Learning Transformation Groups and their Invariants

Taco Cohen (10137408)
Supervisor: prof. Max Welling
42 EC

University of Amsterdam

A thesis submitted in conformity with the requirements for the degree of

MSc. in Artificial Intelligence

2013

Acknowledgements

I would like to thank Max Welling for his supervision and for the freedom and encouragement he gave me during the last half year. Despite his busy schedule, he has always made time to help me when needed.

I would also like to thank Leo Dorst for teaching me many things and for the many interesting conversations and discussions. Leo has been a great example for me, and the best teacher I have had.

Thanks to my committee Max, Leo and Joris Mooij for agreeing to read my thesis on such short notice.

And finally, I wish to thank my parents for their never-ending support.

Abstract

A fundamental problem in vision is that of invariance: how objects are perceived to be essentially the same despite having undergone various transformations. When it is known a priori to which transformations a representation of the image must be invariant, one can try to construct such an invariant representation by analysis. A potentially more generic solution is to build systems that *learn* how to construct an invariant representation by looking at transformed examples. Such a system can be largely agnostic to the representation of the input and the kind of transformation to which invariance is required, so that it can be applied to many kinds of data.

We approach this problem from a dynamical perspective, by modeling the transformation group itself. This allows us to not only build an invariant representation, but also perform pose estimation and understand motion. We derive an elegant algorithm that learns a so-called maximal toroidal Lie group from pairs of inputs, where each pair is related by an unkown transformation from the group. Given one or more pairs of input images, the model provides a full posterior over elements of the group, describing the inherent uncertainty in the transformation estimate. As it turns out, the ambiguity in the estimate of the transformation from one image to itself (the symmetry of the image with respect to the learned group) provides a maximally informative invariant representation of that image.

Contents

Chapter 1

Introduction

> *No man ever steps in the same river twice, for it's not the same river and he's not the same man.* – Heraclitus

Not one image captured by our eyes is exactly the same as another one captured before it, and neither is the visual cortex that interprets it. However, despite this fact and despite Heraclitus' philosophical musings, most of us perceive the river and our selves as having some degree of permanence. In other words, we have the ability to recognize (or confabulate) some invariant essence of a percept, even though the percept is always changing and never exactly the same.

Which changes are considered essential and which ones are considered irrelevant depends on context, but in order to make the distinction at all we need to reduce the holistic percept into distinct properties. Consider figure 1.1. We say that the two letters are essentially the same despite a difference in font, and likewise we recognize the same face with different facial expressions. The properties we use in our description (letter, font, identity, expression) are not explicitly present in the raw representation of the image on our retina, yet they appear completely natural to us. Somehow, we have decomposed the holistic percept (the entire collection of light intensity values) into constituent properties so that we may assess the degree and kind of similarity in a meaningful way.

The previous paragraphs could have been the introduction to a thesis in philosophy, psychology, psychophysics or cognitive science, but here we endeavor to understand this phenomenon from a mathematical perspective, and to devise effective algorithms that learn to see the world in a similar way. That is, we aim to devise learning algorithms that can untangle the different factors of variation, so that observations can be represented in terms of distinct properties.

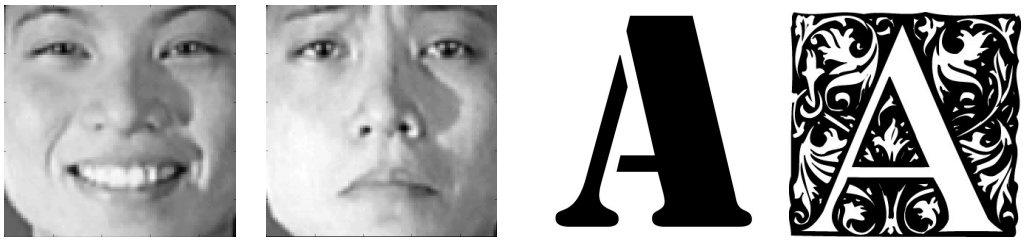

Figure 1.1: The same, yet different.

To make the vague notions of "essentially equivalent despite superficial changes" and "separation into individual properties" precise, we apply two marvelous insights of mathematician Felix Klein (1849-1925) and mathematician/physicist Hermann Weyl (1885-1955). Both ideas are deceptively simple, to the point of seeming almost trivial, but nevertheless have far-reaching consequences and have had a massive impact on the course of mathematics and natural science.

The principle put forward by Klein in his famous Erlangen Program [Klein, 1893] was motivated by problems in geometry. In traditional Euclidean geometry, one considers two figures equivalent if they are related by a Euclidean transformation (a rigid body motion). As we will see in section 1.1, the set of all Euclidean transformations is one example of a type of mathematical object known as a *group*. Groups are the mathematicians' way to describe symmetries, and the Euclidean group describes the symmetries of Euclidean space (this usage of the word "symmetry" will be clarified later). The key idea is this: *instead of taking the Euclidean group as our symmetry group, we can take any group and call it a symmetry group.* Every choice of symmetry group will induce a different notion of equivalence between figures, and will thus result in a different kind of geometry. According to Klein, the *objects of interest* in a particular geometry are the symmetry transformations and any quantity left invariant by these transformations. What Klein showed is that most geometries that mathematicians had cared to study (i.e. those that they had found interesting for some reason) could all be obtained by the right choice of symmetry group. Furthermore, the symmetry groups are related to each other in various ways, and this clarified the relations among the different geometries.

Notice how Klein has given a precise definition of what is to be considered interesting: invariants are worthy of study, other quantities and structures are not. Although paternalistic to the mathematician, such a principle, if representative of our intuitions about what constitutes interesting structure, is invaluable to those who study natural and artificial intelligence.

Applied to the problem of learning representations of data, Klein's principle tells us that the interesting quantities – those that we want to represent explicitly – are invariant to some *group* of transformations. We can be a little more specific, too: we want a *maximal* invariant. Taken together, the features should identify the input up to symmetry transformations [Soatto, 2009].

A maximal invariant could be represented in many ways, because any invertible mapping of a set of features contains the same information and absolute invariance properties. As a further constraint then, we will require that each feature has an invariant meaning that is independent of the other features. But how can we formalize such a vague notion? To approach this problem we employ a basic principle first expounded by Hermann Weyl [Weyl, 1939] and later seen to be of use in computer vision by Kenichi Kanatani [Kanatani, 1990], that states that:

Weyl's principle (intuitive version)
A set of transformed measurement values may rightly be called a single physical observable with an invariant meaning *if and only if* the transformations from a symmetry group do not mix this observable with other observables.

In other words, if two sets of measurements, made at time t and $t+1$, are related by a symmetry transformation, the state of one particular observable (which is a function of the measurements) at time $t+1$ should be completely determined by its state at time t, and the symmetry transformation that was applied. Information about other observables should not be necessary to determine its state, because representing distinct properties of the observed phenomenon, they transform independently. This is not to say that different properties cannot interact, but if the transformation is just "a change in perspective" or "a renaming of points in the measurement space" (i.e. it is a symmetry transformation), then the real phenomenon that is being observed is the same (hence no interactions), and we should be able to tell the next state from the current one for each observable independently. The precise definition of Weyl's principle stated in terms of the irreducible reduction of a group representation is presented in section 3.3, and a rudimentary introduction to group theory required to understand it is given in chapter 2.

From these basic principles and a small number of additional assumptions, we derive a representation learning algorithm which we call Toroidal Subgroup Analysis. This algorithm learns from a stream of transforming data to represent that data in an invariant-equivariant way. That is, the inputs are re-represented in terms of the

learned features (Weyl's observables), from which an absolutely invariant representation can easily be computed. At the same time, due to the reduction into independent observables, transformations in the data can be described explicitly and compactly in a disjunct pathway. While distinctly geometrical, the algorithm is expressed in the language of probabilistic graphical models, enabling us to make sound probabilistic inferences about geometrical quantities. When trained on randomly shifting images, the algorithm learns to perform a discrete Fourier transform of the signal; other types of transformations result in analogous computations. We then extend the algorithm to learn subgroups of the learned toroidal group. Using the theory of Lie algebras, we find that these subgroups correspond to linear subspaces of a space of latent variables of the model. The resulting representation of transformations in the learned subgroup consists of a small number of parameters that are easily interpretable.

We will use examples from visual perception as a way to motivate and explain the TSA algorithm, and the experiments we perform are all done on image patches. However, we will argue that the principles we adopt are useful for structuring acquired knowledge in general, because they facilitate generalization and aim at objectivity. In the following sections of this chapter, we explain what it is exactly that we mean by "symmetry", "preservation" and "group", and show how these notions appear to play a role in structuring human knowledge. Then, we motivate our work on learning transformation groups and invariant representations from the perspective of machine learning. In section 1.5, we outline the contributions of this thesis.

1.1 The conception of symmetry

The usage of the word "symmetry" in this thesis may appear somewhat unusual to those who do not have a background in mathematics or physics. The image that comes to mind upon hearing this word could be one of those shown in figure 1.2. These are indeed examples of symmetry, but the meaning we intend to convey is more abstract. Before we give the abstract definition of symmetry, let us make some observations about these concrete geometric symmetries.

In each of the panels in figure 1.2, notice that there are some transformations that we can apply to the figure so that it remains unchanged. We call these transformations symmetries of the figure. Notice further that there may sometimes be "non-essential" or "superficial" aspects of the figure that *are* changed by the symmetry transformations. For example, the colored patches on the triangle are permuted by reflections and rotations, even if these transformations leave the triangle shape

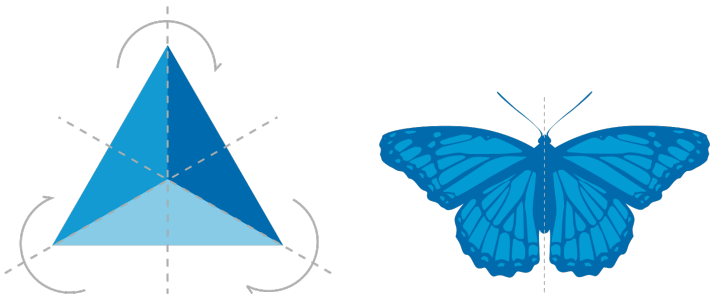

Figure 1.2: Examples of concrete (non-abstract) symmetry.

invariant. For both figures, the symmetry transformations are a subset of the Euclidean transformations of rotation, translation, or reflection (or some combination of them).

Obviously, when we have two transformations that both leave the essence of the figure unchanged, the sequential application of the two will do the same. It is this trivial observation that forms the basis of the mathematical notion of a group of transformations, from which surprisingly abundant structure emerges. Formally, a group G is a *set* of transformations[1] together with a product that composes transformations. To be a group, the set and product must satisfy the following axioms:

1. Closure: if $\mathbf{A} \in G$ and $\mathbf{B} \in G$, then $\mathbf{AB} \in G$.

2. Inverse: if $\mathbf{A} \in G$ then $\mathbf{A}^{-1} \in G$, where $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$ and $\mathbf{I}$ is the identity transformation.

Juxtaposition of two transformations is taken to mean subsequent application, which corresponds to matrix multiplication when the transformations are linear and are represented by matrices in some fixed basis. Normally, the additional axioms of associativity ($(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$ for all $\mathbf{A}, \mathbf{B}, \mathbf{C} \in G$) and identity ($\mathbf{I} \in G$, where $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$ for all $\mathbf{A} \in G$) are included in the definition, but these are superfluous when it is assumed that the elements of G are transformations.

We can read the definition of a group as a recipe for generating one: given a number of symmetries of an object, we can generate a whole group of symmetries by repeatedly multiplying elements and taking inverses, adding them to the group until we find no more new transformations. So we could say that groups are born from symmetries.

[1]The elements of a group need not be interpreted as transformations in order to study groups abstractly, but for now we will not make the distinction between abstract groups and group representations, in order to simplify the exposition.

The key idea that leads to the *abstract* notion of symmetry is the following: instead of considering the Euclidean group of transformations to be the symmetry group of space, we can take any group of transformations and call them symmetries. This means that we now have to identify all the objects that can be transformed into each other by some element of the group, even if this means that objects can be distorted in non-rigid ways. They are considered "essentially the same" with respect to the chosen group.

1.2 Structure preservation in human knowledge representations

Pure geometry is far from the only application of group theory. In mathematics more generally, but also in physics and chemistry, symmetries and the groups that describe them play an important role. While this fact is widely appreciated, it is less often recognized that more mundane propositions and non-scientific proto-theories too, are often of the form "there exists a type of change that leaves some aspect of an object invariant" or "some structure x is preserved when we do y". In the next few sections, we look at examples from vision, developmental psychology, cognitive science and physics. We do not claim that all these instances of preservation of structure must necessarily be modeled using transformation groups, but we do believe that the general concept of "preservation of structure" is fundamentally important in our perception and cognition.

1.2.1 Vision

Visual perception is the main area of application in this thesis, and it is there that the relevance of the concept of abstract symmetry is most obvious. A lot of research in computer vision is geared towards constructing or learning image descriptions that are invariant to some kind of transformation. It has been suggested that the central computational goal of the visual cortex is to learn to discard transformations [Leibo et al., 2011], both generic ones such as translations and rotations, as well as class-specific or local ones that only apply to a particular region of the input space, such as transformations of facial expressions. We subscribe to this view, but emphasize that besides invariance, equally interesting are the maps to which we build invariance themselves. In analogy to Klein's perspective on geometry, we believe that the most natural way to define static object classes in vision is to say that they are the invariants of various transformation groups that act on the visual scene.

1.2.2 Children's conception of the world

The knowledge acquired by young children can often be framed in terms of transformations and invariants. These transformations act not on the perceptual arrays directly (as in vision, for example), but on slightly more abstract mental representations. Consider the following classical experiment by developmental psychologist Jean Piaget [Piaget, 1952]. The child is shown two identical glasses, filled with the same amount of liquid. The child is asked whether the two glasses have the same amount of liquid or if the amount is different. At the ages tested, children are typically able to appreciate that the amount is the same. Then, the experimenter pours the liquid from one of the containers over to a taller, thinner glass, in plain view of the child. The child is then asked whether there is still the same amount of liquid in both glasses, to which a young child will answer "no, there is more water in the tall, thin glass".

We do not know how this sequence of events is represented in the brain of the child, but certainly the information about the height of the liquid at each point in time is represented somehow. Hence there is a transformation of this mental representation as the experiment unfolds. As the child develops, it learns to recognize that many wildly different image sequences all constitute the continuous transformation of the scene called "pouring over" and that this transformation has an invariant: the *volume* of the liquid. Piaget invented many more such conservation tasks, showing that many of the elementary concepts learned by a child (and carried into adulthood) should be viewed in terms of transformations and conserved quantities.

1.2.3 Analogy-making

An analogy is a mapping between two (more or less disparate) cognitive or perceptual domains, that preserves the structure of the source domain. For example, one can make the (somewhat naive) analogy that "the atom is like the solar system": both have a central object (the sun $\sim$ atomic nucleus) and objects orbiting it (planets $\sim$ electrons). The "analogy-function" A maps sun to nucleus and planets to electrons. The map A radically changes the features of the object: the atomic nucleus is very different from the sun in terms of size, for instance. But the *relational structure* is preserved: $\texttt{orbits}(\texttt{planet}, \texttt{sun}) \rightarrow \texttt{orbits}(A(\texttt{planet}), A(\texttt{sun}))$

We can again make an analogy to geometry. In geometry, an object is defined in terms of the *relations* between the points (these relations being distances and angles

in Euclidean geometry), and is considered equivalent (or analogous) to a transformed object if both objects have the same relational structure.

Once an analogical mapping has been established, it can be used for inference about the target domain. This process is known as analogical inference or relational generalization [Holyoak and Morrison, 2005]. As an example of analogical inference, consider the following situation: having already acquired the concept of gravity and found an analogical mapping between the solar system and the atomic nucleus, one could conjecture the existence of some force keeping electrons with the nucleus. Although this mode of reasoning can easily fail, it is nevertheless indispensable for generating plausible conjectures. It allows us to generalize in radically non-local ways; something which machines are currently incapable of.

1.2.4 Physics

In modern physics, practically all laws of nature are derived from symmetry principles. That is, one starts by assuming that the laws governing the behavior of some physical system are invariant to some group of transformations (symmetries) acting on the states of the system. From this assumption alone, one can learn a great deal about the laws themselves, for it is a strong restriction on their form. Furthermore, any continuous symmetry of a system corresponds to a conserved quantity such as mass-energy, momentum, etc – by Noether's theorem. What matters to us at present is that physics is yet another example of a knowledge domain where symmetry and invariance are the main structuring principles.

The grand success of modern physics shows that nature is indeed well described in terms of symmetries and conserved quantities, but the fact that this body of knowledge has come about may also inform us about our way of thinking. The propensity for searching for preserved, unchanging quantities can already be found in the ideas of Thales – who, according to Aristotle was the first naturalistic philosopher. He sought to explain phenomena, in particular the nature of matter, without recourse to gods and deities [Curd, 2012]. In his *Metaphysics*, Aristotle writes of Thales' ideas [Aristotle, 1924]:

> That of which all things that are consist, the first from which they come to be, the last into which they are resolved (the substance remaining, but changing in its modifications), this they say is the element and this the principle of things, and therefore they think nothing is either generated or destroyed, since **this sort of entity is always conserved** [...] Thales,

> the founder of this type of philosophy, says the principle is water
>
> (emphasis mine)

The ancient Greeks did not select such a theory from among many other alternatives because it was empirically successful in describing nature (it was not), but because this is the way we prefer to structure knowledge.

1.3 Group structure as an inductive bias

We have seen some striking examples of symmetry in human knowledge representations. At this point however, it is not quite clear *why* we see this pattern, and whether it is something we should strive to replicate in artificial systems or not. The organization of human knowledge could simply be a reflection of the (often symmetric) structure of the world we perceive, inscribed as it were on the tabula rasa that is the human mind. Or it could be that we have prior knowledge of certain symmetries or even an inductive bias towards group structure in general.

Henri Poincaré wrote on the subject:

> The object of geometry is the study of a particular "group"; but the general concept of group pre-exists in our minds, at least potentially. It is imposed on us not as a form of our sense, but as a form of our understanding.
>
> Only, from among all the possible groups, that must be chosen which will be, so to speak, the standard to which we shall refer natural phenomena.

In his book *Science and Hypothesis* [Poincaré, 1904], Poincaré argued that the concept of a group is innate, and key to reasoning itself. Only, he says, the particular group that we typically use to explain the movement of objects around us – the Euclidean group – is derived from experience. In other words, it is *learned*; chosen among many other possibilities simply because it gives a compact description of some aspects of the movie projected onto our retina, and the impressions arriving at other sensory arrays. The retina itself provides a very non-homogeneous, high-dimensional representation of the visual world, and so it is not obvious at all that a representation of the Euclidean group (of all groups) should act on this space[2]. The fact that from this signal we construct our conception of 3D space is in itself quite astonishing. According to

[2]Strictly speaking, the Euclidean group acts on the sensory arrays augmented with latent variables specifying depth and occlusion, for otherwise transformations are not invertible. We will get back to this topic later.

Poincaré, the most plausible explanation is that we have an a priori concept of a group, the details of which are filled in by experience, using our actions as a sort of supervision for the learning process. The mechanism he proposed can be summed up as follows (quoting again from [Poincaré, 1904]):

1. In the first place, we distinguish two categories of phenomena: The first involuntary, unaccompanied by muscular sensations, and attributed to external objects – they are external changes; the second, of opposite character and attributed to the movements of our own body, are internal changes.

2. We notice that certain changes of each in these categories may be corrected by a correlative change of the other category.

