How to Ground A Language for Legal Discourse In a Prototypical Perceptual Semantics

L. Thorne McCarty
Rutgers University
New Brunswick, New Jersey 08903

ABSTRACT

In a pair of papers from 1995 and 1997, I developed a computational theory of legal argument, but left open a question about the key concept of a "prototype." Contemporary trends in machine learning have now shed new light on the subject. In this paper, I will describe my recent work on "manifold learning," as well as some work in progress on "deep learning." Taken together, this work leads to a logical language grounded in a prototypical perceptual semantics, with implications for legal theory. The main technical contribution of the paper is a categorical logic based on the category of differential manifolds (**Man**), which is weaker than a logic based on the category of sets (**Set**) or the category of topological spaces (**Top**). The paper also shows how this logic can be extended to a full *Language for Legal Discourse* (*LLD*), and suggests a solution to the elusive problem of "coherence" in legal argument.

1. INTRODUCTION

In a pair of papers from 1995 and 1997, I developed a computational theory of legal argument, but left open a question about the key concept of a "prototype." The 1995 paper [31] presents a computational reconstruction of the arguments of Justice Pitney and Justice Brandeis in a seminal corporate tax case from 1920, based on a representation of *prototypes* and *deformations*. The 1997 paper [32] is mostly a critical review of the literature, but it also includes a positive thesis about the nature of legal argument in Section 5 (entitled "The Correct Theory"). Crucially, in Section 5, in the context of a discussion about the role of theory construction in the law, the paper raises a question about the relevance of machine learning:

> . . . 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 [Smith and Medin, 1981], 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. . . .

The paper then proceeds to discuss Ronald Dworkin's thesis in "Hard Cases" [11] and "Law's Empire" [12], and concludes that the mystery can only be solved by developing a computational theory of "coherence" in legal argument.

Contemporary trends in machine learning have now shed new light on the subject. In Section 2 (entitled "Prototype Coding"), I will describe my recent work on "manifold learning" (see Section 2.1), as well as some work in progress on "deep learning" (see Section 2.2). Taken together, this work leads to a logical language grounded in a prototypical perceptual semantics, with implications for legal theory. The main technical contribution of the paper is a categorical logic based on the category of differential manifolds (**Man**), which is weaker than a logic based on the category of sets (**Set**) or the category of topological spaces (**Top**). The details are presented in Section 3 (entitled "A Logical Language"). In Section 4, which is primarily an outline of future work, I will then show how this logic can be extended to a full *Language for Legal Discourse* (*LLD*).

Finally, in Section 5, I will suggest a solution to the elusive problem of "coherence" in legal argument.

2. PROTOTYPE CODING

What is prototype coding? The basic idea is to represent a point in an n-dimensional space by 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 idea has been explored, in various guises, in the literature on "manifold learning" [46, 44, 6], and to some extent in the literature on "deep learning" [7, 17, 42]. In this section, I will review these developments, briefly, with an emphasis on my own recent work.

ICAIL'15, June 08–12, 2015, San Diego, CA, USA
Copyright 2015 ACM 978-1-4503-3522-5/15/06 ...$15.00.
http://dx.doi.org/10.1145/2746090.2746091.

The main reference for Section 2.1 is [34]. Section 2.2 reviews work which is still "in progress," but which will eventually appear in [36] and [35]. These papers are technically complex, but the ideas can often be conveyed with relatively simple graphics, and I have used some of these illustrations in the present paper. For additional illustrations in Section 2.1, the reader may wish to consult the Appendix in [34], which is available on ResearchGate[1]. Additional illustrations for Section 2.2 can be found in a set of slides from a Workshop at the University of San Diego (held in March, 2014), which is also available on ResearchGate[2].

2.1 Manifold Learning

To introduce the fundamental ideas of manifold learning, let's look at a passage in a recent paper [43] in one of the main conferences on neural networks: *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 ... [citations omitted]
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.

Observe that these hypotheses combine geometric concepts ("non-linear sub-manifolds of much lower dimensionality") with probabilistic concepts ("low density regions of the input space"). The authors then show how to exploit these hypotheses to construct a learning algorithm which they call the "Manifold Tangent Classifier (MTC)".

My own approach to manifold learning is presented in a recent paper entitled "Clustering, Coding, and the Concept of Similarity" [34]. In some ways similar to [43], this paper develops a theory of *clustering* and *coding* which combines a geometric model with a probabilistic model in a principled way. The *geometric model* is a Riemannian manifold with a Riemannian metric, $g_{ij}(\mathbf{x})$, which is interpreted as a measure of *dissimilarity*. The *probabilistic model* consists of a stochastic process with an invariant probability measure which matches the density of the sample input data. The link between the two models is a potential function, $U(\mathbf{x})$, and its gradient, $\nabla U(\mathbf{x})$. I use the gradient to define the dissimilarity metric, which guarantees that my measure of dissimilarity will depend on the probability measure. Roughly speaking, the dissimilarity will be small in a region in which the probability density is high, and vice versa. Finally, I use the dissimilarity metric to define a coordinate system on the embedded Riemannian manifold, which gives us a low-dimensional encoding of our original data.

2.1.1 Probabilistic Model

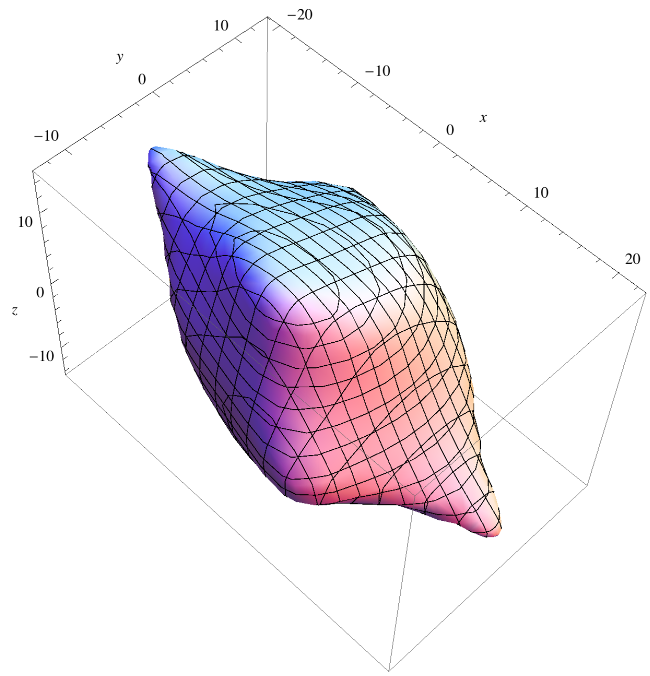

Figure 1: A Curvilinear Gaussian Potential $U(\mathbf{x})$.

