Journal of Mathematical Psychology 70 (2016) 21–34

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

Similarity, kernels, and the fundamental constraints on cognition

Reza Shahbazi [a], Rajeev Raizada [b,1], Shimon Edelman [c,*,1]

[a] Weill Cornell Medicine, Cornell University, United States
[b] Department of Brain and Cognitive Sciences, University of Rochester, United States
[c] Department of Psychology, Cornell University, United States

HIGHLIGHTS

- Cognitive necessities.
- Kernels.
- Similarity measurement.
- Tuning curves.

ARTICLE INFO

Article history:
Received 8 March 2015
Received in revised form
1 November 2015

Keywords:
Kernel
RKHS
Similarity
Dimensionality
Complexity
Linear separability
Nonlinear

ABSTRACT

Kernel-based methods, and in particular the so-called kernel trick, which is used in statistical learning theory as a means of avoiding expensive high-dimensional computations, have broad and constructive implications for the cognitive and brain sciences. An equivalent and complementary view of kernels as a measure of similarity highlights their effectiveness in low-dimensional and low-complexity learning and generalization — tasks that are indispensable in cognitive information processing. In this survey, we seek (i) to highlight some parallels between kernels in machine learning on the one hand and similarity in psychology and neuroscience on the other hand, (ii) to sketch out new research directions arising from these parallels, and (iii) to clarify some aspects of the way kernels are presented and discussed in the literature that may have affected their perceived relevance to cognition. In particular, we aim to resolve the tension between the view of kernels as a method of raising the dimensionality, and the various requirements of reducing dimensionality for cognitive purposes. We identify four fundamental constraints that apply to any cognitive system that is charged with learning from the statistics of its world, and argue that kernel-like neural computation is particularly suited to serving such learning and decision making needs, while simultaneously satisfying these constraints.

1. Motivation and plan

The concept of similarity is widely used in psychology. Historically, in a philosophical tradition dating at least back to Aristotle, it has served as a highly intuitive, unifying slogan for a variety of phenomena related to categorization. Here's how Hume put it in the *Enquiry* (1748):

> ALL our reasonings concerning matter of fact are founded on a species of Analogy, which leads us to expect from any cause the same events, which we have observed to result from similar causes. Where the causes are entirely similar, the analogy is perfect, and the inference, drawn from it, is regarded as certain and conclusive. […] Where the objects have not so exact a similarity, the analogy is less perfect, and the inference is less conclusive; though still it has some force, in proportion to the degree of similarity and resemblance.

In the past century, psychologists have turned similarity into a powerful theoretical tool, most importantly by honing the ways in which similarity can be grounded in multidimensional topological or metric representation spaces (see Osgood, 1949 for an early example) or in situations where a set-theoretic approach may seem preferable (Tversky, 1977).

Sometimes criticized as too loose to be really explanatory (e.g., Goodman, 1972), the concept of similarity has eventually been given a mathematical formulation, including a derivation from first principles of the fundamental relationship between similarity and

* Corresponding author.
E-mail address: edelman@cornell.edu (S. Edelman).

[1] This work was supported in part by US NSF grant # 1228261 to R. Raizada and S. Edelman.

http://dx.doi.org/10.1016/j.jmp.2015.11.004
0022-2496/

generalization, and its empirical validation (Shepard, 1987). The mathematical developments, in particular, have solidified similarity's status as a theoretical-explanatory construct in cognitive science (Ashby & Perrin, 1988; Edelman, 1998; Goldstone, 1994; Medin, Goldstone, & Gentner, 1993; Tenenbaum & Griffiths, 2001; for a recent review, see Edelman & Shahbazi, 2012).

In the present paper, we explore the parallels between the psychological construct of similarity and its recent mathematical treatment in the neighboring discipline of machine learning, where a family of classification and regression methods has emerged that is based on the concept of a kernel (Schölkopf & Smola, 2002). Insofar as kernels (described formally in a later section) involve the estimation of distances between points or functions (Jäkel, Schölkopf, & Wichmann, 2008, 2009), they are related to similarity. At the same time, there seems to be a deep rift between the two.

On the one hand, similarity-based learning and generalization has long been thought to require low-dimensional representations, so as to avoid the so-called "curse of dimensionality" (Bellman, 1961; Edelman & Intrator, 1997, 2002), as well as to promote the economy of information storage and transmission (Jolliffe, 1986; Roweis & Saul, 2000). Moreover, as no two measurements of the state of the environment are likely to be identical, some abstraction is necessary before learning becomes possible, which calls for information-preserving dimensionality reduction (Edelman, 1998, 1999). On the other hand, the best-known kernel methods, based on the Support Vector Machine idea (Cortes & Vapnik, 1995; Vapnik, 1999), involve a massive increase in the dimensionality of the representation prior to solving the task at hand.

We attempt to span this rift by seeking a common denominator for some key ideas – and, importantly, their mathematical treatment – behind similarity and kernels. In service of this goal, we first identify, in Section 2, four fundamental constraints on cognition, having to do with (i) measurement, (ii) learnability, (iii) categorization, and (iv) generalization. In Section 3, we then show that while on an abstract-functional or task level these constraints appeal to the concept of similarity, on an algorithmic computational level they call for the use of kernels. Section 4 revisits some standard notions from the similarity literature in light of this observation. In Section 5, we illustrate the proposed synthesis by pairing the methods that it encompasses with a range of cognitive tasks and suggest some ways in which these methods can be used to further our understanding of computation in the brain. Finally, Section 6 offers a summary and some concluding remarks.

2. Fundamental constraints on cognition

2.1. A fundamental constraint on measurement

Perception in any biological or artificial system begins with some measurements performed over the raw signal (Edelman, 2008, ch.5). In mammalian vision, for instance, the very first measurement stage corresponds to the retinal photoreceptors transducing the image formed by the eye's optics into an array of neural activities. The resulting signal is extensively processed by the retinal circuitry before being sent on to the rest of the brain through the optic nerve.

Effectively, a processing unit at any stage in the sensory pathway and beyond "sees" the world through some measurement function $\phi(\cdot)$. Importantly, the measurement process is, at least in the initial stages of development, *uncalibrated*, in the sense that the precise form of the measurement function is not known – that is, not explicitly available – throughout the system. For example, the actual, detailed weight, timing profiles, and noise properties of the receptive field of a sensory neuron are implicitly "known" to the neuron itself (insofar as these parameters determine its response to various types of stimuli), but not to any other units in the system. Indeed, for the usual developmental reasons, those parameters vary from one neuron to the next in ways that are underspecified by the genetic code shared by all neurons in an organism.

Even if the system learns to cope with this predicament (as suggested by some recent findings; Pagan, Urban, Wohl, & Rust, 2013), such learning can only be fully effective if driven by calibrated stimuli, which are by definition not available in natural settings. Moreover, a system that relies on learning, be it as part of its development or as part of its subsequent functioning, it must either (i) simultaneously learn the structure of the data and its own parameters, or (ii) learn the former while being insensitive to the latter.

These considerations imply the following fundamental challenge:

Measurement Any system that involves perceptual measurement is confronted with unknowns that it must learn to tolerate or factor out of the computations that support the various tasks at hand, such as learning and categorization (see Tables 4 and 5).

To the best of our knowledge, this is the first statement of the measurement constraint in the literature. On a somewhat related note, Resnikoff (1989) observed that the general measurement uncertainty principle, as formulated by Gabor (1946), is important for understanding perception. For a recent review of uncertainty in perceptual measurement and the role of receptive field learning under this uncertainty, see (Jurica, Gepshtein, Tyukin, & van Leeuwen, 2013).

2.2. Three fundamental constraints on learning

In learning tasks, the need to generalize from labeled to unlabeled data (in supervised scenarios) or from familiar to novel data (in unsupervised scenarios) imposes certain general constraints on the computational solutions (Geman, Bienenstock, & Doursat, 1992). Although here we focus on categorization, where the goal is to learn class labels for data points, these constraints apply also to regression, where the goal is to learn a functional relationship between independent and dependent variables (Bishop, 2006).

According to the standard formulation in computational learning science, the problem of learning reduces, on the most abstract level of analysis, to probability density estimation (Chater, Tenenbaum, & Yuille, 2006). Indeed, the knowledge of the joint probability distribution over the variables of interest allows the learner to compute, for a query point, the value of the dependent variable, given the observed values (measurements) of the independent variables.[3] This basic insight serves as a background for the present discussion.

In this section, we briefly discuss the constraints that apply to (i) the computation of *similarity* among stimuli, (ii) to the *dimensionality* of representation spaces, and (iii) to the *complexity* of the decision surfaces.

2.2.1. Similarity

Estimating the *similarity* among stimuli is arguably the most important use to which sensory data could be put. As mentioned in the introduction, similarity constitutes the only principled basis for generalization Shepard (1987). Therefore, any non-trivial learning from experience (Edelman, 1998; Edelman & Shahbazi, 2012; Hume, 1748; Shepard, 1987) faces the following challenge:

[3] In this sense, the joint probability distribution over the representation space is the most that can be known about a problem. To know more – for instance, to know the directions of causal links between variables – observation alone does not usually suffice (Pearl, 2009; Steyvers, Tenenbaum, Wagenmakers, & Blum, 2003). This topic is beyond the scope of the present survey.

Similarity The fundamental challenge confronted by any system that is expected to generalize from familiar to unfamiliar stimuli is how to estimate similarity over stimuli in a principled and feasible manner.

2.2.2. Dimensionality

Given a representation space for which similarity has been defined, a straightforward and surprisingly effective approach to generalizing category labels is to assign to the query point a label derived from its nearest neighbor(s) (Cover & Hart, 1967). Importantly, this approach is nonparametric, in that no particular functional form is assumed for the underlying probability distribution function.

To ensure uniformly good generalization, the nearest neighbor approach requires that the representation space be "tiled" with exemplars, so that any new query point would fall not too far from familiar ones. This requirement gives rise to the so-called "curse of dimensionality" (a concept first formulated, in the context of control theory, by Bellman, 1961): the tiling of the problem representation space with examples, and with it learning to generalize well, becomes exponentially less feasible as the dimensionality of the space grows. Hence, the following constraint:

Dimensionality The fundamental challenge facing any learning system is how to reduce the effective dimensionality of the problem so as to allow learning from the typically sparse available data (Edelman & Intrator, 1997; Intrator & Cooper, 1992).

