JMLR: Workshop and Conference Proceedings 44 (2015) 1-18 NIPS 2015

The 1st International Workshop “Feature Extraction: Modern Questions and Challenges”

A Survey of Modern Questions and Challenges in Feature Extraction

Dmitry Storcheus DSTORCHEUS@GOOGLE.COM
Google Research

Afshin Rostamizadeh ROSTAMI@GOOGLE.COM
Google Research

Sanjiv Kumar SANJIVK@GOOGLE.COM
Google Research

Editor: Neil D. Lawrence

Abstract

The problem of extracting features from given input data is of critical importance for the successful application of machine learning. Feature extraction, as usually understood, seeks an optimal transformation from input data into a (typically real-valued) feature vector that can be used as an input for a learning algorithm. Over time, this problem has been attacked using a growing number of diverse techniques that originated in separate research communities, including feature selection, dimensionality reduction, manifold learning, distance metric learning and representation learning. The goal of this paper is to contrast and compare feature extraction techniques coming from different machine learning areas, discuss the modern challenges and open problems in feature extraction and suggest novel solutions to some of them.

Keywords: feature selection, feature weighting, feature normalization, column subset selection, random projection, clustering, nearest neighbor, dimensionality reduction, manifold learning, distance metric learning, multiple kernel learning, representation learning, embedding, PCA, LDA, RCA, kernel PCA, Isomap, LLE, Laplacian eigenmap, neural networks, autoencoders, restricted Boltzmann machines

1. Introduction

Over the recent years, the volume of data available to machine learning algorithms has grown tremendously, not only in the number of instances, but also in the amount of raw information that each instance contains. Millions or, in some cases, even billions of input features are available in applications such as natural language processing, image recognition, text categorization, audio analysis, bioinformatics and physics. Given such excessive amounts of raw information, the task of *feature extraction*, that is transforming input data into features that can be useful for a learning algorithm, is as critical as ever for the successful application of machine learning.

One major goal of feature extraction is to increase the accuracy of learned models by compactly extracting salient features (understandable to the learning algorithm) from the input data, while also potentially removing noise and redundancy from the input (Guyon et al., 2008). Additional objectives include low-dimensional representations for data visualization and compression for the purpose of reducing data storage requirements as well as increasing training and inference speed (Guyon and Elisseeff, 2003). Note that in this work, depending on the context, we use the term *input data* (or equivalently *input features*) to indicate either potentially unstructured raw data (such as an image, text string, or audio signal) or a well defined set (say a set of real-valued variables), from we wish to extract more valuable features.

In general, without informative features, it is not possible to train a model with low generalization error, but if relevant features can be extracted, then even a simple method can show remarkable results (Yang and Pedersen, 1997; van der Maaten et al., 2009). Accordingly, much effort is spent during practical machine learning in building and maintaining complex feature extraction pipelines (Sculley et al. (2014), Sec. 4.2). This has driven feature extraction research in various industrial and academic fields, making the topic quite broad and diverse. In this survey, we try to give a structured outline on this diverse topic, compare and contrast key approaches, as well as describe how they attack specialized open problems.

The domain of feature extraction has accumulated a diverse set of terminology, and here we work towards clarifying the terms and explaining how they fit together. When referring to *feature extraction*, we mean the most general class of methods that deal with any transformation from *input data* to *features* for machine learning algorithms (Guyon et al., 2008). One particular feature extraction approach is *feature selection*, also known as *variable subset selection*, which is concerned with choosing the best subset from a larger input feature set (and does not synthesize new features) (Blum and Langley, 1997; Kohavi and John, 1997). Common methods for feature selection include variable ranking (Rakotomamonjy, 2003), feature subset selection (Narendra and Fukunaga, 1977) and penalized least squares (Fan and Li, 2001). These methods derive from classic statistics and heavily rely on hypothesis testing framework. A more general approach is *feature (re)weighting*, which aims at finding best weight for each feature (Wettschereck et al., 1997). Variable selection can be viewed as a special case of feature weighting (Wettschereck et al., 1997) when weights restricted to $\{0, 1\}$. An applied study of feature weighting methods is presented in Bai et al. (2010); Hussein et al. (2001).

