Generative Models and Sparse Coding

Zachary Siegel

November 9, 2016

Abstract

Sparse coding makes carefully chosen nonzero coefficients meaningful. The support of a sparse representation can then help classify data, as intersections in the support of data points may indicate a semantic relationship. On an ad-hoc basis, any sparse coding may help classify data, but enforcing or inferring the types of interactions between the support of different data points can further this goal. Hence, the Boltzmann Machine (BM) distribution, which generates sparse data points whose supports' interactions are specified. The parameters specifying these patterns do not have an immediate intuitive interpretation, but the are very closely related to intuitive characteristics of the data. The relationship between the BM parameters and more apparent statistics of sparsely coded data will be explained, as will the usefulness of those statistics, and hence the BM parameters. Intuitive methods for classifying sparsely coded data will thereby be grounded in a generative statistical framework.

Contents

1 Introduction to Sparse Coding and Dictionary Learning **2**
- 1.1 Low-Dimensionality Example . . . 2
  - 1.1.1 One Category: Spheres . . . 2
  - 1.1.2 Noise and Dimensionality Reduction . . . 3
  - 1.1.3 Sparsity and Classification: Shapes . . . 4

2 Sparse Models **6**
- 2.1 An Idea About Models . . . 6
  - 2.1.1 An Example . . . 6
- 2.2 Ways to Model Sparsity . . . 7
  - 2.2.1 Dictionary Learning, Compressive Sensing . . . 7

3 Sparse Coding of Noisy Data **8**

4 The Boltzmann Machine **9**
- 4.1 General Boltzmann Machine . . . 10
  - 4.1.1 Simulated Annealing: a Typical Application of the Boltzmann Machine . . . 10
- 4.2 Sparse Signals and the Boltzmann Machine Prior . . . 11
  - 4.2.1 The BM Generative Model . . . 11
  - 4.2.2 The Normal Distribution of Coefficients Given Support $(X|S)$ . . . 12
  - 4.2.3 Distribution of y around Ax . . . 12
- 4.3 Sparse Recoveries Given Parameters W, b . . . 13
- 4.4 Estimating BM Model Parameters W, b . . . 13
  - 4.4.1 Joint Estimation of Parameters and Representation . . . 14

4.5 Structured Sparsity 14
4.5.1 Block Sparsity 14
4.5.2 Hierarchical Sparsity 16

5 Data Classification **16**
5.1 Anomaly/Saliency Detection Using the Boltzmann Machine 16
5.2 Sensitivity and Specificity: A Rarely Used Atom, the BM Parameter b 17

1 Introduction to Sparse Coding and Dictionary Learning

Models lie in the background of all sparse coding applictions. Points in most data sets wind up geometrically following some pattern that isn't the 'uniform distribution in D-dimensional space'.

No data, even of a specific type, is precisely 'drawn from a distribution'. In the words of George E.P. Box, "all models are wrong, but some are useful." Instead, data points are imagined to come from a given distribution -'modeled' as such - and procedures that 'would work' on data from such a distribution are applied anyway.

One useful model is the assumption of low dimensionality. Under this model, data points may come with many components, i.e. lie in a high ambient dimension, but either some components are not important, or all points are some linear combination of the same few patterns of components.

1.1 Low-Dimensionality Example

1.1.1 One Category: Spheres

Suppose each data vector y represents a sphere of aluminum with density ρ, and the components look like this:

$$y = (\text{radius, diameter, circumference, surface area, volume, weight}).$$

Since

$$\begin{aligned}
\text{diameter} &= 2\times \quad \text{radius} \\
\text{circumference} &= 2\pi\times \quad \text{radius} \\
\text{surface area} &= 4\pi\times \quad \text{radius}^2 \\
\text{volume} &= \frac{4}{3}\pi\times \quad \text{radius}^3 \\
\text{weight} &= \frac{4}{3}\pi\rho\times \text{radius}^3
\end{aligned}$$

all data points are of the form

$$y = (r, 2r, 2\pi r, 4\pi r^2, \frac{4}{3}\pi r^3, \rho\frac{4}{3}\pi r^3)$$

and there is one degree of freedom for all the data points, despite data lying in 6-dimensional space! All data points lie on the one-dimensional curve parameterized by r.

If data is corrupted by noise, then while none of the data points' components will follow this pattern exactly, they should all follow it closely enough that, given a data vector, the real radius can be deduced reasonably well, perhaps by taking the mean of the radii implied by each component.

A dimensionality reduction to one degree of freedom (in this case, r) is desirable, but linear models are much more efficient to work with. Using a linear model, all data can be expressed as

$$\begin{aligned} y &= (r, 2r, 2\pi r, 4\pi r^2, \frac{4}{3}\pi r^3, \rho\frac{4}{3}\pi r^3) \\ &= \underbrace{r}_{a_1}\underbrace{(1,2,2\pi,0,0,0)}_{b_1} + \underbrace{r^2}_{a_2}\underbrace{(0,0,0,4\pi,0,0)}_{b_2} + \underbrace{r^3}_{a_3}\underbrace{(0,0,0,0,\frac{4}{3}\pi,\rho\frac{4}{3}\pi)}_{b_3} \\ &= a_1b_1 + a_2b_2 + a_3b_3 \end{aligned}$$

and all data points are approximated and represented in the span of b_1, b_2, and b_3. In this way, data that seemed to be 6-dimensional is now clearly 3-dimensional. The linear representation corresponding to this dimensionality reduction would be $y = a_1b_1 + a_2b_2 + a_3b_3$, ignoring that $a_2 = a_1^2$ and $a_3 = a_1^3$.

1.1.2 Noise and Dimensionality Reduction

If data is corrupted by noise, dimensionality reduction can reduce its impact! For example, suppose a data vector $y = (y_1, \ldots, y_6)$ is corrupted by white Gaussian noise. Then for noise $e = (e_1, \ldots, e_6) \sim N(0, \sigma^2)^6$,

$$\begin{aligned} y = (y_1, \ldots, y_6) &= a_1b_1 + a_2b_2 + a_3b_3 + e \\ &= a_1(1,2,2\pi,0,0,0) + a_2(0,0,0,4\pi,0,0) + a_3(0,0,0,0,\frac{4}{3}\pi,\rho\frac{4}{3}\pi) + (e_1,\ldots,e_6) \end{aligned}$$

or

$$\begin{aligned} y_1 &= a_1 + e_1 \\ &\vdots \\ y_6 &= \rho\frac{4}{3}\pi a_3 + e_3. \end{aligned}$$

The nature of the data, which is to say the assumed model, dictates that $y = a_1b_1 + a_2b_2 + a_3b_3$, but the six-dimensional noise might force y outside of the span of b_1, b_2, and b_3. How, then, can one infer the true radius of the represented sphere, or, as an easier problem, how can one infer the true values of a_1, a_2, and a_3 (ignoring that $a_1 = r, a_2 = r^2$, and $a_3 = r^3$).