We remark that the effective dimensionality of a problem need not be the same as its nominal dimensionality, which is inherited from the measurement or representation space in which the problem arises. In particular, the parametric form of the decision or regression surface (or, more generally, of the underlying joint probability distribution) may be known independently, in which case the effective dimensionality is determined by that form. Likewise, in the support vector approach to classification (Cortes & Vapnik, 1995), the nominal dimensionality, which is equal to the number of features (dimensions of the representation space), is raised drastically when the problem is remapped into a new space that affords linear discrimination, yet its effective dimensionality is determined by the typically very small number of "support vectors" — key data points that determine the width of the classifier margin. More on this below and in Section 3.3.

2.2.3. Complexity

If the parametric form of the probability distribution is known, or if a particular form is adopted as a working hypothesis, subject to evaluation, then the focus in assuring good generalization shifts from the nominal dimensionality of the representation space to the number of parameters that need to be learned. As noted by Cortes and Vapnik (1995), it was Fisher (1936) who first formalized the two-class categorization problem and derived a Bayesian-optimal solution to it in the form of a quadratic discriminant function, which he recommended to approximate by a linear discriminant in cases where the number of data points is too small relative to the dimensionality of the measurement space — a very common predicament, known in learning theory as the problem of sparse data. Since then, the idea of keeping the number of parameters small – including opting whenever possible for the smallest number of parameters for a given problem, as afforded by the linear classifier – proved to be a manifestation of a very general principle that governs generalization from data.

Support for Fisher's recommendation comes from converging ideas in the theory of information and computation (Solomonoff, 1964), the Minimum Description Length Principle or MDL (Rissanen, 1987), nonparametric estimation (Geman et al., 1992), regularization theory (Evgeniou, Pontil, & Poggio, 2000), and statistical learnability theory based on the concept of Vapnik–Chervonenkis (VC) dimension (Blumer, Ehrenfeucht, Haussler, & Warmuth, 1989), which is in turn founded on empirical risk minimization (Vapnik, 1999). This latter approach, which leads to Support Vector Machines, has been described by Vapnik as follows: "To generalize well, we control (decrease) the VC dimension by constructing an optimal separating hyperplane (that maximizes the margin). To increase the margin we use very high dimensional spaces".

On the face of it, the second desideratum identified by Vapnik – a high-dimensional representation space – runs counter to the *Dimensionality* constraint identified earlier. However, as we shall see in Section 3.1.2, it is made unproblematic by the so-called "kernel trick", which ensures that the *effective* dimensionality of a problem approached in this manner is dictated by the number of data points, rather than by the number of intermediate representation-space "features", which need never be computed explicitly (Jäkel, Schölkopf, & Wichmann, 2007). The windfall from this mathematical fact allows us to focus on the first part of Vapnik's statement:

Complexity The fundamental challenge facing any categorization system is how to remap the problem at hand into a space where it becomes a matter of low-complexity – preferably, linear – discrimination.

3. Kernel-based methods

The four fundamental constraints listed above – *Measurement*, *Similarity*, *Dimensionality*, and *Complexity* – are simultaneously satisfied by a family of computational approaches based on the concept of *kernel*. In this section, we first describe the so-called "kernel trick" and demonstrate its application with an example, then discuss it in more detail for the general case.

3.1. The "kernel trick"

The phrase "kernel trick" (Bishop, 2006) refers to the possibility of enjoying the advantages of a high-dimensional representation space without having to pay the price of explicit computations in that space — a possibility that is an immediate corollary of the definition of inner product and that holds for any use of data where computations over inner products suffice (as in Principal Component Analysis or in Support Vector Machines). In the following exposition, we draw on materials from (Balcan, Blum, & Vempala, 2006; Jäkel et al., 2007, 2008, 2009; Schölkopf & Smola, 2002).

3.1.1. A simple example of the kernel trick

As an example, consider the problem of classifying objects, represented by points in some multidimensional measurement space, into two or more categories. Information that would support such classification may not be available in individual features (dimensions) of the objects or even in their linear combinations. In such cases, one may resort to the use of a polynomial classifier, whose input features include, in addition to the original dimensions, some or all of their products (Boser, Guyon, & Vapnik, 1992).

For instance, suppose the samples consist of the pairs of dimensions (length and width) of a set of rectangles that are to be classified on the basis of their area, i.e., one label for rectangles whose area is greater than some value and a different label for those that are smaller. In other words, the original signal is $x \in \mathbb{R}^2$, and the

feature of interest is contained in the product x_1x_2. If the diagnostic feature (area) is known a priori, we can explicitly compute it for all our samples and use it in training. However, in practice often not much is known about the diagnosticity of features. An easy fix in those cases is to provide the classifier with several candidate features by mapping the samples from their original space to a new one, obtained under some feature map, $\phi(\cdot)$. For the rectangles example, the polynomial $\phi : \mathbb{R}^2 \to \mathbb{R}^3, (x_1, x_2) \to (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ would be one such feature map. Naturally, the higher the dimensionality of the new representations, the more likely it is that the diagnostic feature is present in it.

Unfortunately the computational cost of such mappings can be prohibitive, particularly when the original data reside in a high-dimensional space (as does any set of megapixel-resolution images), as per the fundamental constraint on *Dimensionality*. For instance, evaluating a d-degree polynomial in N dimensions requires computing $(N + d - 1)!/d!(N - 1)!$ monomial terms. Ideally, one would like to keep the advantages of high-dimensional feature spaces while reducing the cost of working with them. This is where kernel-based approaches come in handy.

A kernel is a nonnegative definite, symmetric function of two arguments, $k(x, y) : \mathbb{R}_x \times \mathbb{R}_y \to \mathbb{R}$.[4] It can be shown that evaluating such a kernel corresponds to taking the inner product $k(x, y) = \langle\phi(x), \phi(y)\rangle$, where $\phi(\cdot)$ is some function, defined over the original domain, which may be desirable but expensive to compute (cf. Section 3.1.2). Identifying a kernel $k(\cdot, \cdot)$ for a function $\phi(\cdot)$ makes it possible, therefore, to evaluate for a given x and y the inner product $\langle\phi(x), \phi(y)\rangle$ directly, without having first to compute the expensive $\phi(x)$ and $\phi(y)$. In the above example, opting for $k(x, y) = \langle x, y\rangle^2$, we get:

$$x = (x_1, x_2), \qquad y = (y_1, y_2)$$
$$\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2), \qquad \phi(y) = (y_1^2, y_2^2, \sqrt{2}y_1y_2)$$
$$\langle\phi(x), \phi(y)\rangle = (x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1x_2y_2) = \langle x, y\rangle^2 = k(x, y).$$

In other words, as long as the classifier only requires the inner products of the samples, as in the case of Support Vector Machines (SVM) or methods based on Principal Component Analysis (PCA), it can use the kernel trick to take advantage of the higher dimensional representations without having to pay the price of computing them.

3.1.2. The kernel trick in general

In machine learning, the primary motivation for using kernels is that in many cases the data classes in their original representation space are not linearly separable, preventing the learner from relying on the tried-and-true algorithms that assume linear separability, such as the Perceptron or Fisher's Linear Discriminant. In such cases, one may use a dimension-raising map $\phi(\cdot)$ to embed the data points into a new, higher-dimensional space, in which they may become linearly separable (Cover, 1965). Intuitively, adding dimensions adds ways in which the points may differ, which may result in linear separability. By raising the dimensionality, we may also effectively enrich the original representation with combinations of the existing features. In any case, the separating hyperplane in the new space corresponds to a non-linear boundary in the original space.

This method can be particularly effective if the choice of $\phi(\cdot)$ is insightful. However, attaining the requisite insight, remapping the data, and any subsequent processing in the ϕ-space can all be very expensive, rendering this approach impractical. Kernelization can remedy these problems.

Cost of computation. In 1964, Aizerman, Braverman, and Rozoner observed that a symmetric positive semi-definite kernel $k(\cdot, \cdot)$ can be viewed as the inner product of some function $\phi(\cdot)$ evaluated at two different points, x and y: $k(x, y) = \langle\phi(x), \phi(y)\rangle$ (the proof of this property is given by Mercer's theorem; Mercer, 1909). Further, as long as the learning algorithm only requires the inner products of data points, i.e., $\langle x, y\rangle$, the kernel $k(x, y)$ can be used to obviate the need to remap the data explicitly through $\phi(\cdot)$ before computing their inner product. In other words, instead of first computing $x \to \phi(x), y \to \phi(y)$ and then $\langle\phi(x), \phi(y)\rangle$, one can compute only the less expensive $k(x, y)$ to the same effect. This shortcut, which came to be known as the kernel trick, made it possible for learning algorithms that up to that point were only effective on linear problems, to successfully handle nonlinear data sets as well, with a reasonable computational overhead. However, it was not until 1992 that the seminal paper by Boser et al. on large margin classifiers, also known as Support Vector Machines, made a strong case for the merits of kernelization and introduced it to the mainstream machine learning.

Choice of transformation. While relying on a kernel function can keep the cost of computation under control, one still needs to figure out what transformation $\phi(\cdot)$ to use, and also what kernel $k(\cdot, \cdot)$ corresponds to that particular $\phi(\cdot)$. Answering the latter question is easy: for a given $\phi(\cdot)$, the corresponding kernel is given by taking its inner product with itself: $k(x, y) = \langle\phi(x), \phi(y)\rangle$. The symmetry of the resulting kernel follows from the properties of inner product; its positive definiteness is only required to guarantee existence of a corresponding feature map, which in this case is established independently.

The question of what transformation $\phi(\cdot)$ to use is, however, not as straightforward, because in general not enough is known about the problem to settle it. Consequently, both $\phi(\cdot)$ and $k(\cdot, \cdot)$ are chosen to be flexible enough to accommodate a wide range of possibilities. Specifically, a massive dimensionality increase seems like a good bet: as shown by Cover (1965), any data set with high probability become linearly separable when mapped onto a space with a sufficiently high dimensionality.

In practice, instead of choosing $\phi(\cdot)$ and computing the kernel from it, the designer of a machine learning system decides on an off-the-shelf kernel known to correspond to a dimension-raising $\phi(\cdot)$. Just how high the new dimensionality will depends on the particular choice of kernel, which for some cases, e.g., the Gaussian kernel $k(x, y) = e^{-\gamma\|x-y\|^2}$, will be infinite (Eigensatz & Pauly, 2006).[5] Why this is so is beyond the scope of this paper, especially since we will not pursue the dimension-raising view of the kernels any further; see Table 1 for a summary of the present discussion.