Another approach that is closely related with feature weighting is *feature normalization*, which may involve feature centring, rescaling them to a target range, scaling to a unit balls, etc. (Aksoy and Haralick, 2001; Ekenel and Stiefelhagen, 2006; Barras and Gauvain, 2003). Furthermore, for the purpose of scaling features to a unit ball, a custom metric can be defined that reflects desirable data properties for a specific application (Stolcke et al., 2008).

The creation of new features from input data is usually referred to as *feature construction* (Liu and Motoda, 1998; Guyon et al., 2008). A standard example of feature construction is feature crossing of real-valued input data or constructing a bag-of-words or n-gram based vector from input text data. A more complex example of feature construction is clustering (Duda et al., 2001), which replaces a concentrated subsets of variables by their cluster center, in which case the constructed features are cluster centers instead of original variables.

When the goal of feature construction is to map the input features onto a lower dimensional space, it is referred to as *dimensionality reduction* (van der Maaten et al., 2009). Dimensionality reduction techniques extract features by projecting the input data into a lower dimensional subspace (van der Maaten et al., 2009; Cunningham and Ghahramani, 2014, 2015). Classic dimensionality reduction methods assume that this lower dimensional subspace of features is restricted to the span of input data points: Principal Component Analysis (Pearson, 1901), Random Projection (Hegde et al., 2008) and Linear Discriminant Analysis (Fisher, 1938). Selecting a smaller subset of available features can also be thought of as a very special case of dimensionality reduction.

Dimensionality reduction can be also treated from a probabilistic perspective, which is based on the maximum-likelihood estimation of a latent variable model (see Probabilistic PCA in Tipping and

Bishop (1999)). A classic non-linear extension of probabilistic methods, which allows to represent mode complex high-dimensional data, is the Gaussian Process Latent Variable Model (GP-LVM), developed in Lawrence (2004, 2007). A more general overview of probabilistic dimensionality reduction methods is given in Lawrence (2012).

Over the last decade, *nonlinear dimensionality reduction* methods have been extensively studied, which aim at learning a map from input data onto some nonlinear low-dimensional geometric structure (manifold) (Lee and Verleysen, 2007). This area is also referred to as *manifold learning* (Law and Jain, 2006; Roweis and Saul, 2000). The main goal of mapping input features onto a manifold is to capture non-linear relationships between the input features and reduce dimension by embedding into a low dimensional surface. Thus, these methods are also referred to as feature *embeddings* (Roweis and Saul, 2000). Usually a map from the input data to a manifold must preserve some statistical measure on the data (e.g. variance, reconstruction error). The preservation of different measures gives rise to different manifold learning algorithms. Classic examples are Isometric Feature Mapping (Isomap) (Tenenbaum et al., 2000), locally linear embedding (Roweis and Saul, 2000), and Laplacian eigenmaps (Belkin and Niyogi, 2003). An overview of these and other related methods is provided in (Saul et al., 2006; Venna et al., 2007). A remarkable result is that all of the manifold learning techniques already mentioned and many others are specific instances of the kernel PCA (KPCA) algorithm for different choices of the kernel function (Ham et al., 2004). Techniques that allow to make kernel methods scalable are *kernel approximations*; they involve either approximating the sample kernel matrix or the kernel function, for example with Nyström method (Drineas and Mahoney, 2005), or random Fourier features (Rahimi and Recht, 2007).