3. We distinguish among external changes those that have a correlative in the other category – which we call displacements; and in the same way we distinguish among the internal changes those which have a correlative in the first category.

Let us work out an example. We will use $\mathbf{x}^t$ to indicate the aggregate of perceptual measurements (pixel values, say) at some time instant t. We could model this (as is typical in machine learning) as a vector $\mathbf{x}^t \in \mathbb{R}^N$. Now we perform an experiment: we perform a small eye movement denoted g, and observe the resulting percept $\mathbf{x}^{t+1} = g\mathbf{x}^t$. We can do this over and over, yielding a data set $\{(\mathbf{x}^t, \mathbf{x}^{t+1}; g^t)\}_{t=1\ldots T}$, from which we can try to learn the functions $g\mathbf{x}^t$ for all possible g, that predict the changes in the perceptual space (which we take to be visual space in this example, but we may substitute auditory space, haptic space, etc. when considering other actions). Notice that if we have one saccade g and another saccade h, the effect of two successive saccades hg can also be achieved as one single saccade. Furthermore, for any eye movement g we can do the reverse movement g^{-1}. Eye movements are also associative and we can perform the identity movement, so if we ignore actuator limits[3] (we cannot see into the back of our head), the saccades form a group. As we will see in later chapters, the group formed by these transformations, together with the projection function that generates the retinal image from scene parameters, determine a group representation in the space of pixel values. So we see that in order to predict the effect of our actions on the world as projected onto our senses – at least in this simple example – we must learn (or at least somehow develop) a group representation.

[3]Such practicalities have not hindered physicists in the successful application of group theory to the study of crystal structures either, even though crystals are not actually infinite in extent.

There is some evidence that this function is indeed learned and not hard-wired. In a clever experiment, [Cox et al., 2005] modified the identity of visual objects during the period of transient blindness that accompanies eye movements, and showed that after a training period, this manipulation caused failures in position invariance of human subjects under normal viewing conditions. It has also been shown that this manipulation alters the neuronal representation of objects in primate cortex [Li and DiCarlo, 2008], as would be expected.

Both Poincaré and Piaget argued for an inborn preference for group structure in human cognition. The fact that abundant group structure in our sensorium is certainly an evolutionary constant supports this idea, and we have presented a very rough sketch of how such a bias could guide learning. Of course, it is not possible to establish the claim of innateness of the group concept with any degree of certainty. One step we can take is to give an existence proof: to show that such a learning algorithm can indeed work, and do useful things. Our work on Toroidal Subgroup Analysis can be seen as a small step in this direction.

1.4 Why model transformations and invariants?

The ideas relating to human cognition set forth in the previous sections motivate our work on learning transformations and invariant representations, but we realize that this motivation may not be all too compelling to the well-grounded engineer who is most interested in applications. In this section, we will describe a number of advantages of the group-theoretical approach as seen from a machine learning perspective, and show how an inductive bias towards group structure is useful.

One of the central problems in machine learning is the classification problem. The target function $f(\mathbf{x})$ that maps each point $\mathbf{x} \in \mathbb{R}^N$ to a label $c_1, \ldots, c_K$ partitions the input space $\mathbb{R}^N$ into non-overlapping sets corresponding to the labels[4]. This partition defines an equivalence relation $\sim$, as follows: $\mathbf{x} \sim \mathbf{y} \iff f(\mathbf{x}) = f(\mathbf{y})$; data points $\mathbf{x}$ and $\mathbf{y}$ are equivalent in some sense (f-equivalent) if they have the same label. Every equivalence relation, in turn, corresponds to the action of a group G: $\mathbf{x} \sim \mathbf{y} \iff \exists g \in G : \mathbf{y} = g\mathbf{x}$. It is easy to check that this is well defined and satisfies the axioms reflexivity (because $\mathbf{I} \in G$), symmetry (because $g \in G \implies g^{-1} \in G$) and transitivity (because $g, h \in G \implies gh \in G$) making $\sim$ an equivalence relation. So we see that a target function has a symmetry group that consists of all invertible

[4]Label noise is easily incorporated in this framework by considering f to be the Bayes-optimal decision boundary derived from some probability distribution

maps that permute points within the partitions, but not between them. That is, G is defined implicitly as the largest group that satisfies $\forall g \in G, \forall \mathbf{x} : f(g\mathbf{x}) = f(\mathbf{x})$.

Is this dual perspective on label functions useful? If we can learn or in some other way obtain G, we could build an exemplar-based classifier that classifies a new data point $\mathbf{x}$ as class c_i iff $\exists g \in G : g\mathbf{x} = \mathbf{x}^{(i)}$ for an exemplar $\mathbf{x}^{(i)}$ with $f(\mathbf{x}^{(i)}) = c_i$. Alternatively, we could construct a representation $\mathbf{\Phi} : \mathbb{R}^N \to \mathbb{R}^M$ that is invariant to the group action: $\forall g \in G : \mathbf{\Phi}(g\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})$. To avoid trivial representations, we will require that data variation that cannot be explained by the symmetry group is retained: $\neg(\exists g \in G : g\mathbf{x} = \mathbf{y}) \implies \mathbf{\Phi}(\mathbf{x}) \neq \mathbf{\Phi}(\mathbf{y})$. In both examples, we can learn f from one example per class, once the symmetry group is known; a great reduction in sample complexity.

It is quite clear though, that this is just a different view of the same learning problem, and by itself does not address the fundamental difficulty of learning. The group G, defined implicitly as the set of all transformations that leave f invariant, can be uncountably infinite and may not admit a convenient parameterization. If the decision boundary is highly irregular, then its symmetry group will typically be complicated as well, and so this perspective is no magic bullet. However, as we will now discuss, learning a symmetry group instead of a label function conveys a number of advantages, and provides an integrated approach to learning about both static and dynamic aspects of the input.

First, symmetries may be shared between learning tasks. When this is the case, the group structure need only be learned once, and all tasks that share this symmetry can make use of it. Each of these learning tasks is then simplified, because all the variability in the data that is caused by the action of the group can be taken out, yielding a problem of reduced sample complexity.

There is increasing interest in learning classes without supervision. This is often done with clustering algorithms, but it is not quite clear by what principle observations should be grouped together. Due to the closure property inherent in the definition, groups form clearly defined "units", or "wholes". It is quite clear when a set of transformations form a single unit (a group), but it is not a priori clear at all when a region of the input space corresponds to a single class. So it may well be easier to learn transformation groups without supervision, and have them induce "proto-classes" (the invariants), than it is to use other heuristics to designate certain regions of the input space as belonging a single class.

A model of the dynamics is not only useful in order to induce static object classes; it is also of interest in its own right. It is quite clear that (artificial) intelligence

involves much more than just classification and regression: it involves planning, prediction and action, among many other aspects. All of these involve a dynamic model of the world. A truly scalable approach to AI must learn such a model from data.

Finally, as we will see in section 3.3, the group theoretical perspective gives us a handle on the problem of "untangling" that we discussed in the beginning of this chapter. Bengio describes its significance, [Bengio, 2013]

> Of all the challenges discussed in this paper, this [the disentangling problem] is probably the most ambitious, and success in solving it will most likely have far-reaching impact. In addition to the obvious observation that disentangling the underlying factors is almost like pre-solving any possible task relevant to the observed data, having disentangled representations would also solve other issues, such as the issue of mixing between modes.

1.5 Contributions

The main conceptual contributions of this work are a group-theoretical analysis of learning and the statement of a number of basic principles for representation learning (section 3.3). We give a formal definition of an "observable" for application in representation learning. This doctrine is not new in physics [Weyl, 1939], but has never been applied to machine learning. This principle provides a firm theoretical basis for the untangling problem whose significance we have discussed in section 1.4.

Our main technical contribution, presented in chapter 4, is the Toroidal Subgroup Analysis algorithm. It applies Weyl's principle to the toroidal subgroups of $SO(n)$, the largest compact, connected, Abelian Lie subgroups of the special orthogonal group. TSA is a probabilistic model that learns to split the input signal in an invariant and a covariant part. As a special case (the 2D translation group), the algorithm learns to perform a discrete Fourier transform, and hence provides a novel probabilistic interpretation of the discrete Fourier transform.

TSA relies on a newly derived conjugacy relation between a circularly parameterized Gaussian likelihood and a von Mises prior. Such relations greatly simplify Bayesian analysis, so this result may be of interest to researchers in circular statistics.

Finally, we use the theory of Lie algebras to derive an extension of the basic TSA model that aims to learn a low-dimensional subalgebra of a toroidal Lie algebra (chapter 5). Since all compact Abelian Lie groups are the subgroup of some maximal torus, this work (though incomplete) would complete the compact Abelian case.

Chapter 2

Mathematical Preliminaries

Throughout this thesis, we assume that the reader is familiar with linear algebra, calculus and probability theory. In this chapter, we give a quick introduction to group theory, which we expect to be less well-known in the field of machine learning. The notation, definitions and results given in this chapter will form the basis for later chapters.

2.1 Notation

Scalars x, y, z are written in lower case font, vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ in bold lower case font. Matrices and blades (to be defined shortly) are written as bold upper case roman letters $\mathbf{A}, \mathbf{B}, \mathbf{C}$. The j-th element of a vector $\mathbf{x}$ is denoted x_j, the element in row i, column j of matrix $\mathbf{A}$ is written A_{ij}. The row i of a matrix is written $\mathbf{W}_i$ while the column vector j is written as $\mathbf{W}_{\cdot j}$. We will want to select the elements corresponding to 2D, basis-aligned subspaces from vectors and matrices, which we denote as follows: we write $\mathbf{x_j} = (x_{2j}, x_{2j+1})^T$ and $\mathbf{W_j} = (\mathbf{W}_{\cdot 2j}, \mathbf{W}_{\cdot 2j+1})$. For an $N \times N$ matrix $\mathbf{W}$, $\mathbf{W_j}$ is an $N \times 2$ matrix. Elements of abstract groups G are denoted by lowercase letters $g, h, \ldots$, as opposed to concrete matrix or versor representations $\mathbf{A}, \mathbf{B}$ of these elements.

2.2 Group theory

In this section we introduce some basic group theoretical concepts that we will need later on. This introduction is meant to be easy to understand for those not familiar with advanced mathematics and as such we do not give proofs (for they typically make it harder to see the forest for the trees). For the same reasons, we do not try to make our definitions as general as possible and we do not go far beyond what we will

need directly in the derivation and motivation of the Toroidal Subgroup Analysis algorithm, so this is far from a complete introduction to group theory. Readers interested in a more comprehensive treatment are referred to [Weyl, 1939] (a classic but difficult treatment of Lie groups), [J.P. Elliot, 1985] (applications in physics, somewhat more accessible), [Kanatani, 1990] (applications in computer vision), [Kondor, 2008] (applications in machine learning).

2.2.1 Elementary definitions

A group G is a *set* (finite or infinite), endowed with a product on its elements that satisfies the following axioms:

1. $a(bc) = (ab)c$ for all $a, b, c \in G$ (**Associativity**)
2. $\exists e \in G$ such that $ea = ae = a$ for all $a \in G$ (**Identity**)
3. If $a \in G$ and $b \in G$, then $ab \in G$ (**Closure**)
4. If $a \in G$ then $a^{-1} \in G$, where $a^{-1}a = aa^{-1} = e$ (**Inverse**)

Note that commutativity, $ab = ba$, is *not* required. Groups that do satisfy this additional constraint are called commutative or Abelian groups. Commutative groups are much easier to analyze, so they will be the main object of study for us.

As an example of a finite group, consider the set of all permutations on n letters. If we take multiplication to mean the consecutive application of two permutations, this set forms a group. The set of all rotations in a fixed rotation plane is an example of an infinite commutative group. Other (generally non-commutative) examples include GL_n, the group of all invertible linear transformations in n dimensions, the group O_n and SO_n of orthogonal and special orthogonal transformations (these will be treated in section 2.2.4).

It is often convenient to specify a finite group in terms of its *generators*. A subset $S \subseteq G$ is said to generate G if all elements in G can be obtained by repeated multiplication and inversion of the generators. It is often revealing to display a finite group as a graph where each group element is shown as a node and each edge represents the multiplication by a generator. Such a graph, displayed in figure 2.1, is known as a Cayley graph.

A *subgroup* H of G is a subset of G that is also a group. That is, the subset H must satisfy the group axioms defined above for it to be a subgroup. In figure 2.1, we can spot a subgroup by considering those sets of nodes that include the identity

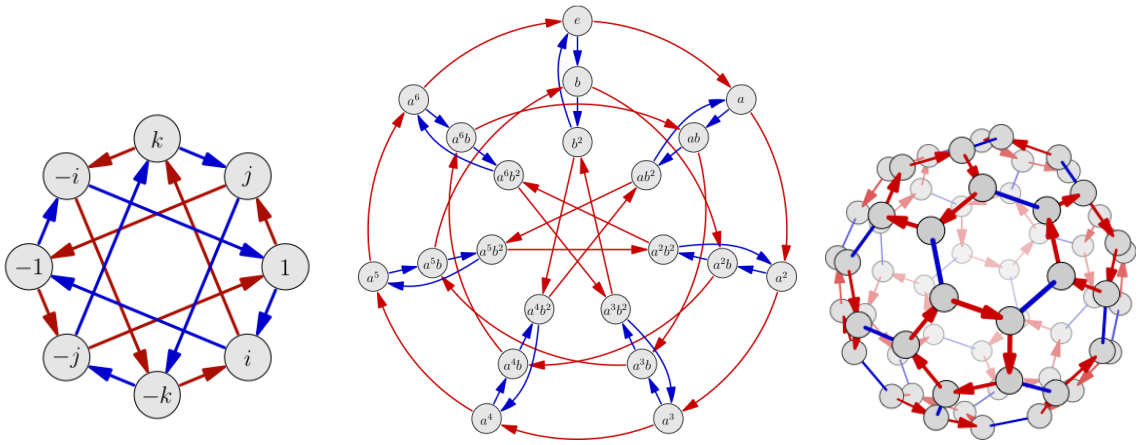

Figure 2.1: Cayley graphs of the Quaternion group Q_4 (left), a group of order 21 (middle) and the alternating group A_5 describing the symmetries of the icosahedron (right). Nodes represent the elements of the group. Arrows represent the left multiplication by a generator, such as i (blue arrow) and j (red arrow) in Q_4. Figures courtesy of Nathan Carter, reproduced with permission [Nathan Carter, 2009].

and are connected by only one color of edges. For example, the sets $\{1, i, -1, -i\}$ and $\{1, j, -1, -j\}$ are subgroups of the group of quaternions shown in figure 2.1.

A group homomorphism is a mapping $\phi : G_1 \to G_2$ that preserves the group structure. More precisely, for any $a, b, c \in G_1$ satisfying $ab = c$ it must be the case that $\phi(a)\phi(b) = \phi(c)$. Notice that the product ab is performed according to the multiplication defined for G_1, while the product $\phi(a)\phi(b)$ is performed according to the multiplication in G_2. The determinant function is an example of a homomorphism from the general linear group (which can be represented by the set of invertible $n \times n$ matrices) to the group of real numbers (excluding zero) under multiplication: for $a, b, c \in GL_n$, we have $ab = c \implies \det(a)\det(b) = \det(c)$. Geometrically, this says that the volume expansions effected by two subsequently applied transformations multiply. Notice how structure has been preserved in the sense that no equation involving only products of group elements can be made to fail by applying the homomorphism, while nevertheless we have lost something because $\phi = \det$ is *surjective*: many matrices have the same determinant.

When a homomorphism is invertible (bijective), it is called an ismorphism. In this case, the groups have exactly the same multiplication structure. This is all the structure that is intrinsic to the group, so we may consider isomorphic groups to be equivalent. Still, it may take some work to see that two groups are indeed isomorphic. For example, the symmetry groups of the cube and the octahedron

(obtained by associating each vertex of the cube with a face and vice versa) have the same symmetry group[1].

A *group action* is an associative map $\phi : G \times X \to X$ with the property that $\phi(e, x) = x$ for all $x \in X$. Here X is a set, which we will typically take to be the vector space $\mathbb{R}^n$. A group action is naturally realized in the setting of *group representation theory*, where we represent abstract groups with elements $a, b, c, \ldots$ by a homomorphic group of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$. That is, a representation of a group G is a group homomorphism $\rho : G \to GL(n)$ from G to the group of invertible $n \times n$ matrices. In this case, the group action can be defined simply as $\phi(g, \mathbf{x}) = \rho(g)\mathbf{x}$.

The *orbit* of an element $x \in X$ under the action of G is the set $\{\phi(a, x) | a \in G\}$. This is an important concept for our applications in representation learning, where $X = \mathbb{R}^n$ is some data space and $\mathbf{x} \in \mathbb{R}^n$ are input vectors. In this setting, the orbits are all the points to which an input $\mathbf{x} \in \mathbb{R}^n$ can be transformed by symmetry transformations in some group G that we wish to disregard. Our goal then, is to collapse the orbit into a single point in the representation space (i.e. to construct an invariant representation) while parameterizing the position on the orbit with a complementary variable. This is formalized by the notion of an *orbit space*, denoted X/G. The points in the orbit space correspond to orbits of G in X.

To make this concrete, consider a 2D space $\mathbb{R}^2$ and a representation of the rotation group $SO(2)$ that rotates about the origin. A matrix representation of $SO(2)$ is given by 2×2 rotation matrices of the form

$$\mathbf{R}(\varphi) = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \tag{2.1}$$

Its action on a vector $\mathbf{x} \in \mathbb{R}^2$ is just $\mathbf{R}(\varphi)\mathbf{x}$. The orbits are concentric circles around the origin, and the orbit space $\mathbb{R}^2/SO(2)$ is given by the radii of the orbits, i.e. $\mathbb{R}$.

2.2.2 Lie groups and Lie algebras

A Lie group (pronounced Lee-group) is a group that is also a differentiable manifold. Intuitively, the elements of a Lie-group lie on a smooth surface on which we can take derivatives. To keep the exposition concrete, we will only consider matrix representations of groups from now on. Most Lie groups can be represented by matrices.

The tangent space at the identity of any Lie group gives rise to a structure known as a Lie algebra. The Lie algebra $\mathfrak{g}$ of a Lie group G is the tangent space to G at the

[1]Interestingly, [Hinton, 1979] has shown that people will confuse these two very different objects with identical symmetries in a mental imagery task.

origin, together with a product on tangent vectors $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ satisfying:

$$\begin{aligned}
[\mathbf{X}, a\mathbf{Y} + b\mathbf{Z}] &= a[\mathbf{X}, \mathbf{Y}] + b[\mathbf{X}, \mathbf{Z}] \\
[a\mathbf{X} + b\mathbf{Y}, \mathbf{Z}] &= a[\mathbf{X}, \mathbf{Z}] + b[\mathbf{Y}, \mathbf{Z}] && \text{(Bilinearity)} \\
[\mathbf{X}, \mathbf{Y}] &= -[\mathbf{Y}, \mathbf{X}] && \text{(Anti-commutativity)} \\
[\mathbf{X}, [\mathbf{Y}, \mathbf{Z}]] + [\mathbf{Z}, [\mathbf{X}, \mathbf{Y}]] + [\mathbf{Y}, [\mathbf{Z}, \mathbf{X}]] &= 0 && \text{(Jacobi Identity)}
\end{aligned} \tag{2.2}$$

For all scalars a, b and elements $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathfrak{g}$. Since we work with matrices, we can be more concrete: the elements of $\mathfrak{g}$ are tangent vectors (matrices) to the group manifold at $\mathbf{I}$, and the Lie bracket can be defined as the commutator: $[\mathbf{X}, \mathbf{Y}] = \mathbf{XY} - \mathbf{YX}$.