The probabilistic model is known in the literature as *Brownian motion with a drift term*. More precisely, it is a *diffusion process* generated by the following differential operator:

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

Brownian motion, by itself, is generated by the differential operator $\frac{1}{2}\Delta$, where Δ is the standard Laplacian expressed in Cartesian coordinates. But Brownian motion "dissipates," that is, it has no invariant probability measure except *zero*. When we add a drift term, which is given here by $\nabla U(\mathbf{x}) \cdot \nabla$, the invariant probability measure is proportional to $e^{U(\mathbf{x})}$. This means that $\nabla U(\mathbf{x})$ is the *gradient of the log of the probability density*, and we can estimate it from sample data.

Figure 1 shows a three-dimensional example in which $U(\mathbf{x})$ is a 6th-degree polynomial and $\nabla U(\mathbf{x})$ is a 5th-degree polynomial. The gradient vector field, $\nabla U(\mathbf{x})$, is shown in Figure 2, restricted to the xy plane at $z = -10$. Notice how the "drift vector" is "transporting probability mass towards the origin" in Figure 2, to counteract the dissipative effects of the Laplacian term in (1), and to maintain an invariant probability measure. This figure also suggests how the "drift vector" can be used to define a *nonlinear coordinate system* in our geometric model.

2.1.2 Geometric Model

To implement the idea of prototype coding, let us define a radial coordinate, ρ, and the directional coordinates $\theta_1, \theta_2, \ldots, \theta_{n-1}$. The radial coordinate will follow the gradient vector, $\nabla U(\mathbf{x})$, and the directional coordinates will be orthogonal to $\nabla U(\mathbf{x})$. But what we really want is a *lower*-dimensional subspace, obtained by projecting our diffusion process onto a $k-1$ dimensional subset of the coordinates

[1] http://bit.ly/1phP8q4
[2] http://bit.ly/1nhOeui

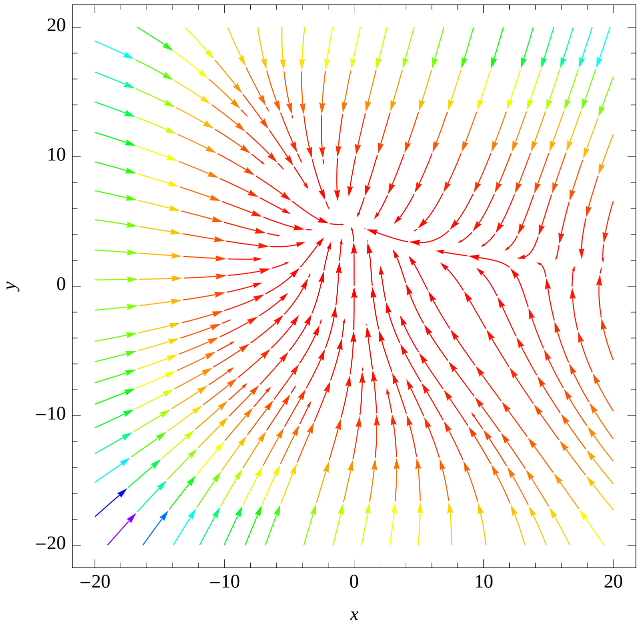

Figure 2: The Gradient Vector Field $\nabla U(\mathbf{x})$.

$\theta_1, \theta_2, \ldots, \theta_{n-1}$. Taken together with the ρ coordinate, we want this projection operation to give us an "optimal" k dimensional subspace.

The mathematical device that we need is a Riemannian metric, $g_{ij}(\mathbf{x})$, which we will use to measure *dissimilarity* on the embedded submanifolds. And 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. We can then take the following steps:

- To find a *principal axis* for the ρ coordinate, we minimize the Riemannian distance, $g_{ij}(\mathbf{x})$, along the drift vector.

- To choose the *principal directions* for the $\theta_1, \ldots, \theta_{k-1}$ coordinates, at a fixed point along the principal axis, we diagonalize the Riemannian matrix, $(\, g_{ij}(\mathbf{x})\,)$, and select the eigenvectors associated with the $k-1$ smallest eigenvalues.

- To compute the *coordinate curves* from any point along the principal axis, we follow the *geodesics* of the Riemannian metric, $g_{ij}(\mathbf{x})$, in each of the $k-1$ principal directions.

Thus, overall, we are minimizing dissimilarity, and maximizing probability. For the details of this procedure, and the precise definition of the Riemannian metric, $g_{ij}(\mathbf{x})$, the reader should consult [34].

Figure 3 shows the optimal two-dimensional subspace for the three-dimensional example in Figures 1 and 2. The coordinates here are labelled ρ and θ_1. Of course, we can often find multiple prototypes in an n-dimensional space, each one with a different k-dimensional subspace. We call these *prototypical clusters*. For an example of two prototypical clusters in a three-dimensional space, the reader should follow the link in footnote 1 and take a look at Figure 12, or follow the link in footnote 2 and take a look at Slide 14.

Figure 3: Geodesic Coordinates for the ρ, θ_1 Surface.

Because of the prominent role of the Riemannian dissimilarity metric in this model, I will often refer to it as a theory of *differential similarity* [36].

2.2 Deep Learning

Let's return to the paper in *NIPS 2011* [43] and look at the authors' first hypothesis, which was omitted in our previous discussion in Section 2.1:

> 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 $p(y|x)$ ... [citation omitted]. 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 ... [citations omitted].
> 2. ...
> 3. ...

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: [7, 17, 42], which initiated the modern field of "deep learning."