It is worth emphasizing that in principle, untangling nonlinearly separable data does not have to involve raising their dimensionality and may be achieved via a dimension-preserving (or perhaps even dimension-reducing) $\phi(\cdot)$. Therefore, raising the dimensionality is merely a practical choice that may be more convenient than searching for the alternative.

Regularization. The linear boundary that is obtained in a high-dimensional space corresponds to a nonlinear boundary in the original space of representation, which raises a concern about

[4] A function is nonnegative definite if all its eigenvalues are greater than or equal to zero; see e.g. (Schleif & Tino, 2015, p. 2071) for the definitions of eigenvalues of a function.

[5] The interested reader may observe that expressing the Gaussian kernel in terms of the corresponding $\phi(\cdot)$'s whose inner product would be $k(\cdot, \cdot)$ involves an infinite expansion. For instance, for $x, y \in \mathbb{R}$ we may have $k(x, y) = e^{-\|x-y\|^2} = e^{-x^2}e^{-y^2}e^{2xy} = e^{-x^2}e^{-y^2}\sum_{i=0}^{\infty}\frac{2^ix^iy^i}{i!}$ where the series results from the Taylor expansion of the last term. Therefore, the feature map is $\phi(t) = e^{-t^2}\sum_{i=0}^{\infty}\sqrt{\frac{2^i}{i!}}t^i$.

Table 1
Summary of the main concepts pertaining to the discussion of the kernel trick in Section 3.1.2. To apply a linear discriminant algorithm to a sample set that is not linearly separable, one can remap the data under a dimension-raising transformation into a higher dimensional space where they are likely to become linearly separable. Furthermore, to bypass the expense of explicit computation in high-dimensional spaces, a symmetric positive semi-definite kernel can be used in place of the inner product of the samples in the new space.

$x_1, x_2, \ldots, x_n \in \mathbb{R}^d$	:	Samples are not linearly separable
$\phi : \mathbb{R}^d \to \mathbb{R}^{d_2}, d_2 > d$	:	Dimension-raising map
$\phi(x_1), \phi(x_2), \ldots \phi(x_n) \in \mathbb{R}^{d_2}$	:	Samples are linearly separable in the new space
$\langle \phi(x_i), \phi(x_j) \rangle$	:	Learning algorithm requires the inner product of the new representations
$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$	:	Operating in the ϕ-space is expensive; use k instead
$\langle \phi(x_i), \phi(x_j) \rangle = k(x_i, x_j)$	:	Using the less expensive k on the original form of x_i and x_j has the same effect as transforming them with ϕ and taking their inner product

generalizability of the learned criterion: a decision boundary with too many degrees of freedom is likely to result in overfitting. The high-dimensional feature maps induced by kernels make it easy to find a combination of parameter values that make the model fit the training data well; most such solutions will, however, generalize poorly to new data. Consequently it is essential that kernel-based learning algorithms constrain the complexity of their model by regularizing it (cf. the question of VC dimension in Section 2.2.3). However, too much regularity will harm the model's ability to fit the available data. In machine learning, this tension between complexity and simplicity is referred to as bias–variance tradeoff (Geman et al., 1992).

In kernel-based settings this issue is addressed by regularization of the decision boundary (Evgeniou et al., 2000). More specifically, during training, the cost function that is being minimized includes a term that penalizes the irregularity of the class boundary, e.g., the norm of the derivative of the decision function, $\|f'\|$, which will be smaller for smoother (i.e., more regular) functions.

3.2. Kernel as a measure of similarity

So far we have focused on the feature map, $\phi(\cdot)$, and its dimension-raising power for untangling data and making them linearly separable. In fact, were it not for the practical difficulties of working explicitly with $\phi(\cdot)$, there would be no place for the kernel in our discussions. However, viewing kernels as a way of raising the dimensionality may not be the most intuitive approach. For instance, in the example of the kernel trick in the previous Section 3.1, it is not quite clear how one would know that the product x_1x_2 is the right choice of feature for the classifier, or why $\phi(y) = (y_1^2, y_2^2, \sqrt{2}y_1y_2)$ should be preferred over the many other mappings that could provide this feature. In this section, we shift our focus from the use of $\phi(\cdot)$ as a means of increasing dimensionality to the use of $k(\cdot, \cdot)$ as a measure of similarity.

As guaranteed by Mercer's (1909) theorem, for any symmetric positive semi-definite kernel, there always exists a function $\phi(\cdot)$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle$. Such a kernel can therefore be viewed as measuring the cosine similarity between data points (vectors) by taking their inner product.[6] However, instead of comparing x and y as they are, k compares a transformed version of them, which for different choices of kernel can be very similar to x and y (e.g., for the linear kernel: $k(x, y) = \langle x, y \rangle$) or very different (e.g., for the Gaussian kernel $k(x, y) = e^{-\gamma \|x-y\|^2}$).

Interestingly, if what we are after is similarity, there is no need to invoke the notion of an implicit map $\phi(\cdot)$: the kernel k can be any function that assigns a non-negative value to a pair of input points $x_j \in X$ regardless of their order, i.e., $k : X \times X \to \mathbb{R}$. If the assigned value can serve as a similarity measure (as in the Gaussian kernel where the assigned value $e^{-\gamma \|x-y\|^2}$ is a nonlinear form of the Euclidean distance between the inputs $\|x - y\|$), then that kernel is useful.

3.2.1. Representation through measurement of similarity

In practice, as in the Chorus of Prototypes approach (Edelman, 1995, 1999; Edelman & Shahbazi, 2012), a subset of the data points can be chosen to serve as landmarks, in terms of distances to which the remaining points are represented and their similarities measured. The representation of any point in the new representation space is a vector whose jth entry is the similarity (inverse distance) of that point to the jth landmark. This new representation can then be used for learning in the usual way. Table 2 summarizes these ideas.

The view of kernels as tools for similarity estimation has been implicit in the kernel approach all along. Recall that the kernel trick is only useful when the learning algorithm relies on the inner products of the data points. That means that the cosine similarity is built into the dimension-raising view as well. The similarity view simply refocuses our interpretation of what the kernel does, shifting attention from ϕ as the goal and k as the "trick" that gets us there to k itself as the goal.

Furthermore, since the kernel approach needs to be paired with a learning algorithm that works with inner products, it is often viewed as part of the learning process, as evidenced by the central role of the kernel trick in SVM methods. In comparison, under the similarity view, the kernel is a means of representing data, which is somewhat independent of the learning algorithm. The advantage of this approach is that it may lead to representations that are better suited to learning. In fact, under the right circumstances even a simple nearest neighborhood search may suffice for learning from data represented via similarity (as in locality-sensitive hashing Arya, Mount, Netanyahu, Silverman, & Wu, 1998; Edelman & Shahbazi, 2012; cf. Section 4.3). Indeed, there now exist kernelized versions of most of the popular classification and regression algorithms in machine learning.

As an example, consider the application of kernels as similarity measures in the Perceptron algorithm. The neurally inspired perceptron decides on the category of the input by comparing a weighted sum of its elements to a threshold: $C(x) = \text{sgn}(\langle w, x \rangle) = \text{sgn}\left(\sum w_j x_j\right)$, with w denoting the weight vector.[7] Being a linear combination of the input feature values, the perceptron's decision boundary is a hyperplane in the input space, which causes the algorithm to fail when the categories are not linearly separable (Minsky & Papert, 1969).

We can try to fix this shortcoming by resorting to kernels (cf. Jäkel et al., 2007), but the kernel trick only works if the learning algorithm relies exclusively on inner products between data points, and never requires the points themselves. The original formulation of the perceptron decision criterion in terms of (a weighted sum of the elements of) the individual points does not

[6] The cosine of the angle between x and $y \in \mathbb{R}^d$ is $\frac{\langle x,y \rangle}{\|x\| \cdot \|y\|}$.

[7] The sign function is defined as $\text{sgn}(t) = \begin{cases} 1 & t \geqslant 0 \\ -1 & t < 0 \end{cases}$. For simplicity, we have omitted the bias term b from the general form of the perceptron decision function, which is $C(x) = \text{sgn}(\langle w, x \rangle + b)$.

Table 2
Summary of the use of kernels as a measure of similarity (Section 3.2). Instead of a shortcut to high-dimensional computations, the kernel can be viewed as a measure of similarity, yielding a new representation of data that might better serve learning from them. First, a subset of data points is chosen as landmarks; then, the kernel is used to compute the similarity of the remaining points to these. The new representation of each point consists of the resulting vector of similarities.

$X : x_1, x_2, \ldots, x_n \in \mathbb{R}^d$	:	Sample set used in learning
$e_1, e_2, \ldots, e_m \subset X$	:	Subset of the samples chosen as exemplars, e_j
$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$	:	Appropriate k for measuring similarity
$\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^m$	:	Re-represent each sample via transformation $\mathcal{T}$. The new dimensionality, m, is decided by the number of exemplars
$\mathcal{T}(x) = (k(x, e_1), k(x, e_2), \ldots, k(x, e_m))$	:	Each sample re-represented as a vector of its similarities to the exemplars
$\hat{X} : \mathcal{T}(x_1), \mathcal{T}(x_2), \ldots \mathcal{T}(x_n) \in \mathbb{R}^m$	:	The new representation of samples used in learning.

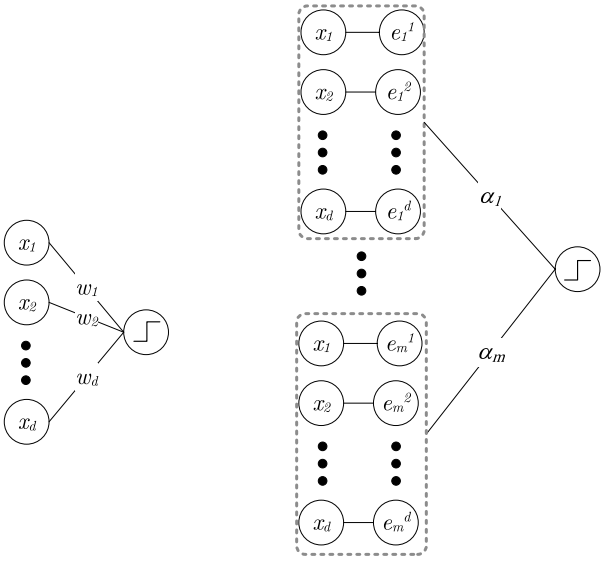

Fig. 1. *Left*: The classical formulation of the perceptron algorithm can only handle linearly separable data. *Right*: The perceptron algorithm can be modified using a kernel as a measure of similarity, which makes it capable of dealing with nonlinearly separable data as well. A subset of data points is chosen as landmarks, and the remaining points are re-represented as their lists of similarities to the those (enclosed by the dashed lines above). The values of α_j, the emphasis put on similarity to landmark j, play the role of the weights in the non-kernelized perceptron (cf. Section 3.2 and Table 2).

meet this requirement. The solution lies in applying the kernel as a measure of similarity. In particular, we may select a subset of the training points as landmarks, $e_1, \ldots, e_m$, and measure against them the kernelized similarity of the test point, x. The rest is the same as in the classical perceptron: $C(x) = \text{sgn}\left(\sum_1^m \alpha_j k(x, e_j)\right)$, or, in the notation of Table 2, $C(x) = \text{sgn}\left(\langle \alpha, \mathcal{T}(x)\rangle\right)$.