Given six-dimensional noise $e \sim N(0, \sigma^2)^6$, we can say

$$P(e) \propto exp\left(\frac{-||e||_2^2}{2\sigma^2}\right).$$

Assuming that the uncorrupted value of the data is $y_0 = a_1b_1 + a_2b_2 + a_3b_3$, we see that $y = y_0 + e$, or $e = y - y_0$. Then, assuming that the noise e is independent of the fixed true value y_0, the probability of the data can be equated to that of the noise

$$P(y|y_0) = P(y_0 + e|y_0) = P(e|y_0) = P(e) \propto exp\left(\frac{-||e||_2^2}{2\sigma^2}\right) = exp\left(\frac{-||y - y_0||_2^2}{2\sigma^2}\right).$$

which is maximized when the MSE (L_2 distance) term in the exponent is minimized. Note that the value of the variance parameter σ^2 does not affect the MLE in this case of Gaussian noise.

To estimate $y_0 = a_1b_1 + a_2b_2 + a_3b_3$ given y, one approach is to use a maximum likelihood estimator (MLE), which is to find the value $\hat{y}$ that maximizes $P(y|y_0 = \hat{y})$ while being a possible

value of $y_0 \in \text{span}(b_1, b_2, b_3)$. Thus, an MLE is $\hat{y}$ that minimizes $||y - \hat{y}||_2^2$ such that $\hat{y} = \hat{a_1}b_1 + \hat{a_2}b_2 + \hat{a_3}b_3$. This amounts to producing estimates $\hat{a_1}, \hat{a_2}$, and $\hat{a_3}$ such that $\hat{y} = \hat{a_1}b_1 + \hat{a_2}b_2 + \hat{a_3}b_3$.

By basic linear algebra, the vector in the span of b_1, b_2, and b_3 with closest L_2 distance to y is the projection of y onto $\text{span}(b_1, b_2, b_3)$. In this case b_1, b_2, and b_3 are orthogonal, so this projection of y onto $\text{span}(b_1, b_2, b_3)$ can be written:

$$\begin{aligned}\widehat{y}_{\text{MLE}} &= (\hat{y_1}, \hat{y_2}, \hat{y_3}, \hat{y_4}, \hat{y_5}, \hat{y_6}) \\ &\approx \underbrace{\frac{< \overbrace{(1,2,2\pi,0,0,0)}^{b_1}, \overbrace{(y_1,y_2,y_3,y_4y_5,y_6)}^{y} >}{||(1,2,2\pi,0,0,0)||_2}}_{\widehat{a_1}} \overbrace{(1,2,2\pi,0,0,0)}^{b_1} + \underbrace{\frac{< b_2, y >}{||b_2||_2}}_{\widehat{a_2}} b_2 + \underbrace{\frac{< b_3, y >}{||b_3||_2}}_{\widehat{a_3}} b_3\end{aligned}$$

Noise is not necessarily Gaussian, and the spanning vectors b_1, b_2, and b_3 are not always known, and even if they are they may not be linearly independent, meaning that there are multiple likelihood-maximizing estimates (MLEs), and other criteria must be introduced to determine the *best* estimate. In this example, the estimate can be further refined by considering that $a_1 = \sqrt{a_2} = \sqrt[3]{a_3}$, which represent the radius of a sphere.

1.1.3 Sparsity and Classification: Shapes

Suppose now that data points can represent aluminum polygons of several types. For example, data points could represent spheres, as above, or cubes, via

$$y = (\text{side-length, face diagonal, cube diagonal, surface area, volume, weight}).$$

Then, as before, y can be placed in three dimensions

$$y = s\underbrace{(1, \sqrt{2}, \sqrt{3}, 0, 0, 0)}_{\text{new basis vector } b_4} + s^2\underbrace{(0,0,0,1,0,0)}_{\in \text{span}(b_2)} + s^3\underbrace{(0,0,0,0,1,\rho)}_{\in \text{span}(b_3)},$$

two of which are spanned by the same bases for the subspace in which spheres lie!

Now a data set representing cubes and spheres can be said to lie in a four-dimensional subspace of $\mathbb{R}^6$, while a subset containing a single type of shape will lie in just three dimensions.

Suppose there are many different types of shapes, and to represent them all as above, several bases are added to our collection, which becomes $\mathcal{B} = \{b_1, \ldots, b_K\} \subset \mathbb{R}^6$, and each polygon's description vector still can be written as a linear combination of three of these spanning vectors. The collection of spanning vectors $\mathcal{B}$ may have $K > 6$ elements, and so it more than spans $\mathbb{R}^6$, meaning they are more than a linearly independent basis. The collection $\mathcal{B}$ can be called an *overcomplete* basis, and points in the span of $\{b_1, \ldots, b_K\}$ do not have a unique representation with respect to $\mathcal{B}$. Thus, there is a high-dimensional subspace of MLEs for a given data point with Gaussian noise. That is, there is more than one way to describe a polygon using $\mathcal{B}$.

Let

- $B = (b_1|\ldots|b_K)$ be the matrix whose columns are (overcomplete) spanning vectors for the space in which descriptions of polygons lie
- $Y = (y_1|\ldots|y_N)$ be the matrix whose columns each represent a polygon
- $A = (\vec{a}_1|\ldots|\vec{a}_N)$ be a matrix whose columns are representations of the data points with respect to $\mathcal{B}$, i.e. $y_i \approx B\vec{a}_i$ or $Y = BA$

The goal is for each y_i, to find a representation $y_i \approx B\vec{a}_i$, i.e. to represent each data point y_i as a linear combination of the vectors in $\mathcal{B}$. Recall the assumption that each polygon can be described by an appropriate choice of *exactly three* of the bases of $\mathcal{B} = \{b_1, \ldots, b_K\}$. The problem then becomes:

$$\min ||Y - BA||_2^2 \text{ such that } ||\vec{a}_i||_0 \leq 3 \ \forall i \in \{1, \ldots, N\}$$

where $||\cdot||_0$ is the zero pseudonorm, which denotes the number of nonzero entries of a vector.

To summarize, the overcomplete basis $\mathcal{B}$ is such that the matrix B whose columns are its spanning vectors yields multiple solutions to the equation $Y = BA$. The desired solution has three nonzero coefficients, and the choice of bases corresponds to the type of polygon.