For a standard example, let's consider the MNIST dataset of handwritten digits, 0 through 9. Each image consists of 28×28 pixels with values in the range $[0, 255]$. (Such

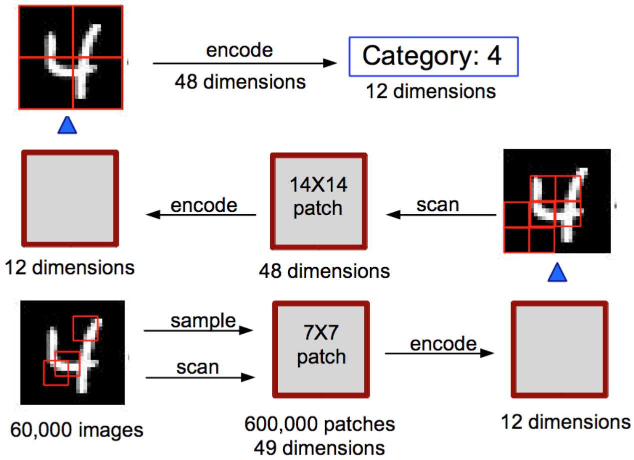

Figure 4: An Architecture for Deep Learning.

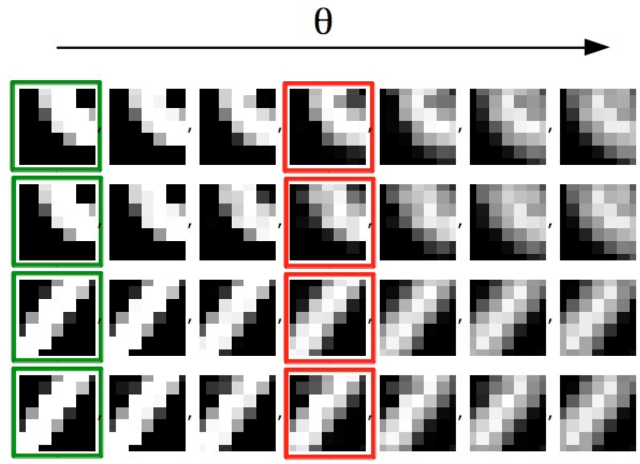

Figure 5: Geodesic Coordinates for Two Prototypes.

images are sometimes called "quasi-binary": The original NIST dataset was binary, but the edges of the digits have been blurred slightly by preprocessing.) The MNIST dataset consists of 60,000 training set images and 10,000 test set images. Historically, it has been used as a benchmark for supervised pattern recognition [23], but we are interested in viewing it as a problem in unsupervised feature learning.

An architecture for deep learning on the MNIST dataset is shown in Figure 4, based on several examples in the recent literature [41, 9]. The process starts in the lower-left corner and follows the arrows to the upper-right corner. The first step is to *scan* and *randomly sample* the images to extract 7×7 "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.

2.2.1 Prototypical 7×7 Patches in the MNIST Dataset

One obvious solution to the problem of dimensionality-reduction is to apply the techniques of "manifold learning" discussed in Section 2.1. The details of this solution will be presented in a forthcoming paper entitled "Differential Similarity in Higher Dimensional Spaces: Theory and Applications" [36].

There are several theoretical results in the paper: First, the theory of differential similarity in [34] can easily be extended to an arbitrary n-dimensional space. Second, the gradient vector, $\nabla U(\mathbf{x})$, can be estimated from sample data, using what is known as the *mean shift algorithm* [15, 8, 10]. Third, the Euler-Lagrange equations for the Riemannian dissimilarity metric can be derived analytically from the estimator for $\nabla U(\mathbf{x})$, and thus the geodesic coordinates for an optimal low-dimensional subspace can be computed in a reasonable amount of time, even when the original Euclidean space has as many as 49 dimensions.

Figure 5 shows some of the empirical results in the paper. We have chosen 35 prototypical clusters to represent the 600,000 patches, using the following criteria: (i) $\nabla U(\mathbf{x})$ should be equal to 0 at the center of each cluster, and (ii) the clusters should partition the space. We have then computed the optimal lower-dimensional subspaces exactly as described in Section 2.1.2. For example, Figure 5 depicts two geodesic coordinate curves for each of two prototypical clusters, corresponding to the geodesic coordinate curves shown in Figure 3. 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° along the coordinate curves; and the patches on the right are points at approximately 180°. Notice that the coordinate curves converge to essentially the same patch at 180°, but they are distinct at 90°. This is one of the properties that we would expect in our particular nonlinear coordinate system.

2.2.2 Product Manifolds with Reduced Dimensionality

Suppose we choose the 12 most informative coordinate curves for each prototype, and use these to *encode* the 7×7 patches. This step is shown in the lower-right corner of Figure 4. We can then assemble four adjacent 7×7 patches into a 2×2 matrix, and *resample* the image using the larger 14×14 patch. If we use the encoded values of the 7×7 patches in the resampling procedure, each 14×14 patch would be represented as a point in a 48-dimensional space. And since there are only 9 distinct scans of our 2×2 matrix across a 28×28 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: (i) construct the *product manifold* from the encoded values of the smaller patches, and then (ii) construct a *submanifold* using the Riemannian dissimilarity metric.

We can now repeat this procedure, as shown in the top line of Figure 4. The submanifolds from the prior step are assembled into a new 2×2 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 4.

A detailed analysis of this system will be presented in a forthcoming paper entitled "Deep Learning with a Riemannian Dissimilarity Metric" [35]. It is still "work in progress" and we do not yet have a full evaluation, but the initial results are positive.

3. A LOGICAL LANGUAGE

Look at the operation in the top line of Figure 4: Four "patches" are combined into an "image" of the digit "four" and the dimensionality of the image is reduced from 48 to 12. The four patches are shown again in Figure 6, arranged as a *logical product*. Using the syntax of my *Language for Legal Discourse* (*LLD*) [28, 31, 33], the image might be represented as follows:

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

In this expression, `?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.

In this section of the paper, we will see how to encode this interpretation of `ImageFOUR` in a logical language based on *category theory*, or what is known as a *categorical logic*. In category theory, we work with *objects* and *morphisms*, which are mappings between objects satisfying three conditions: (i) each morphism is assigned a unique object called its *domain* and a unique object called its *codomain*; (ii) for each pair of morphisms, f and g, if the codomain of f is the same as the domain of g, then there exists a morphism $g \circ f$ from the domain of f to the codomain of g called the *composition* of f and g, and the operation of composition, $\circ$, is associative; (iii) for each object A, there exists an *identity* morphism, ι_A, which is the *unit* of the composition operation, i.e., $f \circ \iota_A = f = \iota_B \circ f$ for all $f : A \to B$. The basic concepts of category theory also include *functors*, which are mappings between categories, and *natural transformations*, which are morphisms in functor categories satisfying certain "naturality" conditions, but we will be making only limited use of these concepts in the present paper. For references on category theory and categorical logic, see [22, 21, 4, 1, 19].