A vast majority of feature extraction methods, including those described above, rely on the notion of a *distance metric* defined on the space of input instances (Yang and Jin, 2006). Apart from using standard distance metrics (L_p, kernel functions), an appropriate metric can be learned from a parametrized class of metric functions (Weinberger et al., 2005; Xing et al., 2002). This brings us to an important area of feature extraction - *distance metric learning* (Yang and Jin, 2006). Common algorithms for metric learning include local LDA (Hastie and Tibshirani, 1996), relevance component analysis (Bar-Hillel et al., 2003), large margin nearest neighbor (Weinberger et al., 2005), and Bayesian active distance metric learning (Yang et al., 2012a). Since a distance metric can be derived from a kernel function (Davis et al., 2007), the methods found in *(multiple) kernel learning* highly overlap with the field of metric learning (Cortes et al., 2009, 2010a).

Finally, breakthroughs in deep networks (as well as growth in amounts of data and computational resources) have had a significant impact on feature extraction (Hinton and Salakhutdinov, 2006; Bengio et al., 2007; Poultney et al., 2006). Neural networks essentially learn a composition of multiple non-linear transformations (Bengio et al., 2013) of the input features, thus creating a new feature representation. Extracting features via layers of a neural network is commonly referred to as *representation learning*.

The paper is structured as follows: in Section 2 we formulate a general scenario for feature extraction and also outline common algorithms and explain how they fit into the general scenario defined. In Section 3 we discuss open challenges and questions in feature extraction and compare how the algorithms that we reference attempt solving those.

2. Overview and comparison of feature extraction methods

Here, we describe the learning scenario of feature extraction in a general framework and explain how existing feature extraction techniques from different fields can be represented using this framework. Let $\mathcal{X}$ denote the input feature space and $\mathcal{Y}$ the set of output labels (if we are dealing with a supervised task). We assume that the learner receives a sample of size m, $S = ((x_1, y_1), \ldots, (x_m, y_m))$, drawn i.i.d. according to some distribution $\mathcal{D}$ over $\mathcal{X} \times (\mathcal{Y} \cup \{\emptyset\})$, where $y_i = \emptyset$ indicates that ith point is unlabelled.

Let $\mathcal{H}$ be another space, referred to as a *feature space* and let $\mathbb{F}$ be a set of functions that map from $\mathcal{X}$ to $\mathcal{H}$. Given $f \in \mathbb{F}$ and for any $i \in [1, m]$ define the feature extraction of an input data point $x_i \in \mathcal{X}$ as $f(x_i) \in \mathcal{H}$. In this framework $\mathbb{F}$ describes a set of feature extraction functions (methods) available to the learner. The learner picks some feature extraction method $f^\star \in \mathbb{F}$ (we will discuss below how the choice of $f^\star$ is performed) and applies it to every x_i in the sample S, thus producing a feature extracted sample $f(S) = ((f(x_1), y_1), \ldots, (f(x_m), y_m))$.

The subsequent classification stage involves training a classifier on the extracted features $f(S)$. For that, given some hypothesis set $\mathbb{H}$ of mappings from $\mathcal{H}$ to $\mathcal{Y}$, a classifier $h^\star \in \mathbb{H}$ is learned on the training set $f(S)$. In order to apply the classifier $h^\star$ to a test point $x \in \mathcal{X}$, we first apply the feature extraction method $f^\star$ chosen at extraction stage to x and then classify it with $h^\star$, thus the application of classifier to a test point x is the composition $h^\star(f^\star(x))$.

One of the key discussions that we address is how to choose the “best” feature extraction function $f^\star \in \mathbb{F}$. For example, if $\mathbb{F}$ is parametrized, how to choose the best values of the parameters. Traditionally, the selection of best feature extraction method has been done in an unsupervised fashion, independently from the training of a classifier $h^\star \in \mathbb{H}$, which will be further referred to as *unsupervised uncoupled* feature extraction. For that, a loss function related to the feature extraction problem is introduced $L(f, x_1, ..., x_m)$ that depends only on $f \in \mathbb{F}$ and unlabelled $x \in \mathcal{X}$. Then, the best feature extraction method is selected as a loss minimization problem $f^\star = \operatorname{argmin}_{f \in \mathbb{F}} L(f, x_1, ..., x_m)$. A classic example for which $f^\star$ can be derived analytically is when $\mathcal{X} = \mathbb{R}^n$ and $\mathbb{F}$ is the set of orthogonal projections onto some $r-$dimensional subspaces of $\mathbb{R}^n$. Then, if the loss function maximizes variance, $f^\star$ is PCA (Pearson, 1901), if it preserves distances along a fixed manifold, then $f^\star$ is Isomap (Tenenbaum et al., 2000) and when it preserves angles, then $f^\star$ is Maximum Variance Unfolding (Weinberger and Saul, 2006).