Learning then consists of optimizing the weights α_j, which denote the emphasis attached to the similarity of x to each of the preset landmarks (Fig. 1; cf. Freund & Schapire, 1999). To see that this new decision rule corresponds to a kernelized version of the linear perceptron, note that the weight vector can be expressed as a linear combination of the exemplars, $w = \sum \alpha_j \phi(e_j)$; thus, optimizing α has the same effect as optimizing w. Therefore, applying the kernel as a measure of similarity has the same effect as its application in the dimension-raising view: $C(x) = \text{sgn}\left(\langle w, \phi(x)\rangle\right) = \text{sgn}\left(\langle \sum_j \alpha_j \phi(e_j), \phi(x)\rangle\right) = \text{sgn}\left(\sum \alpha_j \langle \phi(e_j), \phi(x)\rangle\right) = \text{sgn}\left(\sum_j \alpha_j k(e_j, x)\right)$. For a more recent example of kernelizing an existing algorithm, see the deep learning method of (Cho & Saul, 2009).

3.2.2. Kernels, the measurement constraint, and representation of similarities

Having discussed the views of kernels as a shortcut to the inner product and as a measure of similarity, we now return to the fundamental constraint on *Measurement*, introduced earlier. Recall that the issue here is access to information about the world on part of any neuron that is at least once removed from the sensory transduction stage. The energy of photons captured by the retinal photoreceptors speaks to the present state of the environment, and may carry information vital to the animal. Yet at any stage inside the nervous system this information is only available in a form that depends on the transfer function of possibly many preceding stages.[8] In principle (albeit probably not in practice), this transfer function may be estimated if calibration data are somehow made available. However, the need for such estimation can be avoided if the unit in question learns to be sensitive only to *second-order* properties (Shepard, 1968; Shepard & Chipman, 1970) of the data points, such as their pairwise similarities (Edelman, 1998).

Suppose x and y are two data points fed into a processing stage described by a function $f(\cdot)$, with the outcome $f(x)$ and $f(y)$ (Fig. 2). The next processing stage has access only to these latter values, that is, it "sees" the input only through f; it may not, therefore, rely on computations that require an explicit knowledge of x and y. However, if the system is only interested in the similarity between x and y (a reasonable assumption in cognition), and if the kernel defined by $k(\cdot, \cdot) = \langle f(\cdot), f(\cdot)\rangle$ is a suitable measure of similarity, then all is well. As long as the f-transformed version of the signal can be accessed, all the information of interest about x and y is present. Thus, while the similarity view of kernels addresses their relevance to cognitive tasks, the view that emphasizes the inner product aspects of feature maps suggests how a cognitive system may circumvent the difficulty arising from the fundamental constraint on measurement.

3.3. Regarding the dimensionality of kernel solutions

In this section we first reiterate the significance of dimensionality reduction for learning from experience, then discuss how this issue is addressed in learning systems that rely on kernels.

3.3.1. Do kernels increase or decrease dimensionality?

While measurements performed by a sensory system are typically high-dimensional (to boost the system's ability to make fine distinctions), the amount of data available for learning is always too small, compared to the exponential demands of tiling a high-dimensional representation space with examples — a consequence of the curse of dimensionality (Bellman, 1961), mentioned in Section 2.2.2. What can kernel methods contribute to the solution of this problem?

The relationship between kernels and dimensionality is a somewhat confusing topic, perhaps because in themselves kernels are indifferent to the dimensionality of their domain of application: whether that gets raised, reduced, or left unchanged is for the most part up to the practitioner. Nonetheless, since in machine learning the usefulness of kernels is often attributed to the dimension-raising power of ϕ, cognitive scientists may see the kernel trick

[8] This observation, which these days is likely considered a truism, can be traced back to Johannes Müller's "law of specific nerve energies", which he formulated in 1835.

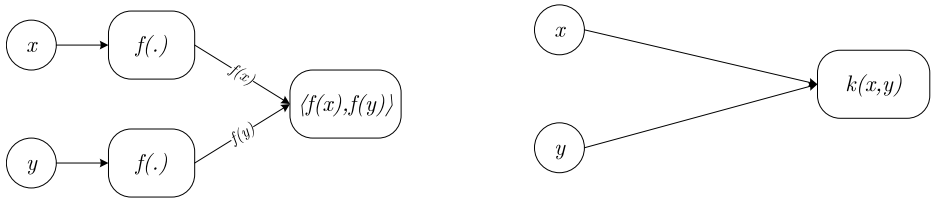

Fig. 2. *Left*: The rightmost processing unit has only access to x and y as seen ("distorted") through $f(\cdot)$ and cannot perform any computation that requires an undistorted version of the signal. *Right*: If, however, the goal is solely to estimate the similarity of x and y in the form of $k(x, y)$, it can use $\langle f(x), f(y)\rangle$ to the same effect, as if it were accessing x and y directly, and effectively bypassing the intermediate stage, $f(\cdot)$ (cf. Section 3.2.2).

exclusively as a method of inflating the dimensionality of data, and thereby dismiss it as irrelevant to the behavioral tasks, where ultimately a reduction of dimensionality is needed (cf. the fundamental constraint on *Dimensionality* in Section 2.2.2). Our aim here is to offer a more balanced view of the relationship between kernels and dimensionality.

As we discussed earlier, there are two main views of kernels: one that emphasizes the feature map $\phi(\cdot)$ and the other — the notion of similarity. Regarding the former view, it is not a necessity that $\phi(\cdot)$ should raise dimensionality; rather, the decision to do so is a matter of convenience, reflecting the designer's imperfect knowledge about the problem. Furthermore, the feature map is *implicit* and never actually computed (which only further complicates this issue). In fact, the data points never leave their native space, where the eventual solution will also reside. Consequently, the onus of determining the effective dimensionality of the kernelized solution is on the learning algorithm, not the feature map. This is made especially clear by the need to introduce regularization into the learning algorithm. For instance, in the SVM algorithm, the effective dimensionality of the decision surface is independent of the dimensionality of $\phi(\cdot)$ and is determined instead by the support vectors, in conjunction with tight regularization.

Under the similarity view, the dimensionality of the data does change, but according to the number of landmarks chosen, and, again, independently of the corresponding $\phi(\cdot)$. Consulting Table 2, we see that following the re-representing transformation $\mathcal{T}$, the dimensionality of the data becomes equal to m, the number of landmarks. The good news is that m is usually much smaller than d, the dimensionality of the original data (Section 3.3.2 explores this idea in more detail).

In summary, kernel methods do not violate the fundamental constraints of *Dimensionality* and *Complexity*; rather, they comply with them by bounding the dimensionality of the eventual solution, thereby increasing the generalizability of the learned decisions. This is achieved by regularizing the decision surface, or, to the same effect, by remapping data using a transformation $\mathcal{T}$ that relies on landmarks whose number is smaller than the original number of dimensions. This bring us to the next question: how do we choose the landmarks?

3.3.2. Random projections and feature selection

Good dimensionality reduction methods preserve as much as possible the relative similarities among items in the new representation space, because these similarities are what categorization is to be based upon (Edelman, 1998, 1999; Edelman & Shahbazi, 2012; Shepard, 1987). As suggested by Edelman (1999), the Johnson–Lindenstrauss (1984) lemma offers a computationally simple and inexpensive way for embedding data into a lower dimensional space while approximately preserving their relative distances, under certain conditions. As long as the number of data points is small relative to their dimensionality (a situation that arises often in perceptual processing), projecting them onto a randomly chosen space of logarithmically lower dimensionality will suffice. Formally, for $x_j \in \mathbb{R}^d$, a linear map $l : \mathbb{R}^d \to \mathbb{R}^m$ with $d \gg m$, and $0 < \varepsilon < 1$, the distance distortion is bounded as follows:

$$(1-\varepsilon)\,\|x_i - x_j\|^2 \le \|l(x_i) - l(x_j)\|^2 \le (1+\varepsilon)\,\|x_i - x_j\|^2.$$

In view of the above considerations, Balcan et al. (2006) observe that the kernel method can be thought of as a lower dimensional embedding of the data. Suppose that a data set, when remapped under ϕ corresponding to some kernel k, becomes linearly separable. By the Johnson–Lindenstrauss lemma, projecting the remapped data onto a subspace spanned by randomly chosen vectors r_j should nearly preserve their linear separability. However, a straightforward application of the lemma, as per $(\langle r_1, \phi(\cdot)\rangle, \langle r_2, \phi(\cdot)\rangle, \ldots, \langle r_m, \phi(\cdot)\rangle)$,[9] would be too expensive to compute, because r_j are of the same dimensionality as ϕ. Instead, one can draw e_1 through e_m from the original data at random to serve as landmarks and remap any other point x using $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^m$ with $m \ll d$ as

$$\mathcal{T}(x) = (k(x, e_1), k(x, e_2), \ldots, k(x, e_m))$$

where $k(\cdot, e_j)$ corresponds to a random projection from ϕ-space along the jth dimension of $\mathcal{T}$ — similar to $\langle r_j, \phi(\cdot)\rangle$, but without the explicit computation of ϕ and $\langle r_j, \phi(\cdot)\rangle$. In other words, $\mathcal{T}$ provides an inexpensive way of embedding x in a lower dimensional subspace while preserving approximately its linear separability under ϕ. Formulated in this manner, k can be thought of as a measure of similarity, and e_j as landmarks, prototypes, or features against which k measures the similarity of x (Balcan et al., 2006; Blum, 2006). In visual object recognition, a recent example of successful application and further development of this method, introduced by Edelman (1999), is the work of Anselmi et al. (2014).