The most important fact about Lie groups is that their structure is almost completely determined by the Lie algebra. In other words, the global structure is determined by the infinitesimal structure. Every element of the group can be expressed as the exponential of an element of the Lie algebra, and the product of two elements of the group can be expressed as a sum of elements in the Lie algebra:

$$\exp(\mathbf{X}) \exp(\mathbf{Y}) \approx \exp(\mathbf{X} + \mathbf{Y} + [\mathbf{X}, \mathbf{Y}] + \ldots) \tag{2.3}$$

The continuation of this series (which is quite involved) is known as the Baker-Campbell-Hausdorff formula. For commutative groups, the commutator (Lie bracket) is always zero, and hence the matrix exponential behaves exactly like the scalar exponential: $\exp(\mathbf{X}) \exp(\mathbf{Y}) = \exp(\mathbf{X} + \mathbf{Y})$.

A Lie subalgebra $\mathfrak{h}$ of $\mathfrak{g}$ is a subspace of $\mathfrak{g}$ that is closed under the Lie bracket. Again, things are simpler in the commutative case, for then the commutator is always zero and hence a subalgebra is just a subspace.

2.2.3 Reduction of a representation

Two representations ρ and π are called equivalent if there exists a single matrix $\mathbf{W}$ such that $\mathbf{W}\rho(g)\mathbf{W}^{-1} = \pi(g)$ for all $g \in G$. In other words, equivalent representations are related by a change of basis.

Let ρ be a representation of G in $\mathbb{R}^n$. We say that subspace V of $\mathbb{R}^n$ is invariant if $\rho(g)\mathbf{v} \in V$, for any $g \in G$ and $\mathbf{v} \in V$. If ρ has a non-trivial invariant subspace, it is called *reducible* (the trivial ones being $\{\mathbf{0}\}$ and $\mathbb{R}^n$). Otherwise it is called *irreducible*. If the orthogonal complement to an invariant subspace V of ρ, V^*, is also an invariant subspace of ρ, then there exists a matrix $\mathbf{W}$ that decomposes each element of the representation into the same block structure:

$$\mathbf{W}\rho(g)\mathbf{W}^{-1} = \begin{bmatrix} \rho_V(g) & 0 \\ 0 & \rho_{V^*}(g) \end{bmatrix} \tag{2.4}$$

where ρ_V is a representation of G in V and ρ_{V^*} is a representation of G in V^*. This process is called the *reduction* of a representation. If it is possible to repeat this process of reduction recursively until we get to the irreducible representations, the representation is called *fully reducible*. All compact groups are fully reducible.

2.2.4 Orthogonal & toroidal groups

The toroidal subgroup analysis algorithm derived in chapter 4 makes use of orthogonal transformations. These are the linear transformations that do not expand or contract the space on which they act. That is, they leave lengths (and angles) invariant, and this defines them fully. Formulaically, for orthogonal $\mathbf{Q}$, $\|\mathbf{Qx}\| = \|\mathbf{x}\|$ and $(\mathbf{Qx})^T(\mathbf{Qy}) = \mathbf{x}^T\mathbf{y}$. A useful computational property of the orthogonal transformations is that they can be inverted for free, because $\mathbf{Q}^{-1} = \mathbf{Q}^T$. This implies that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, which may also be taken as the defining property.

All orthogonal transformations can be generated by repeated reflection. An even number of reflections makes a rotation, an odd number of reflections makes an anti-rotation or a plain reflection. Rotations have a determinant of $+1$, while reflections and anti-rotations have a determinant of -1. The group of all orthogonal transformations in n-dimensions is denoted $O(n)$, while the group of rotations (determinant $+1$), is denoted $SO(n)$, and called the special orthogonal group.

It can be shown that any orthogonal matrix $\mathbf{Q}$ in an even-dimensional space can be brought into the following canonical form, by an orthogonal change of basis $\mathbf{W}$:

$$\mathbf{WQW}^T = \begin{pmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_M \end{pmatrix} \tag{2.5}$$

where each $\mathbf{R}_j$ is a 2×2 orthogonal matrix. In odd-dimensional spaces, there will be an additional 1×1 "block" with value ± 1. We can understand this decomposition easily in three dimensions. For an odd number of reflections, one direction will be flipped, giving a -1 on the singular 1×1 block. For an even number of reflections (a rotation), the axis is left invariant and so there is a $+1$ in the 1×1 block. Furthermore, the orthogonal complement to this direction is a 2D subspace. For a pure rotation (two reflections) or an anti-rotation (three reflections), this 2D space is rotated, while for a reflection it is rotated by 0 degrees, i.e. $\mathbf{R}_1 = \mathbf{I}$. Apparently, the concept that generalizes to higher dimensions is not the rotation *axis*, but the rotation *plane*. The axis is just an artifact that appears in odd-dimensional spaces.

A *toroidal subgroup* of a compact Lie group G is a compact, connected, commutative subgroup of G. It is easy to see that for the special orthogonal group $SO(2M)$, toroidal groups take the form of eq. 2.5, where the 2×2 blocks are rotation matrices (some of which may be equal to the identity block). The matrix $\mathbf{W}$ performs a complete reduction of the toroidal group, by construction. A maximal toroidal subgroup of G is one that is not contained in any other toroidal subgroup of G. We will denote a toroidal subgroup of $SO(2M)$ in a $2M$-dimensional space by $\mathbb{T}^M(\mathbf{W})$, where $\mathbf{W}$ is the reducing basis as in eq. 2.5. Because a toroidal group is commutative, its Lie algebra is trivial, i.e. $[\mathbf{X}, \mathbf{Y}] = 0$ for all $\mathbf{X}, \mathbf{Y} \in \mathfrak{t}$.

2.2.5 Relative & absolute invariants, weights and degeneracy

Let $\rho : SO(2) \to GL(2M)$ be a representation of the rotation group $SO(2)$, parameterized by an angle θ. It can be shown [Kanatani, 1990] that such a representation can always be reduced into block-diagonal form with 2×2 blocks by an orthogonal matrix $\mathbf{W}$ (or equivalently into 1D-complex irreducible representations). Let $\mathbf{R}(\theta)$ denote one of these 2D representations. Clearly, we have the following facts:

$$\mathbf{R}(\theta)\mathbf{R}(\theta') = \mathbf{R}(\theta + \theta') \tag{2.6}$$

$$\mathbf{R}(0) = \mathbf{I} \tag{2.7}$$

$$\mathbf{R}(\theta + 2\pi) = \mathbf{R}(\theta). \tag{2.8}$$

From these, it follows that

$$\mathbf{R}(\theta) = \exp(n\theta\mathbf{B}), \tag{2.9}$$

where n is an integer and

$$\mathbf{B} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{2.10}$$

The integer n is called the *weight* of the representation. A vector in the 2D subspace on which a particular representation $\mathbf{R}$ acts, is called an absolute invariant if $n = 0$. Otherwise it is called a relative invariant. Notice that while an absolute invariant will have the same value after the group acts on it, a relative invariant will change, but will nevertheless be invariant as a vector (only the basis is changed).

If a representation has multiple irreducible representations of the same weight, the reduction is not unique. That is, given k representations of the same weight, any basis for the $2k$-dimensional subspace spanned by these representations defines a valid irreducible reduction. Furthermore, the sum of two relative invariants of the same weight is another relative invariant of the same weight.

Chapter 3

Representation Learning

It is widely recognized that the representation of data is an important factor in the success of machine learning algorithms [Bengio et al., 2013]. Typically, there is some kind of "nuisance variability" in the raw data whose removal can reduce the complexity of the learning task. But even when two representations have the same information content (i.e. they are in bijection), one can be a more effective substrate for learning than another (see figure 3.1). Strictly speaking (and as can be seen in this figure), the quality of a representation can only be defined with respect to a particular task and a particular choice of learning algorithm. For example, a representation that makes two classes linearly separable is well suited to a linear classifier such as SVM, while a decision tree classifier may fare poorly on it, because the splits required are not aligned with the coordinate axes (features). Conversely, a representation where the classes are easily separated by a few splits on each coordinate may be a good choice for a decision tree classifier but a very bad choice for a linear classifier. Furthermore, multiple learning tasks can be defined on the same input domain, and different tasks may suggest different representations of that domain. Nevertheless, it is often possible and desirable to find one representation that works well for a range of classifiers and learning tasks in the same input domain.

The observation that representations are important has spurred a great deal of research on various descriptors and feature extraction techniques for visual and audio data (e.g. [Bruna and Mallat, 2013], [Lowe, 2004], [Davis and Mermelstein, 1980]). These work well for the tasks and input domains they were designed for, but hand-crafting such features is a formidable task, making it infeasible to repeat this for every input domain. This is especially so if the input domain itself is a more or less abstract (possibly learned) representation of some lower-level perceptual domain, because in such a situation it may not be easy to understand the input domain analytically. A black-box that finds useful representations on any input domain would be preferable.

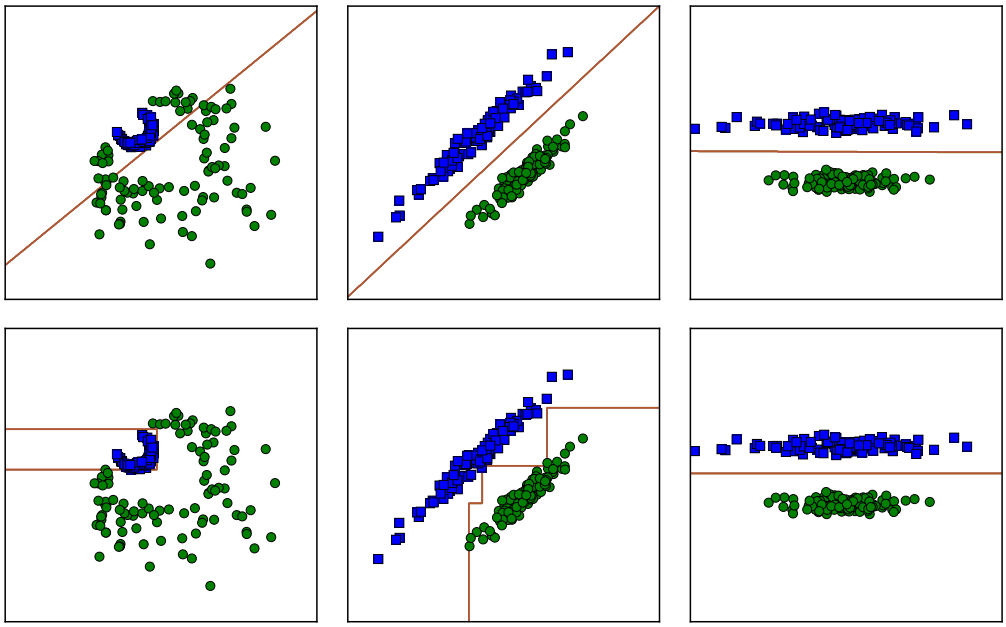

Figure 3.1: Two classifiers trained on three different data representations. Left: a somewhat difficult to separate data set consisting of two classes (blue squares, green circles). Middle: re-representation after performing a Möbius transformation (translation conjugated by circle inversions) to the data. Right: re-representation after performing an additional rotation. Top: decision boundary (solid line) learned by a linear SVM. Bottom: decision boundary learned by a decision tree. Clearly, the rightmost representation is superior for both classifiers. Other representations (left, middle) may work for one classifier but not so much for another.

Recently, there have been many attempts at *learning* representations with desirable qualities, using only unlabeled data. The objectives of this type of work often go beyond merely learning representations that are useful for classification. According to the website of the International Conference on Learning Representations 2013 [Bengio and Lecun, 2013],

> The rapidly developing field of representation learning is concerned with questions surrounding how we can best learn *meaningful* and *useful* representations of data.

Other common objectives of this type of research include: learning representations that are useful across a range of tasks (transfer learning), learning *invariant* representations, learning hierarchical representations that are progressively more *abstract* and learning representations that explicitly *untangle* the factors of variation present in the data. All of these are commendable goals, but at this point seem ill-posed,

or even "not-posed", because we have no formal definition for such intuitive ideas as "meaningfulness", "abstractness" and "untangling".

Despite the paucity of formal concepts, the field has been very successful, achieving the stated goal of "usefulness" in a spectacular way. Many benchmarks for perceptual classification tasks are now dominated by approaches that learn several layers of internal representations of the data. On what is arguably the most challenging image recognition data set, ImageNet, [Krizhevsky et al., 2012] trained a deep, 60-million parameter convolutional neural network, winning the competition by a large margin (however, this approach did not involve unsupervised learning). Using a multi-layer network trained by Independent Subspace Analysis (section 3.1.1), [Le et al., 2011] made significant improvements on the KTH, Hollywood2, UCF and YouTube action recognition data sets. In speech recognition, large gains were reported on the TIMIT data set using a deep belief network [Mohamed et al., 2012]. See [Hinton et al., 2012] for an overview of deep neural networks in speech recognition. The architecture of [Le et al., 2012], based on sparse autoencoders was able to learn fairly high-level object classes such as faces and cats using only unsupervised image data.

Let us look at the goals of representation learning in a little more detail. We (and [Bengio and Lecun, 2013]) want a meaningful representation, but what makes a representation meaningful? As a first approximation, we can consider the degree to which the representation makes explicit such concepts as feel natural to a human observer. If the model produces units that are uniquely sensitive to a region of pixel-space that a human would label `cat`, or `face` (e.g. [Le et al., 2012]), then it has learned a meaningful representations because these appear to be completely natural object classes. We can try out various models and algorithms, and keep the ones that learn representations that "make sense" to us. To actually guide learning towards such representations however, we need a formal definition of what it is we seek. The regions of pixel-space that correspond to natural classes are not arbitrary, so we should seek to characterize them in a way that does not refer to human judgment or intuition about what are the natural classes. Clearly, human judgment itself requires such an objective principle, although supervision (i.e. deferring to other humans' judgment) can play some role.

The related notion of "untangling" is used in two related but distinct meanings. The first, as used in [DiCarlo and Cox, 2007] and [DiCarlo et al., 2012], involves flattening of object manifolds. The pixel space manifold generated by continuously changing scene parameters such as object position, lighting position, lighting intensity, etc., is highly convoluted, and appears tangled when projected onto three dimensions (see

figure 3 in [DiCarlo and Cox, 2007]). The authors of this paper suggest that the central problem in object recognition is the flattening of these manifolds.

Bengio uses the word in a slightly different way, to mean the isolation of distinct factors of variation at each point on the object manifold [Bengio et al., 2013]. For example, some images can be factored (disentangled) into subject identity, pose, position, camera position, etc. It was noted that current representation learning algorithms seem to do some amount of disentangling, but (quoting from [Bengio, 2013]):

> Although these observations [of successful disentangling] are encouraging, we do not yet have a clear understanding as to *why* some representation algorithms tend to move towards more disentangled representations, and there are other experimental observations suggesting that this is far from sufficient.

These two uses can be clarified by considering the notion of a manifold in a little more detail. A (presentation of a) differentiable manifold is a set of differentiable, invertible maps called charts ϕ_α that map some open subset U_α of a flat parameter space $\mathbb{R}^n$ to a curved subset of $\mathbb{R}^m$. In the machine learning setting, $\mathbb{R}^m$ is where we make observations, $\mathbb{R}^n$ is a space of latent or hidden variables. Flattening a manifold, in the sense of [DiCarlo and Cox, 2007], means to learn the map(s) ϕ_α^{-1}, so that we can move around the data manifold by controlling latent variables in a linear space $\mathbb{R}^n$. Untangling in the sense of [Bengio et al., 2013] is more specific, in that it should also prefer certain bases for this linear space to other bases. That is, we prefer a representation where we can assign some familiar names to the coordinate axes; we prefer a basis $(\texttt{width}, \texttt{height}, \texttt{x-pos}, \texttt{y-pos})^T$ over a linear transformation ("linear tangling") of these, even though such a transformed space is equally flat and equally informative about the percept. This is essentially the same issue as that of learning meaningful representations that we discussed before, only applied to a factorial representation (object properties) instead of an object class based representation (`cat`, `face`, ...). We can restate the fundamental question in the language of manifolds: what principle can we use to express a preference for one basis over another?

In his review, [Bengio, 2013] lists several priors (or rather, inductive biases) that may help solve the problem of disentangling. These include: smoothness, multiple explanatory factors, hierarchical organization, natural clustering, sparsity and several others (see section 6.2 of this paper for details). In our view, only the prior of sparsity and the related notion of statistical independence gets at the core of the

problem as described above – the others seem to be generally useful ideas in statistical modeling, but not aimed specifically at disentangling. Two variables x, y are statistically independent when $p(x, y) = p(x)p(y)$ so that $p(x|y) = p(x)$; variable y provides no information about x and vice versa. Thus these variables may be said to constitute different factors of variation (literally, in the sense that the joint density factorizes), and this is the basis for the Independent Component and Independent Subspace Analysis algorithms, which we will discuss in more detail in section 3.1.1.

However, we do not believe that independence is ultimately the right way to formalize disentangling. The reason is that many distinct factors of variation (the natural properties we use to describe objects), are in fact correlated. We want to separate factors that can *in principle* be varied independently, even if in a particular data set (the totality of experience of an intelligent agent, for instance) these factors are correlated and hence not independent. We think statistical independence is a reasonably good proxy for a better notion of separation, which we will present in section 3.3.

In the rest of this chapter, we review a number of representation learning algorithms, highlighting similarities and differences with the Toroidal Subgroup Analysis algorithm that will be presented in chapter 4. We distinguish three representation learning paradigms: supervised, unsupervised and transformation learning. Classical neural networks can be viewed as learning an internal representation in a supervised manner, and will not be reviewed here. In section 3.1 we will review more recent work on unsupervised representation learning, and in section 3.2, we will look at transformation learning algorithms. Finally, in section 3.3 we present a number of basic principles for representation learning that are informed by group theory.

3.1 Representation learning algorithms

3.1.1 ICA and ISA

Independent Component Analysis (ICA) refers to a class of techniques for separating a signal into independent components. The main equation of the ICA model is

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{3.1}$$

where $\mathbf{x}$ is the input signal, $\mathbf{A}$ is the mixing matrix that is to be learned, and $\mathbf{s}$ is a vector of source coefficients. The goal of learning is to obtain coefficients s_j that are as independent as possible. Typical objectives include minimization of mutual information and maximization of kurtosis [Hyvärinen and Oja, 2000].

If the signal conforms reasonably well with the model assumptions, such as when the signal consists of overlapping speech signals, ICA is very effective in separating the sources. However, it is not always possible to find independent components by taking a linear combination of the input variables, so that the found sources show some residual dependencies. This has led to the development of Independent Subspace Analysis (ISA) algorithms [Cardoso, 1998], where dependencies *within* groups of source variables (subspaces) are allowed but dependencies *between* groups are discouraged. When trained on natural images using 2D subspaces, this algorithm finds subspaces that are locally invariant under image shifts [Hyvarinen and Hoyer, 2000].

3.2 Transformation learning algorithms

3.2.1 Anti-symmetric Covariance Analysis

In a very interesting but largely overlooked paper, [Bethge et al., 2007] provide an analysis of unsupervised learning of invariant representations, and derive an anti-symmetric version of canonical correlation analysis for this purpose. Their algorithm is based on the observation that for wide-sense stationary processes, the optimal linear predictor is given by the time-lagged covariance matrix. Furthermore, they show that for whitened data, this matrix is orthogonal.