A feature extraction function can also be selected in a *supervised uncoupled* way, where the labels of the input data are introduced into the loss function $L(f, x_1, ..., x_m, y_1, ..., y_m)$. A simple example would be feature selection based on correlation with the label, such as the MTFS algorithm (Argyriou et al., 2008). Finally, the most holistic feature extraction is done in a *supervised coupled* way. For that a loss function is formulated with both $f \in \mathbb{F}$ and $h \in \mathbb{H}$ as arguments $L(f, h, x_1, ..., x_m, y_1, ..., y_m)$, which is minimized *simultaneously* to obtain a coupled solution for feature extraction and classification function $(f^\star, h^\star) = \operatorname{argmin}_{f \in \mathbb{F}, h \in \mathbb{H}} L(f, h, x_1, ..., x_m, y_1, ..., y_m)$. One example of this is supervised training of deep neural networks, which jointly learns a feature representation as well as a labeling function. It has been argued that optimizing a feature extraction function jointly with a classifier can significantly improve classification accuracy (Gönen, 2014), though it usually costs more in terms of algorithmic complexity.

In the following subsections, we give more concrete examples of feature extraction techniques and how they fit into the framework described above.

2.1 Feature selection

Traditionally, feature selection methods have been divided into three groups (Guyon and Elisseeff, 2003; Liu and Motoda, 2007): *filter methods*, *wrapper methods*, and *embedded methods*. *Filter methods* select a subset of features based on the notion of an importance score that is computed independent of the classifier (Gu et al., 2012; Boutsidis et al., 2009; He et al., 2005; Chen and Lin, 2006). Such methods may be supervised or unsupervised, depending on the choice of importance score, but in any either case, they are certainly uncoupled feature extraction methods. The supervised coupled analogue are called *wrapper* methods (Kohavi and John, 1997), which select a subset of features that provides the best classification accuracy according to some underlying black-box model. A class of more specialized supervised coupled techniques are called *embedded methods* (Breiman et al., 1984), which select features (possibly in an online manner) simultaneously with the process of model training.

As discussed in (Wettschereck et al., 1997), feature selection is a special case of the more general feature weighting approach. Referring back to the framework defined in Section 2, we define the input feature space in this scenario as $\mathcal{X}^d \times \mathcal{T}^d$, where the first component in the Cartesian product defines a general set of d features and the second component $\mathcal{T}^d \subseteq \mathbb{R}^d$ corresponds to the space of input weights for the d features. A general set of feature extraction functions here is $\mathbb{F} = \{(x,t) \mapsto (x, w(t)) \colon x \in \mathcal{X}^d, t \in \mathcal{T}^d\}$, where $w \colon \mathcal{T}^d \to \mathcal{T}^d$. Thus, feature weighting is concerned with constructing or learning the weighting function w. Note, we are implicitly assuming that out learning algorithm can consume *weighted features*, i.e. elements of $\mathcal{X}^d \times \mathcal{T}^d$. This can, for example, be accomplished by weighting an instance's empirical loss proportional to its weight. Clearly, in the special case of feature selection, where $\mathcal{T} = \{0,1\}$, this coincides with simply removing the feature from the learning problem when it's weight is zero.

