to see why this is important. It means that we can use the machinery of our prototypical clusters, our differential similarity metric, our geodesic coordinate system, etc., for the *Actor* manifold, the *Corporation* manifold, the *Control* manifold, and so on. And this is, concretely, what I mean by a prototypical perceptual semantics.

Returning to the proof theory, what are these two sequents saying? The first sequent says: “From the proposition **1** you can prove the proposition *Control* ...” Well, **1** means *true*. So, if you can prove this particular proposition from *true*, then the proposition itself is true, that is, “Myrtle H. Macomber controls Standard Oil.” The second sequent says: “From the proposition *Control* ... you can prove the proposition **0**.” Well, **0** means *false*. So, if from this particular proposition you can prove *false*, then the proposition itself is false, that is, “Myrtle H. Macomber does not control Standard

---

33. See McCarty, *supra* note 2.

Oil.” The final remark at the bottom of Figure 11 is also important: *A sequent is interpreted as a morphism*. In the category **Set**, remember, a morphism is an arbitrary mapping between abstract sets, so this interpretation does not tell us very much in **Set**. But in the category **Man**, a morphism is a smooth mapping between differential manifolds, and this means that a sequent is interpreted as a smooth mapping from the manifold on the left-hand side to the manifold on the right-hand side. My ICAIL 2015 paper gives you more information about how this works. For the two sequents at the top of Figure 11, you have to know that **1** is a *terminal object* in the category **Man**, which is the 0-dimensional manifold consisting of a single point,  $\{0\}$ , and **0** is an *initial object* in the category **Man**, which is the empty manifold,  $\{\}$ , but you will have to consult the paper for the technical details.

The third sequent in Figure 11 is a slightly more complicated example. In this example, there is an atomic formula on the left-hand side of the turnstile as well as an atomic formula on the right-hand side. Furthermore, instead of constant individuals, *macomber* and *so*, we have variables:  $?a$  is a variable that can be instantiated to a particular *Actor*, and  $?c$  is a variable that can be instantiated to a particular *Corporation*. There is also another technical detail of the proof theory illustrated here, but you will have to consult the paper to see how it works: The turnstile has a *context*,  $C$ , attached to it, which lists all the variables in the two formulas. What is this sequent saying? Well, we don’t know what the predicate  $Q$  means, but whatever it means, the sequent is saying that from the relation  $Q$  between  $?a$  and  $?c$  you can prove the relation *Control* between  $?a$  and  $?c$ . And, since a sequent is interpreted as a morphism, this means that there exists a smooth mapping in the category **Man** from the point  $?q$  on the manifold  $Q$  to the point  $?r$  on the manifold *Control*. Again, the details are in the paper. Let me emphasize, though, that these details are not novel, although they may be somewhat unfamiliar, even to some logicians. They are all features of categorical logic,<sup>34</sup> and all I have done is to apply these ideas to the category of differential manifolds.

For the full sequent calculus, we need to add *proof rules* to our system, in order to derive sequents from sequents and to interpret the logical connectives. See Figure 12. The structural rule for *cut* is just a rewriting of the rule for the composition of morphisms. It says: If

---

34. See, e.g., AWODEY, *supra* note 30; Abramsky & Tzevelekos, *supra* note 30; JACOBS, *supra* note 30.



from  $\phi$  you can prove  $\theta$  and from  $\theta$  you can prove  $\psi$ , then from  $\phi$  you can prove  $\psi$ . The rule for *conjunction* is just a rewriting of the definition (see the paper) of the categorical product listed on Figure 10. This is a bidirectional rule, with the  $\wedge$ -introduction rule reading from top to bottom, and the  $\wedge$ -elimination rule reading from bottom to top. Given the cut rule and the conjunction rule, we can now add *horn axioms* as shown on the figure. This says: From the conjunction of the  $Q$ s you can prove  $P$ , and any variables in the context  $C$  will be universally quantified, implicitly, at the top level. If you are familiar with the programming language PROLOG, you will understand that adding these horn axioms to the two rules on Figure 12 gives us the basics of *horn clause logic programming*.<sup>35</sup> But we can go even further. I do not have time to cover this material in my talk, but in the paper I show how to add introduction and elimination rules for explicit *existential* and *universal quantifiers*, and for *implication*, as well as axioms for *simple embedded implications*. The end result is an extended logic programming language, based on intuitionistic logic, which I proposed and analyzed a number of years ago,<sup>36</sup> and which provides the foundation for my *Language for Legal Discourse (LLD)*.<sup>37</sup> I was surprised when I discovered this fact, because I was not thinking about differential manifolds at the time.