The analysis is simplest in the noiseless case. Consider the time-series $\mathbf{x}^t$, $t = 1 \dots T$ generated by $\mathbf{x}^t = \mathbf{R}\mathbf{x}^{t-1}$. In their experiments, the data $\mathbf{x}$ are vectorized image patches and the matrix $\mathbf{R}$ is a representation of an image rotation operator in the space of pixels. Let $\mathbf{X}$ and $\mathbf{Y}$ be matrices with column t equal to $\mathbf{x}^{t-1}$ and $\mathbf{x}^t$, respectively. Assuming whiteness of the instantaneous covariance, the time-lagged covariance is equal to the optimal linear predictor;

$$\begin{aligned} \mathbf{C}_{t,t-1} &= \frac{1}{T}\mathbf{Y}\mathbf{X}^{\mathrm{T}} \\ &= \frac{1}{T}\mathbf{Y}(\mathbf{R}^T\mathbf{Y})^{\mathrm{T}} \\ &= \left(\frac{1}{T}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right)\mathbf{R} \\ &= \mathbf{R} \end{aligned} \tag{3.2}$$

This matrix, estimated either as the time-lagged covariance or simply by least-squares, is then decomposed to yield invariant filter pairs. As discussed in section 2.2.4, an orthogonal matrix can always be reduced into block-diagonal form with 2×2 rotation blocks by an orthogonal change of basis. A pair of basis vectors corresponding

to one of these 2×2 blocks spans an invariant subspace of the transformation $\mathbf{R}$, so that the norm of the projection of a signal onto this subspace provides an invariant representation of the signal. [Bethge et al., 2007] show that this basis matrix $\mathbf{W}$ can be as the eigenvectors of the squared anti-symmetric part of $\mathbf{R}$.

The ACA algorithm assumes that there is a single transformation $\mathbf{R}$ relating subsequent observations, possibly with added noise. In the case of image rotations, this means that $\mathbf{R}$ represents a rotation by only one particular angle. Even though image rotations by any angle can be represented by a single basis matrix $\mathbf{W}$ with varying block-diagonal matrices, the ACA algorithm cannot learn $\mathbf{W}$ in this scenario. This limitation is a significant barrier to application of this learning algorithm in real-world settings, where the stimulus is not tightly controlled. In the TSA algorithm, we weaken this assumption to the assumption that the subsequent observations are related by an unknown and variable element of a compact commutative group.

3.2.2 Gating models

A more flexible class of models are the so-called gating models, such as the gated Boltzmann machine [Memisevic and Hinton, 2010], gated autoencoder [Memisevic, 2011], synchrony-kmeans [Konda et al., 2013] and the closely related spatiotemporal energy models [Adelson and Bergen, 1985]. For concreteness and brevity, we focus on the gated autoencoder. This model encodes the transformation between two input vectors by computing products of filter responses. One such factor signals the presence of one feature (filter) in the first image and another in the second image. If the filter pairs learn to span the shared invariant 2D subspaces of a group of transformations, then using appropriate (possibly learned) pooling matrices, the factor responses can be turned into transformation-invariant representations of image content and content-invariant representations of the transformation.

While effective and easy to train, gating models are parameterized inefficiently. A single filter pair can detect one particular angle in one particular 2D subspace. Therefore, to effectively model transformations from a commutative group, the model has to tile each invariant subspace with many filter pairs, effectively discretizing the transformations.

Gating models were initially motivated by the wish to incorporate multiplicative or "gating" interactions in neural networks; the relation with the decomposition of orthogonal transformations into rotations in invariant 2D subspaces was found later [Memisevic, 2012]. The gating perspective greatly increases the expressive power of neural network models using a neurally realistic operation (taking the product of two

scalars), but we feel that this is too low a level of analysis. Useful mathematical abstractions such as Lie groups exist, and we believe using them can guide the development of new transformation learning models as well as help us interpret existing models and the representations they produce.

3.2.3 Phase-amplitude models

Motivated by findings in natural image statistics, [Cadieu and Olshausen, 2012] proposed a model for learning representations of form and motion. The first finding is that responses of pairs of filters learned by sparse coding of natural images tend to show circularly symmetric kurtotic distributions. The fact that these dependencies occur even though the prior in such models is factorial, suggests that a better model should incorporate dependencies among pairs of filter responses. In particular, the circular symmetry of these distributions suggests that a polar-coordinate representation is the right way to describe the signal in each 2D subspace spanned by filter pairs. The second finding is that the log-amplitudes tend to show linear dependencies. That is, if a signal is projected onto two learned 2D subspaces, the logarithms of the projection amplitudes show linear dependencies.

The model projects the image patch onto a set of complex filters, each of which is decomposed into an amplitude and phase. This is similar to the TSA model presented in chapter 4, although the basis of [Cadieu and Olshausen, 2012] is not orthogonal as in TSA, and the phase variables represent the absolute phase, i.e. phase relative to an arbitrary real axis, instead of using phase offsets between subsequent patches. This first layer is trained by optimizing an objective function that encourages sparse and slowly changing amplitudes, and phases that allow the signal to be reconstructed properly.

Once trained, the log amplitudes and phase differences of the data are computed, and used as input data for a second layer. The log-amplitudes are modeled using a sparse and slow objective, similar to the first layer. The phase differences are modeled by von Mises distributions whose means are linear functions of a number of sparse and slow components.

While the model is specified in probabilistic language, only point estimates for the latent variables are obtained, in contrast to full posteriors we obtain in our model. Furthermore, the values for the latent variables are obtained by a gradient descent procedure as opposed to a one-shot computation, precluding real-time applications.

3.2.4 Lie group methods

Several researchers have looked at the problem of learning Lie groups, the earliest of which is [Rao and Ruderman, 1999], later improved in [Miao and Rao, 2007]. The basic idea in each case is to learn a set of generator matrices $\mathbf{G}_j$, that are linearly combined using a set of latent variables s_j and exponentiated to obtain the transformations relating observed vectors $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{y} = \exp\left(\sum_j s_j \mathbf{G}_j\right)\mathbf{x}. \tag{3.3}$$

The model of [Miao and Rao, 2007] is a probabilistic model trained using EM. In the E-step, the expected complete-data log likelihood with respect to the posterior over s_j is computed. In the M-step, this function is optimized with respect to the generators $\mathbf{G}_j$ (among other parameters), where the matrix exponential is approximated linearly. Once trained, the model can generate arbitrarily large transformations by computing the matrix exponential of an arbitrary linear combination of the generators, but training itself can only be done using infinitesimally small transformations due to the linear approximation.

[Sohl-Dickstein et al., 2010] avoid having to compute a full matrix exponential, by using an eigendecomposition:

$$\exp \mathbf{A}s = \exp\left(\mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}s\right) = \mathbf{W}\exp\left(\mathbf{\Lambda}s\right)\mathbf{W}^{-1}. \tag{3.4}$$

where $\mathbf{A} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}$ denotes the eigen decomposition of $\mathbf{A}$ into a complex matrix $\mathbf{W}$ and a complex diagonal matrix Λ. The matrix exponential of a diagonal matrix $\mathbf{\Lambda}s$ is simply the elementwise exponential of the diagonal entries. This is much faster than computing a full matrix exponential as in the model of [Miao and Rao, 2007], but precludes a simple linear combination of generators in the Lie algebra, without recombining $\mathbf{A} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}$. Instead, [Sohl-Dickstein et al., 2010] use a multiplicative combination of one-parameter groups in a fixed order:

$$\mathbf{y} = \left(\prod_j \mathbf{W}_j \exp\left(\mathbf{\Lambda}_j s_j\right)\mathbf{W}_j^{-1}\right)\mathbf{x}. \tag{3.5}$$

The parameters and latent variables are obtained by gradient descent on an objective based on 3.5, with additional smoothing and regularization terms.

3.3 Basic principles for representation learning

In this chapter's introduction, we looked at the goals of representation learning, and saw that there are currently no solid foundations on which to build a theory for this field. With a clear picture of the goals of- and current approaches to representation learning, and a rudimentary understanding of group theory we are now ready to lay down some basic principles which will form the basis of our work in later chapters, and which we hope others will find useful.

The object classes we recognize do not partition the perceptual space arbitrarily. If we are to build representation learning algorithms that can discover these classes without much supervision, we should try to understand the structure of this partitioning. As we have seen in the introduction, all equivalence relations (partitionings of a set, classes) correspond to a group action. If we can learn the effect of our own movements (or more generally "actions") on the input space, we have defined the object classes implicitly:

The most natural proto-classes are defined by a group G and its action on a data space X, as the points in the orbit space X/G.

Typically, we take take $X = \mathbb{R}^N$ to be some feature space consisting of pixels, optical flow vectors, the output of a representation learning algorithm, etc. – see [Kanatani, 1990] for many examples of group representations in such feature spaces. As discussed in section 1.4, when the orbits are aligned with the actual class-boundaries, a representation in terms of orbits can drastically reduce the sample complexity, because we can learn the label of an entire orbit from one or a few examples.

Besides volitional actions, we can consider actions that are unobserved or externally generated. In the next chapter, we show that assuming the changes are caused by a toroidal group acting linearly on the input space, we can learn the group representation without being told which transformation was responsible for each observed change. In general we will want to learn a group action that can explain as many observed changes as possible using as few generators as possible. That is, we want the group to act transitively on high-density regions of input space (typically manifolds) so that we will often have some element of the group that can explain an observed change. At the same time, the group should not contain elements that can transform a high-density observation into a very low-density observation (at least for small transformations).

The ultimate completion of this program (learning to explain every observed change, and considering two inputs equivalent when they are in one orbit) leads to a system that will see every input as one and the same thing. However, nothing prevents us from performing analysis ("to break up into parts"), i.e. to consider only subgroups at a time, and to consider the classes induced by these subgroups.

When we observe a change $\mathbf{x} \to \mathbf{y}$, and seek to explain it using some element of a group $g \in G$, there can be many candidates that are consistent with this observations. Typically, the set of candidates forms a group. A special case of this is when we ask which transformations $g \in G$ can transform $\mathbf{x}$ into itself. These are the symmetries of $\mathbf{x}$ with respect to G. It is clearly desirable to know the ambiguities in our transformation estimate, and to describe this set of possibilities (using a posterior distribution over the group elements, for example). So we have,

A representation of data $x \in X$ should include a characterization of the symmetries of x with respect to the learned group G.

A concrete example that shows why this is useful can be found in the widely studied problem of estimating camera motion and 3D scene structure from point correspondences. As shown in [Ma et al., 1999], typical camera motions allow us to reconstruct the scene geometry and camera motion only up to subgroups of the Euclidean group. The utility of characterizing the ambiguity is obvious in this example, but it is no less useful in the case of a learned group representation acting on high dimensional observations instead of 3D points.

Up until now, we have ignored the issue of invertibility. Many changes of practical relevance are not strictly invertible when one considers only the observation at one instant in time. For example, by translating an image, pixels fall of on one side and these cannot be recovered through an inverse translation. While this can be seen as a limitation of the proposed approach, we believe it is actually a strength. We can always assume that non-invertibility implies incomplete information; it signals the need to introduce latent variables or invoke some memory mechanism to make the transformation on the union of observed and latent variables invertible. Our notion of Euclidean space is an example of this. We never directly observe Euclidean space, in the sense that sensory measurement values do not come in the form of points in a 3D space. However, if we construct a more abstract representation that does involve some kind of 3D-description and remember past observations, any Euclidean motion in a scene becomes invertible on this augmented space (and thus an understanding

of object permanence is achieved). This idea is taken to the extreme in the field of physics, where all quantities of interest are inferred, and all transitions are invertible at the elementary level.

We can take this idea of molding the representation $\mathbf{\Phi}(\mathbf{x})$ to conform to prior assumptions of group structure one step further. As we have seen in chapter 2, an abstract (Lie) group has infinitely many representations, so we can pick one with useful computational properties – as long as we can learn a way to embed $\mathbf{x}$ into a representation space $\mathbf{\Phi}(\mathbf{x})$ that transforms according to the group representation of choice. One important computational requirement is that we can compute inverses cheaply, which suggests that we seek a data representation that makes the group representation orthogonal. All compact groups (and more generally the amenable groups) are unitarizable, and hence orthogonalizable (by a procedure described in section 11.4.1 of [Doran and Lasenby, 2003]). Orthogonal transformations leave invariant a (not necessarily Euclidean) metric in the representation space, and so they are sometimes called isometries.

In the representation space, symmetries should become isometries.

We have taken this maxim from [Dorst et al., 2007], where it is applied to model Euclidean and conformal geometry. Besides free invertibility, this approach has the important advantage that all metric and non-metric geometric relations between the represented objects in this geometry are covariantly preserved by the symmetry group (as long as these relations can be expressed in geometric algebra; see [Dorst et al., 2007] chapter 17 for details).

The next problem is untangling. The group theoretic point of view provides a clear principle for disentangling, which we have called Weyl's principle [Weyl, 1939] after [Kanatani, 1990]:

A k-tuple $\mathbf{z}_j$ of latent variables can rightly be called a *single* observable if and only if it defines an inequivalent irreducible representation of a symmetry group of the phenomenon being observed.

By "symmetry group of the phenomenon" we are referring to the previous principles: it can be anything we wish, but for practical purposes we use a group that can often be invoked to explain observed transformation sequences, and that appears to a symmetry of one or more functions we are interested in. By "defining an irrep" we

mean that the vectors $\mathbf{z}_j$ (for $j = 1 \ldots J$) are obtained by a linear change of variables from the measurements $\mathbf{x} \in \mathbb{R}^N$, such that all elements of the group only transform the elements of each $\mathbf{z}_j$ among themselves. See section 2.2.3 for details on irreducible representations.

More generally, the goal can be to separate the variables into sets that transform independently (as an ordered set), using some class of non-linear functions. This can in principle be done, as was shown by [Kanatani, 1990] for camera rotation reconstruction problems. Kanatani achieved this non-linear untangling by analysis, and not by a learning algorithm. To our knowledge non-linear untangling by something like Weyl's principle has never been tried in machine learning.

Weyl's principle also provides a new way to understand and construct convolutional neural networks [LeCun and Bottou, 1998], [Serre et al., 2007]. This classical architecture is characterized by two features: convolution and pooling. Several learned filters are convolved with the image, which is equivalent to saying the weights of filters at each location in the image are shared. After a layer of filtering, the network locally pools several filter response values at each location, by squaring and summing, or by taking maxima. This can be done within one feature map or between feature maps.

As was discussed in section 2.2.5, the sum of two relative invariants is a relative invariant if and only if they define *equivalent* representations, i.e. if they have the same weight. So if $\mathbf{z}_j = \mathbf{W_j}^T\mathbf{x}$ (for some matrix of filters $\mathbf{W}$) defines an irreducible representation, we may pool the values in $\mathbf{z}_j$ at nearby locations. We should never sum-pool the values of inequivalent representations, because the result will not be an invariant. However, we can perform max-pooling among absolute invariants computed from inequivalent representations, because the result is again an invariant.

Chapter 4

Toroidal Subgroup Analysis

In this chapter we describe a model and an algorithm for learning toroidal groups from examples. These groups are a natural place to start developing a catalog of learnable group structures, because their defining properties (compact, connected, commutative) make them easy to deal with, both analytically and computationally.

4.1 The TSA model

4.1.1 The likelihood function

The TSA model assumes that observed pairs of vectors $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1\ldots N}$ are each related by some unknown transformation from a toroidal subgroup $\mathbb{T}^M(\mathbf{W})$ of $SO(2M)$. Recall (section 2.2.4) that every transformation $\mathbf{Q}$ in a toroidal group $\mathbb{T}^M(\mathbf{W})$ can be written as $\mathbf{Q} = \mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T$, where $\mathbf{Q}$ and $\mathbf{W}$ are $2M \times 2M$ orthogonal matrices, and $\mathbf{R}(\varphi)$ is block-diagonal with M rotation blocks of size 2×2. The matrix $\mathbf{W}$ performs the reduction of $\mathbb{T}^M(\mathbf{W})$ into irreducible representations, which are the 2×2 blocks of $\mathbf{R}(\varphi)$. We will parameterize the group in reduced form, by considering $\mathbf{W}$ as the parameters to be learned. The toroidal group is determined by $\mathbf{W}$, while a particular transformation from the group is identified with the phase-vector $\varphi^{(i)}$. This gives the following data model (suppressing indices i from now on):

$$\mathbf{y} = \mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x} + \epsilon, \tag{4.1}$$

where $\varphi = (\varphi_1, \ldots, \varphi_M)^T$ is a latent vector of phase-variables (one vector per data pair) and ϵ is isotropic Gaussian noise with mean zero and variance σ_n^2. It follows that

$$\mathbf{y} \sim \mathcal{N}(\mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x},\ \sigma_n^2). \tag{4.2}$$

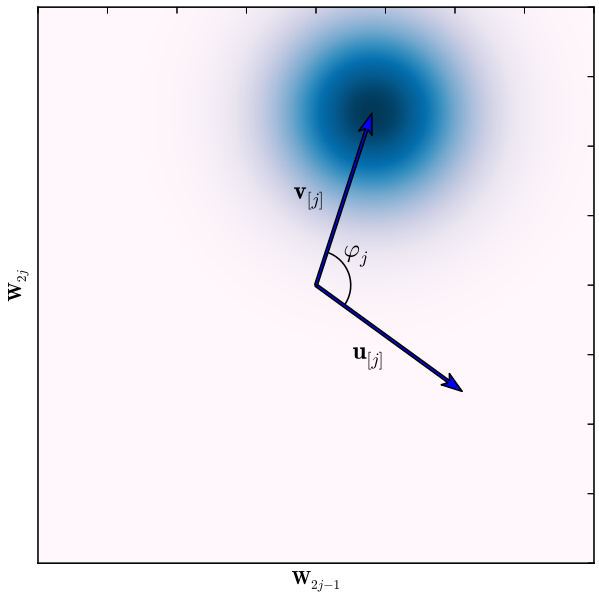

Figure 4.1: Likelihood function $p(\mathbf{v_j}|\mathbf{u_j})$.

By design, the transformation $\mathbf{Q}$ is more easily described in the $\mathbf{W}$-basis. We define $\mathbf{u}^{(i)} = \mathbf{W}^T\mathbf{x}^{(i)}$ and $\mathbf{v}^{(i)} = \mathbf{W}^T\mathbf{y}^{(i)}$, so that equation 4.2 can be written as

$$\mathbf{v} \sim \mathcal{N}(\mathbf{R}(\varphi)\mathbf{u},\ \sigma_n^2). \tag{4.3}$$

In general, obtaining the pdf of a transformed random vector involves scaling by a Jacobian determinant factor (see [DeGroot and Schervish, 2002], section 3.9), but in our case this reduces to $|\det(\mathbf{W})| = 1$.

The distribution over $\mathbf{v}$ has diagonal covariance (for it is again isotropic), so it factors into M independent distributions over each invariant subspace:

$$p(\mathbf{v}|\mathbf{u}) = \prod_{j=1}^{M} p(\mathbf{v_j}|\mathbf{u_j}) = \prod_{j=1}^{M} \mathcal{N}(\mathbf{v_j}|\mathbf{R}(\varphi_j)\mathbf{u_j},\ \sigma_n^2), \tag{4.4}$$

where we use a bold index $\mathbf{j}$ to denote the part of $\mathbf{u}$ or $\mathbf{v}$ in the j-th invariant 2D subspace (as opposed to simply taking the j-th coordinate v_j) and $\mathbf{R}(\varphi_j)$ is a 2×2 rotation matrix that rotates by angle φ_j. The distribution $p(\mathbf{v_j}|\mathbf{u_j})$ is shown in figure 4.1.