That is, suppose a given data point

$$y_i = a_{i1}b_1 + a_{i2}b_1 + a_{i3}b_3 = \underbrace{(b_1|b_2|b_3)}_{\text{Columns of } B} \quad \underbrace{\begin{pmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \end{pmatrix}}_{\text{nonzero terms of } \vec{a}_i}$$

then the choice of $\{b_1, b_2, b_3\} \subset \mathcal{B}$ indicates exactly that this shape is a sphere, which may be more relevant than the values themselves of the coefficients a_{i1}, a_{i2}, a_{i3}. This is especially true in this example where, given one of those 3 coefficients, the others may be its square or cube, meaning their information is not particularly telling.

To organize, given all the data points $y_1, \ldots, y_N$, each data point corresponds to some subset of (specifically 3, in this example) the overcomplete basis $\mathcal{B} = \{b_1, \ldots, b_K\}$. That is, every data point can be written $y_j = a_{j1}b_1 + \ldots + a_{jK}b_K$, or

$$y_j = B\vec{a}_j = B \begin{pmatrix} a_{j1} \\ \vdots \\ a_{jK} \end{pmatrix},$$

where only 3 of $a_{j1}, \ldots, a_{jK}$ are nonzero, and as we've seen, the indices of these 3 nonzero terms can carry valuable information. So, for each data point y_j, define the *support set* (or just *support*) $S_j \subset \{1, \ldots, K\}$ such that

$$S_j = \{k | a_{jk} \neq 0\}.$$

Similarly, for each data point y_j, define a *support vector* $\vec{s}_j$ whose k'th entry is

$$s_{jk} = \begin{cases} 1 & : k \in S_j \ (\Leftrightarrow a_{jk} \neq 0) \\ 0 & : k \notin S_j \ (\Leftrightarrow a_{jk} = 0) \end{cases}$$

and a *support matrix*

$$S = (\vec{s}_1| \ldots |\vec{s}_N).$$

Then there is a correspondence

$$A = (\vec{a}_1| \ldots |\vec{a}_N) = \begin{pmatrix} a_{11} & \cdots & a_{N1} \\ \vdots & \ddots & \vdots \\ a_{1K} & \cdots & a_{NK} \end{pmatrix} \sim \begin{pmatrix} s_{11} & \cdots & s_{N1} \\ \vdots & \ddots & \vdots \\ s_{1K} & \cdots & s_{NK} \end{pmatrix} = \vec{s}_i : (\vec{s}_1| \ldots |\vec{s}_N) = S$$

where S is a binary version of A indicating nonzero-ness and, in particular $\forall \ i, ||\vec{s}_i||_0 = ||\vec{a}_i||_0$ ($= 3$ in this example).

Given S, that is, given the support of A in the representation $Y = BA$, finding an appropriate A might not be so difficult. In this example, this amounts to knowing which shape constitutes each data point. There may be a unique choice of $\vec{a}_i$ given $\vec{s}_i$, which is the case in this running examplel. Section 1.1.2 explains how despite noise, there may be an obvious representation (choice of coefficients given support), which can even reduce the effect of the noise. The support of a sparse representation can carry much of the information that classifies data, often referred to as *semantic meaning*.

The most difficult problem then, is recovering S, given Y and A. Although this is accomplished by solving $Y \approx AX$, the support S of A is the difficult component. There are numerous sparse recovery algorithms, the most common of which are Orthogonal Matching Pursuit, Matching Pursuit, and Basis Pursuit.

2 Sparse Models

2.1 An Idea About Models

To extrapolate, compress, or classify (i.e. process) data, that data is often assumed to adhere to some model, in terms of which the data may then be expressed for some advantage. There are two important connections in this process:

1. The data conforming to a model.
2. Utilizing the properties of that model to achieve a goal.

The first connection is usually in the form of an assumption, or an assertion about qualities of the data, and the second involves exploiting characteristics of those qualities. If a procedure makes no assumptions on its data, then its gains are limited: no simplifications can be made and exploited, making the data effectively untenable and unpredictable - it could be anything! Data that doesn't conform to an assumed model is likely to 'misbehave' and either will not be represented efficiently or will be represented incorrectly.

2.1.1 An Example

If one-dimensional data points represent the lengths of the fingernails of a random person in a room, the points might come from a uniform distribution, meaning any length in some known or unknown range is equally likely (including the most and least extreme lengths in that range).

If the model is *known to be correct*, where the range $[a, b] \subset \mathbb{R}$ is known, the lengths can be written using less information: if the precision of the measurements is 0.1 in, then there are only $\frac{b-a}{0.1}$ possible values, which can each be uniquely represented by exactly $\log_2(\frac{b-a}{0.1})$ binary bits, exploiting the model.

If points are represented by $\frac{b-a}{0.1}$ bits but the model that data points lie in $[a, b]$ is *incorrect*, then a point outside that range might arise and be represented incorrectly or not at all.

Not using a model amounts to representing the values as arbitrary real numbers, for which a computer might use any number of unnecessary bits by default.

Alternatively, all numbers involved with any computer must be small enough to be represented at all: a number that cannot be stored by all of a computer's transistors (i.e. a number greater than $2^{\text{number of transistors}}$, though in practice much smaller) is completely unusable. In this way, a computer exploits the basic model that its input is within the usable range.

Of course the model could be correct with a and/or b unknown, in which case the weaker model can *still* be exploited using statistics, but always with some uncertainty.

2.2 Ways to Model Sparsity

Recent advances in signal processing have exploited the *sparsity* model. In one approach, data points in $\mathbb{R}^D$ are assumed to lie close to the union of K linear subspaces $\pi_1, ..., \pi_K \subset \mathbb{R}^D$ of dimensions $d_1, ..., d_K$, with $d_i \ll D$ [5]. This means that, represented as a linear combination of some bases that span these subspaces, the number of nonzero coefficients representing any given data point $x_i \in \pi_i$ is no more than $d_i \ll D$, so the non zeros are *sparse*.

More classically, all data points have been assumed to lie around a *single* low-dimensional subspace of dimension d. In this case, the best such subspace is often the span of the first d principal components of the data set, which minimizes the mean-squared-error of the representation. That is, the orthogonal directions of maximum variance span the subspace around which data lies most closely. Finding these mutually orthogonal directions of maximum variance is called principal component analysis (PCA)

In a generalization of the previous paragraph, data assumed to lie on the union of low-dimensional subspaces can be addressed in two ways. First, a *dictionary* - a set of bases whose subsets span the subspaces - can be assumed, and *sparse coding* algorithms can be applied to minimize error while constraining each data point to one such low-dimensional subspace. Secondly, a suitable dictionary can be inferred from the data and then sparse coding can be applied in the same way. This problem is referred to as *dictionary learning*.