Figure 12

<p><b>Structural Rule for cut:</b></p> $\frac{\Gamma, \phi \vdash_C \theta \quad \Gamma, \theta \vdash_C \psi}{\Gamma, \phi \vdash_C \psi}$ <p><b>Introduction and Elimination Rules for conjunction:</b></p> $\frac{\Gamma \vdash_C \psi_1 \quad \Gamma \vdash_C \psi_2}{\Gamma \vdash_C \psi_1 \wedge \psi_2}$ <p><b>Horn Axioms:</b></p> $Q_1 \wedge Q_2 \wedge \dots \wedge Q_n \vdash_C P$ <p><b>This is sufficient for <i>horn clause logic programming</i>.</b></p>
--

35. See, e.g., JOHN W. LLOYD, FOUNDATIONS OF LOGIC PROGRAMMING 10 (2d ed. 1987).

36. L. Thorne McCarty, *Clausal Intuitionistic Logic: I. Fixed-Point Semantics*, 5 J. LOGIC PROGRAMMING 1 (1988); L. Thorne McCarty, *Clausal Intuitionistic Logic: II. Tableau Proof Procedures*, 5 J. LOGIC PROGRAMMING 93 (1988); L. Thorne McCarty, *Circumscribing Embedded Implications (Without Stratifications)*, 17 J. LOGIC PROGRAMMING 323 (1993).

37. See McCarty, *supra* note 27.



Another surprise was the discovery of two novel properties in a categorical logic based on the category **Man**:

1. I have already explained that a sequent in categorical logic is interpreted as a morphism, but a proof is a chain of inference rules applied to sequents, and this means that a proof is interpreted as a *composition* of morphisms. Thus, in the category **Man**, a proof is a smooth mapping of differential manifolds, starting with the atomic formulas, which can be chosen to represent prototypical clusters. Proofs in classical logic do not have this property, since they are interpreted invariably as arbitrary mappings of abstract sets.
2. In **Set** and **Top**, every subset of an object is a subobject. But in the category **Man**, not every subset of a manifold is a submanifold. (A standard counter-example is shown in Figure 7 of my ICAIL 2015 paper.<sup>38</sup>) This has possible implications for the logic:
  - In standard *second-order* logic, based on **Set** or **Top**, the predicate variables range across the set of all subsets of the first-order domains. But in a second-order logic based on the category **Man**, the predicate variables would range across the set of all submanifolds, which is strictly less than the set of all subsets. Does this fact have implications for Gödel's Theorem?
  - Similarly, in the category **Man**, the search space for learning first-order predicates would be strictly less than set of all subsets of the first-order domains. Does this fact have implications for Learnability?

I have framed these points as questions, because I consider them to be very speculative. Nevertheless, I think they are speculations worthy of further investigation.

### III. DEFINING THE ONTOLOGY OF LLD

In my original paper on a *Language for Legal Discourse (LLD)*, the goals of the work were described as follows:

There are many common sense categories underlying the representation of a legal problem domain: space, time, mass, action, permission, obligation, causation, purpose, intention, knowledge, belief, and so on. The idea is to select a small set of these common sense categories, . . . and . . . develop a *knowledge representation language* that faithfully mirrors the structure of

---

38. See McCarty, *supra* note 1, at 94.

this set. The language should be formal: it should have a compositional syntax, a precise semantics and a well-defined inference mechanism.<sup>39</sup>

This research programme has now been incorporated into a more general research programme on *legal ontologies*,<sup>40</sup> in which logic plays a prominent role. But what happens when we shift the foundations of *LLD* from a language based on **Set** (which is classical), or **Top** (which is intuitionistic), to a language based on **Man**? There are several questions to consider:

1. One important feature of *LLD* is the distinction between *count terms* and *mass terms*. For example, *Actor* and *Corporation* are count terms, whereas “20 shares of common stock” is a mass term. A standard way to represent this distinction in a knowledge representation language is to use a second-order logic, in which the second-order mass terms have a *measure* attached to them. Does it make a difference whether this logic is based on **Set**, or **Top**, or **Man**?
2. One of the most important features of *LLD* is the representation of events and actions,<sup>41</sup> and the various modalities over actions,<sup>42</sup> the most significant of which, in a legal context, are the modalities of *permission* and *obligation*.<sup>43</sup> In my previous work, the underlying action language was based on either a classical or an intuitionistic logic, and it was clumsy, at best. A better approach, I think, would be to take differential manifolds seriously, and represent actions (initially) by the *Lie group* of *rigid body motions* in the category **Man**.<sup>44</sup> We can then apply the theory of differential similarity to the manifold of physical actions, and generalize from there to a manifold of abstract