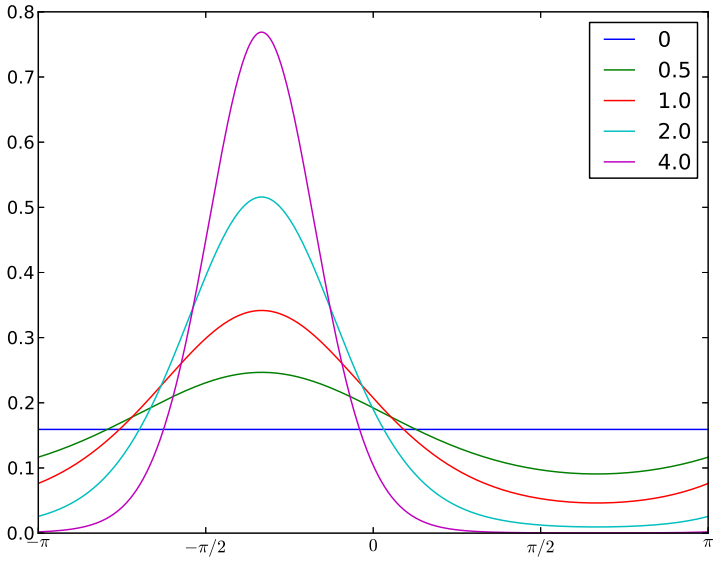

Figure 4.2: The von Mises distribution for $\mu = \pi/3$ and various values of κ (shown in the legend).

4.1.2 The prior over phases

For each data pair, we have M latent phase variables φ_j, $j = 1 \ldots M$ that describe the rotation in each invariant subspace. Since these are periodic variables, we use the von Mises distribution ([Mardia and Jupp, 1999, Mardia, 1975]) which we denote by $\mathcal{VM}$. The von Mises distribution is the maximum entropy distribution for a periodic variable with given circular mean and circular variance [Jammalamadaka and Sengupta, 2001]. It is given by

$$p(\varphi_j|\mu,\kappa) = \mathcal{VM}(\varphi_j|\mu,\kappa) = \frac{e^{\kappa\cos(\varphi_j-\mu)}}{2\pi I_0(\kappa)}, \tag{4.5}$$

where I_0 is the order-0 modified Bessel function of the first kind [Abramowitz and Stegun, 1965], μ is the preferred angle and κ is a precision parameter. We have plotted the von Mises distribution for $\mu = -\pi/3$ and various values of κ in figure 4.2.

We assume (for now) that φ_j are independent, i.e. $p(\varphi) = \prod_j p(\varphi_j)$. The conditional independence structure is shown in figure 4.3.

4.1.3 The posterior distribution

For EM learning (section 4.2), we need to evaluate the posterior distribution over the latent variables φ_j given the data $\mathbf{x}, \mathbf{y}$. The posterior distribution in invariant

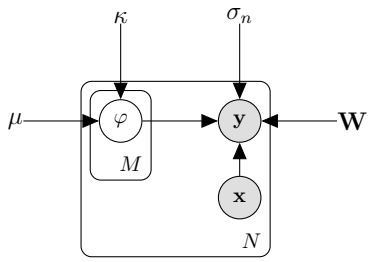

Figure 4.3: TSA as a probabilistic graph. Shaded nodes indicate observed variables, unshaded ones denote latent variables. The plates denote repetition of the variable for each subspace $j = 1, \ldots, M$ and data pair $i = 1, \ldots, D$. It is also possible to have separate variables κ_j and μ_j for each subspace j.

subspace j is

$$\begin{aligned} p(\varphi_j|\mathbf{u_j}, \mathbf{v_j}) &= \frac{p(\mathbf{v_j}|\mathbf{u_j}, \varphi_j)p(\varphi_j)}{\int_0^{2\pi} p(\mathbf{v_j}|\mathbf{u_j}, \varphi_j)p(\varphi_j)d\varphi_j} \\ &\propto \exp\left(\kappa\cos(\varphi_j - \mu) - \frac{1}{2\sigma_n^2}\|\mathbf{v_j} - \mathbf{R}(\varphi_j)\mathbf{u_j}\|^2\right). \end{aligned} \tag{4.6}$$

While it is not obvious from the form of eq. 4.6, the posterior is again a von Mises distribution. That is, $p(\varphi_j|\mathbf{u_j}, \mathbf{v_j}) \propto \exp\left(\kappa'_j\cos(\varphi_j - \mu'_j)\right)$. As far as we know, this conjugacy relation has not been published before. Conjugacy relations greatly simplify sequential Bayesian inference, because the form of the distribution is invariant under Bayesian updating and therefore remains tractable indefinitely.

We list the update equations for μ and κ below. Because the computation is somewhat tedious, the details are left for Appendix A. Leaving out index j for readability, writing $\mathbf{v}$ for $\mathbf{v_j}$,

$$\begin{aligned} \kappa' &= \sqrt{\frac{2\kappa\sigma_n^2\left(\mathbf{v}^{\mathrm{T}}\mathbf{R}(\mu)\mathbf{u}\right) + \|\mathbf{v}\|^2\|\mathbf{u}\|^2}{\sigma_n^4} + \kappa^2}, \\ \mu' &= \tan^{-1}\left(\frac{(v_1u_2 - v_2u_1) + \kappa\,\sigma_n^2\,\sin(\mu)}{(v_1u_1 + v_2u_2) + \kappa\,\sigma_n^2\,\cos(\mu)}\right). \end{aligned} \tag{4.7}$$

We can gain some insight into eq. 4.7 by using the outer product $\mathbf{v} \wedge \mathbf{u}$ from geometric algebra [Dorst et al., 2007]. When applied to vectors, this product constructs an element called a 2-blade that represents the weighted subspace spanned by its arguments. The weight of a blade is analogous to the magnitude of a vector, and can be retrieved by geometric multiplication by a unit-weight 2-blade of the same orientation, here given by $\mathrm{I} = \mathbf{W}_{2j} \wedge \mathbf{W}_{2j+1}$, the unit 2-blade representing span of the unit vectors $\mathbf{W}_{2j}, \mathbf{W}_{2j+1}$.

The coordinates of a 2-blade $\mathbf{v} \wedge \mathbf{u}$ in an orthonormal frame can be computed as the upper triangle of $\mathbf{v}\mathbf{u}^T - \mathbf{u}\mathbf{v}^T$ [Bethge et al., 2007]. In the 2D case, this is a single scalar. The numerator and denominator in the equation for μ' then become

$$\begin{aligned}(v_1u_2 - v_2u_1) = \mathrm{I}(\mathbf{v} \wedge \mathbf{u}) &= \|\mathbf{v}\|\|\mathbf{u}\| \sin(\theta) \\ (v_1u_1 + v_2u_2) = (\mathbf{v} \cdot \mathbf{u}) &= \|\mathbf{v}\|\|\mathbf{u}\| \cos(\theta).\end{aligned} \tag{4.8}$$

Equation 4.8 shows the symmetry in eq. 4.7, and makes it clear that to compute the posterior mean, we should convert the prior mean μ and observed angle θ to unit vectors $(\cos(\mu), \sin(\mu))^T$ and $(\cos(\theta), \sin(\theta))^T$, compute a weighted average of those vectors and convert the result back to an angle using the arctangent function.[1] The averaging weights are the strength of our prior belief in μ, the precision κ, and the strength of belief we should have in θ, which is $\|\mathbf{u}\|\|\mathbf{v}\|\sigma_n^{-2}$. This geometrically intuitive result follows simply from consistent application of Bayes rule, as shown in Appendix A.

The update equation for the precision parameter κ captures the propagation of measurement noise ϵ through the angle computation process. When the vectors $\mathbf{u}$ and $\mathbf{v}$ in eq 4.7 are short relative to the noise variance σ_n^2, or when the prior μ does not fit well to the observed data, the posterior precision will be small. Another way to read the κ-update in equation 4.7 is that we must divide the data vectors $\mathbf{u}$ and $\mathbf{v}$ by the standard deviation of the noise: $\mathbf{p} = \mathbf{u}/\sigma_n$ and $\mathbf{q} = \mathbf{v}/\sigma_n$. The update then becomes

$$\kappa' = \sqrt{2\kappa\ \mathbf{q}^{\mathrm{T}}\mathbf{R}(\mu)\mathbf{p} + \|\mathbf{q}\|^2\|\mathbf{p}\|^2 + \kappa^2}, \tag{4.9}$$

This obviates the need for excluding vectors $\mathbf{u_j}$, $\mathbf{v_j}$ whose norm is below some arbitrary threshold, as is necessary in the model of Cadieu & Olshausen to avoid instabilities [Cadieu and Olshausen, 2012]. TSA doe not exclude any μ_j, but instead provides each one with a confidence κ_j.

4.1.4 Relation to discrete Fourier transform

If we make no prior assumptions, i.e. if we set $\kappa = 0$, we can simplify even further:

$$\kappa'_j = \|\mathbf{q_j}\|\|\mathbf{p_j}\|. \tag{4.10}$$

Let us assume the filters $\mathbf{W}$ are sinusoids. As we will demonstrate in chapter 6, these emerge when TSA is trained on image shifts. Taking the input vectors to constitute discrete samplings of a 1D signal, we can describe the discretely sampled

[1] In practice, the atan2 function is preferred.

sinusoid filters analytically as follows. Let $\mathbf{W_j}$ have columns equal to the real and imaginary parts of ω^{-jk} (for $k = 1 \ldots N$), where $\omega = e^{2\pi i/N}$ is the N-th primitive root of unity. The DFT of $\mathbf{x}$ is given by:

$$X_j = \sum_{n=0}^{N-1} x_n \omega^{-jn}. \tag{4.11}$$

So the real and imaginary part of X_j are given by the two components of $\mathbf{W_j}^T\mathbf{x}$ for the first $N/2$ frequencies. The other half are complex conjugates of the first when $\mathbf{x}$ is real, and so they carry no information.

Now suppose we are interested in the transformation taking some arbitrary fixed vector $\mathbf{e} = (\frac{1}{\sqrt{2}}, \ldots, \frac{1}{\sqrt{2}})^T$ to $\mathbf{p} = \mathbf{W}^T\mathbf{x}$. The vector $\mathbf{e}$ is to be interpreted in the $\mathbf{W}$-basis, given by sinusoids. By eq. 4.10, we find $\kappa'_j = \|\mathbf{p_j}\| = \|\mathbf{W_j}^T\mathbf{x}\| = |X_j|$ (the modulus of the Fourier transform). Summarizing: using a flat prior and a Fourier basis for $\mathbf{W}$, the TSA representation of the transformation from $\mathbf{e}$ to $\mathbf{x}$ in terms of μ_j and κ_j is equal to the DFT representation of $\mathbf{x}$ in terms of complex phase $\arg(X_j)$ and amplitude $|X_j|$. So the model we have specified provides a probabilistic interpretation of the DFT, and generalizes it.

4.2 Learning by expectation-maximization

The model defined in the previous section contains a latent vector of angles φ that relates each input pair $\mathbf{x}, \mathbf{y}$. We saw that given some estimate of $\mathbf{W}$, we can compute the posterior over φ. As we will see in this section, given the posterior over φ, we can optimize the expected likelihood of the data with respect to $\mathbf{W}$. Situations like this are a common occurrence, and are typically solved by the Expectation-Maximization (EM) algorithm [Dempster et al., 1977].

The EM algorithm finds a maximum-likelihood estimate of the parameters in the presence of latent variables, by iterating through an E-step and an M-step. In the E-step, the posterior over latent variables $p(\varphi|\mathbf{x}, \mathbf{y})$ given the current estimate of the parameters $\mathbf{W}$ is evaluated. In the M-step, the expected complete-data log-likelihood under the posterior over latent variables is optimized with respect to the parameters. The details for the TSA models are given below.

We work with a data set $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1\ldots D}$ of pairs of vectors. We will sometimes store the observed and latent variables column-wise in matrices $\mathbf{X}, \mathbf{Y}, \mathbf{\Phi}$. We assume the pairs i, i' are independent from one another.

4.2.1 E-step

In the E-step, we compute the following expectation, which is to serve as an objective in the M-step:

$$\begin{aligned}\mathcal{Q}(\mathbf{W}, \mathbf{W}') &= \mathbb{E}\left[\ln p(\mathbf{Y}, \mathbf{\Phi}|\mathbf{X}, \mathbf{W})\right]_{\mathbf{\Phi}|\mathbf{X},\mathbf{Y},\mathbf{W}'} \\ &= \sum_i \mathbb{E}[\ln p(\mathbf{y}^{(i)}|\varphi^{(i)}, \mathbf{x}^{(i)}, \mathbf{W}) p(\varphi^{(i)}|\mu, \kappa)]_{\varphi^{(i)}|\mathbf{x}^{(i)},\mathbf{y}^{(i)},\mathbf{W}'}.\end{aligned} \tag{4.12}$$

where $\mathbf{W}$ is the optimization variable and $\mathbf{W}'$ is our current parameter estimate, which is fixed.

Only the likelihood term $p(\mathbf{y}^{(i)}|\varphi^{(i)}, \mathbf{x}^{(i)}, \mathbf{W})$ in eq. 4.12 depends on $\mathbf{W}$, so we can instead maximize[2] $\sum_i \mathbb{E}[\ln p(\mathbf{y}^{(i)}|\varphi^{(i)}, \mathbf{x}^{(i)}, \mathbf{W})]_{\varphi^{(i)}|\mathbf{x}^{(i)},\mathbf{y}^{(i)},\mathbf{W}'}$. Taking logarithms and writing out the expectation, we find this to be equal to

$$-\frac{1}{2\sigma_n^2}\sum_i \int_{\varphi\in\mathbb{T}^M} \left[\prod_j \mathcal{VM}(\varphi_j|\mu_j^{(i)}, \kappa_j^{(i)})\right] \|\mathbf{y}^{(i)} - \mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x}^{(i)}\|^2 d\varphi, \tag{4.13}$$

where we integrate over the whole torus $\mathbb{T}^M = \{(\varphi_1, \ldots, \varphi_M) \mid \varphi_j \in [0, 2\pi]\}$. Notice that the product of von Mises distributions $\prod_j \mathcal{VM}(\varphi_j|\mu_j^{(i)}, \kappa_j^{(i)})$ is equal to $p(\varphi|\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{W}')$, although the dependencies on $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{W}'$ are implicit in the use of posterior parameters $\kappa_j^{(i)}$ and $\mu_j^{(i)}$.

As shown in Appendix B.2, we can remove terms that do not depend on $\mathbf{W}$ and simplify, yielding the following maximization objective:

$$\begin{aligned}&\sum_{i,j} \mathbf{y}^{(i)T}\mathbf{W_j}\int_0^{2\pi}\left[\frac{\exp\left(\kappa_j^{(i)}\cos(\varphi_j - \mu_j^{(i)})\right)}{2\pi I_0(\kappa_j^{(i)})}\mathbf{R}(\varphi_j)\right] d\varphi_j \; \mathbf{W_j}^T\mathbf{x}^{(i)} \\ &= \sum_{i,j} \frac{I_1(\kappa_j^{(i)})}{I_0(\kappa_j^{(i)})}\mathbf{y}^{(i)T}\mathbf{W_j}\mathbf{R}(\mu_j^{(i)})\mathbf{W_j}^T\mathbf{x}^{(i)}\end{aligned} \tag{4.14}$$

where $\mathbf{W_j}$ denotes columns $2j-1$ and $2j$ of $\mathbf{W}$ spanning subspace j, and $\mathbf{R}(\varphi_j)$ is a 2×2 matrix that rotates by an angle of φ_j.

The result is rather intuitive. For each data pair i and invariant subspace j, we rotate the part of $\mathbf{x}^{(i)}$ that lies in subspace j using the posterior mean angle $\mu_j^{(i)}$ and then compute a dot product with the part of $\mathbf{y}^{(i)}$ that lies in subspace j. This will be at a maximum when the rotated $\mathbf{x}$ is fully aligned with $\mathbf{y}$, which is what we want to achieve. The ratio of Bessel functions $\frac{I_1(\kappa_j^{(i)})}{I_0(\kappa_j^{(i)})}$, plotted in figure 4.4, acts to weigh

[2]The term "E-step" is somewhat of a misnomer for us, because we do not evaluate the true expectation but instead evaluate an objective that is proportional to it.

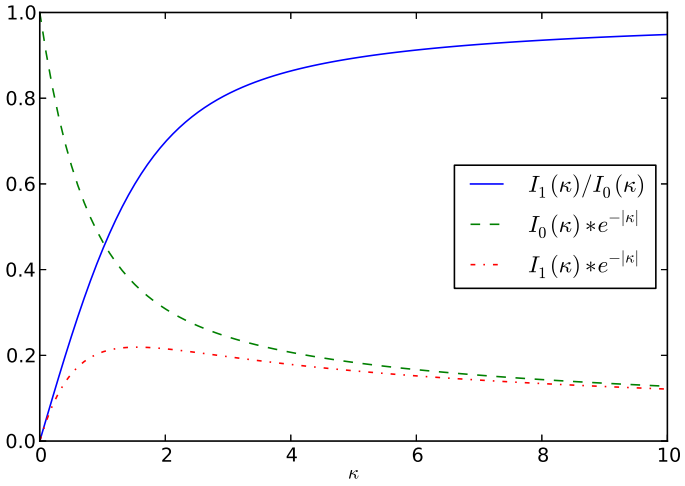

Figure 4.4: Ratio of Bessel functions $I_1(\kappa)/I_0(\kappa)$, computed from the exponentially scaled Bessel functions.

the terms so that those terms with low uncertainty (high κ) are weighted higher than those with high uncertainty (low κ). In practice, it is important for numerical stability to use the exponentially-scaled Bessel functions (available in most scientific computing libraries) to compute this ratio:

$$\frac{I_1(\kappa)}{I_0(\kappa)} = \frac{I_1(\kappa)\exp(-|\kappa|)}{I_0(\kappa)\exp(-|\kappa|)}. \tag{4.15}$$

4.2.2 M-step

To optimize the objective in eq. 4.14, we employ a simple gradient ascent scheme. The gradient is

$$\frac{d}{d\mathbf{W_j}}\mathcal{Q}(\mathbf{W_j}, \mathbf{W}') = \sum_i \frac{I_1(\kappa_j^{(i)})}{I_0(\kappa_j^{(i)})\sigma_n^2}\left[\mathbf{x}^{(i)}\mathbf{y}^{(i)\mathrm{T}}\mathbf{W_j}\mathbf{R}(\mu_j^{(i)}) + \mathbf{y}^{(i)}\mathbf{x}^{(i)\mathrm{T}}\mathbf{W_j}\mathbf{R}(\mu_j^{(i)})^{\mathrm{T}}\right]. \tag{4.16}$$

Clearly, $\mathbf{W_j} = 0$ will make the gradient zero, but this is typically not a maximum and does not take into account the orthogonality constraint on $\mathbf{W}$. So we optimize $\mathcal{Q}$ by gradient ascent, and use a simple SVD procedure to enforce the orthogonality constraint. After each gradient update, we compute $\mathbf{USV}^T = \mathbf{W}$ and then set $\mathbf{W} := \mathbf{UV}^T$. More sophisticated methods for optimization over the orthogonal group exist [Edelman et al., 1998, Plumbley, 2004], but our approach is simple and works well in practice. Alternatively, we can normalize the length of each column of $\mathbf{W}$ and orthogonalize column $2j$ against column $2j+1$ by a single step of Gram-Schmidt.

This approximate procedure appears to work reasonably well; the subspaces learn to become near-orthogonal although this constraint is not strictly enforced.

In practice, we do not completely optimize the objective in each M-step, but instead perform a single gradient-based update per M-step. Such a procedure has the same convergence guarantees as regular EM, and works well in practice.