We are primarily interested in the category **Man**, in which the objects are *differential manifolds* and the morphisms are *smooth mappings*. By a "smooth" mapping we mean either a C^k *mapping*, which is a k-fold continuously differentiable function, or a C^∞ *mapping*, which is a C^k *mapping* for all $k \geq 0$. For comparison, we will also consider the categories **Set** and **Top**, which are defined as follows:

	objects	*morphisms*
Set	abstract sets	arbitrary mappings
Top	topological spaces	continuous mappings
Man	differential manifolds	smooth mappings

Each of these categories has an *initial object*, $\mathbf{0}$, defined as the object for which there exists exactly one morphism $\mathbf{0} \to X$ for every object X in the category, and a *terminal object*, $\mathbf{1}$, defined as the object for which there exists exactly one morphism $X \to \mathbf{1}$ for every object X in the category. In **Man**, the initial object is the empty manifold, $\{\}$, and the terminal object is the 0-dimensional manifold consisting of a single point, $\{0\}$. It is evident that there exists a mapping $Y \to \mathbf{0}$ if and only if Y is also the empty manifold, and thus we interpret $\mathbf{0}$ as **false** in our logic. By contrast, the

Figure 6: The Product of Four Patches

mappings $\mathbf{1} \to Y$ pick out the "points" of the manifold Y, implying that Y is nonempty, and we therefore interpret $\mathbf{1}$ as **true**.

We now turn to the definition of a categorical product, and introduce the important idea of a *Universal Mapping Property*. Instead of defining a mathematical object by an explicit construction from simpler objects (e.g., sets, tuples, sets of tuples, ...), we characterize it *up to isomorphism* by a universal property of its maps to other objects. Thus, for two objects A and B in a given category, we define the *abstract binary product* as follows:

Definition 1. **Product.** *The* binary product of A and B, *written* $\langle P, \{\pi_A, \pi_B\}\rangle$, *is an object* P *together with two projection morphisms,* $\pi_A : P \to A$ *and* $\pi_B : P \to B$*, such that, for any object* X *and any morphisms* $f : X \to A$ *and* $g : X \to B$*, there exists a unique morphism* $h : X \to P$ *for which the following diagram commutes:*

$$\begin{tikzcd} & & A \\ X \arrow[urr, bend left, "f"] \arrow[drr, bend right, "g"'] \arrow[r, dashed, "h"] & P \arrow[ur, "\pi_A"] \arrow[dr, "\pi_B"'] & \\ & & B \end{tikzcd}$$

It is straightforward to verify that any two abstract binary products, $\langle P, \{\pi_A, \pi_B\}\rangle$ and $\langle P', \{\pi'_A, \pi'_B\}\rangle$, satisfying this definition would be isomorphic, using the definition of isomorphism appropriate to the given category. Moreover, in the category **Set**, the abstract product would be isomorphic to the ordinary Cartesian product; in the category **Top**, it would be isomorphic to the topological product, i.e., to the Cartesian product endowed with the product topology; and in the category **Man**, it would be isomorphic to the usual C^k or C^∞ product manifold. Note also that the abstract product in **Man** (or **Top** or **Set**) is associative, and that the empty product in **Man** using Definition 1 would reduce to $\mathbf{1} = \{0\}$. We can thus define a product with any number of factors, using a simple recursive construction.

The concept of a *subobject* in these categories can also be characterized by a universal mapping property:

Definition 2. **Subobject.** *The* subobject $\langle S, i\rangle$ of C *is an object* S *together with an inclusion morphism* $i : S \to C$ *such that, for any object* X *and any morphism* $f : X \to C$ *with* $f(X) \subseteq S$*, there exists a unique morphism* $h : X \to S$ *for which the following diagram commutes:*

$$\begin{tikzcd} & C \\ X \arrow[ur, bend left, "f"] \arrow[r, dashed, "h"] & S \arrow[u, hook, "i"'] \end{tikzcd}$$

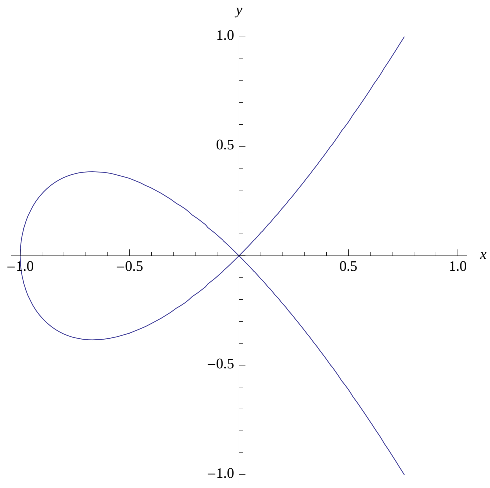

Figure 7: A Subspace but not a Submanifold.

In the category **Set**, of course, an abstract subobject is an ordinary subset; in **Top**, it is a subset endowed with the subspace topology; and in **Man**, it is a submanifold. See [20], Chapter II, Section 2. But there is now an important difference between **Man** and the other two categories. In **Top**, any subset of C can be endowed with the subspace topology, and thus, in both **Set** and **Top**, the subobjects of C correspond to the *powerset object* of C. But this is not the case in **Man**, since not every subset is a submanifold. A standard example is shown in Figure 7. Let

$$T \;=\; \{\, \langle x, y\rangle \in \mathbb{R}\times\mathbb{R} \mid y^2 = x^2(x+1) \,\}$$

be a topological space with the subspace topology inherited from $\mathbb{R}^2$, and assume that there exists a manifold M with the same underlying set. Then there would be a C^k *chart* defining a coordinate system for some neighborhood of $\langle 0, 0\rangle \in M$, and this can be shown to lead to a contradiction. (Because of space limitations, the proof will be left as an exercise for the reader.) Thus M cannot be a submanifold of $\mathbb{R}^2$.