The small set of randomly selected prototypes in the formulation of (Balcan et al., 2006) serves well for a binary classification setting. While the significance of feature selection for similarity-based learning is still being explored (Pękalska, Duin, & Paclík, 2006), it appears that in a more involved setting, with multiple classes and complex feature sets, it may be preferable to (i) carefully select the exemplars so as to better reflect prior knowledge about the structure of the data, (ii) select them via optimization of some objective function (Klare & Jain, 2012), or (iii) increase the number of randomly chosen prototypes, thus increasing the probability of including some nearly optimal ones.

3.4. Kernels and the fundamental constraints

We are now in a position to make the following observation: the kernel trick can serve as a conceptual basis for an approach to representation and learning from data that would satisfy all four fundamental constraints listed earlier:

[9] The projection of a vector v onto the subspace spanned by $r_1, \ldots, r_m$ is given by $\mathcal{R}v$, where $\mathcal{R}$ is the projection matrix whose jth column is r_j.

Measurement By relying on kernels both as a measure of similarity and as an implicit feature map, a system can gain direct access to information that would otherwise remain implicit in the data (Section 3.2.2).

Similarity A unit that needs to compute the similarity of x and y can do so using $k(x, y)$ (Section 3.2).

Dimensionality Using the kernel trick, a learning problem can be embedded into a space spanned by the data points, whose effective dimensionality is typically much lower than the nominal dimensionality of the data space (Section 3.3.2).

Complexity By keeping the effective dimensionality of the solutions low, either by regularizing the solution or by relying on a few landmark data points, kernel-based methods keep complexity under control (Section 3.3.1).

3.5. A probabilistic angle

Similarity-based approaches to learning and generalization, as well as kernel methods in general, have received extensive probabilistic treatment. A particularly prominent example is Shepard's (1987) derivation, from first principle and using the Bayes Theorem, of a universal law of generalization, according to which the probability of a novel and a familiar stimulus sharing the same "consequence" decays exponentially with their dissimilarity or distance in the perceiver's representation space.[10] Tenenbaum and Griffiths (2001) subsequently offered an explicitly Bayesian formulation and extension of Shepard's law, which included generalization from multiple familiar examples.

Shepard's law can be seen as giving rise to a family of exponential radial basis function (RBF) kernels: the probability of assigning the same label L to two stimuli, x and y, is given by $P(L_x = l_0 \mid L_y = l_0) \propto e^{-\|x-y\|_p}$, where p is the parameter that selects the l_p metric in the representation space. More recently, an explicitly kernelized Bayesian approach was developed by Smola, Gretton, Song, and Schölkopf (2007), who proposed a way to "embed [probability] distributions in a Hilbert space" by constructing a mapping between the two and treating each point in the Reproducing Kernel Hilbert Space or RKHS[11] as the mean of a distribution. This method was further generalized by Song, Huang, Smola, and Fukumizu (2009) to include conditional distributions, leading eventually to an explicitly Bayesian formulation of nonparametric posterior point estimation in RKHS by Fukumizu, Song, and Gretton (2011).

3.6. Are Gaussian kernels special?

Learning algorithms that make use of the kernel trick require that the kernel be chosen carefully, so as both to accommodate the particular set of data at hand and avoid overfitting (see the discussion of regularization in Section 3.1.2). In machine learning practice, kernels are often chosen by hand to fit the problem. For instance, while a polynomial kernel may suffice for certain types of data, other cases may require a sigmoid or a Gaussian kernel. Choosing the kernel and tuning its parameters are therefore aspects of model selection — a machine learning task that is generally very difficult (Burges, 1998; Howley & Madden, 2005; Lowe, 1995). Computational modelers therefore often resort to kernels that are known to be powerful and versatile, such as RBFs, which are commonly assumed to be Gaussian: $k(x, y) \propto e^{-\|x-y\|^2}$ (Belkin & Niyogi, 2003; Hegde, Sankaranarayanan, & Baraniuk, 2012).

This choice is sometimes motivated by pointing out parallels between RBFs and neuronal response profiles. In the mammalian primary visual cortex, for instance, there are neurons whose response falls off with the distance, in the appropriate feature space, between the actual stimulus x and the preferred one y as $e^{-\gamma\|x-y\|^p}$ (Daugman, 1980; Kang, Shapley, & Sompolinsky, 2004; Rose, 1979). Similar tuning profiles are observed in other cortical areas (Dayan & Abbott, 2001) and animal species (Miller, Jacobs, & Theunissen, 1991; Theunissen, Roddey, Stufflebeam, Clague, & Miller, 1996). Some of the properties of the Gaussian kernel that may contribute to its apparently widespread use are:

- The Gaussian tuning curves of a collection of units can serve as a basis function set for the synthesis of versatile mappings, for instance between different sensory modalities (Pouget & Sejnowski, 2001).
- With regard to basis function decomposition, it is interesting to note that the Fourier transform of a Gaussian consists of Gaussian basis functions (e.g., for $f(t) = e^{-\alpha t^2}$ we have $\mathcal{F}(s) = \sqrt{\pi/\alpha}e^{-\pi^2 s^2/\alpha}$).
- Multidimensional Gaussian basis functions can be synthesized as products of lower-dimensional ones (Poggio & Edelman, 1990).
- The Gaussian kernel is self-similar: convolving two Gaussians yields another Gaussian. This property may be helpful in that in a cascade of cells with Gaussian tuning curves little information is lost to those units that do not have direct access to each other; cf. the *Measurement* constraint.
- The Gaussian can arise from the collective activity of a large population, none of whose profiles are necessarily Gaussian, as per the Central Limit Theorem.
- The feature map corresponding to a Gaussian kernel is infinite dimensional (Eigensatz & Pauly, 2006), offering more flexibility where little is known about the nonlinearities present in the data.

4. Issues and ideas related to kernels

In this section, we discuss the main headings under which the relevant material is typically found in the literature. As we shall see, there is considerable overlap among the headings, which underscores the need for a unified framework.

4.1. Manifolds and linearization

A central goal of perceptual processing is to extract from the sensory measurements the presumably much lower-dimensional representations that are most pertinent to the various tasks faced by the cognitive system (Edelman & Intrator, 1997). For instance, in object recognition, a major challenge is to separate the dimensions of shape variation (along which different objects differ) from those that correspond to variable viewing conditions, such as object pose relative to the viewer. Both these sets of dimensions are best thought of as low-dimensional manifolds: the shape space, which captures shape variation, and the view space, which captures object pose (Edelman, 1999). For a rigid object with two rotational degrees of freedom (for instance), the latter is a smooth two-dimensional manifold.

The measurement-space manifold corresponding to all the images of a given object is, generally, highly nonlinear (even if locally smooth; Edelman, 1999). Moreover, it may be closely entangled in the measurement space with the manifolds corresponding

[10] Note that the decay in Shepard's law is inverse exponential in the distance, or $e^{-\|x-y\|}$, rather than a Gaussian, or $e^{-\|x-y\|^2}$.

[11] A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space associated with a kernel that "reproduces" every function in the space, in the sense that the inner product of the function with the kernel, $\langle f, k(\cdot, x)\rangle$ results in the point evaluation of the function at x, $f(x)$. Furthermore, every such function can be written as a linear combination of kernels, evaluated at a chosen set of data points.

to other objects of interest (DiCarlo & Cox, 2007; Edelman & Duvdevani-Bar, 1997). To tolerate changes in pose and other viewing conditions, a visual recognition system must separate the manifold of the object of interest from those of potential distractors. This can be done either by approximating the manifolds directly (Edelman & Duvdevani-Bar, 1997; Lando & Edelman, 1995) or by first transforming the representation space so that the manifolds become simply – that is, linearly – separable (DiCarlo & Cox, 2007). The latter approach seems to be more in line with the fundamental constraint on *Complexity* (Section 2.2.3), which states that good generalization requires that the decision surfaces be as simple as possible. There is some evidence to the effect that the primate visual system carries out progressive linearization of the representations, as they are transformed by the successive processing stages in the ventral stream (Pagan et al., 2013).

From the computational standpoint, kernelizing the metric used to measure similarity between data points can be effective in separating nonlinear manifolds. For instance, while ordinary PCA can only extract linear subspaces, its kernelized version, kPCA (Schölkopf, Smola, & Müller, 1997), is capable of uncovering nonlinear trends as well. Graph-based methods, such as Isomap and Laplacian eigenmaps (discussed below in Section 4.2) can also use kernels to approximate and untangle nonlinear manifolds. As in other contexts, the effectiveness of kernel methods in manifold learning stems in part from their ability to avoid expensive computations. For instance, Hegde et al. (2012) note that most visual manifold learning methods unquestioningly apply Euclidean distance to measure image similarity and suggest that the Earth Mover Distance is a better alternative — but only if its computational cost is kept in check through the use of kernels.

4.2. Graph methods

Some of the graph-theoretic approaches to low-dimensional manifold discovery in high dimensional data spaces are related to kernels. Consider, for instance, isometric feature mapping, or Isomap (Tenenbaum, 1998), which treats the data points in a small neighborhood as vertices of a weighted graph whose edges correspond to the pairwise Euclidean distances of data points. The geodesic distance of any two points along the manifold is then defined as the length of the shortest path between them in the graph. By applying multidimensional scaling to these geodesic distances, Isomap can reveal underlying nonlinear manifolds in a computationally tractable fashion. Furthermore, the adjacency matrix representing the geodesic distances can be interpreted as a kernel matrix, in which the weight of the edge connecting vertices i and j is $k(x_i, x_j)$ (Ham, Lee, Mika, & Schölkopf, 2004). This suggests that Isomap is related to the kernel eigenvalue problem, with the caveat that the matrix in question may not necessarily be positive semi-definite (a property required of kernel matrices).

Similarly, in the Laplacian eigenmap method (Belkin & Niyogi, 2003), data points become the vertices of a Gaussian-weighted graph, such that the weight of the edge connecting points x_i and x_j is $W_{ij} = e^{-\|x_i - x_j\|^2/2\sigma^2}$. The target manifold is then computed via eigendecomposition of the graph Laplacian (Wilson & Zhu, 2008).[12] Ham et al. (2004) point out that these graph-based methods, Isomap and the Laplacian eigenmap, as well as the closely related Locally Linear Embedding (LLE), can be viewed as special cases of the general family of kernel PCA.