Chapter 5

Low-dimensional subgroups

It is often the case that the transformations that act on the data can be described with far fewer parameters than the M angles used in the basic TSA model of the previous chapter. For example, when rotations act on a data set consisting of image patches, the transformations can be described with a single parameter: the rotation angle. When (cyclic) translations act on the data, we should need only two parameters: the amount of shift in each direction. If the data set consists of images of human faces with changing facial expressions, the transformations could be described by parameters like changes in smiling, opening-closing of the eyes, raisedness of the eyebrows, etc [Susskind et al., 2011].

These examples show that we are typically interested in groups that are of lower dimensionality than the M-dimensional Toroidal groups. Since every compact, connected commutative Lie-group is a subgroup of a maximal toroidal group, we can first learn the toroidal group and then search for the appropriate low-dimensional subgroup. The subspace angles φ_j are coordinates in the Lie algebra of $\mathbb{T}^M(\mathbf{W})$, so instead of looking for subgroups, we can look for subalgebras. The advantage of this approach is that we can seek *linear* subspaces in the Lie algebra instead of non-linear manifolds in the space of matrices.

5.1 The stabilizer subgroup

For a data point $\mathbf{x}$, we can ask the question: what is the set of transformations from the learned toroidal group $\mathbb{T}^M(\mathbf{W})$ that will leave $\mathbf{x}$ unchanged? This set is a group called the stabilizer subgroup of $\mathbf{x}$ in $\mathbb{T}^M(\mathbf{W})$,

$$\text{stab}(\mathbf{x}) = \{\mathbf{Q} \in \mathbb{T}^M(\mathbf{W}) \mid \mathbf{Q}\mathbf{x} = \mathbf{x}\}. \tag{5.1}$$

For a given $\mathbf{x}$, which transformations are in stab($\mathbf{x}$)? Since we are working with a toroidal group, the answer is pretty simple. When the component of $\mathbf{x}$ in subspace j is zero, $\mathbf{W}_{\mathbf{j}}^T\mathbf{x} = \mathbf{u_j} = \mathbf{0}$, then we can rotate it by any angle and the result will be the same (still $\mathbf{0}$). Conversely, when $\mathbf{x}$ has a non-zero projection $\mathbf{u_j}$, rotation by any non-zero (mod 2π) angle will change the vector. So we see that a transformation $\mathbf{Q} = \mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T$ is in stab($\mathbf{x}$) iff $\varphi_j = 0$ whenever $\mathbf{u_j} \neq 0$. In practice, we will be mostly concerned with transformations $\mathbf{Q}$ for which the constraint $\mathbf{Q}\mathbf{x} = \mathbf{x}$ is approximately satisfied, i.e. those that rotate in subspace j only when $\|\mathbf{u_j}\|$ and the posterior precision κ'_j are small.

The relevance of the stabilizer subgroup to our discussion is that we can only determine the transformation in $\mathbb{T}^M(\mathbf{W})$ that takes $\mathbf{x}$ to $\mathbf{y}$ modulo the stabilizer subgroup. That is, if $\mathbf{Q}\mathbf{x} = \mathbf{y}$ for $\mathbf{Q} \in \mathbb{T}^M(\mathbf{W})$ then also $\mathbf{Q}\mathbf{S}\mathbf{x} = \mathbf{y}$ for $\mathbf{S} \in$ stab($\mathbf{x}$). The stabilizer subgroup expresses the ambiguity in the transformation estimate. One nice property of the TSA model is that it can naturally incorporate new evidence to diminish the (simultaneous) stabilizer subgroup down to $\{e\}$. That is, given a set of correspondence pairs $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1\ldots D}$, we can use Bayesian updating to get the optimal toroidal rotation relating the $\mathbf{x}$'s to the $\mathbf{y}$'s (assuming they are all related by the same transformation, and the torus determined by $\mathbf{W}$ is fixed). In a more traditional estimation setting, one would always need at least M correspondence pairs in order to estimate the M angles, but by virtue of the conjugacy relation we can easily incorporate new information sequentially and give a best guess at any point in time, while being explicit about the remaining uncertainty.

The phenomenon of low-norm subspace projections appears in bilinear gating models too, where it has been called the subspace aperture problem by Memisevic [Memisevic, 2012, Memisevic and Exarchakis, 2013]. In this perspective, the low-norm subspaces pose a problem because they make it hard to estimate the "true" transformation from only one correspondence pair. The solution offered is to create a new layer in the network that pools over multiple factors, thus sharing information and yielding a representation of the transformation that is supposed to be independent of image content. But can this goal be achieved at all? What is the "true" transformation that is to be represented, and can it always be retrieved from one correspondence pair?

The group theoretical perspective provides clarity: if we are estimating an element of an M-parameter group such as $\mathbb{T}^M(\mathbf{W})$, it cannot generally be done given only one correspondence pair, no matter what clever trick is used. We need M pairs. In order to deal properly with the case where we have insufficient information to determine

the transformation uniquely with high precision, we should describe the uncertainty (both due to noise and the symmetry of the data) by a posterior distribution. If the transformation group under consideration contains fewer parameters, we need fewer data, so we should seek to diminish the number of parameters. Apparently the toroidal group is not the real symmetry group of interest; it is merely the centralizer in $SO(N)$ of the group we really are interested in. If the number of parameters to be estimated equals one (as is the case for image rotations), this single parameter can be estimated from one correspondence pair. Whether it is actually represented as a single number or as a distributed representation as in the gating models is not relevant. Already for two-parameter groups such as x, y-translation of an image, it becomes impossible to estimate the transformation parameters from a single correspondence pair. This leads to the well-known aperture problem, which says that for images with translational symmetry, the x, y-translation parameters cannot be determined uniquely.

In the section 5.3, we discuss how subgroups can be learned in order to reduce the ambiguity in the transformation estimates as obtained from a single correspondence pair. However, a non-trivial stabilizer is not only a problem: it also provides valuable information about the signal. We can ask, "what are the transformations in $\mathbb{T}^M(\mathbf{W})$ that take $\mathbf{x}$ to $\mathbf{x}$?" The answer is $\texttt{stab}(\mathbf{x})$, characterized by κ' inferred from the pair $(\mathbf{x}, \mathbf{x})$. So even if we are really interested in a subgroup $G \triangleleft \mathbb{T}^M(\mathbf{W})$, the symmetries of $\mathbf{x}$ with respect to the centralizer of G in $SO(N)$ (a maximal torus, $\mathbb{T}^M(\mathbf{W})$), provide a maximal invariant to transformations from G. For example, the equilateral triangle in figure 1.2, call it $\mathbf{x}$, has three rotational symmetries. If we rotate the image, our uncertainty about the transformation that took $\mathbf{x}$ to $\mathbf{x}$ remains the same: it is an invariant representation of the triangle.

5.2 Lie subalgebras and phase wrapping

When the subspace angles φ_j are dependent, as they must be when the correspondence pairs are related by transformations from a non-trivial subgroup of the toroidal group, we can try to predict the angles in underdetermined subspaces from those where $\|\mathbf{u_j}\|$ is large. What model should we use for these dependencies? A cursory look at the data would suggest that we need a fairly complex model, and indeed all related work where similar problems are encountered use high-dimensional filter banks to model the relations between subspace angles [Cadieu and Olshausen, 2012, Memisevic, 2013] (see section 3.2.2 and 3.2.3 for details).

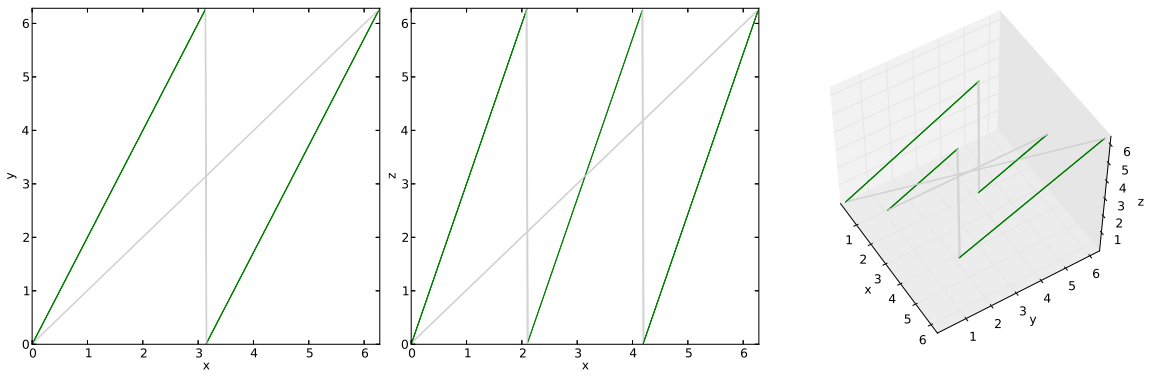

Figure 5.1: Example of phase-wrapping. Points on the dark green line correspond to vectors φ in three dimensions. The x, y, z axes correspond to $\varphi_1, \varphi_2, \varphi_3$. The slope of the line is $(1, 2, 3)$, corresponding to the values of the weights a_1, a_2, a_3 (see text). Left, middle: projection onto the x-y and x-z planes, right: 3D x-y-z space.

Can we do better? Remember that while any group can serve as a group of symmetries, it must indeed be a group; requesting invariance to some arbitrary set of transformations does not make sense. So while we are free to model high density regions of φ-space, *the set of transformations corresponding to this region must form a group*. As we have seen in section 2.2.2, subgroups H of a toroidal group $\mathbb{T}^M(\mathbf{W})$ correspond to linear subspaces $\mathfrak{h}$ of the Lie algebra $\mathfrak{t}$ of $\mathbb{T}^M(\mathbf{W})$. Since the vector φ corresponds to an element on the Lie algebra $\mathfrak{t}$, linear subspaces of φ-space correspond to subgroups of $\mathbb{T}^M(\mathbf{W})$ through the exponential map $\mathbf{Q} = \exp \sum_j \varphi_j \mathbf{B}_j = \mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T$, where $\mathbf{B}_j$ is the generator of rotations in subspace j: $\mathbf{B}_j = \mathbf{W}_{\cdot 2j}\mathbf{W}^T_{\cdot 2j+1} - \mathbf{W}_{\cdot 2j+1}\mathbf{W}^T_{\cdot 2j}$.

There is however one major complication. For a fixed $\mathfrak{h}$, any $\varphi \in \mathfrak{h}$ corresponds to an element in H through the exponential map, but doing bottom up inference of φ from correspondence pairs $\mathbf{y} = \mathbf{Q}\mathbf{x}$ will not necessarily yield $\varphi \in \mathfrak{h}$. The reason is that an element of $\mathfrak{h}$ may correspond to an angle φ_j that is outside $[0, 2\pi]$, but bottom-up inference will always yield an angle in that range. Consider a 1-D subalgebra of $\mathfrak{t}$, generated by an element $\mathbf{A} \in \mathfrak{t}$: $\mathbf{A} = \sum_j a_j \mathbf{B}_j$. The elements of the subgroup are given by $\mathbf{H} = \exp s\mathbf{A} = \exp\left(s \sum_j a_j \mathbf{B}_j\right)$. Whenever $sa_j > 2\pi$, the inferred phase φ_j wraps, and so the observed vectors φ will not lie on a line. Figure 5.1 shows a visual example in 3 dimensions.

Roughly speaking, current algorithms such as [Memisevic, 2012] deal with the piecewise linear space of angles using a large filter bank. A more parameter-efficient approach that will also yield a more comprehensible representation of the transformation, is to learn the weights a_j and introduce a new random variable corresponding to the group parameter s. Recall from section 2.2.5, that the weights must be integers

in the case of a representation of $SO(2)$, where $s \in [0, 2\pi]$. If we let the weights be real numbers, the learned subgroup can become non-compact and non-periodic by choosing weights that are irrational. (Of course, truly irrational numbers do not exist in a finite machine, but nearly-irrational weights can still yield impractically large periods.)

We have experimented with various approaches for learning the weights, both in the real-valued case (which may be easier to deal with even if we know we are only interested in integral weights) and integral case. The first approach we tried (section 5.3.1) is to explicitly represent the unwrapped phase variables, who in turn are connected to a group parameter. Another approach is to connect the means of the wrapped phases directly to a group parameter. We present this approach in section 5.3.2.

So far, these attempts have been moderately but not completely successful. Learning the weights reliably and correctly turns out to be surprisingly difficult. The main difficulty is in the exponentially large search space. When each weight is allowed to take values $1, \ldots, K$, there are K^M possible joint assignments to the weights $a_1, \ldots a_M$ for a single 1-parameter subgroup. For a 16×16 image patch and $K = 10$, this gives $128^{10} = 2^{70}$ possibilities. For the continuous case, we can use gradient descent in the 128 dimensional space, but this space appears to have many local optima, due to the periodicities of the objective function.

So far, we have only worked with 1-parameter subgroups, but it is conceptually straightforward to generalize to higher dimensional subgroups.

5.3 Learning one-parameter subgroups of a torus

5.3.1 Real-valued weights with explicit phase unwrapping

We introduce a new vector of real-valued parameters $\mathbf{a} = (a_1, \ldots, a_M)^T$ that will represent the weights of the irreducible representations. Next, we introduce a set of unwrapped phases $\bar{\varphi}_j$ that determine the mean of the von Mises distribution over wrapped phases φ_j: $p(\varphi_j) = \mathcal{VM}(\varphi_j|\bar{\varphi}_j, \kappa)$. The unwrapped phases are assumed to be normally distributed with mean determined by $\mathbf{a}$ and a latent variable s corresponding to the group parameter: $p(\bar{\varphi}|s, a_j, \bar{\sigma}^2) = \mathcal{N}(\bar{\varphi}|a_j s, \bar{\sigma}^2)$. Since the group parameter need not be periodic, we use a Gaussian prior on it: $p(s) = \mathcal{N}(s|\mu_s, \sigma_s^2)$.

To perform EM learning, we need to optimize an expectation with respect to the posterior $p(\varphi, \bar{\varphi}, s|\mathbf{x}, \mathbf{y}, \mathbf{a}, \mathbf{W})$. This is probably not possible analytically, so we use a

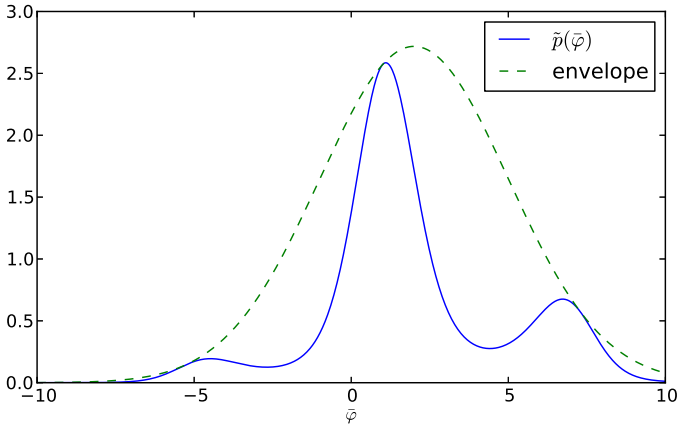

Figure 5.2: The unnormalized conditional distribution of $\bar{\varphi}$ given its neighbors, wrapped by a scaled Gaussian.

sampling approximation

$$\mathbb{E}[\ln p(\varphi, \bar{\varphi}, s, \mathbf{y}|\mathbf{x}, \mathbf{\Theta})]_{\varphi,\bar{\varphi},s|\mathbf{x},\mathbf{y},\mathbf{\Theta}'} \approx \frac{1}{L}\sum_{l=1}^{L} \ln p(\varphi^{(l)}, \bar{\varphi}^{(l)}, s^{(l)}, \mathbf{y}|\mathbf{x}, \mathbf{\Theta}), \tag{5.2}$$

where $\mathbf{\Theta}$ represents all the parameters in the model and the samples l are drawn from the posterior over latent variables given data and old parameter estimates. This algorithm is known as Monte Carlo EM (MC-EM), [Levine and Casella, 2001].

An easy way to obtain samples from the posterior is to use Gibbs sampling. That is, we iteratively sample from the conditionals of each variable given its neighbors in their current state. The conditional $p(\varphi|\bar{\varphi}, \mathbf{x}, \mathbf{y})$ is a von Mises distribution derived using the update equations in eq. 4.7, using $\mu = \bar{\varphi}_j$. The conditional $p(s|\bar{\varphi})$ is a Gaussian and,

$$p(\bar{\varphi}_j|\varphi_j, s) \propto \exp\left(\kappa\cos(\varphi_j - \bar{\varphi}_j) - \frac{1}{2\bar{\sigma}^2}(\bar{\varphi}_j - a_j s)^2\right). \tag{5.3}$$

Because the periodic part can be bounded, this distribution can be enveloped by a scaled Gaussian, as shown in figure 5.2. This allows us to perform rejection sampling of $\bar{\varphi}$.

5.3.2 Integral weights

Since we are interested in *compact* subgroups of the torus, it is arguably better to constrain the a_j to be integer valued. We will also connect the group parameter s directly to the φ, making explicit phase-unwrapping unnecessary.

The prior on φ_j in the basic TSA model is replaced by a von Mises distribution whose mean is determined by a_j and s:

$$p(\varphi_j|s, a_j, \kappa_j) = \mathcal{VM}(\varphi_j|a_j s, \kappa_j). \tag{5.4}$$

The prior on s must now also be a periodic distribution, so we use a von Mises: $p(s|\nu, \gamma) = \mathcal{VM}(s|\nu, \gamma)$.

For EM learning, we again require samples from the posterior $p(\varphi, s|\mathbf{x}, \mathbf{y}, \mathbf{\Theta})$. It is possible to integrate out φ, but the resulting density involves a Bessel function of a complicated function of s, making rejection sampling difficult. We have not tried it yet, but importance sampling may work well.

We now describe a Gibbs sampling scheme for this model. Suppressing parameters, we have $p(\varphi|\mathbf{x}, \mathbf{y}, s) \propto p(\varphi|s)p(\mathbf{y}|\mathbf{x}, \varphi)$, which is a von Mises times a polar-Gaussian, and hence is again a von Mises with parameters μ', κ' (as derived in appendix A). The other conditional required for Gibbs sampling is

$$p(s|\varphi) \propto p(\varphi|s)p(s) \propto \exp\left(\gamma \cos(s - \nu) + \sum_j \kappa_j \cos(\varphi_j - a_j s)\right). \tag{5.5}$$

In general, there can be multiple a_j with the same integer value, and we can group these together using the formula for a linear combination of cosines (see appendix A). This yields a distribution of the form $p(s|\varphi) \propto \exp\left(\sum_{j=1}^{F} \rho_j \cos(j(s - \delta_j))\right)$, which is known as a generalized von Mises distribution [Gatto and Jammalamadaka, 2007, Gatto, 2008].

We can again use rejection sampling to sample from this unnormalized distribution, because it can be bounded by a constant function $\exp\left(\sum_{j=1}^{F} \rho_j\right)$. If we want to sample from the same distribution specified in the form of eq. 5.5, the bounding constant becomes $\exp\left(\gamma + \sum_{j=1}^{M} \kappa_j\right)$. Rejection sampling then proceeds as follows:

1. Sample a proposal sample $s_0 \sim \mathcal{U}(0, 2\pi)$.

2. Sample $u_0 \sim \mathcal{U}(0, \exp\left(\gamma + \sum_{j=1}^{M} \kappa_j\right))$. The pair (s_0, u_0) is distributed uniformly under the uniform proposal distribution.

3. Let $\tilde{p}(s)$ denote the unnormalized distribution from which we wish to sample. If $u_0 > \tilde{p}(s_0)$, reject the sample and repeat from step 1. Otherwise accept s_0 as a sample.