Now that we have defined the product and the subobject for the category of differential manifolds, **Man**, it should be clear how to interpret the `ImageFOUR` example at the beginning of this section. But, for the sake of variety, let's switch to a legal example suggested by [31] and [33], and let's simultaneously introduce the notion of a *sequent calculus* for our logical language. Suppose we write:

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

What does this mean? Clearly, we are interpreting `Actor` and `Corporation` as differential manifolds, and we want to interpret `macomber` (≡ Myrtle H. Macomber) and `so` (≡ Standard Oil) as points on these manifolds. To represent "points" in category theory, we use morphisms from the terminal object, $\mathbf{1}$, and thus `(Actor macomber)` would be represented by the morphism `macomber` $: \mathbf{1} \to$ `Actor`, and likewise for `(Corporation so)`. What is `Control`? As in the `ImageFOUR` example, `Control` would be represented by a submanifold of the product manifold of `Actor` and `Corporation`, which means that `?r` would refer to the image of the pair $\langle$`macomber`, `so`$\rangle$ under the subobject mapping given by Definition 2, if such a value for `?r` exists. How should we interpret the overall sequent? In categorical logic, a sequent $X \vdash Y$ is interpreted as a morphism $X \to Y$. Thus, to assert the truth of the preceding sequent for `Control`, we would simply assert that there exists a morphism $\mathbf{1} \to$ `Control` which picks out the point referred to by `?r`. The dual situation would be exemplified by the following sequent:

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

Since the morphism `Control` $\to \mathbf{0}$ exists if and only if `Control` is the empty manifold, this sequent asserts that there is no value for `?r` that satisfies the given constraints.

For an example of a more general sequent, consider the following:

`(Q ?q (Actor ?a) (Corporation ?c))`
$\quad\vdash_{\mathcal{C}}$ `(Control ?r (Actor ?a) (Corporation ?c))`

We are now using variables `?a` and `?c` to refer to points on the manifolds `Actor` and `Corporation`, respectively, and we are required by the rules of our sequent calculus to enter these variables into the *context*, $\mathcal{C}$, associated with the sequent turnstile $\vdash$. Thus $\mathcal{C} = \{$`?a`, `?c`$\}$ in this example. (We will see later how these contexts can change during the course of a proof.) The left hand side of this sequent would be represented by a submanifold, `Q`, which might be different, in general, from the submanifold `Control`. But again, the sequent `Q` $\vdash$ `Control` would be interpreted as a morphism `Q` $\to$ `Control`. Thus, to assert that this sequent is true is equivalent to asserting that the morphism `Q` $\to$ `Control` maps the point referred to by `?q` to the point referred to by `?r`, and that the manifold `Control` is nonempty whenever `Q` is nonempty.

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

$$\frac{\Gamma,\, \phi \vdash_{\mathcal{C}} \theta \qquad\qquad \Gamma,\, \theta \vdash_{\mathcal{C}} \psi}{\Gamma,\, \phi \vdash_{\mathcal{C}} \psi}$$

The rule for conjunction is just a rewriting of Definition 1 for the categorical product:

$$\frac{\Gamma \vdash_{\mathcal{C}} \psi_1 \qquad\qquad \Gamma \vdash_{\mathcal{C}} \psi_2}{\Gamma \vdash_{\mathcal{C}} \psi_1 \wedge \psi_2}$$

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 add "horn axioms" in the following form:

$$Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \;\vdash_{\mathcal{C}}\; P$$

and we have the basics of *horn clause logic programming* [24]. Note that these horn axioms are universally quantified at the top level, implicitly, because of the occurrence of free variables in the context $\mathcal{C}$.

To add explicit quantifiers to our language, we will follow the standard approach in categorical logic, which is summarized by the slogan: "Quantifiers as Adjoints". To explain this idea intuitively, let's first work through an example in the category **Set**.

Let $A \times B \times C$ be a product in **Set**, i.e., the ordinary Cartesian product, and let $\pi_{B\times C}$ be the projection onto the last two factors. Construct the *inverse image* function on the powerset of $B\times C$,

$$\Pi^{-1}_{B\times C} : \mathcal{P}(B\times C) \to \mathcal{P}(A\times B\times C)$$

by defining, for each $T \subseteq B\times C$,

$$\Pi^{-1}_{B\times C}(T) = \{\, x \in A\times B\times C \;\mid\; \pi_{B\times C}(x) \in T\,\}.$$

We now want to construct a function that operates in the opposite direction,

$$\exists_A : \mathcal{P}(A\times B\times C) \to \mathcal{P}(B\times C)$$

and which, for all $S \subseteq A\times B\times C$ and $T \subseteq B\times C$, satisfies the following condition:

$$\exists_A(S) \;\subseteq\; T \iff S \;\subseteq\; \Pi^{-1}_{B\times C}(T). \tag{2}$$

It is straightforward to verify that the desired function can be written explicitly as

$$\exists_A(S) = \{y \in B\times C \mid \exists x \in A\times B\times C : \pi_{B\times C}(x) = y \wedge x \in S\}$$

We can therefore interpret $\exists_A$ as the representation of *existential quantification* in **Set**.

We will now modify the example slightly. Let $A\times B\times C$ be a product in **Set**, but this time let $\pi_{A\times B}$ be the projection onto the first two factors, and construct the inverse image function $\Pi^{-1}_{A\times B}$ on the powerset of $A\times B$. By analogy to the previous example, we want to construct a function that operates in the opposite direction,

$$\forall_C : \mathcal{P}(A\times B\times C) \to \mathcal{P}(A\times B)$$

and which, for all $S \subseteq A\times B\times C$ and $T \subseteq A\times B$, satisfies the following condition:

$$\Pi^{-1}_{A\times B}(T) \;\subseteq\; S \iff T \;\subseteq\; \forall_C(S). \tag{3}$$

It is straightforward to verify that the desired function can be written explicitly as

$$\forall_C(S) = \{y \in A\times B \mid \forall x \in A\times B\times C : \pi_{A\times B}(x) = y \;\Rightarrow\; x \in S\}$$

We can therefore interpret $\forall_C$ as the representation of *universal quantification* in **Set**.