4.3. Locality-sensitive hashing

Our next example of a popular technique that is related to kernels is Locality Sensitive Hashing (LSH), a similarity-based approach to content-addressable memory and classification, which extends and improves upon the k-nearest-neighbor (k-NN) idea (Arya et al., 1998; Cover & Hart, 1967). Elsewhere, we have discussed content-addressable memory and hashing in the context of similarity (Edelman & Shahbazi, 2012). While a conventional hash function is designed to minimize collisions (that is, different entries being mapped to the same hash key), in LSH the aim is to allow collisions among entries that are similar enough according to some metric. By relaxing the requirement for an exact match, albeit with a predictably bounded error, LSH offers a k-NN implementation whose cost grows logarithmically in the number of points and their dimensionality (Andoni & Indyk, 2008; Indyk & Motwani, 1998). This efficiency is made possible in part by partitioning the search space into small regions, so that similar entries get binned together, and in part by relying on simple features, or even randomly selected ones (Charikar, 2002), for deciding how to partition the search space.

Because it clusters items by similarity and does not require carefully designed features, LSH is a useful tool for similarity-based classification purposes, especially when dealing with large and high-dimensional data sets, for which other methods may be too computationally expensive (Grauman & Darrell, 2007; Shakhnarovich, Indyk, & Darrell, 2006). Moreover, LSH admits a natural kernelization, which leads to an improved performance compared to the non-kernelized versions (Kulis & Grauman, 2009).

4.4. Overcomplete representations

We note that incorporating nonlinearities of the type discussed thus far can potentially address a range of questions about the response nonlinearities observed in the visual system. Recordings from the early stages of the visual pathway in cats and monkeys indicate certain "non-classical" nonlinearities in the patterns of neuronal responses to stimuli (Zetzsche & Rhrbein, 2001). For instance, while stimuli whose representation is orthogonal to the synaptic weight profile of a neuron are expected not to elicit any response from it, they in fact result in the weakening of that neuron's activity. Olshausen and Field (1997) argue that this phenomenon may be due to the use of a representational scheme that is sparse and overcomplete.[13]

At some of the early stages of the mammalian cortical visual stream (e.g., in layer IV of the primary visual area V1), the number of neurons exceeds the number of projections that arrive from the lower stage. This suggests an overcomplete basis representation scheme, which would result in linear dependency among the firing of different units. Nonlinearly transforming the tuning of the units counters such dependencies, thus promoting sparseness (Olshausen & Field, 2004), which is computationally and metabolically desirable. A thorough investigation of the connections between kernels and such nonlinear transformations is, however, outside the scope of the present paper.

5. Similarity and kernels in neuroscience and psychology

In this section, we briefly review some of the questions arising in neuroscience and psychology that lead to similarity- and kernel-based formulations and approaches.

[12] Belkin and Niyogi (2003) refer to their kernel as the heat kernel, due to the analogy between their method and the solution to the diffusion differential equation.

[13] A basis set is overcomplete if the number of basis functions that comprise it exceeds the dimensionality of the representation space. A distributed representation is sparse if only a few of the units in the relevant population respond to any given stimulus.

5.1. Behavioral needs vs. computational means

In investigating the ways in which similarity – and with it kernel methods – can contribute to the understanding of behavior and the brain, a good place to start is considering the tasks that a cognitive agent faces in its day to day problem solving, corresponding to the topmost level of the Marr (1982) hierarchy (cf. Table 3). Many of these tasks – for instance, distinguishing predator from prey, edible fruit from inedible, or a promising from a hopeless conspecific match for mating – are typically formalized as classification. This may be used to recognize previously encountered members of a class or to categorize new items; the class itself can be broad (e.g., predators), or narrow (e.g., trustworthy and cooperative members of the opposite sex with good hunting skills).

At the algorithm level, a common characteristic of the various conceivable strategies that the cognitive system may resort to the service of classification (e.g., nearest neighborhood search, large margin separation, explicitly probabilistic inference) is that they all require some possibly implicit measure of similarity. For instance, while k-NN methods explicitly measure distances between data points, a perceptron's treatment of similarity is implicit in taking the inner product between the input and its weight profile. Furthermore, as discussed earlier, the continually changing conditions under which useful environmental cues are encountered (e.g., pose, illumination, climatic conditions, etc.) require that the measure of similarity used in classification reflect the animal's behavioral needs, by striving for veridical representation of relevant dimensions of variation and suppression of irrelevant ones (Edelman, 1999).

Furthermore, it is not only categorical decision making that can benefit from a nonlinear measure of similarity. Many of the tasks that an animal performs routinely fall under the rubric of ordinal decision making. Questions such as "How much …", "How fast …", "How many …", and the like need to be answered in a manner that respects order. Solutions to these problems often take the form of regression, which can also benefit from kernelization (Rosipal & Trejo, 2002).

Classification and regression are effective means of decision making when labeled examples are abundant, a situation that is rare for tasks faced by animals in the wild. When explicit supervision information is not available, the cognitive system must resort to learning from unlabeled examples, typically in the form of discovering, and later exploiting, statistical structure in the data. Algorithms devised for this purpose (e.g., PCA, Isomap, and mixture models) commonly rely on similarity among data points, which is why they can be made more effective by incorporating a kernel-like nonlinearity into their measurement of similarity (e.g., Schölkopf et al., 1997).

The examples of tasks in the right column of Table 4 and the corresponding strategies in Table 5 suggest how the many seemingly different cognitive problems lead to related similarity-based algorithms. Considering the generality of the four fundamental constraints on cognition discussed earlier in this paper, it is not surprising that solutions to most behavioral problems can benefit from a kernelized formulation, as summarized in Table 5.

5.2. Perceptual and conceptual decision making

There is a long standing debate in psychology over the mechanisms that support learning of high level concepts, e.g., birds, furniture, fruits, etc. Of the existing theories, two popular ones maintain that the category of a stimulus is decided by measuring its similarity to certain landmarks. Specifically, in the *prototype* theory, each landmark is an abstraction or prototype of multiple previously encountered exemplars (Posner & Keele, 1968; Rosch, 1978), while in the *exemplar* theory, the landmarks are individual exemplars themselves (Medin & Schaffer, 1978; Nosofsky, 1986). A third theory posits a *decision boundary* that separates the representation space into regions corresponding to the categories in question (Ashby & Gott, 1988; see Ashby & Maddox, 2005 for a survey).

The differences among these three theories notwithstanding,[14] at their core they all call for strategies that are familiar to us from the foregoing discussion. In particular, algorithms inspired by exemplar and prototype theories closely resemble the methods outlined in Section 3.2 and Table 2, according to which perceptual items are re-represented as their vectors of similarities to previously encountered samples, measured using some kernel. The resultant representation is often good enough to support effective decision making by searching the near neighborhood of the input (cf. Section 4.3). Likewise, the decision boundary approach can be implemented by constructing linear or nonlinear decision surfaces similar to perceptrons or large-margin classifiers.

Mechanisms similar to those involved in the learning of concepts can also support perceptual decision making. Indeed, insofar as both situations involve classification, they give rise to related computational problems. Consequently, here too both the reliance on similarity to landmarks and the partitioning of the input space by a decision surface are good candidate solutions. In fact, the elementary computations performed "natively" by cortical pyramidal neurons are well-suited for implementing either strategy. In perception, for instance, one of these computations is the construction of a graded receptive field that is tuned to a "best" stimulus that serves as a landmark (Edelman, 2008, p.42; cf. Section 3.6). At the same time, a neuron that estimates and then thresholds the projection of its input vector onto the vector of its synaptic weights (Edelman, 2008, p.57) effectively draws a decision surface in its input space, just like a perceptron or a Support Vector Machine.

The dichotomy between landmark and decision boundary computations may be less strict than previously thought. Recent advances in machine learning suggest that remarkable classification performance can be achieved by combining the merits of both these strategies. In a deep convolutional network (Krizhevsky, Sutskever, & Hinton, 2012), artificial neurons arranged in a cascade of many layers are trained to be selective for progressively more invariant features of the input. Following training, each convolutional unit estimates the similarity between its immediate input and the learned feature; it then applies a nonlinear transformation and reports the outcome to the units downstream from it (see Fig. 3 and the accompanying discussion at the end of Section 5.3). After several such stages, the results are fed into a multilayer perceptron that classifies using a learned decision boundary (e.g., Szegedy et al., 2015). In other words, "[the] present-day DCNs (Deep Convolutional Networks) can be exactly equivalent to a hierarchy of kernel machines with pooling and non-pooling layers" (Anselmi, Rosasco, & Poggio, 2015).

5.3. Behavioral findings and the brain angle

In light of the discussion of kernels and linear separability offered earlier, it is worth asking whether or not subjects perform better in tasks where the stimuli classes are linearly separable in the given (or in a readily computable) representation space. Researchers in visual psychophysics have for some years been exploring the effects of linear separability of simple stimuli, defined usually in a low-dimensional parameter space. For instance, Vighneshvel and Arun (2013) used line segments differing only in their

[14] Another popular approach, *rule-based*, is less concerned with natural kinds and more with conventional categories, e.g., "uncle" or "bank teller".

Table 3
A hierarchy of tasks arising in visual object and scene processing.
Source: Reproduced from Edelman and Shahbazi (2012).

Type of task	*What needs to be done*	*What it takes*
Recognition	Dealing with novel views of shapes	Tolerance to extraneous factors (pose, illumination, etc.)
Categorization	Dealing with novel instances of known categories	Tolerance to within-category differences
Open-ended representation	Dealing with shapes that differ from familiar categories	Representing a novel shape without necessarily categorizing it
Structural analysis	Reasoning about (i) the arrangement of parts in an object; (ii) the arrangement of objects in a scene	Explicit coding of parts & relationships of objects and scenes

Table 4
A non-exhaustive list of tasks that can help an animal survive (left column), and examples of situations in which they play out (right column).

Means of survival	*Examples*
Deciding on an appropriate response to a novel stimulus	"Is this food?" "Is this a dangerous animal?" "Can I outrun this predator?" "How much water do I need for this trip?"
Veridical representation	Judging the similarity of a red apple to a green apple Judging the similarity of a red apple to a red flower
Dealing with noise and confounding factors	Detecting a lion's roar from a distance on a windy day Telling apart a dog from a wolf
dealing with ambiguity and missing information	Recognizing prey in the fog Recognizing an occluded pig by its tail
Generalizing learned skills to new tasks	Learning to hunt boar can help better hunt deer Learning tree climbing can help rock climbing Figuring out what a ripe cherry looks like can help figure out what a ripe apricot looks like

Table 5
A non-exhaustive list of visual tasks that can help an animal survive (left column), possible ways these tasks can be undertaken (middle column), and the kernel-based machine learning techniques implementing them (right column).