In a further generalization, data points can be clustered by proximity and locally assigned to low-dimensional affine subspaces. Nearby points by definition lie throughout the full dimension of their local affine subspaces, and that is how those subspaces and their dimensions are determined. This amounts to imposing a union of piecewise-linear manifolds over the data, and is described in [5] as Multiscale SVD. This technique yields the location and dimensions of the manifolds on which data lie, which is extremely specific and thereby informative. However, the use of topological classifications is nascent, and the terminology surrounding current applications of this technique may be misleading: in practice, local topological representations are the same as those of dictionary learning, which captures affine translations as dictionary atoms.

The most general possible assumption is that data lies on an arbitrary manifold or union of manifolds in $\mathbb{R}^D$. Multiscale SVD (and the previous paragraph's framework) determine the local dimensions of manifolds on which data lie. This might help piece together complete spaces and infer topology, but [1] does that directly by determining the basis-free topology of a data set, and thereby the support of its distribution. The breakthrough in [1] is to determine that the topology of a set of 3×3 high-contrast natural image patches in $\mathbb{R}^9$ is that of the Klein bottle, which is a 2-manifold, and hopes to compress images using a Klein bottle dictionary. The Klein bottle cannot be embedded without self-intersection in any dimension lower than 4, which illustrates how the embedding of some topologies might render their low intrinsic dimension inaccessible to the previously mentioned algorithms. Of course an arbitrary manifold could be constructed to encompass any sampled data. Thus, simplicity must be imposed or complexity penalized in order to usefully determine topology of data. This approach may yield important insights into data sets some day as the usefulness of linear clustering in machine learning is exhausted and large, densely-sampled data sets become available and computationally viable.

2.2.1 Dictionary Learning, Compressive Sensing

Many models implicitly exploit sparsity, though not using the general term. Instead, they focus on the advantages of modeling data in various *specific*, scientifically-derived ways that usually require fewer terms. Periodicity of signals is exploited by representations in the Fourier basis, in terms of

which they are sparse; locally periodic or transient signals are exploited using wavelets. These are desirable because data which is 'spread out' in one domain, like sound or images sampled over time or space, maybe accumulate in another, like the frequency- or wavelet-domain.

Assumptions about sparsity may, however, have less to do with their relationship to a given, scientifically selected basis than their inherent sparsity with respect to an unknown basis. The most striking examples of this are compressive sensing and dictionary learning.

In compressive sensing, a signal *is* assumed to like on a union of low-dimensional subspaces. However, recovery of the bases of those subspaces is not attempted. A best-possible sparse representation with respect to some random or general overcomplete basis is then determined. Sparsity will with high probability be achievable assuming the bases aren't conspiring against the process. That is, an arbitrary overcomplete basis encompassing sufficiently many directions will with high probability contain bases that span arbitrary low-dimensional planes. Then, assuming data lies on *some* low-dimensional planes, the closest surrogates spanned by small subsets of the overcomplete basis are used. The basis employed is fixed, such as the discrete cosine transform (DCT) basis, or any random overcomplete, and (therefore) non-orthogonal basis [7, 10].

Dictionary learning, in contrast to compressive sensing, concentrates on finding an optimal overcomplete basis, subsets of which span low-dimensional planes on which the data primarily lies [2, 4–6, 13, 14]. The data may also be assumed to lie on some non-linear manifold, which entails learning several local dictionaries (amounting to a piecewise-linear dictionary) [4, 5].

3 Sparse Coding of Noisy Data

Typically, assumptions of sparsity assert that data $y \in \mathbb{R}^D$ can be written

$$y = Ax + e, \tag{1}$$

where A is an $D \times K$ matrix called a 'dictionary' whose K columns are 'dictionary atoms', x is a sparse representation, and e represents noise.

As discussed in section 1, the above formulation usually stands in for a generative model, with algorithmic rather than probabilistic parameters. Specifically, sparse coding algorithms like matching pursuit (MP) and orthogonal matching pursuit (OMP) find the closest possible approximation to data y given a basis A and a sparsity constraint T, solving either of the equivalent problems:

$$\min_x ||x||_0 \qquad \text{s.t. } ||y - Ax||_2^2 \leq \epsilon \tag{2}$$

$$\min_x ||y - Ax||_2^2 \qquad \text{s.t. } ||x||_0 \leq T \tag{3}$$

$$\min_x ||y - Ax||_2^2 + \lambda ||x||_0 \tag{4}$$

These three problems are *equivalent* in that for any $y \in \mathbb{R}^D$, and any $T \in \mathbb{N}$, $\epsilon > 0$, or $\lambda \in \mathbb{R}$, there exists an appropriate $\epsilon > 0$, $\lambda \in \mathbb{R}$ or $T \in \mathbb{N}$ such that the same solution x minimizes all three objective functions [20]. Therefore, the three problems can be solved by the same algorithm, although different formulations may be desirable for different applications.

These problems constitute *models* in the sense that they define desirable ways to present data. The first formulation assumes that data clusters around low dimensional subspaces up to (specifically) Gaussian noise with variance ϵ, but does not specify exactly the dimensions of those subspaces. The second formulation assumes that data clusters around subspaces of a specific low dimension, but does not specify the variance of the noise.

If data points are indeed generated as sparse linear combinations of columns of A plus noise, then estimating that linear combination despite the noise may be meaningful for classification or anomaly-detection. At an appropriate scale, columns of A may represent *semantically meaningful* 'structural primitives' [16], meaning pieces of images that each correspond to identifiable features of those images. Whether or not dictionary atoms do noticeably correspond to primitive units, any algorithmically successful solution indicates that the data comprises some such primitive units; that is, these objective functions demand some meaning from dictionary atoms because so few are used to represent each data point. Any solution achieved with a sufficiently low value of the objective function indicates that the dictionary atoms are in fact meaningful.

Structural primitives of small natural image patches, for example, may be corners and edges. Images can be divided into patches - of e.g. 3 by 3 pixels - whose pixels' values are ordered into vectors - e.g. in $\mathbb{R}^9$. A small natural image patch is likely to contain relatively constant values or have just a few features - corners and edges. A collection of such features constitutes a dictionary and a sparse coding accurately represents patches as linear combinations of these features. Constant values, corners, and edges, then, are structural primitives for small natural image patches. Larger natural image patches may have differently scaled structural primitives, which may be more semantically meaningful. Of course, there are fewer patches of larger size in any image data set, and more complicated features will require *more* data to stick out in data. Thus, the larger the image patches in partitioned data, the more data is necessary to assign semantic meaning to sparse representations.

Non-probabilistic, algorithmic approaches impose structure onto data, but it do not specify the distribution of data around that structure. This is most clear in the case of the third formulation of the sparse coding problem (4), which is known as the LASSO problem, and which does not impose a specific dimension *or* a specific level of noise.