---

39. McCarty, *supra* note 27, at 180.

40. See, e.g., Rinke Hoekstra et al., *The LKIF Core Ontology of Basic Legal Concepts*, in PROCEEDINGS OF THE WORKSHOP ON LEGAL ONTOLOGIES AND ARTIFICIAL INTELLIGENCE TECHNIQUES 43 (P. Casanovas et al. eds., 2007).

41. See L. Thorne McCarty & Ron van der Meyden, *Reasoning About Indefinite Actions*, in PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING: PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE (KR '92) 59 (1992).

42. See L. Thorne McCarty, *Modalities Over Actions*, in PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING: PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE (KR '94) 437 (1994).

43. See L. Thorne McCarty, *Permissions and Obligations*, in PROCEEDINGS OF THE EIGHTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI '83) 287 (1983).

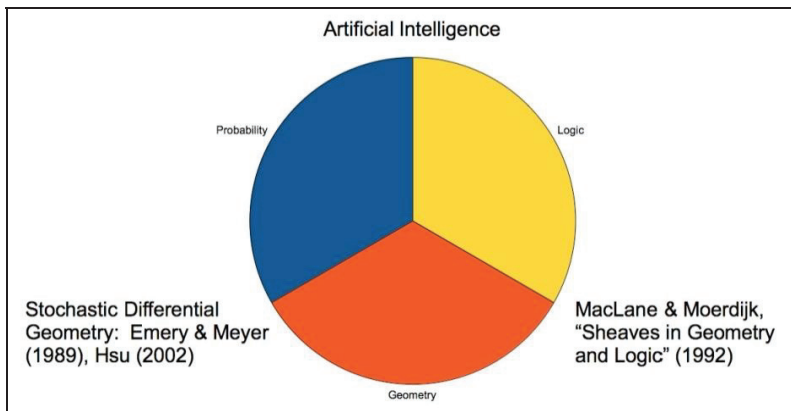
44. See, e.g., JEAN GALLIER, *GEOMETRIC METHODS AND APPLICATIONS FOR COMPUTER SCIENCE AND ENGINEERING* 459-511 (2d ed. 2011).

actions. Does this formalization lead to a distinctly different theory of the deontic modalities?

3. The best current model of the epistemic modalities, such as *knowledge* and *belief*, can be found in the literature on *justification logics*.<sup>45</sup> A justification logic adds the annotation  $t : X$  to the proposition  $X$ , and interprets this compound term as “ $X$  is justified by reason  $t$ .” Essentially,  $t$  is a proof of  $X$ . This is currently an active area of research, and there are justification logics that correspond to many different proof systems. I have introduced a new logical language here, of course, based on the category **Man**. Does this language and its proof system lead to a new variant in the family of justification logics?

These are all interesting research questions, and there are probably many more. The plan is to reconstruct my *Language for Legal Discourse* from the ground up, and to study the implications of this new approach.

**Figure 13**



#### IV. TOWARD A THEORY OF COHERENCE

The theory of differential similarity is a hybrid drawn from three areas of mathematics: Probability, Geometry, Logic. Historically, these three fields were distinct, but their boundaries

45. See, e.g., Sergei N. Artemov, *The Logic of Justification*, 1 REV. SYMBOLIC LOGIC 477 (2008); Melvin Fitting, *Reasoning with Justifications*, in TOWARDS MATHEMATICAL PHILOSOPHY, PAPERS FROM THE STUDIA LOGICA CONFERENCE 107 (D. Makinson, J. Malinowski & H. Wansing eds., 2009).

have been blurred in the past twenty or thirty years. Figure 13 shows one way to understand these relationships.