Having obtained a set of samples $\{\varphi^{(l)}, s^{(l)}_{l=1\ldots L}\}$, we optimize the expected complete data log-likelihood,

$$\frac{1}{L}\sum_{l=1}^{L} \ln p(\varphi^{(l)}, s^{(l)}, \mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \frac{1}{L}\sum_{l=1}^{L} -\frac{1}{2\sigma_n^2}\|\mathbf{y} - \mathbf{W}\mathbf{R}(\varphi^{(l)})\mathbf{W}^T\mathbf{x}\|^2 - 2M\ln\sigma_n \quad (5.6)$$

$$+ \sum_{j=1}^{M} \kappa_j \cos(\varphi_j^{(l)} - a_j s^{(l)}) - \ln 2\pi I_0(\kappa_j) \quad (5.7)$$

$$+ \gamma\cos(s^{(l)} - \nu) - \ln 2\pi I_0(\gamma). \quad (5.8)$$

This objective must be summed over the whole data set $i = 1 \ldots D$. The objective is easily optimized by gradient descent with respect to σ_n, κ_j, ν, γ. The gradient with respect to $\mathbf{W}$ is the same as derived in the previous chapter, except that it is now summed over l. Optimization with respect to a_j can be done by exhaustively checking every value in some range $a_j = 0 \ldots F$. However, in practice, when κ_j is large the samples of φ_j tend to be closely aligned with $a_j s$ so that the optimal value of a_j is not changed, and when κ_j is small the values of a_j fluctuate randomly. We have various alternatives, such as making a_j a random variable and sample it (either by itself or jointly with s), but have so far not found an algorithm that consistently yields good results. We hope to improve this in the future, but for now we will only perform experiments with the continuous-weight version of the algorithm presented in section 5.3.1.

Chapter 6

Experiments

6.1 Random data

To check the correctness of the basic TSA algorithm described in chapter 4, we generated random 18-dimensional vectors $\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $i = 1 \dots 10000$ and rotated each one by a random matrix of the form

$$\mathbf{R}^{(i)} = \begin{pmatrix} \mathbf{R}(\varphi_1^{(i)}) & & \\ & \ddots & \\ & & \mathbf{R}(\varphi_9^{(i)}) \end{pmatrix}, \tag{6.1}$$

where $\varphi^{(i)} \sim \mathcal{U}(0, 2\pi)$ and $\mathbf{R}(\varphi_j^{(i)})$ is a 2×2 rotation matrix that rotates by angle $\varphi_j^{(i)}$. The matrices $\mathbf{R}^{(i)}$ are elements of the torus $\mathbb{T}^9(\mathbf{I})$ since $\mathbf{W} = \mathbf{I}$. We initialized W at a random orthogonal matrix and trained the TSA model until convergence. Figure 6.1 shows the learned matrix $\mathbf{W}$ with high values corresponding to light pixels and low values corresponding to dark pixels. As expected, there are two types of non-identifiability in the matrix $\mathbf{W}$. First, the blocks are permuted in a random order and second, because there is no inherent ordering of 2D subspaces. Secondly, each 2×2 block is orthogonal but not necessarily equal to the identity $\mathbf{I}_{2\times 2}$, because only the *subspace* but not the *basis* can be identified. Finally, notice that in each block, either the diagonal or anti-diagonal values are equal, while the other (anti-diagonal, resp. diagonal) values are opposite. This corresponds to a pure rotation and an anti-rotation (also known as a roto-reflection), which is a reflection (which is orthogonal) followed by a rotation.

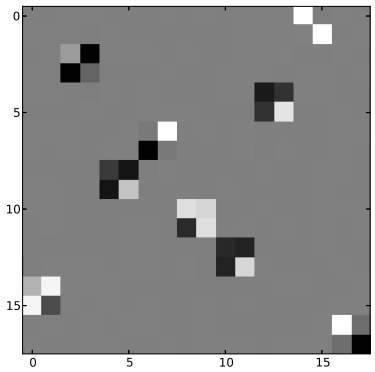

Figure 6.1: Retrieved block-matrix $\mathbf{W}$. Lighter pixels correspond to higher values.

6.2 Learning image filters

Next, we trained the model on randomly generated image patches, that were rotated or cyclicly shifted by random amounts. To be precise, a random patch was sampled from an isotropic Gaussian, $\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then the corresponding patch $\mathbf{y}^{(i)}$ was generated from $\mathbf{x}^{(i)}$ by rotation by a uniformly random angle $\theta \sim \mathcal{U}(0, 2\pi)$ or shifted by $s_x, s_y \sim \mathcal{U}(0, w)$ where $w = 16$ is the width of the patch. The result is shown in figure 6.2.

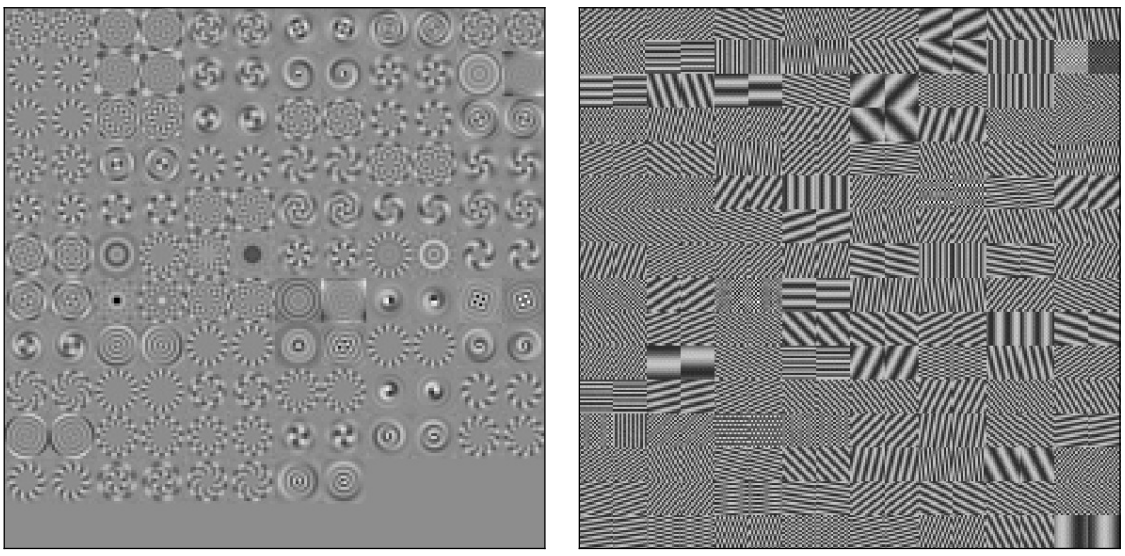

Figure 6.2: Filters learned from random rotations and shifts.

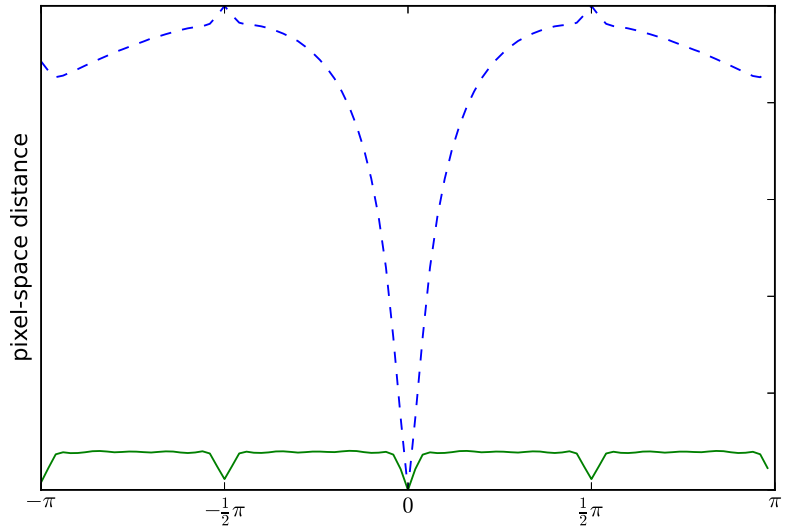

Figure 6.3: Blue, dashed line: distance of a reference patch (center) to rotated copies of it. Green, solid line: distance between a reference patch and the back-rotation by $\mathbf{WR}(\mu')\mathbf{W}^T$ of a rotated copy (where μ' is the vector of inferred subspace angles).

6.3 Testing rotation invariance

To test the theoretically derived invariance properties of the representation, we performed the following experiment. We trained the basic TSA model on randomly rotated image patches, as in section 6.2. Then, we selected 10.000 reference patches from the MNIST data set of handwritten digits [LeCun and Bottou, 1998], scaled them to 16×16 pixels, and rotated each of them by an angle that increased from $-\pi$ to π in 100 steps. For each one of the 100 angles θ, we computed the average distance between the reference patches $\mathbf{x}^{(i)}$ and their rotated copies $\mathbf{y}^{(i,\theta)}$ $(i = 1 \ldots 10000)$. Next, we computed the posterior mean angle $\mu^{(i,\theta)}$ for each pair $\mathbf{x}^{(i)}, \mathbf{y}^{(i,\theta)}$, and computed $\mathbf{z}^{(i,\theta)} = \mathbf{WR}(\mu^{(i,\theta)})\mathbf{W}^T\mathbf{x}^{(i)}$. Figure 6.3 shows the average distance $\|\mathbf{x}^{(i)} - \mathbf{y}^{(i,\theta)}\|$ and $\|\mathbf{x}^{(i)} - \mathbf{z}^{(i,\theta)}\|$ for each angle θ.

6.4 Rotation invariant classification

We performed an experiment to see if the learned features are useful for classification. We classified rotated handwritten digits from the MNIST-rot data set [Larochelle et al., 2007], which were rescaled from 28×28 to 16×16, because we did not have time to train a full basis $\mathbf{W}$ of size 784×784. This data set consists of 10 classes of handwritten digits, which were rotated by random angles $\theta \in [0, pi]$. The data set has 12000 training examples and 50000 test examples.

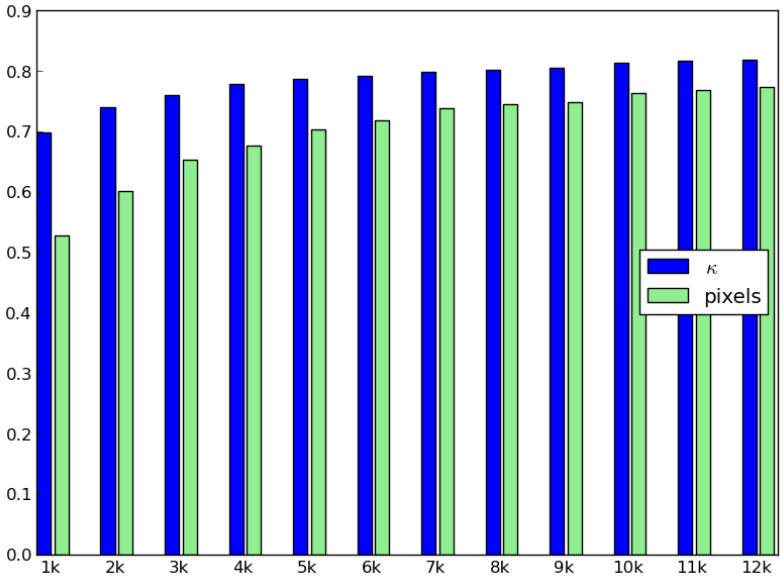

Figure 6.4: Classification results obtained on rescaled MNIST-rot, using the raw pixel representation and the invariant κ representation.

Using a basis $\mathbf{W}$ trained as in section 6.2, we obtained an invariant representation from each digit $\mathbf{x}^{(i)}$ by computing the posterior precision vector $\kappa^{(i)}$ of the transformation from $\mathbf{x}^{(i)}$ to itself. The prior means and precisions were set to 0. As explained in section 5.1, the vector $\kappa^{(i)}$ obtained in this way describes the symmetries of the image with respect to the toroidal group. Since a transformation from the learned group (which includes all image rotations) does not change the symmetries of the figure, this representation is invariant.

Figure 6.4 shows classification results for various training set sizes and two data representations: the invariant representation κ and the raw pixel values. These were obtained using a random forest classifier with 100 trees. The invariant representation performs better for every training set size, but the benefit is most pronounced for small training sets. It should be noted that our results are lower than those reported in [Larochelle et al., 2007], both for the invariant and raw pixel representations, which is most likely due to having rescaled the images.

6.5 Subgroup learning

We tested the subgroup-learning algorithm presented in section 5.3.1 on image rotations. The 2D rotation group $SO(2)$, being compact, connected and commutative, is

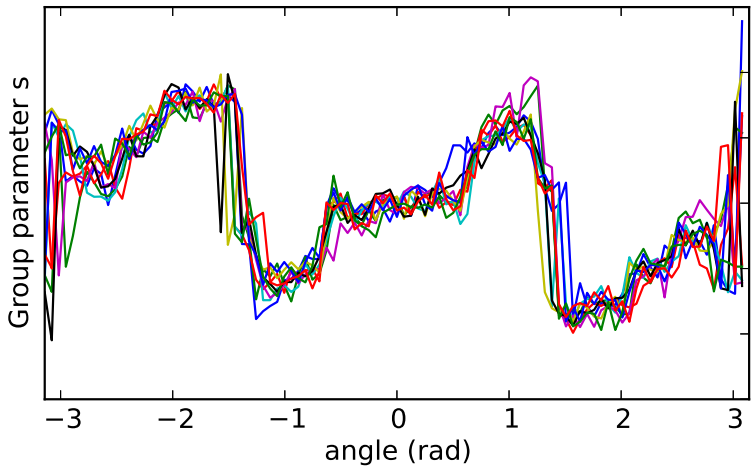

Figure 6.5: Inferred s-value for various rotation angles. Each line corresponds to a digit image.

a subgroup of some maximal torus (which is in $SO(N)$, where N is the dimensionality of the pixel space).

We trained the algorithm on random rotations of random 16×16 image patches, reduced by PCA to 64 dimensions[1]. Then, we chose one MNIST digit from each class 09, and rotated it by 100 angles in $[-\pi, \pi]$e. Figure 6.5 shows one line for each digit, with the angle between the reference digit and the rotated copy shown on the x-axis, and the sampled value of s after 10 steps of Gibbs sampling on the y-axis. Notice that the lines are largely overlapping, showing that the same representation (s-value) is used to represent the same rotation on different images. Furthermore, notice that there is still a phase-wrap at $\frac{\pi}{2}$ and $-\frac{\pi}{2}$, which shows that the model has failed to make use of the lowest-frequency filters. This means that it will have the same internal representation of two different rotations, and therefore will not be able to apply this transformation properly to a new image. Other than that, the model has linearized the 1-parameter group; it has discovered the natural concept "rotation angle".

[1]The training data will indeed lie in a low-dimensional subspace, but this will not be so for new randomly drawn images. However, as the results indicate it does work well enough on digit images.

Chapter 7

Conclusion and Outlook

We have presented an analysis of group theoretical methods in representation learning, and a model and algorithm for learning representations toroidal groups. Our work on learning subgroups of a maximal toroidal group has so far been only moderately successful, and we hope to improve this in the future. Doing so would make it possible to create convolutional networks whose architecture is informed by group representation theory, thus opening the door for application of our work to more serious problems such as object and action recognition in full-size images and videos. A sample of other interesting topics for future work:

1. Modeling non-commutative groups, possibly using toroidal groups as building blocks.

2. In our work so far we have assumed that the pairs $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ are all independent, but it is also possible to consider sequence models with the same conditional independencies as HMM/LDS models but using emission probabilities defined by TSA.

3. Build sparse versions of TSA using a kurtotic prior on κ and an overcomplete dictionary $\mathbf{W}$.

4. A full Bayesian analysis, i.e. using priors over the parameters and sampling them from their posterior (taking into account orthogonality constraints).

Appendix A

Derivation of von Mises-Polar Gaussian conjugacy

Assume a von Mises prior on φ:

$$p(\varphi|\mu,\kappa) = \mathcal{VM}(\varphi|\mu,\kappa) = \frac{e^{\kappa\cos(\varphi-\mu)}}{2\pi I_0(\kappa)} \tag{A.1}$$

$$\tag{A.2}$$

where I_0 is the order-0 modified Bessel function of the first kind. Furthermore, let $\mathbf{u}$ and $\mathbf{v}$ be 2D vectors and assume a circularly parameterized Gaussian distribution on $\mathbf{v}$:

$$p(\mathbf{v}|\mathbf{u},\varphi,\sigma) = \mathcal{N}(\mathbf{v}|\mathbf{R}(\varphi)\mathbf{u},\sigma^2) \tag{A.3}$$

where

$$\mathbf{R}(\varphi) = \begin{bmatrix}\cos(\varphi) & -\sin(\varphi)\\ \sin(\varphi) & \cos(\varphi).\end{bmatrix} \tag{A.4}$$

The posterior distribution in is given by Bayes rule

$$\begin{aligned} p(\varphi|\mathbf{u},\mathbf{v}) &= \frac{p(\mathbf{v}|\mathbf{u},\varphi)p(\varphi)}{\int_0^{2\pi} p(\mathbf{v}|\mathbf{u},\varphi)p(\varphi)d\varphi} \\ &\propto \exp\left(\kappa\cos(\varphi-\mu) - \frac{1}{2\sigma^2}\|\mathbf{v}-\mathbf{R}(\varphi)\mathbf{u}\|^2\right) \end{aligned} \tag{A.5}$$

Notice that

$$\|\mathbf{v}-\mathbf{R}(\varphi)\mathbf{u}\|^2 = \mathbf{v}^{\mathrm{T}}\mathbf{v} + \mathbf{u}^{\mathrm{T}}\mathbf{u} - 2\mathbf{v}^{\mathrm{T}}\mathbf{R}(\varphi)\mathbf{u}. \tag{A.6}$$

The terms in eq. A.6 that do not depend on φ can be factored out of the normalizing integral of eq. A.5 as well as the numerator, and so they cancel. This gives us the posterior:

$$p(\varphi|\mathbf{u},\mathbf{v}) \propto e^{\kappa\cos(\varphi-\mu)+\sigma^{-2}\mathbf{v}^{\mathrm{T}}\mathbf{R}(\varphi)\mathbf{u}} \tag{A.7}$$

It turns out this is again a von Mises distribution, although it is not obvious in the current form. To get the update equations for μ and κ, we write out the simplified exponent of in terms of sines, as follows:

$$\begin{aligned}
&\kappa\cos(\varphi-\mu)+\sigma^{-2}\mathbf{v}^{\mathrm{T}}\mathbf{R}(\varphi)\mathbf{u}\\
&=\kappa\cos(\varphi-\mu)+\sigma^{-2}\left(v_1\cos(\varphi)u_1+v_2\sin(\varphi)u_1-v_1\sin(\varphi)u_2+v_2\cos(\varphi)u_2\right)\\
&=\kappa\sin(\varphi-\mu+\frac{1}{2}\pi)+\sigma^{-2}\left(v_1u_1\sin(\varphi+\frac{1}{2}\pi)+v_2u_1\sin(\varphi)-v_1u_2\sin(\varphi)+v_2u_2\sin(\varphi+\frac{1}{2}\pi)\right).
\end{aligned} \tag{A.8}$$

Now we apply the following identity:

$$\sum_i a_i\sin(\varphi+\delta_i)=a\sin(\varphi+\delta), \tag{A.9}$$

where $a^2=\sum_{ij}a_ia_j\cos(\delta_i-\delta_j)$ and

$$\tan\delta=\frac{\sum_i a_i\sin(\delta_i)}{\sum_i a_i\cos(\delta_i)}, \tag{A.10}$$

making the substitution and simplifying, we find the parameters of the von Mises posterior:

$$\begin{aligned}
\kappa'&=\sqrt{\frac{2\kappa\sigma^2\left(\mathbf{v}^{\mathrm{T}}\mathbf{R}(\mu)\mathbf{u}\right)+\|\mathbf{u}\|^2\|\mathbf{v}\|^2}{\sigma^4}+\kappa^2},\\
\mu'&=\tan^{-1}\left(\frac{(v_1u_2-v_2u_1)\sigma^{-2}+\kappa\sin(\mu)}{(v_1u_1+v_2u_2)\sigma^{-2}+\kappa\cos(\mu)}\right).
\end{aligned} \tag{A.11}$$

Appendix B

EM Equations

B.1 Expected rotation matrix under von Mises distributed angle

Here we derive the formula for the expected 2×2 rotation matrix under a von Mises distributed rotation angle. This result will be useful later.