4 The Boltzmann Machine

One probabilistic model that can describe sparse data is the Boltzmann machine (BM). The Boltzmann machine does not specify the basis within which data can be represented sparsely, nor does it specify the values of sparse nonzero coefficients, but rather describes sparse yes-no pattern of the support. Finding such a basis amounts to dictionary learning or just science. Dictionary learning often iterates between finding sparse representations and updating the basis - or 'dictionary' - until some level of convergence, and therefor the BM model can take part in dictionary learning well. The discussion here will only address the BM's relevance to representations with respect to a pre- or un-specified basis.

The work in [10] explains almost every consequence of the Boltzmann Machine as a prior for modeling sparse data. This distribution constitutes a *prior* distribution for the support of a representation with respect to the *posterior* distribution of those representations given data. The BM prior asserts that some sparse representations of data may be less likely than others due to patterns in their sparsity. Specifically, the parameter b renders non-sparse representations improbable, with different penalties for allowing the different dictionary atoms into the support of a signal's representation. The interactions matrix W renders representations improbable if their interactions are not of an assumed form.

A main goal of this paper is to use that probabilistic framework to explain and justify intuitive data classification techniques.

4.1 General Boltzmann Machine

The BM describes a network of nodes $\{s_1, \ldots, s_n\}$, each of which can be 'on' or 'off' - that is, $s_i \in \{-1, 1\} \forall i$. Each node s_i is assigned a bias b_i and each pair of nodes s_i, s_j is assigned an connection strength w_{ij}. The system can be assigned a global 'energy' determined by the biases and interactions of the nodes:

$$E(\{s_1, \ldots, s_n\}) = -(\sum_{i<j} w_{ij} s_i s_j + \sum_i b_i s_i). \tag{5}$$

Energy is high when nodes with low bias are 'turned on', or have value 1, and nodes with low or negative interactions are turned on concurrently. Let

- the $n \times 1$ state vector $S \in \{-1, 1\}^n$ represent a possible state of the system
- the symmetric $n \times n$ interactions matrix $W = (w_{i,j})_{i,j=1,\ldots,n}$ contain the interaction coefficients
- the $1 \times n$ bias vector $b = (b_i)_{i=1,\ldots,n}$ contain the biases.

Then (5) becomes

$$E(S) = -(S^T W S + bS)$$

As in the Boltzmann distribution from which the Boltzmann Machine gets its name, the probability of a state is determined by its energy:

$$Pr(S) \propto exp(\frac{-E(S)}{kT}) = e^{(\frac{S^T W S + bS)}{kT})} \tag{6}$$

where k is a fixed constant and T represents an artificial 'temperature', and the two are often varied together as one constant[1].

A high temperature means that a large change in energy has a lesser effect on the probability of the state. Thus a low temperature indicates a more sensitive system, with high-energy states being much less likely than low-energy states. At each temperature, a constant normalizes the distribution.

4.1.1 Simulated Annealing: a Typical Application of the Boltzmann Machine

The Boltzmann machine might be used on its own to find a highly likely configuration of 'yes' and 'no' states throughout a system. Given the parameters W and b, representing interactions between nodes and biases, respectively, 'simulated annealing' is a common practice to find such a state.

Simulated annealing (SA) is a metaheuristic (type of strategy) that amounts to iterative stochastic gradient ascent of the BM probability density function constrained to 'adjacent' states of lower energy, while the 'temperature' parameter is constantly lowered. The stochasticity of the gradient ascent entails a nonzero probability that a successive state's energy will have increased, decreasing its likelihood.

As the temperature lowers, high-energy states become increasingly improbable, and the process is less likely to enter a state of higher energy, even if that is necessary to find the global optimum. In practice, this amounts to turning 'on' and 'off' nodes of the Boltzmann machine that will lower the energy function, with some iteratively decreasing likelihood of selecting nodes that increases the energy, in order to both avoid local minima and eventually converge to a local maximum. With

[1]The Boltzmann Distribution is used in thermodynamics to model the energy strata occupied by particles at a given temperature, or, alternatively, to describe temperature by the occupied energy states.

parameters W and b that induce sparsity, this amounts to finding a small subset of nodes that yield an optimal global state by being turned on concurrently.

The BM will be used to model sparse representations, and an algorithm similar to simulated annealing will be used to find likely sparse representations given parameters W, b. The BM model constrains the *structure* of a sparsity pattern, rather than just favoring representations with small support. The BM distribution is used to construct a useful objective function yielding desirable sparse representations.

4.2 Sparse Signals and the Boltzmann Machine Prior

The BM abstractly corresponds to patterns of yes and no, which can naturally describe the support of sparse signals. A generative model is described, and then its estimators are outlined. Next, these estimators' geometric significance is associated with heuristics for data classification and anomaly detection.

The BM is considered as the distribution of the *support* of a representation X (such that $Y \approx AX$). Then, the prior probability of a representation given its support is defined, as is the probability of data given its representation to then quantify the the quality (likelihood) of a representation:

$$Pr(A|Y) \propto Pr(S) \times P(X|S) \times Pr(Y|X)$$

4.2.1 The BM Generative Model

The following constitutes a complete model specifying the distribution from which signals are drawn, and how the BM prior induces sparsity. A pseudolikelihood (PL) stands in for a proper probability distribution function. That is, to apply the BM model to a data set with more than one point, the probability of simultaneous behaviors of data points is approximated by the products of their individual probabilities. Crucially, [10] proves the consistency of many PL estimators, meaning that estimates based on the PL in many cases hold for the actual implicit distribution of data following this model.

The Boltzmann machine is used in this paper as a prior for sparse representations of data, and posterior likelihoods of representations are determined by the data. More explanation of parameters is given in 4.1.

Data points in the $D \times N$ matrix $Y = (y_1, \ldots, y_N)$ are represented as $Y \approx AX$, where A is a $D \times K$ matrix whose columns, called 'dictionary atoms' are bases for the planes $\pi_1, \ldots, \pi_{K'}$, and $X = (x_1, \ldots, x_N)$ represents Y with respect to the basis D. The planes $\pi_1, \ldots, \pi_{K'}$ may be of dimension $\neq 1$, so $K' \neq K$ in general.

Let s_i be the *support set* of x_i, meaning $s_i = \{i | x_i \neq 0\}$. Then let the *support vector* S be the $K \times N$ matrix whose entries correspond to the entries of X in the following way:

$$\begin{aligned} S_{i,j} = &\begin{cases} 1 & : x_{i,j} \neq 0 \\ -1 & : \text{otherwise.} \end{cases} \\ =&1 - 2 \cdot \mathbb{1}(x_{i,j} = 0) \end{aligned}$$

Alternatively, this could read:

$$S_{i,j} = \begin{cases} 1 & : \text{the } i\text{'th atom is used to represent the } j\text{'th data point (or } i \in s_j) \\ -1 & : \text{otherwise (or } i \notin s_j) \end{cases}$$

which helps conceptualize the entries of S as corresponding to Bernoulli random variables indicating whether or not dictionary atoms are in the support of given data points.