At the top of the figure, I have listed the field of Artificial Intelligence. The earliest work in AI was based on logic, while more recent work has been based primarily on probability theory. There have been numerous attempts through the years to combine these two approaches,<sup>46</sup> but it is fair to say that there is still no single unified theory that has garnered universal support. The new entry, at the bottom of the figure, is geometry. How is geometry related to probability theory? The hybrid field, which has emerged only in the past twenty or thirty years, is *stochastic differential geometry*, as represented by the two books cited.<sup>47</sup> In fact, the theorems that I have been using on the properties of Brownian motion on Riemannian manifolds come from the field of stochastic differential geometry. How is geometry related to logic? There is a rather famous book by MacLane and Moerdijk, cited in the figure,<sup>48</sup> which studies this relationship at a very abstract level. (Saunders MacLane is the co-founder, along with Samuel Eilenberg, of category theory.) In fact, what I have been doing with my own logic based on the category of differential manifolds is a concrete (and very elementary!) version of what MacLane and Moerdijk were doing in their book.

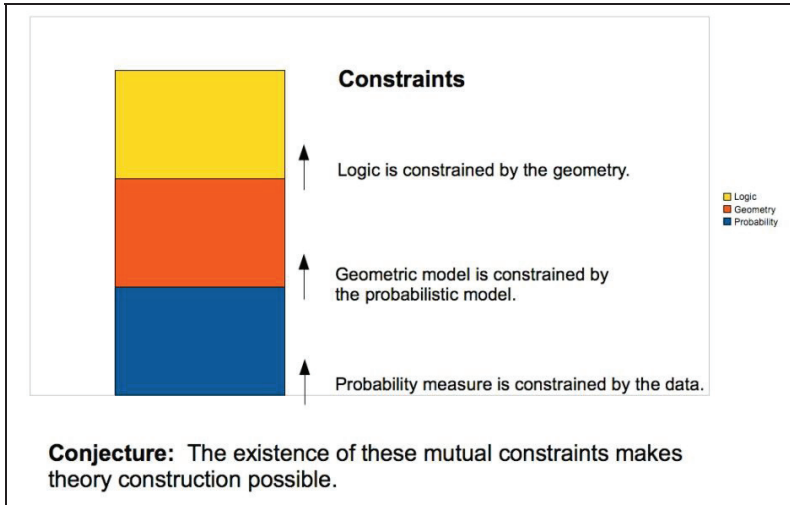
---

46. See, e.g., Nils J. Nilsson, *Probabilistic Logic*, 28 ARTIFICIAL INTELLIGENCE 71 (1986); Matthew Richardson & Pedro Domingos, *Markov Logic Networks*, 62 MACHINE LEARNING 107 (2006).

47. MICHEL EMERY & P.A. MEYER, *STOCHASTIC CALCULUS IN MANIFOLDS* (1989); ELTON P. HSU, *STOCHASTIC ANALYSIS ON MANIFOLDS* (2002).

48. SAUNDERS MAC LANE & IEKE MOERDIJK, *SHEAVES IN GEOMETRY AND LOGIC: A FIRST INTRODUCTION TO TOPOS THEORY* (1992).

Figure 14



What I am suggesting in Figure 13 is that Artificial Intelligence needs all three fields: Probability, Geometry, Logic. Furthermore, we should be proceeding *from* probability *through* geometry *to* logic, as in my theory of differential similarity. This means that we need to unfold the pie chart in Figure 13 to produce the stack of blocks in Figure 14.

Figure 14 depicts a high-level view of a cognitive model constructed from the bottom up, and a view of the constraints in this model from the top down, according to the theory of differential similarity. First, the logic is constrained by the geometry, as we have seen. Second, the geometric model is constrained by the probabilistic model, since the Riemannian dissimilarity metric depends on the probability measure. Third, the probability measure is constrained by the distribution of sample data in the actual world. These three propositions lead us to the conjecture at the bottom of the figure: *It is the existence of these mutual constraints that makes theory construction possible.*

Let's now return to the passage from Section 5 of my ICAIL 1997 paper, where we were trying to understand legal reasoning as a form of theory construction. I have reproduced part of this passage in Figure 15.

Figure 15

**ICAIL-'97, Section 5:**

... Somehow, the requirement that the exemplar of a concept must be “similar” to a prototype (a kind of “horizontal” constraint) seems to reinforce the requirement that the exemplar must be placed at some determinate level of the concept hierarchy (a kind of “vertical” constraint). How is this possible?

**This is one of the great mysteries of cognitive science.**

**It is also one of the great mysteries of legal theory.**

**Q: Is the mystery now solved?**

Recall that we were looking for a *learnable* knowledge representation language, based on prototypes. How is this possible? This is (or was) one of the great mysteries of cognitive science, and one of the great mysteries of legal theory.

Is the mystery now solved? For me, that’s just a rhetorical question, and the answer is: Yes.