An illustrative example of a loss function for feature selection is given in Geng et al. (2007), where they filter features based on precomputed importance and similarity scores. Here, the input feature space is simply $\mathbb{R}^d \times \{0,1\}^d$ and $w(t) = [w_1 t_1, \ldots, w_d t_d]$, where $w_i \in \{0,1\}$. Then, given v_i, the importance score of the i-th feature, and $e_{i,j}$, the similarity score between features i and j, the feature selection problem is defined as $f^\star = \mathrm{argmin}_{w_i \in \{0,1\}, \sum_i w_i < n} C \sum_i \sum_{i \neq j} e_{i,j} x_i x_j - \sum_i v_i w_i$, where n is the threshold on the number of selected features and $C > 0$ is a tuned parameter that trades off the diversity of selected features with the total importance of selected features.

2.2 Embeddings

Linear dimensionality reduction methods are based on projecting input data to a lower dimensional subspace, usually using an objective that is uncoupled with the classifier. Many popular linear dimensionality reduction problems can be formulated as a minimization problem with orthogonal matrix constraints (Cunningham and Ghahramani, 2014, 2015). Assuming that the input space is $\mathbb{R}^n$ and the output feature space has dimension d, where $d < n$, the set of linear dimensionality reduction functions is $\mathbb{F} = \{x \mapsto \Pi x : \mathrm{rank}(\Pi) = d, x \in \mathbb{R}^n\}$, where $\Pi : \mathbb{R}^n \mapsto \mathbb{R}^d$ is an orthogonal projection. If the loss function for choosing a feature extraction method is the reconstruction error $L(f, x_1, ..., x_m) = \sum_{i=1}^m \|x_i - f(x_i)\|^2$, then $f^\star = \mathrm{argmin}_{f \in \mathbb{F}} L(f, x_1, ..., x_m)$ is rank-d PCA projection, which is by far the most popular unsupervised linear dimensionality reduction technique (van der Maaten et al., 2009).

If the objective is to preserve distances in the projected d-dimensional feature space, for example $L(f, x_1, ..., x_m) = \sum_{i,j=1}^m \left(\|\tilde{x}_i - \tilde{x}_j\| - \|f(x_i) - f(x_j)\|\right)^2$, then the solution $f^\star$ is the multidimensional scaling algorithm (Torgerson, 1952). There are plenty of other dimensionality reduction methods based on projections that derive from various loss functions; for a detailed overview refer to (Cunningham and Ghahramani, 2015).

In the same way that popular linear dimensionality reduction methods can be generalized as projections in Euclidean space, common nonlinear techniques can be generalized as projections in the reproducing kernel Hilbert space (Ham et al., 2004). All these methods can be thought of as first mapping input vectors into a reproducing kernel Hilbert space and then conducting a low-rank projection within that space. Conceptually this can even be treated as a composition of feature extraction functions, where an orthogonal projection is composed of a reproducing space map.

A well known theorem (Aronszajn, 1950) states that if $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a PDS kernel, then there exists a Hilbert space $\mathcal{H}_K$ and a mapping $\Phi_K : \mathcal{X} \mapsto \mathcal{H}_K$ such that $(\forall x, y \in \mathcal{X}) : K(x, y) = \langle \Phi_K(x), \Phi_K(y) \rangle_{\mathcal{H}_K}$. The space $\mathcal{H}_K$ is referred to as the associated reproducing kernel Hilbert space (RKHS). Thus, the set of feature extractors for nonlinear dimensionality reduction can be defined as $\mathbb{F} = \{x \mapsto \Pi\Phi_K(x) \colon x \in \mathcal{X}\}$, where Π is a rank-d orthogonal projection in $\mathcal{H}_K$. Assuming the kernel function K is fixed in advance, let us consider one of the feature extraction losses mentioned previously, but in the feature space $\mathcal{H}_K$: $L(f, x_1, ..., x_m) = \sum_{i=1}^{m} \|\Phi_K(x_i) - \Pi\Phi_K(x_i)\|^2_{\mathcal{H}_K}$. In this case $\operatorname{argmin}_{f \in \mathbb{F}} L(f, x_1, ..., x_m)$ is the rank-d Kernel PCA (KPCA) algorithm (Schölkopf et al., 1997; Blanchard et al., 2007), which is the projection onto the top-d eigenspace of the empirical covariance operator in $\mathcal{H}_K$.