For $i = 1, \ldots, N$, the support vector S_i of the representation $y_i \approx Ax_i$ has the Boltzmann distribution:

$$Pr(S_i) := \frac{1}{Z} \exp(\vec{b}^T S_i + \frac{1}{2} S_i^T W S_i). \tag{7}$$

The vector $\vec{b}$ corresponds to the probabilistic biases of each of the atoms (columns of A) being included in the support of x_i, and the matrix W corresponds to the dependencies of between pairs of atoms regarding mutual inclusion or exclusion from the support. This relatively simple model is actually very flexible and the two parameters b and W can induce radically different sparse structures.

The full support of a representation $Y \approx AX$ can be defined by considering the support x_i of each data point y_i as an i.i.d. sample from the BM distribution, and so

$$Pr(S_1, \ldots, S_N) = Pr(S_1) \ldots Pr(S_N).$$

4.2.2 The Normal Distribution of Coefficients Given Support $(X|S)$

Concentrating on a single signal $y = Ax + e$, which is the case $N = 1$ of the above model, denote the nonzero coefficients of x by x_s, where $s = \{s_1, \ldots, s_{|s|}\} = \{i | x_i \neq 0\}$ is the support set of x. Let $S \in \{-1, 1\}^K$ be the support vector of x as described above ($S_i = 1 - 2 \cdot \mathbb{1}(x_{i,j} = 0)$).

As in [10], assume the entries of x_s have independent Gaussian distribution with zero mean and variance σ^2_{x,s_i}. Note that the s_i subscript indicates that the components of x_s - each of which corresponds to a different dictionary atom - may have different variances.

As with any independent, normally distributed random variables,

$$P(x_s|s) \propto \exp(-\frac{1}{2} x_s^T \Sigma_s^{-1} x_s) \tag{8}$$

where Σ_s is an $|s| \times |s|$ diagonal matrix whose diagonal elements are $(\Sigma_s)_{i,i} = \sigma_{x,s_i}$. The normalizing constant that gives equality is $\frac{1}{\det(2\pi\Sigma_s)^{1/2}}$.

4.2.3 Distribution of y around Ax

The noise $e = y - Ax$ is assumed to be Gaussian, and so the distribution of y about its sparse representation Ax is normal, meaning

$$P(y|x_s, s) \propto \exp(-\frac{1}{2\sigma_e^2} ||y - A_s x_s||_2^2) \tag{9}$$

where $A_s x_s = Ax$ is written to remind that, given s, only some columns of A are used in the representation Ax, which columns can be called A_s. The normalizing constant that gives equality is $\frac{1}{(2\pi\sigma_e^2)^{n/2}}$.

An expression for $P(y|s)$ is easily derived, and comprises an integral over a computationally feasible dimension, thanks to the sparsity constraint. That is,

$$Pr(y|s) = \int_{x_s \in \mathbb{R}^{|s|}} P(y|x_s, s) P(x_s|s) dx_s \tag{10}$$

and [10] algebraically solves for this expression in closed form ON PAGE 6, EQUATION (9). This expression can be used to derive a *maximum a posteriori* estimator for the support s. That is, given an expression for $P(y|s)$, [10] derives a closed form expression for

$$\hat{s}_{MAP} = \text{argmax}_s \, P(s|y) = \text{argmax}_s \, P(y|s) P(s) \tag{11}$$

utilized in section 4.4.1.

4.3 Sparse Recoveries Given Parameters W, b

The goal is given data in the columns of matrix Y, a dictionary A, and the parameters of the BM W and b, to obtain the representation X such that $Y \approx AX$ and $P(X|Y)$ is high. In terms of the pieces above,

$$P(X|Y) \propto \underbrace{P(S)}_{\substack{\text{BM dist'n} \\ (b,W)}} \times \underbrace{P(X|S)}_{(x_{ij}|s_{ij}=1) \sim N(0,\sigma)} \times \underbrace{P(Y|X)}_{\sim N(X,\sigma_\epsilon)} \tag{12}$$

Orthogonal matching pursuit (OMP) iteratively introduces columns of A into the support of the representation $y \approx Ax$ to optimize the LASSO function 4. To find a maximum likelihood estimator for 12, a similar approach is used, outlined in [10], along with several algorithmic variants for different purposes.

4.4 Estimating BM Model Parameters W, b

Given a data matrix $Y = (y_1|\dots|y_N)$, a sparse representation $AX = A(x_1|\dots|x_N)$, whose support is $S = (s_1|\dots|s_N)$, where

$$s_{ij} = \begin{cases} -1 & : \quad x_{ij} = 0 \\ 1 & : \quad \text{otherwise} \end{cases},$$

the goal is to recover the BM parameters b and W, the bias vector and interactions matrix, respectively.

In [10], two algorithms are introduced to estimate these parameters, and a survey of other techniques is given. This is a difficult problem, and is actually not possible unless significant restrictions are placed upon the interactions matrix W.

The (i, j)'th entry of the interactions matrix parameter W can be understood very intuitively to relate to the likelihood of the i'th and j'th atoms simultaneously representing the same data point. However, the entries defined by the BM distribution do not exactly correspond to this likelihood, but rather are indicators of it.

To help understand what the entries in fact represent, estimates of W from a sparse representation of data can be compared to a covariance matrix of the atoms. That is, let C be a symmetric matrix corresponding to the sparse representation $Y \approx AX$ such that, for some data point y whose support vector is S,

$$\begin{aligned} c_{i,j} &= P(\text{atom i is used to represent data point y}|\text{atom j is used to represent data point y}) \\ &= P(S_i = 1|S_j = 1) \end{aligned} \tag{13}$$

Given the matrix $S = (S_1|\dots|S_N)$ whose columns are the support vectors of the sparse representation $Y \approx A(x_1|\dots|x_N)$, C can be expressed as a function of S. Recall that $S_{i,j} \in \{-1, 1\}$; let S' be the related matrix such that $S'_{i,j} = 2(S_{i,j} - 1) \in \{0, 1\}$. Then

$$C = \frac{1}{N} S' S'^T. \tag{14}$$

Calculating C given X is then trivial: find S' (entries are indicators of nonzero entries) and use 14. How is C related to W? Beyond their both representing similar quantities, their relationship is most easily seen by plotting corresponding entries, as in Figure 1. The entries of C are actually all multiples of their corresponding entries of W plus one of just a few constants, despite W being recovered in a completely different manner described in [10].