Let's now transpose these concepts to the category of differential manifolds. As we have seen, there are no powerset objects in **Man**, but we can still construct the set of all submanifolds of X, denoting this by $\mathcal{M}(X)$, and we can define a partial order, $S \le Y$, which holds whenever S is a submanifold of Y. Now let $A\times B\times C$ be a product in **Man**. We can always construct the inverse image function, $\Pi^{-1}_{B\times C}$, and thus what we want for the existential quantifier is the function $\exists_A$ in the following diagram:

$$\begin{tikzcd}
\mathcal{M}(A\times B\times C) \arrow[rr, bend left, "\exists_A"] & & \mathcal{M}(B\times C) \arrow[ll, "\Pi^{-1}_{B\times C}"]
\end{tikzcd}$$

for which the following condition is true:

$$\exists_A(S) \;\le\; T \iff S \;\le\; \Pi^{-1}_{B\times C}(T). \tag{4}$$

Similarly, we can always construct the inverse image function, $\Pi^{-1}_{A\times B}$, and thus what we want for the universal quantifier is the function $\forall_C$ in the following diagram:

$$\begin{tikzcd}
\mathcal{M}(A\times B\times C) \arrow[rr, bend right, "\forall_C"'] & & \mathcal{M}(A\times B) \arrow[ll, "\Pi^{-1}_{A\times B}"']
\end{tikzcd}$$

for which the following condition is true:

$$\Pi^{-1}_{A\times B}(T) \;\le\; S \iff T \;\le\; \forall_C(S). \tag{5}$$

In categorical terms, (2) and (4) express the fact that $\exists_A$ is the *left adjoint* to the functor $\Pi^{-1}_{B\times C}$ and (3) and (5) express the fact that $\forall_C$ is the *right adjoint* to the functor $\Pi^{-1}_{A\times B}$. In each case, the categories to which these functors are applied consist of the subobjects of $A\times B\times C$ with either $\subseteq$ or $\le$ as the morphisms, i.e., they are *partially ordered sets*. Observe also that both the existential and the universal quantifiers are identified with the object that is *omitted* in the projections: A is omitted from the projection $\pi_{B\times C}$ and C is omitted from the projection $\pi_{A\times B}$. For more information on adjoint functors and the categorical interpretation of "Quantifiers as Adjoints," see [4], Chapter 9.

Let's see how this formalism would work in our legal example of `Control`. Suppose we wanted to represent the proposition: "For every `Corporation ?c` there exists an `Actor ?a` such that `?a Controls ?c`." In the category **Man**, `Control` is a submanifold of the binary product `Actor`$\times$`Corporation`. Alternatively, we could work with the product $A\times B\times C$ from our previous example, identifying $A =$ `Actor` and $C =$ `Corporation`, and taking B to be the empty product $\mathbf{1} = \{0\}$. Let's take the second approach, and see what happens when we apply the quantifiers in sequence. First, we apply the function $\exists_A$ to `Control`, producing a subobject of $B\times C$. Second, we apply the function $\forall_C$ to $\exists_A$(`Control`), producing a subobject of B. But B has only two subobjects, $\{0\}$ and $\{\}$. If the final result is $\mathbf{1} = \{0\}$, then our proposition is true. Otherwise, if the final result is $\mathbf{0} = \{\}$, then our proposition is false.

The proof rules for existential and universal quantification can be designed to mimic these categorical operations very closely. Here is a bidirectional rule for the existential quantifier:

$$\frac{\Gamma,\ \phi[\texttt{Actor ?x}] \ \vdash_{\mathcal{C}\cup\{\texttt{?x}\}}\ \psi}{\Gamma,\ \exists\,\texttt{Actor ?x} : \phi\ \vdash_{\mathcal{C}}\ \psi}$$

Reading this rule from bottom to top, which makes it a $\exists$-elimination rule, notice that the context is expanded from $\mathcal{C}$ to $\mathcal{C}\cup\{\texttt{?x}\}$ and then ψ is derived from $\phi[\texttt{Actor ?x}]$. Since the context below the line is just $\mathcal{C}$, however, ψ cannot contain `?x` as a free variable. Here is a bidirectional rule for the universal quantifier:

$$\frac{\Gamma,\ \phi\ \vdash_{\mathcal{C}\cup\{\texttt{?x}\}}\ \psi[\texttt{Corporation ?x}]}{\Gamma,\ \phi\ \vdash_{\mathcal{C}}\ \forall\,\texttt{Corporation ?x} : \psi}$$

Reading this rule from bottom to top, which makes it a $\forall$-elimination rule, notice that the context is expanded from $\mathcal{C}$ to $\mathcal{C}\cup\{\texttt{?x}\}$ and then $\psi[\texttt{Corporation ?x}]$ is derived from ϕ. Again, since the context below the line is just $\mathcal{C}$, ϕ cannot contain `?x` as a free variable. These are the usual rules for the elimination of quantifiers, of course, but notice how the treatment of free variables in these rules corresponds to the adjoint functor conditions in (4) and (5).

Let's now consider the implication connective, $\Leftarrow$ or $\Rightarrow$, which raises a subtle question, as we will see. Categorical logic interprets the implication $X \Rightarrow Y$ as an object that "internalizes" a morphism from X to Y, in the following way:

Definition 3. **Exponential.** *The* exponential of C by B, *written* $\langle C^B, \mathbf{eval}\rangle$, *is an object* C^B *together with an evaluation morphism* $\mathbf{eval} : C^B \times B \to C$ *such that, for any object* A *and any morphism* $f : A \times B \to C$, *there exists a unique morphism* $\tilde{f} : A \to C^B$ *for which the following diagram commutes:*

$$\begin{tikzcd} C^B \times B \arrow[r, "\mathbf{eval}"] & C \\ A \times B \arrow[u, dashed, "\tilde{f} \times \iota_B"] \arrow[ur, bend right, "f"'] & \end{tikzcd}$$

To see how the universal mapping property in this definition leads to an isomorphism, choose any $g : A \to C^B$ and define

$$\bar{g} \;=\; \mathbf{eval} \circ (g \times \iota_B) \;:\; A \times B \to C.$$

Then, by the uniqueness of the induced morphism in Definition 3, $\tilde{\bar{g}} = g$ and $\bar{\tilde{f}} = f$ for any $f : A \times B \to C$. In other words, the operation that transforms $f : A \times B \to C$ into $\tilde{f}$ is inverse to the operation that transforms $g : A \to C^B$ into $\bar{g}$. Denoting the set of all morphisms from X to Y by $\mathrm{Hom}(X, Y)$, this establishes the bijection:

$$\mathrm{Hom}(A \times B, C) \;\cong\; \mathrm{Hom}(A, C^B),$$

which translates directly into a bidirectional proof rule for implication:

$$\frac{\Gamma,\, \phi \;\vdash_{\mathcal{C}}\; \psi}{\Gamma \;\vdash_{\mathcal{C}}\; \phi \Rightarrow \psi}$$

Reading this rule from bottom to top, it says that $\phi \Rightarrow \psi$ can be derived from Γ if ψ can be derived from Γ with ϕ as an additional assumption.