When performing KPCA, the choice of the kernel function is crucial. As already mentioned, for most known manifold learning algorithms, a kernel function can be constructed so that the solution to KPCA is equivalent to that of the desired manifold learning algorithm. As shown by (Ham et al., 2004), if the kernel function is $K = -\frac{1}{2}(I - ee^T)S(I - ee^T)$, where S is the squared distance matrix and $e = \frac{1}{\sqrt{m}}(1, ..., 1)^T$, then KPCA solution recovers Isomap. Similar arguments are shown for LLE and Laplacian eigenmap (Ham et al., 2004).

Since the definition of the kernel map is a vital part of many embeddings, approximate kernel methods have been developed to make these embeddings scalable. These methods include approximating the kernel matrix, for example with Nyström method (Drineas and Mahoney, 2005; Cortes et al., 2010c), or approximating the kernel function with random Fourier features (Rahimi and Recht, 2007; Pham and Pagh, 2013; Hamid et al., 2013). A comparison of these two fundamentally different approaches is attempted in (Yang et al., 2012b), who show that Nyström approximations provide better generalization guarantees than random features when the eigen-gap of the sample kernel matrix is significant.

Nonlinear dimensionality reduction is not restricted to KPCA, in fact it can involve the construction of a more general projection in $\mathcal{H}_K$ or learn a spectral projection of a combination of operators in reproducing space (Mohri et al., 2015). This ability of kernel function to generalize nonlinear methods of dimensionality reduction provides great opportunities and at the same time a hard challenge - how to best choose a kernel function? That is a vital question in the area of feature extraction and it has been heavily studied in machine learning literature where it is commonly referred to as *(multiple) kernel learning*. We will expand more in the next section.

2.3 Metric learning

We have seen above that feature extraction techniques crucially rely on the definition of distance metric between data points, especially in case of Mahalanobis distance. A different distance metric may result in a completely different extraction algorithm. As argued in Globerson and Roweis (2005) “there is a close link between distance learning and feature extraction”, since extracting features and then using a natural metric in the feature space can be thought of as metric learning. The question of how to pick a distance function, or a feature space with natural metric, from a discrete set of available options has been well studied before. Now research is focused on a more general and challenging problem - how to learn a distance metric? For a detailed distance metric learning review refer to Kulis (2012).

Following our general framework, distance metric learning as feature extraction can be represented as $\mathbb{F} = \{\tilde{x} \mapsto (x, d) \colon \tilde{x} \in \mathcal{X}, x \in \mathcal{H}\}$, where $\mathcal{X}$ is the input space and $\mathcal{H}$ is the feature space with associated metric d. Oftentimes, the metric is implicitly defined by the space itself (i.e. via the inner-product), but we make it explicit for this formulation.

Many nonlinear distance learning methods can be obtained by kernalization (Kulis, 2012), that is, by mapping raw data to a RKHS $\mathcal{H}_K$. Since $\mathcal{H}_K$ is a Hilbert space, it has the natural metric $\|\cdot\|_{\mathcal{H}_K}$ induced by the inner product. Specifying different kernel functions K equivalently specifies different metrics on $\mathcal{H}_K$. Thus, learning a kernel function can be thought of as learning a particular class of metrics, which may be non-linear with respect to raw input data. This reveals a close connection between metric learning and *multiple kernel learning*. In the case of learning a kernel, the feature extractors can be represented as $\mathbb{F} = \{\tilde{x} \mapsto (x, \|\cdot\|_{\mathcal{H}_K}) \colon \tilde{x} \in \mathcal{X}, x \in \mathcal{H}_K\}$.