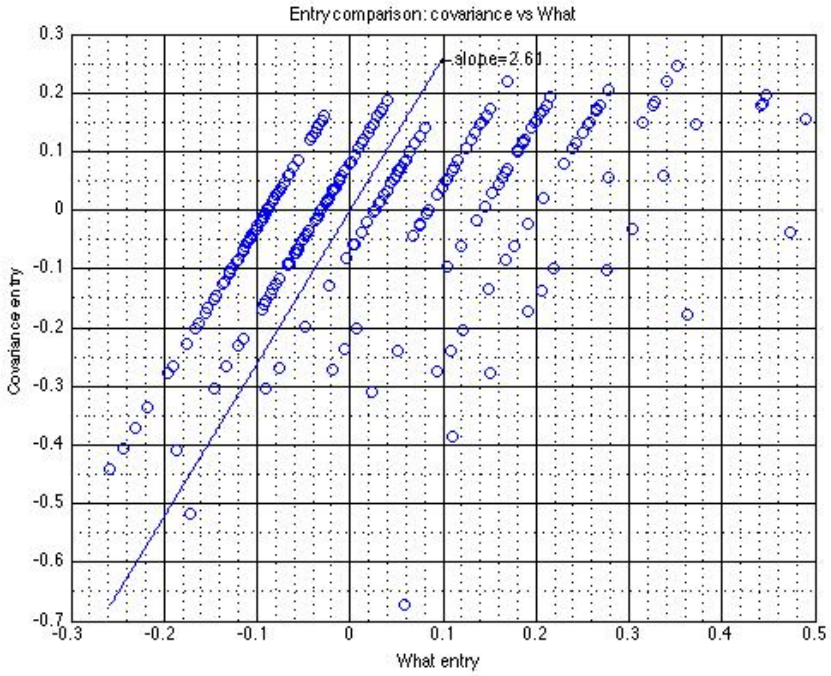

Figure 1: Entries of atom-usage covariance matrix C vs corresponding entries of BM interaction matrix W for a sparse coding

4.4.1 Joint Estimation of Parameters and Representation

Given data points as the columns of matrix $Y = (y_1|\dots|y_N)$ and a dictionary A, both a sparse representation $Y \approx AX$ and model parameters b and W may be desirable for data classification. To accomplish this task, [10] explains how an iterative process of generating sparse representations and from those estimating the model parameters b and W.

4.5 Structured Sparsity

Sparse coding models that posit dependancies between atoms' use usually employ one of two models, block sparsity and hidden Markov trees, the latter also known as hierarchical sparsity. Both these types of interactions between atoms are representable through the BM model's interactions parameter W. That is, data generated from the BM model can be induced to obey assumptions of either block sparsity or tree-dependencies through the interactions matrix W, as outlined below. In this way, the BM prior is quite general, making estimation procedures within this framework very robust.

4.5.1 Block Sparsity

Block sparsity refers to the natural assumption that atoms' use will strongly cluster, and that many subsets of atoms will usually either be used concurrently or not at all. For example, any data classification problem lends itself naturally to a block sparsity assumption. If data points belonging to different categories exhibit geometric differences reflected in sparsity patterns, those differing patterns can be assigned to blocks of atoms that interact in a given way.

For example, image segmentation procedures may, through supervision or not, only allow certain image features to be represented by certain dictionary atoms. Thus, an image of a nose is not allowed to be represented as a linear combination of any atoms other than those corresponding to noses, unless the nose is incorrectly assigned to a different category; intuitively, though, preventing that problem is easier to avoid when blocks are highly distinctive, as may result from robust learning on large data sets.

Geometrically, block sparsity amounts to clustering data onto different subspaces of various dimensions. This is in fact the general assumption outlined in Section 1, where data clusters around the K' planes $\pi_1, \ldots, \pi_{K'}$ of dimensions $d_1, \ldots, d_{K'}$, as described in [5].

Throughout the rest of this paper, data is assumed to lie on the union of the low-dimensional subspaces spanned by the K columns of a $D \times K$ dictionary A. Low-dimensional is referred to as $T \ll D, \ll K$, meaning that data clusters around the $\binom{K}{T}$ subspaces of dimension T spanned by subsets of size T of the columns of A. This binomial coefficient highlight a strange implicit assumption made by *not* specifying a block-sparse model.

In this way, the block-sparsity model is actually more general than its omission, as the bases for different blocks may have some similar or identical elements, and thus subsumes the general model above by specifying the above $\binom{K}{T}$ subspaces as blocks. In a general block sparsity framework, however, blocks' associated subspaces may intersect only in $\{0\}$, and block-sparsity is not subsumed by the general model, which always allows possible subspaces to intersect nontrivially by sharing support.

Despite the intuition underlying block sparsity, [10] determined that it is not a particularly accurate model for small natural image patches with a discrete cosine transform (DCT) dictionary. This is a patch size-, dictionary-, and data set-specific observation, however.

To generate the support for block-sparse data using the BM model, the interactions matrix W can encode all the interactions between atoms. Note that the support matrix S has entries in ± 1, and so the probability of a single data point v_i's support, which is

$$P(S_i) \propto e^{-(bS_i + S_i^T W S_i)},$$

experiences both inhibitory and positive interactions between atoms. To encode block sparsity, atoms in different blocks must have inhibitory interactions, while those in the same block must have positive interaction. That is, partition the K columns of A into K' blocks, with $n_1, \ldots, n'_K$ entries, respectively.

$$A = (a_1 | \ldots | a_K) = (a_{1,1} | a_{1,2} | \ldots | a_{1,n_1} | a_{2,1} | \ldots | a_{2,n_2} | \ldots | a_{K',1} | \ldots | a_{K',n'_K})$$

For notation, suppose $a_j \in \{a_{i,1}, \ldots, a_{i,n_i}\}$ - then denote that subset of the columns of A by A_j. Then, to encourage block sparsity, $P(S_i)$ must be high when for all i, j, k, ($S_{i,j} \neq 0$ and $S_{i,k} \neq 0$) $\Leftrightarrow A_j = A_k$. To encourage the generation of such support, the entries of W can be tailored such that $A_j = A_k \Rightarrow W_{j,k} = -1$ and $W_{j,k} = 1$ otherwise. Then a BM interactions matrix parameter W might have block-diagonal structure in the algebraic sense of the word, where off-block-diagonal entries are 1 (inhibitory interaction) and diagonal block entries are all -1 (positive interaction), as illustrated in figure 2.