The question, though, is whether exponential objects actually exist in any particular category. If C^B exists for every object B and C, we say that the category is *cartesian closed*. The category **Set** is cartesian closed, since every set of set-theoretic mappings is itself a set! The category **Top** is not cartesian closed, but there is a cartesian closed subcategory of **Top**, called **kTop**, which consists of the *compactly generated Hausdorff* spaces [45]. Likewise, the category **Man** is not cartesian closed. For our purposes, however, there is a simple solution to this problem. Consider an implication in the following form:

$$Q \;\Leftarrow\; R_1 \wedge R_2 \wedge \ldots \wedge R_k$$

We can require that the Q and the R_i be interpreted always as *compact finite-dimensional* manifolds, which is consistent with our discussion of prototypical clusters in Section 2.1.2. Then the morphism that represents this implication would be a smooth mapping between finite-dimensional manifolds, and we can always construct an *infinite-dimensional* manifold on the space of such smooth mappings. See, e.g., [20]. This means that $R_1 \times R_2 \times \ldots \times R_k$ is an *exponentiable object* in the category **Man**, and we can therefore interpret the preceding implication as $Q^{R_1 \times R_2 \times \ldots \times R_k}$ using Definition 3. Of course, we can only repeat this process once, since the antecedent in our implication must be represented by a finite-dimensional manifold, but we can live with such a restriction. The logical language that results is the language of *simple embedded implications*, in which the "horn axioms" discussed previously are augmented by "embedded implication axioms" in the following form:

$$\ldots \wedge (Q \;\Leftarrow\; R_1 \wedge R_2 \wedge \ldots \wedge R_k) \wedge \ldots \;\vdash\; P$$

Fortuitously, this is a logic programming language which has been studied extensively in the literature, by myself [26, 27, 29] and others [40].

With this last step, our language includes the logical connectives: $\wedge$, $\exists$, $\forall$, $\Leftarrow$. We could add disjunction to this list, but an alternative approach is to do "indefinite reasoning" with "definite rules" as advocated in [37]. We can also add negation by defining $\neg P$ as $P \Rightarrow \mathbf{0}$, which is the standard approach in intuitionistic logic. Finally, we can add "negation-as-failure" using the techniques described in [39] and [29]. We have thus reconstructed, with a semantics grounded in the category of differential manifolds, **Man**, the full *intuitionistic logic programming* language [26, 27] which has always been at the core of my *Language for Legal Discourse* (*LLD*) [28].

Technical Note: Most mathematicians are unhappy with the limitations we have noted in the category **Man**, and they have attempted to find an extended category in which these limitations disappear. A recent example is [5]. Baez and Hoffnung study a category of "smooth spaces" which has the properties: (i) that every subset of a smooth space is itself a smooth space, and thus a *weak subobject classifier* exists in the category; and (ii) that the *parametrized mapping space* of a pair of smooth spaces is also a smooth space, and thus the category is *locally cartesian closed*. But this category has some strange objects in it, such as the Cantor set, which does not match our intuitions about what a smooth space should be. And, for our purposes, this is exactly the wrong approach. If we are looking for a knowledge representation language that is *learnable*, we want it to be as restrictive as possible. Thus the limitations in the category **Man** constitute a feature, not a bug!

4. 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 this set. The language should be formal: it should have a compositional syntax, a precise semantics and a well-defined inference mechanism. ... [28]

This research programme has now been incorporated into a more general research programme on *legal ontologies* [18], in which logic plays a prominent role. In this new framework, however, the theory of differential similarity raises several questions about the formalization of a legal ontology: How do these logics change when we switch from a language based on **Set** (which is classical) or **Top** (which is intuitionistic) to a language based on **Man**, which is the category of differential manifolds?

For example, one important feature of *LLD* is the distinction between *count terms* and *mass terms*. To represent the definition of "control" that appears in [31] and [33], we actually need something like the following sequent:

$$\begin{aligned}&\texttt{(Own ?o (Actor ?a) (Stock76 ?s1))} \wedge \\ &\texttt{(Issue ?i (Corporation ?c) (Stock76 ?s2))} \\ &\quad\vdash_{\mathcal{C}} \texttt{(Control ?r (Actor ?a) (Corporation ?c))}\end{aligned}$$

Here, `?s1` and `?s2` refer to submanifolds of a manifold called `Stock76`, which represents a *mass* and can have a *measure*. Syntactically, this is an expression in second-order logic, and we should be able to reason about it using the techniques in [40]. Does it make a difference, semantically, that `Stock76` is a manifold with a measure, rather than an abstract set or a topological space?

One of the most important features of *LLD* is the representation of *events* and *actions* [38], and the various modalities over actions, such as *permissions* and *obligations* [25]. The most thorough treatment of the deontic modalities in my previous work can be found in [30], but the representation of actions in that paper leaves much to be desired. We can do better, I think, if we take differential manifolds seriously, and represent an action (initially) by the Lie group of *affine isometries* in $\mathbb{R}^3$, also known as *rigid body motions*. See, e.g., [16], Chapter 18. We can then apply the theory of differential similarity to the manifold of physical actions, and generalize from there to a manifold of abstract actions. Does this formalization lead to a distinctly different theory of the deontic modalities?

The best current model of the epistemic modalities, such as *knowledge* and *belief*, can be found in the literature on *justification logics* [13, 3]. Based on the work of Artemov on the *Logic of Proofs (LP)* [2], 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, and it can be extracted from a provable formula $\Box X$, the traditional modal proposition in a Hintikka-style logic of knowledge, by what is known as a "Realization Theorem." This is currently an active area of research, and there are justification logics that correspond to the modal systems **K**, **T**, **K4**, **S4**, **K45**, **S5** and others. There is also now a first-order version of the logic of proofs [14]. We have introduced a new logical language in Section 3, of course, based on the category of differential manifolds. Does this lead to a new variant in the family of justification logics?

The answer would seem to depend on what a proof looks like in our system, which is a question addressed in the following section.