In multiple kernel learning, $\mathbb{F}$ is usually parametrized by a kernel function K that is restricted to some pre-defined set of PSD kernel functions $\mathcal{K}$ defined with respect to fixed base kernels $(K_1, \ldots, K_p)$, e.g. $\mathbb{F} = \{x \mapsto \Phi_K(x) : K \in \mathcal{K}\}$. Various kernel families $\mathcal{K}$ have been studied (Bach et al., 2004; Weinberger et al., 2004; Cortes et al., 2010b, 2013; Lanckriet et al., 2004; Cortes et al., 2010b; Bach et al., 2004), but the most widely used family is that of non-negative linear combinations of base kernels $\mathcal{K} = \{\sum_{i=1}^{p} \mu_i K_i : \boldsymbol{\mu} \in \mathbb{R}^p, \|\boldsymbol{\mu}\|_q \leq 1\}$, where the weight vector $\boldsymbol{\mu}$ is regularized within a unit ball with some L_q norm. Generalization bounds for learning a linear combination of kernels have been rigorously analyzed. Particularly, for $q = 1$, it is shown (Cortes et al., 2010b) that the generalization bound is logarithmic in the number of base kernels, which suggests to use more variety in the choice of base kernels (hence base distance metrics).

With linear combinations of kernels, distance metric learning reduces to fitting a weight vector $\boldsymbol{\mu}$. This can be done in an uncoupled manner, for instance tuning $\boldsymbol{\mu}$ to maximize some statistical measure on sample kernel matrix $K = \sum_{i=1}^{p} \mu_k K_k$, such as kernel alignment (Cortes et al., 2012). Alternatively, a potentially more effective, albeit complex, approach is to use a supervised coupled method where $\boldsymbol{\mu}$ is learned jointly with a classifier or regression in $\mathcal{H}_K$. For example, in the case of kernel ridge regression (Saunders et al., 1998), if a linear combination of kernels is learned jointly, then the resulting supervised coupled problem requires solving following minimax problem $\min_{\|\boldsymbol{\mu}\|_q \leq 1} \max_{\alpha} L(\boldsymbol{\mu}, \boldsymbol{\alpha}) = -\lambda \alpha^T \alpha - \sum_{k=1}^{p} \mu_k \alpha^T K_k \alpha + 2\alpha^T y$ (Cortes et al., 2009).

2.4 Representation learning

In the most general sense, *representation learning* could refer to the entire literature of extracting features from input data, however, in practice it is usually associated with extracting features via multi-layer neural networks and is studied within neural network research community. Here, we will treat representation learning as the subset of feature extraction methods that leverage neural networks and its specific realizations, such as autoencoders and Boltzmann machines. The holy grail of representation learning is to completely automate the feature extraction step from raw input data (e.g. images) so that no human engineering is necessary for hand crafting features (Bengio and Courville, 2013). Recently, a remarkable success on video, image and speech data has been demonstrated by deep networks, which attempt to learn multiple levels of representations of increasing complexity and abstraction (Bengio et al., 2013). The main aspect that distinguishes deep nets from other techniques is that, unlike previously mentioned feature extraction methods (clustering, feature selection, PCA, manifold learning) they learn *multiple* levels of representation. Usually the higher level of representations are more abstract and nonlinear, capturing structures that are not obvious from the input data.

To define the deep learning representation problem in the framework of this paper, we define the family of feature extraction functions as composition of functions $\mathbb{F} = \{x \mapsto f_n(...f_1(f_0(x))...)\}$, where $f_i : \mathcal{H}_i \mapsto \mathcal{H}_{i+1}$, and $\mathcal{H}_i$ is the data representation at ith layer. As for the target loss function, the idea was to learn f_i sequentially from $i = 0$ to $n + 1$ in an unsupervised uncoupled manner (for example, by reducing reconstruction error) that first revolutionized deep architectures (Hinton and Salakhutdinov, 2006; Bengio et al., 2007; Poultney et al., 2006). The supervised analogue of the loss function depends not only on input data representation at nth level, but also on prediction made by the last layer f_{n+1} as well as the data labels for the supervised task (Lee et al., 2009a,b). Examples of representation learning using deep architectures are autoencoders (Bengio et al., 2007; Goodfellow et al., 2009), denoising autoencoders (Vincent et al., 2008)), contractive autoencoders (Rifai et al., 2011), restricted Boltzmann machines (Hinton and Salakhutdinov, 2006).