The entries of W are not constrained in general, and can contain entries other that ± 1. The bias parameter b adjusts the likelihood of an atom's use in general, and so scaling W by a factor is equivalent to adjusting b and scaling the constant factor that normalizes the distribution as a whole. Thus, without loss of generality interactions matrices W with small entries can be considered.

$$W = \begin{pmatrix} \begin{matrix} -1 & \dots & -1 & 1 & \dots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & \dots & -1 & 1 & \dots & 1 \\ 1 & \dots & 1 & -1 & \dots & -1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \dots & 1 & -1 & \dots & -1 \end{matrix} & & \\ & \begin{matrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{matrix} & \\ & & \ddots \\ & & \begin{matrix} -1 & \dots & -1 & 1 & \dots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & \dots & -1 & 1 & \dots & 1 \\ 1 & \dots & 1 & -1 & \dots & -1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \dots & 1 & -1 & \dots & -1 \end{matrix} \end{pmatrix}$$

Figure 2: A BM interactions matrix encouraging block-sparsity.

4.5.2 Hierarchical Sparsity

Tree-structures are similar to block structures conceptually, though different atoms' concurrent use is dictated by a hierarchical model. This is most often applied to wavelet dictionaries with natural image patches, where wavelets' multiple scales may be reliant on each other in a hierarchical structure.

Despite the intuition underlying hierarchical sparsity, [10] determined that it is not a particularly accurate model for small natural image patches with a discrete cosine transform (DCT) dictionary. This is a patch size-, dictionary-, and data set-specific observation, however.

5 Data Classification

Given a set of data, a common goal is to classify them into categories. This amounts to partitioning the set by either clustering data points to each other or comparing them to criteria of fixed categories. Many data processing tasks fall under the category of data classification, including anomaly/saliency detection and feature detection. This paper will focus primarily on data-driven anomaly detection.

5.1 Anomaly/Saliency Detection Using the Boltzmann Machine

Given a data set $\{y_1, \dots, y_N\} \subset \mathbb{R}^D$ drawn from some distribution, the questions driving anomaly/saliency detection are:

- Which, if any, and how many data points deviate from a specified behavior?
- Which, if any, and how many data points deviate from the behavior of the other data points?

Answering the first question amounts to *supervised learning*, while the second involves *unsupervised learning*, but these approaches are related. Under the unsupervised approach, a model is assumed and parameters are inferred from a single data set. To classify data using supervised

learning, pre-classified or 'labeled' data is used to determine parameters that are used to classify unlabeled data.

In either case, the BM-based likelihood function can quantify which sparse codings are unusual, and thus anomaly detection amounts to finding sparsely coded data sets whose atom usage does not conform to the preferences expressed by b and W. This contrasts to the more traditional approach of finding data points that lie far away from the others.

Section 4.4 discusses the relationship between the BM parameter W and the covariance matrix C of atoms' use in a sparse coding. C is easy to calculate given a sparse coding, and is easy to interpret as well. Then, given a sparse coding, calculating C and comparing it to the matrix W amounts to quantifying how closely data is adhering to the BM generative model. Data that does not conform may be anomalous.

5.2 Sensitivity and Specificity: A Rarely Used Atom, the BM Parameter b

In [10], "an atom is labeled as "rarely used" if it is active in less than 0.3% of the data samples. This is an arbitrary definition, but it helps in showing that the estimated parameters tend to be close to correct."

The need to define rare usage arose in [10] to explain the scope of their estimation technique. Specifically, to estimate the interactions between atoms in the matrix W in their generative model, an atom must exhibit sufficiently many interactions in general. This amounts to an atom representing sufficiently many data points. The authors are indicating that their interactions model will not be influenced by atoms used less than 0.3% of the time.

In [8], anomalous data points are detected using dictionary learning by observing data points that are represented by "rarely used" dictionary atoms under a slightly different definition. However, the relevance of rarity translates directly: in [10], rarity means that the interactions concerning an atom cannot be effectively inferred; in [8], rarity suggests that those interactions can be assumed abnormal. In [8], the matrix C and rarity, a similar statistic to the estimation of b in [10], are used to detect anomalous data points.

References

[1]

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: Design of dictionaries for sparse representation. *Proceedings of SPARS*, 5:9–12, 2005.

[3] T. Broderick, M. I. Jordan, and J. Pitman. Beta processes, stick-breaking, and power laws. *ArXiv e-prints*, June 2011.

[4] Guangliang Chen and M. Maggioni. Multiscale geometric wavelets for the analysis of point clouds. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6, 2010.

[5] Guangliang Chen and M. Maggioni. Multiscale geometric and spectral analysis of plane arrangements. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2825–2832, 2011.

[6] Namgook Cho and C.-C.J. Kuo. Sparse music representation with source-specific dictionaries and its application to signal separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(2):326–337, 2011.

[7] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ?1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[8] Mark Eisen, Mengjie Pan, Zachary Siegel, and Sara Staszak.

[9] M. Elad. Sparse and redundant representation modeling - what next? *Signal Processing Letters, IEEE*, 19(12):922–928, 2012.

[10] Tomer Faktor, Yonina C. Eldar, and Michael Elad. Exploiting statistical dependencies in sparse representations for signal recovery. *CoRR*, abs/1010.5734, 2010.

[11] Stefano Gonella and Jarvis Haupt. Automated defect localization via low rank plus outlier modeling of propagating wavefield data. 2012. Revised draft for Transactions on Ultrasonics, ferroelectrics and frequency control, IEEE.

[12] J.M. Hughes, D.N. Rockmore, and Yang Wang. Bayesian learning of sparse multiscale image representations. *Image Processing, IEEE Transactions on*, 22(12):4972–4983, 2013.

[13] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[14] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research (JMLR)*, 11:19–60, 2010.

[15] S. Mukhopadhyay and F. Liang. Bayesian Analysis of High Dimensional Classification. In P. M. Goggans and C.-Y. Chan, editors, *American Institute of Physics Conference Series*, volume 1193 of *American Institute of Physics Conference Series*, pages 243–250, December 2009.

[16] Bruno A. Olshausen and David J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997.

[17] John Paisley and Lawrence Carin. Nonparametric factor analysis with beta process priors.

[18] M.R. Schroeder. Period histogram and product spectrum: New methods for fundamental frequency measurement. *Journal of the Acoustic Society of America*, 1968.

[19] Steven K. Tjoa, Matthew C. Stamm, W. Sabrina Lin, and K. J. Ray Liu. Harmonic variable-size dictionary learning for music source separation. In *ICASSP*, pages 413–416. IEEE, 2010.

[20] Ivana Tosic and Pascal Frossard. Dictionary learning. *Signal Processing Magazine, IEEE*, 28(2):27–38, 2011.

[21] Mingyuan Zhou, Haojun Chen, J. Paisley, Lu Ren, Lingbo Li, Zhengming Xing, D. Dunson, G. Sapiro, and L. Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *Image Processing, IEEE Transactions on*, 21(1):130–144, 2012.

[22] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Guillermo Sapiro, and Lawrence Carin. Non-parametric bayesian dictionary learning for sparse image representations. *Neural Information Processing Systems (NIPS)*, 2009.