5. TOWARD A THEORY OF COHERENCE

The theory of differential similarity is a hybrid drawn from three areas of mathematics: Probability, Geometry, Logic. The probabilistic model and the geometric model were combined in [34] to produce a theory of *clustering* and *coding*, and the (forthcoming) papers [36] and [35] show how a hierarchy of *prototypical clusters* can support a form of unsupervised learning. In Section 3 of the present paper, we added the logical component: The prototypical clusters are actually *differential manifolds*, and they can be used to provide the semantic interpretation of the atomic formulas in a categorical logic. This means that a proof in our logic is more than just a chain of inference rules. It is a *composition* of *morphisms* in the category **Man**, i.e., it is a *smooth mapping* of differential manifolds, and it can carry the prototypical structure of a concept up through the ontological ladder of a language such as *LLD*.

This is the key to a theory of "coherence" in legal argument. The logic is constrained by the geometry, as we have seen in Section 3. In our model, concepts are not arbitrary sets, but smooth manifolds linked by smooth mappings. The geometric model, in turn, is constrained by the probabilistic model, since the Riemannian dissimilarity metric depends on the probability measure. And, finally, the probability measure is constrained by the distribution of sample data in the actual world. It is the existence of these mutual constraints, I suggest, which makes theory construction possible.

6. REFERENCES

[1] S. Abramsky and N. Tzevelekos. Introduction to categories and categorical logic. *Lecture Notes*, arXiv:1102.1313 [math.CT], 2011.

[2] S. N. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, 2001.

[3] S. N. Artemov. The logic of justification. *The Review of Symbolic Logic*, 1(4):477–513, 2008.

[4] S. Awodey. *Category Theory*. Oxford Logic Guides, Volume 49. Oxford University Press, 2006.

[5] J. C. Baez and A. E. Hoffnung. Convenient categories of smooth spaces. *Transactions of the American Mathematical Society*, 363(11):5789–5825, 2011.

[6] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[7] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, volume 19, pages 153–160, 2006.

[8] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[9] A. Coates. *Demystifying Unsupervised Feature Learning*. PhD thesis, Stanford University, 2012.

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[11] R. Dworkin. Hard cases. *Harvard Law Review*, 88:1057–1109, 1975.

[12] R. Dworkin. *Law's Empire*. Harvard University Press, 1986.

[13] M. Fitting. Reasoning with justifications. In D. Makinson, J. Malinowski, and H. Wansing, editors, *Towards Mathematical Philosophy, Papers from the Studia Logica conference Trends in Logic IV*, pages 107–123. Springer, 2009.

[14] M. Fitting. Possible world semantics for first-order logic of proofs. *Annals of Pure and Applied Logic*, 165(1):225–240, 2014.

[15] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in

pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[16] J. Gallier. *Geometric methods and applications: For computer science and engineering. 2nd ed.* Texts in Applied Mathematics 38. Springer, 2011.

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[18] R. Hoekstra, J. Breuker, M. D. Bello, and A. Boer. The LKIF core ontology of basic legal concepts. In P. Casanovas, M. A. Biasiotti, E. Francesconi, and M. T. Sagri, editors, *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, 2007.

[19] B. Jacobs. *Categorical logic and type theory*. Studies in logic and the foundations of mathematics. Elsevier, Amsterdam, 2001.

[20] S. Lang. *Differential and Riemannian Manifolds*. Graduate Texts in Mathematics, 160. Springer-Verlag, 1995.

[21] F. Lawvere and R. Rosebrugh. *Sets for Mathematics*. Cambridge University Press, 2003.

[22] F. W. Lawvere and S. H. Schanuel. *Conceptual Mathematics: A First Introduction to Categories*. Cambridge University Press, 2nd edition, 2009.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1984.

[25] L. T. McCarty. Permissions and obligations. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 287–294, 1983.

[26] L. T. McCarty. Clausal intuitionistic logic. I. Fixed-point semantics. *Journal of Logic Programming*, 5(1):1–31, 1988.

[27] L. T. McCarty. Clausal intuitionistic logic. II. Tableau proof procedures. *Journal of Logic Programming*, 5(2):93–132, 1988.

[28] L. T. McCarty. A language for legal discourse. I. Basic features. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, pages 180–189. ACM Press, 1989.

[29] L. T. McCarty. Circumscribing embedded implications (without stratifications). *Journal of Logic Programming*, 17:323–364, 1993.

[30] L. T. McCarty. Modalities over actions, I. Model theory. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR94)*, pages 437–448. Morgan Kaufmann, 1994.

[31] L. T. McCarty. An implementation of Eisner v. Macomber. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 276–286. ACM Press, 1995.

[32] L. T. McCarty. Some arguments about legal arguments. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 215–224, Melbourne, 1997.

[33] L. T. McCarty. Ownership: A case study in the representation of legal concepts. *Artificial Intelligence and Law*, 10(1-3):135–161, 2002.

[34] L. T. McCarty. Clustering, coding, and the concept of similarity. *Preprint*, arXiv:1401.2411 [cs.LG], 2014.

[35] L. T. McCarty. Deep learning with a Riemannian dissimilarity metric. Forthcoming, 2015.

[36] L. T. McCarty. Differential similarity in higher dimensional spaces: Theory and applications. Forthcoming, 2015.

[37] L. T. McCarty and R. van der Meyden. Indefinite reasoning with definite rules. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 890–896, 1991.

[38] L. T. McCarty and R. van der Meyden. Reasoning about indefinite actions. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR92)*, pages 59–70. Morgan Kaufmann, 1992.

[39] L. T. McCarty and R. van der Meyden. An intuitionistic interpretation of finite and infinite failure. In L. Pereira and A. Nerode, editors, *Logic Programming and NonMonotonic Reasoning: Proceedings of the Second International Workshop*, pages 417–436. MIT Press, 1993.

[40] D. Miller and G. Nadathur. *Programming with Higher-Order Logic*. Cambridge University Press, 2012.

[41] M. Ranzato. *Unsupervised Learning of Feature Hierarchies*. PhD thesis, New York University, 2009.

[42] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*, volume 19, pages 1137–1144, 2006.

[43] S. Rifai, Y. Dauphin, P. Vincent, Y. Bengio, and X. Muller. The manifold tangent classifier. In *Advances in Neural Information Processing Systems*, volume 24, pages 2294–2302, 2011.

[44] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[45] N. E. Steenrod. A convenient category of topological spaces. *The Michigan Mathematical Journal*, 14(2):133–152, 1967.

[46] J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.