Means of survival	*Possible strategy*	*Kernel based ML technique*
Deciding on an appropriate response to a novel stimulus	Judge similarity to familiar examples	k-NN with kernel metric
	Judge similarity to random examples	Kernel re-representation $\mathcal{T}(\cdot)$, LSH
	Find a decision boundary based on previous examples	SVM, RBF networks
	Discover and exploit structure within collected examples	Isomap, kPCA, spectral clustering
	Quantify output in terms of input	Regression, Gaussian processes
Veridical representation	Preserve pairwise distances	MDS with kernel metric
Dealing with noise and confounding factors	Allow for variance	Regularization
Dealing with ambiguity and missing information	Use co-occurring information,	Explicit coding of structure
	Top-down processing	Generative models — not kernel based (but see Section 3.5)
Generalizing learned skills to new tasks	Domain adaptation and transfer of learning	Hierarchical mixture models — not kernel based, deep convolutional networks — Implicitly kernel based

orientation as stimuli in a visual search task. In this setting, the task of finding a segment tilted at 0° among 20° and 40° distractors is linearly separable, whereas the task of finding a segment tilted at 20° among 0° and 40° distractors is not. The results were described as "refuting the widely held belief that linear separability influences visual search". We remark, however, that natural perceptual categorization problems never reside in such simple spaces. A more realistic approach should vary the layout of stimuli parametrically in some appropriately complex "hidden" space (Cutzu & Edelman, 1996; Op de Beeck, Wagemans, & Vogels, 2001; cf. Blair & Homa, 2001). Indeed, in higher-level tasks, linear separability may facilitate learning (Blair & Homa, 2001; Medin & Schwanenflugel, 1981; Wattenmaker, Dewey, Murphy, & Medin, 1986).

The issue of linear separability, which figures prominently in the machine learning literature on kernels, has become a major research question in computational modeling of the visual processing in the brain (DiCarlo & Cox, 2007; DiCarlo, Zoccolan, & Rust, 2012; Pagan et al., 2013). The chief rationale offered for linearization is improved invariance: "At higher stages of visual processing, neurons tend to maintain their selectivity for objects across changes in view; this translates to manifolds that are more flat and separated" (DiCarlo et al., 2012). Linear separability is also desirable on the grounds of complexity. As discussed in Section 2.2.3, simpler decision criteria make for better generalization. Recall, however, that the linear separation attained by kernels in ϕ-space does not by itself guarantee simplicity, and thereby generalizability: were it not for the tight grip of the regularizing term, it could easily result in a disastrously overfitted solution.

At the same time, we may observe that the "untangled" neural representations do not reside in the implicit ϕ-space. Rather, they occupy a very explicit space comprised of signals that are re-represented by way of a certain "flattening" transformation, which may or may not relate to the implicit feature map (cf. Section 3.3.1).

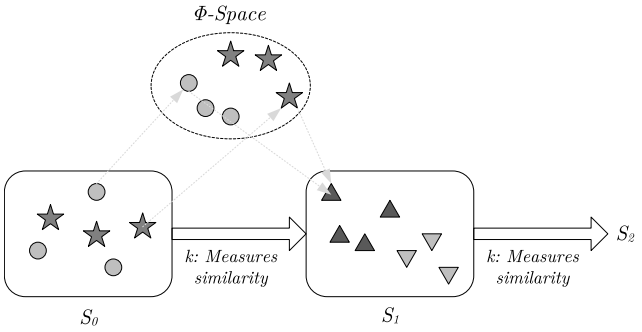

Fig. 3. Linear separability in neural representations. Raw sensory measurements reside in S_0, where categories are typically not linearly separable. Through the application of a Gaussian kernel that measures their similarities, they may become linearly separable in the new space S_1. However, this linear separability would be different from the one corresponding to the ϕ-space of the Gaussian. While each point in the ϕ-space corresponds to a sensory measurement, each point in S_1 denotes the similarity between two sensory measurements. Finally, another application of the kernel to S_1 yields S_2, where second-order similarities are represented. See text for details.

For a concrete example, suppose that the nervous system uses the Gaussian kernel to remap the sensory signal residing in S_0 into a new space, S_1, which is better suited for the perceptual needs of the organism. Now, while each point in the ϕ-space (corresponding to the Gaussian kernel) stands for an individual point from S_0, each point in S_1 corresponds to the *similarity between two points* from S_0, as measured by the Gaussian kernel. Consequently, when the shape of decision boundaries is discussed, one must consider those in the S_0, ϕ, and S_1 spaces, which may or may not be hyperplanes (Fig. 3). Furthermore, representations in S_2, computed as before via a Gaussian kernel but with S_1 as input, would correspond to the similarity of similarities. Such higher order measures of similarity, particularly when combined with hierarchical abstraction to reflect the similarity of parts and wholes, have been shown effective in shape and string matching (Egozi, Keller, & Guterman, 2010; On & Lee, 2011).

6. Summary and conclusions

Cognitive agents whose survival and flourishing depend on making effective decisions in uncertain environments face a number of fundamental challenges that stem from the basic computational principles that underlie cognition. In this paper, we have discussed four fundamental constraints that apply to cognitive information processing, having to do, respectively, with *Measurement* (Section 2.1), *Similarity* (Section 2.2.1), *Dimensionality* (Section 2.2.2), and *Complexity* (Section 2.2.3). With the exception of *Measurement*, these challenges have been discussed previously in the literature. Here, we attempted to pull together various aspects of these discussions, using mathematical concepts from the theory of Reproducing Kernel Hilbert Spaces to relate the existing ideas and approaches to one another.

The main insight from this attempt is that the "kernel trick", which was originally conceived as a means of bypassing expensive computations in high-dimensional spaces, serves to simultaneously address all four constraints. By acting as a measure of *Similarity*, kernels offer solutions that are of low *Complexity* and reside in spaces of lower *Dimensionality* than the original data space. Furthermore, as long as at each stage of processing the required information about earlier stages is limited to comparisons and similarities, relying on kernel-like computations obviates the need for a system to have a detailed knowledge of its own measurement "front end", as per the *Measurement* constraint.

These observations are consistent with findings from a range of neurobiological studies and behavioral experiments. At the neural level, estimating the similarity of the stimulus to reference or landmark items is among the most common types of operation. Furthermore, electrophysiological recordings and brain imaging support the notion that low-complexity representations, such as those afforded by kernel-like computations, emerge in later processing stages in the brain. In a separate project, we are using fMRI and multivoxel pattern analysis to study the linear separability of cortical representations in humans. Much work, however, remains to be done. A particularly interesting direction to explore is the possibility of extending similarity- and kernel-based methods to sequential symbolic data (e.g., Clark, Costa Florêncio, & Watkins, 2006; Clark, Florêncio, Watkins, & Serayet, 2006; Lodhi, Shawe-Taylor, Cristianini, & Watkins, 2001) and from stimulus/response mapping to sequential behaviors (Edelman, 2015; Kolodny & Edelman, 2015).

References

Aizerman, A., Braverman, E. M., & Rozoner, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, *25*, 821–837.

Andoni, A., & Indyk, P. (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, *51*, 117–122.

Anselmi, F., Leibo, J. Z., Rosasco, L., Mutch, J., Tacchetti, A., & Poggio, T. A. (2014). *Unsupervised learning of invariant representations with low sample complexity: the magic of sensory cortex or a new framework for machine learning? Technical report*. Center for Brains, Minds and Machines, MIT.

Anselmi, F., Rosasco, L., & Poggio, C. T. T. (2015). *Deep convolutional networks are hierarchical kernel machines. Technical report*. Cambridge, MA, USA: Center for Brains, Minds and Machines, McGovern Institute, Massachusetts Institute of Technology.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., & Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, *45*, 891–923.

Ashby, F. G., & Gott, R. E. (1988). Decision rules in the perception and categorization of multidimensional stimuli. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*(1), 33.

Ashby, F. G., & Maddox, W. T. (2005). Human category learning. *Annual Review of Psychology*, *56*, 149–178.

Ashby, F. G., & Perrin, N. A. (1988). Toward a unified theory of similarity and recognition. *Psychological Review*, *95*(1), 124–150.

Balcan, M.-F., Blum, A., & Vempala, S. (2006). Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, *65*, 79–94.

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

Bellman, R. E. (1961). *Adaptive control processes*. Princeton, NJ: Princeton University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.

Blair, M., & Homa, D. (2001). Expanding the search for a linear separability constraint on category learning. *Memory & Cognition*, *29*, 1153–1164.

Blum, A. (2006). Random projection, margins, kernels, and feature-selection. In C. Saunders, M. Grobelnik, S. Gunn, & J. Shawe-Taylor (Eds.), *Lecture notes in computer science*: *3940*. *Subspace, latent structure and feature selection* (pp. 52–68). Springer.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1989). Learnability and the Vapnik–Chervonenkis dimension. *Journal of the ACM*, *36*, 929–965.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). ACM.

Burges, C. J. C. (1998). Geometry and invariance in kernel based methods. In B. Schoelkopf, C. J. C. Burges, & A. Smola (Eds.), *Advances in kernel methods — support vector learning*. Cambridge, MA: MIT Press.

Charikar, M. S. (2002). Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on theory of computing* (pp. 380–388). ACM.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*, 287–291.

Cho, Y., & Saul, L. K. (2009). Kernel methods for deep learning. In *Advances in neural information processing systems* (pp. 342–350).

Clark, A., Costa Florêncio, C., & Watkins, C. (2006). Languages as hyperplanes: grammatical inference with string kernels. In *Proceedings of the European conference on machine learning, ECML* (pp. 90–101).

Clark, A., Florêncio, C.C., Watkins, C., & Serayet, M. (2006). Planar languages and learnability. In *Proceedings of the 8th international colloquium on grammatical inference, ICGI* (pp. 148–160).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, *14*, 326–334.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*, *IT-13*, 21–27.

Cutzu, F., & Edelman, S. (1996). Faithful representation of similarities among three-dimensional shapes in human vision. *Proceedings of the National Academy of Sciences*, *93*, 12046–12050.

Daugman, J. G. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, *20*, 847–856.

Dayan, P., & Abbott, L. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Taylor & Francis.

DiCarlo, J. J., & Cox, D. D. (2007). Untangling invariant object recognition. *Trends in Cognitive Sciences*, *11*, 333–341.