Let φ be a scalar angle, and

$$\mathbf{R}(\varphi) = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \tag{B.1}$$

We show that

$$\int_0^{2\pi} \frac{\exp\left(\kappa \cos(\varphi - \mu)\right)}{2\pi I_0(\kappa)} \mathbf{R}(\varphi)\, d\varphi = \frac{I_1(\kappa)}{I_0(\kappa)} \mathbf{R}(\mu). \tag{B.2}$$

First, substitute $u = \varphi - \mu$:

$$
\begin{aligned}
&\int_0^{2\pi} \frac{\exp\left(\kappa\cos(u)\right)}{2\pi I_0(\kappa)} \mathbf{R}(u+\mu)\, du \\
&= \frac{1}{2\pi I_0(\kappa)} \left[\int_0^{2\pi} \exp\left(\kappa\cos(u)\right)\mathbf{R}(u)\, du\right] \mathbf{R}(\mu) \\
&= \frac{1}{2\pi I_0(\kappa)} \left(\left[\int_0^{2\pi} \exp\left(\kappa\cos(u)\right)\cos(u) du\right] - \left[\int_0^{2\pi} \exp\left(\kappa\cos(u)\right)\sin(u) du\right] \mathbf{B}\right) \mathbf{R}(\mu) \\
&= \frac{1}{2\pi I_0(\kappa)} \left(\left[\int_0^{2\pi} \frac{d}{d\kappa}\exp\left(\kappa\cos(u)\right) du\right] + \left[\int_0^{2\pi} \exp\left(\kappa\cos(u)\right) d\cos(u)\right] \mathbf{B}\right) \mathbf{R}(\mu) \\
&= \frac{1}{2\pi I_0(\kappa)} \left[\frac{d}{d\kappa}\int_0^{2\pi} \exp\left(\kappa\cos(u)\right) du\right] \mathbf{R}(\mu) \\
&+ \frac{1}{2\pi I_0(\kappa)} \left[\frac{\exp\left(\kappa\cos(u)\right)}{\kappa}\right]_0^{2\pi} \mathbf{B}\mathbf{R}(\mu) \\
&= \frac{1}{I_0(\kappa)} \frac{d}{d\kappa} I_0(\kappa)\mathbf{R}(\mu) + 0 \\
&= \frac{I_1(\kappa)}{I_0(\kappa)}\mathbf{R}(\mu),
\end{aligned}
\tag{B.3}
$$

Where we defined

$$
\mathbf{B} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{B.4}
$$

We also used the standard results:

$$
\begin{aligned}
\int_0^{2\pi} \exp(\kappa\cos(u)) du &= 2\pi I_0(\kappa) \\
\frac{d}{d\kappa} I_0(\kappa) &= I_1(\kappa).
\end{aligned}
\tag{B.5}
$$

B.2 Derivation & optimization of EM objective

We now derive the equations required for EM learning in the basic TSA model.

Let $\mathbf{X}$, $\mathbf{Y}$, Φ, M, K be matrices containing $\mathbf{x}^{(i)}$, $\mathbf{y}^{(i)}$, $\varphi^{(i)}$, $\mu^{(i)}$, $\kappa^{(i)}$ in the i-th column. Let $\mathbf{W_j}$ denote the two columns in $\mathbf{W}$ spanning subspace j.

At step t of the algorithm, do:

E step: compute the parameters $\mu_j^{(i)}$ and $\kappa_i^{(i)}$ of the posterior $p(\varphi_j^{(i)}|\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{W_j}^{(t-1)}, \mu, \kappa)$, using the update equations derived in appendix A.

M step: we first define $\mathcal{Q}$ as the expected complete data log-likelihood under the posterior over φ, and then optimize it with respect to $\mathbf{W}$.

$$\begin{aligned}\mathcal{Q}(\mathbf{W}, \mathbf{W}^{(t-1)}) &= \mathbb{E}[\ln p(\mathbf{Y}|\Phi, \mathbf{X}, \mathbf{W}) p(\Phi|\mu, \kappa)] \\ &= \sum_i \mathbb{E}[\ln p(\mathbf{y}^{(i)}|\varphi^{(i)}, \mathbf{x}^{(i)}, \mathbf{W}) + \ln p(\varphi^{(i)}|\mu, \kappa)]\end{aligned} \tag{B.6}$$

where the expectation is taken with respect to the posterior of φ, which depends on $\mathbf{W}^{(t-1)}$.

B.2.1 Optimization with respect to W

Only the likelihood term has a dependence on $\mathbf{W}$, so we leave out the prior term of equation B.6:

$$\mathbb{E}[\ln p(\mathbf{y}^{(i)}|\varphi^{(i)}, \mathbf{x}^{(i)}, \mathbf{W})] = \int_{\varphi^{(i)} \in \mathbb{T}^M(\mathbf{W})} p(\varphi^{(i)}|\mu^{(i)}\kappa^{(i)}) \ln p(\mathbf{y}^{(i)}|\varphi^{(i)}, \mathbf{x}^{(i)}, \mathbf{W}) d\varphi^{(i)} \tag{B.7}$$

Writing out the defining conditionals we find B.7 to equal

$$-\int_{\varphi^{(i)} \in \mathbb{T}^M(\mathbf{W})} \left[\prod_j \frac{\exp\left(\kappa_j^{(i)} \cos(\varphi_j^{(i)} - \mu_j^{(i)})\right)}{2\pi I_0(\kappa_j^{(i)})} \right] \frac{\|\mathbf{y}^{(i)} - \mathbf{W}\mathbf{R}(\varphi^{(i)})\mathbf{W}^T\mathbf{x}^{(i)}\|^2}{2\sigma_n^2} d\varphi^{(i)}. \tag{B.8}$$

We expand the squared norm to find the φ-dependent terms (suppressing i-indices):

$$\begin{aligned}\|\mathbf{y} - \mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x}\|^2 &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{y}^T\mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x} \\ &\quad + \mathbf{x}^T\mathbf{W}\mathbf{R}(\varphi)^T\mathbf{W}^T\mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x} \\ &= \|\mathbf{y}\|^2 + \|\mathbf{x}\|^2 - 2\mathbf{y}^T\mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x}\end{aligned} \tag{B.9}$$

The last step assumes that $\mathbf{W}$ is orthogonal, and so we must enforce this constraint later on. Since the first two terms $\|\mathbf{y}\|^2$ and $\|\mathbf{x}\|^2$ do not depend on φ, they can be taken out of the expectation. Since they don't depend on $\mathbf{W}$ either, they do not affect the position of the optimum and so we leave them out.

Since $\mathbf{R}(\varphi^{(i)})$ is block diagonal, we can split the bilinear form into a sum of M smaller bilinear forms:

$$\mathbf{y}^T\mathbf{W}\mathbf{R}(\varphi)\mathbf{W}^T\mathbf{x} = \sum_j \mathbf{y}^T\mathbf{W}_j\mathbf{R}(\varphi_j)\mathbf{W}_j^T\mathbf{x}, \tag{B.10}$$

where each term in the sum only depends on one of the M angles φ_j for data point i.

Leaving out the constant terms $\|\mathbf{x}\|^2$ and $\|\mathbf{y}\|^2$ and the constant factor $\frac{1}{2\sigma_n^2}$ out of eq. B.8, we are left to maximize

$$
\begin{aligned}
&\sum_i \int_{\varphi^{(i)} \in \mathbb{T}^M(\mathbf{W})} \left[\prod_j \frac{\exp\left(\kappa_j^{(i)} \cos(\varphi_j^{(i)} - \mu_j^{(i)})\right)}{2\pi I_0(\kappa_j^{(i)})} \right] \left[\sum_j \mathbf{y}^{(i)T} \mathbf{W_j} \mathbf{R}(\varphi_j^{(i)}) \mathbf{W_j}^T \mathbf{x}^{(i)} \right] d\varphi^{(i)} \\
&= \sum_{i,j} \int_{\varphi^{(i)} \in \mathbb{T}^M(\mathbf{W})} \left[\prod_l \frac{\exp\left(\kappa_l^{(i)} \cos(\varphi_l^{(i)} - \mu_l^{(i)})\right)}{2\pi I_0(\kappa_l^{(i)})} \right] \left(\mathbf{y}^{(i)T} \mathbf{W_j} \mathbf{R}(\varphi_j^{(i)}) \mathbf{W_j}^T \mathbf{x}^{(i)} \right) d\varphi^{(i)} \\
&= \sum_{i,j} \int_0^{2\pi} \left[\frac{\exp\left(\kappa_j^{(i)} \cos(\varphi_j^{(i)} - \mu_j^{(i)})\right)}{2\pi I_0(\kappa_j^{(i)})} \right] \left(\mathbf{y}^{(i)T} \mathbf{W_j} \mathbf{R}(\varphi_j^{(i)}) \mathbf{W_j}^T \mathbf{x}^{(i)} \right) d\varphi_j^{(i)} \\
&= \sum_{i,j} \mathbf{y}_i^T \mathbf{W_j} \int_0^{2\pi} \left[\frac{\exp\left(\kappa_j^{(i)} \cos(\varphi_j^{(i)} - \mu_j^{(i)})\right)}{2\pi I_0(\kappa_j^{(i)})} \mathbf{R}(\varphi_j^{(i)}) \right] d\varphi_j^{(i)} \;\; \mathbf{W_j}^T \mathbf{x}^{(i)} \\
&= \sum_{i,j} \frac{I_1(\kappa_j^{(i)})}{I_0(\kappa_j^{(i)})} \mathbf{y}^{(i)T} \mathbf{W_j} \mathbf{R}(\mu_j^{(i)}) \mathbf{W_j}^T \mathbf{x}^{(i)}.
\end{aligned}
\tag{B.11}
$$

The last step uses the result derived in section B.1.

Finally we evaluate the gradient:

$$
\frac{d}{d\mathbf{W_j}} \mathcal{Q}(\mathbf{W_j}, \mathbf{W_j}^{(t-1)}) = \sum_i \frac{I_1(\kappa_j^{(i)})}{2I_0(\kappa_j^{(i)})\sigma_n^2} \left[\mathbf{x}^{(i)} \mathbf{y}^{(i)\mathrm{T}} \mathbf{W_j} \mathbf{R}(\mu_j^{(i)} + \mathbf{y}^{(i)} \mathbf{x}^{(i)\mathrm{T}} \mathbf{W_j} \mathbf{R}(\mu_j^{(i)\mathrm{T}} \right].
\tag{B.12}
$$

(we reintroduced the factor $2\sigma_n^2$ that we left out before) Keep in mind that feasible solutions must satisfy orthogonality, so we optimize this by projected gradient descent.

Bibliography

[Abramowitz and Stegun, 1965] Abramowitz, M. and Stegun, I. A. (1965). *Handbook of Mathematical Functions*. Dover Pub.

[Adelson and Bergen, 1985] Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *J Opt Soc Am A*, 2(2):284–299.

[Aristotle, 1924] Aristotle (1924). *Metaphysics*. Clarendon Press.

[Bengio, 2013] Bengio, Y. (2013). Deep learning of representations: Looking forward. *arXiv preprint arXiv:1305.0445*.

[Bengio et al., 2013] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE transactions on pattern analysis and machine intelligence*, pages 1–30.

[Bengio and Lecun, 2013] Bengio, Y. and Lecun, Y. (2013). International Conference on Learning Representations.

[Bethge et al., 2007] Bethge, M., Gerwinn, S., and Macke, J. H. (2007). Unsupervised learning of a steerable basis for invariant image representations. *Proceedings of SPIE Human Vision and Electronic Imaging XII (EI105)*, pages 64920C–64920C–12.

[Bruna and Mallat, 2013] Bruna, J. and Mallat, S. (2013). Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–86.

[Cadieu and Olshausen, 2012] Cadieu, C. F. and Olshausen, B. a. (2012). Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–66.

[Cardoso, 1998] Cardoso, J. (1998). Multidimensional independent component analysis. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4.

[Cox et al., 2005] Cox, D. D., Meier, P., Oertelt, N., and DiCarlo, J. J. (2005). 'Breaking' position-invariant object recognition. *Nature Neuroscience*, 8(9):1145–1147.

[Curd, 2012] Curd, P. (2012). Presocratic Philosophy. *The Stanford Encyclopedia of Philosophy (Winter 2012 Edition), Edward N. Zalta (ed.),*.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4).

[DeGroot and Schervish, 2002] DeGroot, M. H. and Schervish, M. J. (2002). *Probability and Statistics*. Addison Wesley, 3rd edition.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

[DiCarlo and Cox, 2007] DiCarlo, J. J. and Cox, D. D. (2007). Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–41.

[DiCarlo et al., 2012] DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–34.

[Doran and Lasenby, 2003] Doran, C. and Lasenby, A. (2003). *Geometric algebra for physicists*. Cambridge University Press.

[Dorst et al., 2007] Dorst, L., Fontijne, D., and Mann, S. (2007). *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Edelman et al., 1998] Edelman, A., Arias, T. a., and Smith, S. T. (1998). The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353.

[Gatto, 2008] Gatto, R. (2008). Some computational aspects of the generalized von Mises distribution. *Statistics and Computing*, 18(3):321–331.

[Gatto and Jammalamadaka, 2007] Gatto, R. and Jammalamadaka, S. (2007). The generalized von Mises distribution. *Statistical Methodology*, 4(3):341–353.

[Hinton, 1979] Hinton, G. (1979). Some Demonstrations of the Effects of Structural Descriptions in Mental Imagery*. *Cognitive Science*, 3(3):231–250.

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 29(6):82–97.

[Holyoak and Morrison, 2005] Holyoak, K. J. and Morrison, R., editors (2005). *The Cambridge Handbook of Thinking and Reasoning*. Cambridge University Press, Cambridge, UK.

[Hyvarinen and Hoyer, 2000] Hyvarinen, A. and Hoyer, P. (2000). Emergence of Phase- and Shift-Invariant Features by Decomposition of Natural Images into Independent Feature Subspaces. *Neural Computation*, 1720:1705–1720.

[Hyvärinen and Oja, 2000] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks : the official journal of the International Neural Network Society*, 13(4-5):411–30.

[Jammalamadaka and Sengupta, 2001] Jammalamadaka, S. R. and Sengupta, A. (2001). *Topics in Circular Statistics*. World Scientific Pub Co Inc, har/dskt edition.

[J.P. Elliot, 1985] J.P. Elliot, P. D. (1985). *Symmetry in Physics: Principles and Simple Applications, Volume 1*. Oxford University Press, USA.

[Kanatani, 1990] Kanatani, K. (1990). *Group Theoretical Methods in Image Understanding*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Klein, 1893] Klein, F. (1893). A comparative review of recent researches in geometry. *Bulletin of the American Mathematical Society*, 1.

[Konda et al., 2013] Konda, K., Memisevic, R., and Michalski, V. (2013). The role of spatio-temporal synchrony in the encoding of motion. *arXiv preprint arXiv:1306.3162*, pages 1–9.

[Kondor, 2008] Kondor, R. (2008). *Group theoretical methods in machine learning Risi Kondor*. PhD thesis, Columbia University.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1–9.

[Larochelle et al., 2007] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*.

[Le et al., 2012] Le, Q., Ranzato, M., Monga, R., Devin, M., Kai, C., Corrado, G. S., Dean, J., and Ng, A. Y. (2012). Building high-level features using large scale unsupervised learning. *Proceedings of the 29th International Conference on Machine Learning*.

[Le et al., 2011] Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. Comput. Sci. Dept., Stanford Univ., Stanford, CA, USA, IEEE.

[LeCun and Bottou, 1998] LeCun, Y. and Bottou, L. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

[Leibo et al., 2011] Leibo, J., Mutch, J., and Poggio, T. (2011). Learning to discount transformations as the computational goal of visual cortex. *Nature Precedings*, pages 1–3.

[Levine and Casella, 2001] Levine, R. and Casella, G. (2001). Monte Carlo EM Algorithm. *Journal of Computational & Graphical Statistics*, 10(3):422–439.

[Li and DiCarlo, 2008] Li, N. and DiCarlo, J. J. (2008). Unsupervised Natural Experience Rapidly Alters Invariant Object Representation in Visual Cortex. *Science*, 321(5895):1502–1507.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Ma et al., 1999] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. (1999). Euclidean reconstruction and reprojection up to subgroups. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 773—-780 vol.2. California Univ., Berkeley, CA, IEEE.

[Mardia, 1975] Mardia, K. (1975). Statistics of directional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 37(3):349–393.

[Mardia and Jupp, 1999] Mardia, K. V. and Jupp, P. E. (1999). *Directional Statistics*. John Wiley & Sons, 1 edition.

[Memisevic, 2011] Memisevic, R. (2011). Gradient-based learning of higher-order image features. *2011 International Conference on Computer Vision*, pages 1591–1598.

[Memisevic, 2012] Memisevic, R. (2012). On multi-view feature learning. *International Conference on Machine Learning*.

[Memisevic, 2013] Memisevic, R. (2013). Learning to Relate Images. *IEEE transactions on pattern analysis and machine intelligence*, pages 1–19.

[Memisevic and Exarchakis, 2013] Memisevic, R. and Exarchakis, G. (2013). Learning invariant features by harnessing the aperture problem. *International Conference on Machine Learning*, 28.

[Memisevic and Hinton, 2010] Memisevic, R. and Hinton, G. E. (2010). Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines. *Neural Computation*, 22(6):1473–1492.

[Miao and Rao, 2007] Miao, X. and Rao, R. P. N. (2007). Learning the Lie groups of visual invariance. *Neural computation*, 19(10):2665–93.

[Mohamed et al., 2012] Mohamed, A.-r., Dahl, G., and Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE transactions on Audio, Speech, and Language Processing*, 20(1):14–22.

[Nathan Carter, 2009] Nathan Carter (2009). Visual Group Theory website.

[Piaget, 1952] Piaget, J. (1952). *The Origins of Intelligence in Children*. International University Press, New York.

[Plumbley, 2004] Plumbley, M. (2004). Lie group methods for optimization with orthogonality constraints. *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*.

[Poincaré, 1904] Poincaré, H. (1904). *Science and hypothesis*. The Walter Scott Publishing Co.

[Rao and Ruderman, 1999] Rao, R. and Ruderman, D. (1999). Learning Lie groups for invariant visual perception. *Advances in neural information processing systems*, 816:810–816.

[Serre et al., 2007] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE transactions on pattern analysis and machine intelligence*, 29(3):411–26.

[Soatto, 2009] Soatto, S. (2009). Actionable information in vision. *Machine learning for computer vision*, pages 2138–2145.

[Sohl-Dickstein et al., 2010] Sohl-Dickstein, J., Wang, J., and Olshausen, B. (2010). An unsupervised algorithm for learning lie group transformations. *http://arxiv.org/abs/1001.1027*.

[Susskind et al., 2011] Susskind, J., Hinton, G., Memisevic, R., and Pollefeys, M. (2011). Modeling the joint density of two images under a variety of transformations. *Cvpr 2011*, pages 2793–2800.

[Weyl, 1939] Weyl, H. (1939). *The classical groups: their invariants and representations*. Princeton University Press.