DiCarlo, J. J., Zoccolan, D., & Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, *73*, 415–434.

Edelman, S. (1995). Representation, similarity, and the chorus of prototypes. *Minds and Machines*, *5*, 45–68.

Edelman, S. (1998). Representation is representation of similarity. *Behavioral and Brain Sciences*, *21*, 449–498.

Edelman, S. (1999). *Representation and recognition in vision*. Cambridge, MA: MIT Press.

Edelman, S. (2008). *Computing the mind: how the mind really works*. New York, NY: Oxford University Press.

Edelman, S. (2015). The minority report: some common assumptions to reconsider in the modeling of the brain and behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, *27*.

Edelman, S., & Duvdevani-Bar, S. (1997). Similarity-based viewspace interpolation and the categorization of 3D objects. In *Proc. similarity and categorization workshop* (pp. 75–81). Dept. of AI, University of Edinburgh.

Edelman, S., & Intrator, N. (1997). Learning as extraction of low-dimensional representations. In D. Medin, R. Goldstone, & P. Schyns (Eds.), *Mechanisms of perceptual learning* (pp. 353–380). Academic Press.

Edelman, S., & Intrator, N. (2002). Models of perceptual learning. In M. Fahle, & T. Poggio (Eds.), *Perceptual learning* (pp. 337–353). MIT Press.

Edelman, S., & Shahbazi, R. (2012). Renewing the respect for similarity. *Frontiers in Computational Neuroscience*, *6*, 45.

Egozi, A., Keller, Y., & Guterman, H. (2010). Improving shape retrieval by spectral matching and meta similarity. *IEEE Transactions on Image Processing*, *19*, 1319–1327.

Eigensatz, M., & Pauly, M. (2006). *Insights into the geometry of the gaussian kernel and an application in geometric modeling*. (Ph. D. thesis), Zurich: Swiss Federal Institute of Technology.

Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization networks and support vector machines. *Advances in Computational Mathematics*, *13*, 1–50.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 111–132.

Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*, 277–296.

Fukumizu, K., Song, L., & Gretton, A. (2011). Kernel Bayes' rule. In *NIPS* (pp. 1737–1745).

Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers*, *93*, 429–441.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*, 1–58.

Goldstone, R. L. (1994). The role of similarity in categorization: providing a groundwork. *Cognition*, *52*, 125–157.

Goodman, N. (1972). *Seven strictures on similarity*. Indianapolis: Bobbs Merill.

Grauman, K., & Darrell, T. (2007). Pyramid match hashing: Sub-linear time indexing over partial correspondences. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.

Ham, J., Lee, D. D., Mika, S., & Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on machine learning* (p. 47). ACM.

Hegde, C., Sankaranarayanan, A.C., & Baraniuk, R.G. (2012). Learning manifolds in the wild. Preprint.

Howley, T., & Madden, M. G. (2005). The genetic kernel support vector machine: Description and evaluation. *Artificial Intelligence Review*, *24*, 379–395.

Hume, D. (1748). An enquiry concerning human understanding. Available online at http://eserver.org/18th/hume-enquiry.html.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on theory of computing* (pp. 604–613). ACM.

Intrator, N., & Cooper, L. N. (1992). Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks*, *5*, 3–17.

Jäkel, F., Schölkopf, B., & Wichmann, F. A. (2007). A tutorial on kernel methods for categorization. *Journal of Mathematical Psychology*, *51*, 343–358.

Jäkel, F., Schölkopf, B., & Wichmann, F. A. (2008). Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin & Review*, *15*, 256–271.

Jäkel, F., Schölkopf, B., & Wichmann, F. A. (2009). Does cognitive science need kernels? *Trends in Cognitive Sciences*, *13*, 381–388.

Johnson, W. B., & Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, *26*, 189–206.

Jolliffe, I. T. (1986). *Principal component analysis, vol. 487*. New York, NY: Springer.

Jurica, P., Gepshtein, S., Tyukin, I., & van Leeuwen, C. (2013). Sensory optimization by stochastic tuning. *Psychological Review*, *120*, 798–816.

Kang, K., Shapley, R. M., & Sompolinsky, H. (2004). Information tuning of populations of neurons in primary visual cortex. *The Journal of Neuroscience*, *24*, 3726–3735.

Klare, B., & Jain, A. (2012). Heterogeneous face recognition using kernel prototype similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*, 1410–1422.

Kolodny, O., & Edelman, S. (2015). The problem of multimodal concurrent serial order in behavior. *Neuroscience and Biobehavioral Reviews*, *56*, 252–265.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In *Proc. advances in neural information processing systems*, vol. 25 (pp. 1097–1105).

Kulis, B., & Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *Proc. 12th international conference on computer vision, ICCV* (pp. 2130–2137).

Lando, M., & Edelman, S. (1995). Receptive field spaces and class-based generalization from a single view in face recognition. *Network*, *6*, 551–576.

Lodhi, H., Shawe-Taylor, J., Cristianini, N., & Watkins, C. J. (2001). Text classification using string kernels. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems 13* (pp. 563–569). MIT Press.

Lowe, D. G. (1995). Similarity metric learning for a variable-kernel classifier. *Neural Computation*, *7*(1), 72–85.

Marr, D. (1982). *Vision*. San Francisco, CA: W. H. Freeman.

Medin, D. L., Goldstone, R. L., & Gentner, D. (1993). Respects for similarity. *Psychological Review*, *100*, 254–278.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.

Medin, D. L., & Schwanenflugel, P. J. (1981). Linear separability in classification learning. *Journal of Experimental Psychology: Human Learning and Memory*, *7*, 355–368.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, Series A, CCIX*, 415–446.

Miller, J. P., Jacobs, G. A., & Theunissen, F. E. (1991). Representation of sensory information in the cricket cercal sensory system. I. Response properties of the primary interneurons. *Journal of Neurophysiology*, *66*, 1680–1689.

Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, *115*, 39–61.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, *37*(23), 3311–3325.

Olshausen, B. A., & Field, D. J. (2004). What is the other 85% of v1 doing. *Problems in Systems Neuroscience*, *4*(5), 182–211.

On, B.-W., & Lee, I. (2011). Meta similarity. *Applied Intelligence*, *35*, 359–374.

Op de Beeck, H., Wagemans, J., & Vogels, R. (2001). Inferotemporal neurons represent low-dimensional configurations of parameterized shapes. *Nature Neuroscience*, *4*, 1244–1252.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. *Psychological Review*, *56*, 132–143.

Pagan, M., Urban, L. S., Wohl, M. P., & Rust, N. C. (2013). Signals in inferotemporal and perirhinal cortex suggest an untangling of visual target information. *Nature Neuroscience*, *16*, 1132–1139.

Pearl, J. (2009). Causal inference in statistics: an overview. *Statistics Surveys*, *3*, 96–146.

Pękalska, E., Duin, R. P. W., & Paclík, P. (2006). Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, *39*, 189–208.

Poggio, T., & Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature*, *343*, 263–266.

Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, *77*, 353–363.

Pouget, A., & Sejnowski, T. J. (2001). Simulating a lesion in a basis function model of spatial representations: comparison with hemineglect. *Psychological Review*, *108*, 653–673.

Resnikoff, H. L. (1989). *The illusion of reality*. New York, NY: Springer.

Rissanen, J. (1987). Minimum description length principle. In S. Kotz, & N. L. Johnson (Eds.), *Encyclopedia of statistic sciences, vol. 5* (pp. 523–527). J. Wiley and Sons.

Rosch, E. (1978). Principles of categorization. In E. Rosch, & B. Lloyd (Eds.), *Cognition and categorization* (pp. 27–48). Hillsdale, NJ: Erlbaum.

Rose, D. (1979). Mechanisms underlying the receptive field properties of neurons in cat visual cortex. *Vision Research*, *19*, 533–544.

Rosipal, R., & Trejo, L. J. (2002). Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research*, *2*, 97–123.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Schleif, F.-M., & Tino, P. (2015). Indefinite proximity learning: a review. *Neural Computation*, *27*, 2039–2096.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K.-R. (1997). Kernel principal component analysis. In *ICANN'97* (pp. 583–588). Springer.

Shakhnarovich, G., Indyk, P., & Darrell, T. (2006). *Nearest-neighbor methods in learning and vision: theory and practice*. MIT Press.

Shepard, R. N. (1968). Cognitive psychology: A review of the book by U. Neisser. *American Journal of Psychology*, *81*, 285–289.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, *237*, 1317–1323.

Shepard, R. N., & Chipman, S. (1970). Second-order isomorphism of internal representations: Shapes of states. *Cognitive Psychology*, *1*, 1–17.

Smola, A., Gretton, A., Song, L., & Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *Algorithmic learning theory* (pp. 13–31). Springer.

Solomonoff, R. J. (1964). A formal theory of inductive inference, parts A and B. *Information and Control*, *7*, 1–22. 224–254.

Song, L., Huang, J., Smola, A., & Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th annual international conference on machine learning* (pp. 961–968). ACM.

Steyvers, M., Tenenbaum, J. B., Wagenmakers, E. J., & Blum, B. (2003). Inferring causal networks from observations and interventions. *Cognitive Science*, *27*, 453–489.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D. et al. (2015). Going deeper with convolutions. In *The IEEE conference on computer vision and pattern recognition, CVPR*.

Tenenbaum, J. B. (1998). Mapping a manifold of perceptual observations. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.), *Advances in neural information processing, vol. 10* (pp. 682–688). MIT Press.

Tenenbaum, J. B., & Griffiths, T. L. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, *24*, 629–641.

Theunissen, F., Roddey, J. C., Stufflebeam, S., Clague, H., & Miller, J. P. (1996). Information theoretic analysis of dynamical encoding by four identified primary sensory interneurons in the cricket cercal system. *Journal of Neurophysiology*, *75*, 1345–1364.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327–352.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*, 988–999.

Vighneshvel, T., & Arun, S. P. (2013). Does linear separability really matter? Complex visual search is explained by simple search. *Journal of Vision*, *13*, 10.

Wattenmaker, W. D., Dewey, G. I., Murphy, T. D., & Medin, D. L. (1986). Linear separability and concept learning: Context, relational properties, and concept naturalness. *Cognitive Psychology*, *18*, 158–194.

Wilson, R. C., & Zhu, P. (2008). A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, *41*, 2833–2841.

Zetzsche, C., & Rhrbein, F. (2001). Nonlinear and extra-classical receptive field properties and the statistics of natural scenes. *Network: Computation in Neural Systems*, *12*(3), 331–350.