It is important to stress that there is a connection between representation learning and manifold learning, especially for shallow architectures. For instance, linear autoencoders with square loss function are equivalent (up to rotation) to PCA (Bourlard and Kamp, 1988), as they essentially learn to represent data in the subspace with smallest squared reconstruction error. Moreover, approximate kernel feature maps (Rahimi and Recht, 2007; Pham and Pagh, 2013; Hamid et al., 2013) provide the representation of data, the inner product of which approximates kernel function. Such approximate feature maps can also be learned, in which case representation learning is shown (Yu et al., 2015) to be equivalent to training a shallow neural network. It is a major open challenge to study connections between representation learning and manifold learning, especially for deep architectures.

3. Questions and challenges in feature extraction

3.1 Supervised and unsupervised approaches

Classic manifold learning methods such as LLE, Isomap, Laplacian Eigenmap have proven to be able to efficiently extract patterns in unlabeled data. Various supervised extensions of these methods have emerged that make use of labels for adjusting the distance between distinct classes (de Ridder et al., 2003; Zhao et al., 2005; Kouropteva et al., 2003; Geng et al., 2005; Yang et al., 2006; Li and Guo, 2006). Distance metric learning also has efficient supervised extensions (Hoi et al., 2008). Moreover, supervised learning of projections in a fixed RKHS have proven to be effective for nonlinear feature extraction (Blanchard and Zwald, 2008; Fukumizu et al., 2004) as well as for multiple kernel learning (Gönen and Alpaydın, 2011; Gönen, 2014). This progress in supervised methods and compelling empirical evidence raises an important debate: should supervised and coupled feature extraction always be preferred to uncoupled and/or unsupervised methods? It is an open debate, since even though supervised methods often improve classification accuracy, this comes at a cost flexibility and scalability. It is often the case that unlabeled examples can be gathered much more easily than their labeled counterparts. Additionally, supervised coupled methods generally require much more computationally intensive algorithms. Can we shed more light on the trade-off between supervised and unsupervised methods? Can we understand, which methods are most useful for particular settings and why?

There are multiple interesting attempts to address the open question described above. In Mohri et al. (2015) the authors show examples where conducting dimensionality reduction blindly (i.e. in an unsupervised manner) adversely affects subsequent classification stage. Thus, they argue for supervised coupled dimensionality reduction by providing a general framework of learning a combination of multiple kernels jointly with projection in the reproducing space for classification. The argument is supported by a favorable generalization bound for such a hypothesis set that depends on the number of base kernels logarithmically. On the other hand, it is shown that for some data types and settings, unsupervised methods work better. An interesting example is the work of (Guerra et al., 2011), where they show on clinical test data that unsupervised techniques work better than supervised for very high dimensional representations. Also, (Wang et al., 2010) claims that kernel based unsupervised dimensionality reduction by maximizing information in covariates is preferable to supervised approach for the purpose of clustering and visualization as well as embedding data into very few dimensions.

3.2 Scalability

Tremendous growth in the size of datasets for machine learning makes many feature extraction methods infeasible, especially the more complex nonlinear methods. Almost ten years ago (Guyon and Elisseeff, 2003) it was a challenge to run a feature selection algorithm on thousands of features, now the challenge is to do so for millions of input dimensions. This raises a significant challenge: how can we scale up feature extraction? Which methods can be parallelized? How can we balance bottlenecks between the number of features and the number of instances?

The scaling of linear feature extraction methods mostly relies on matrix approximations and parallel linear algebra algorithms. Significant progress has been made in column subset selection (Tropp, 2009), where scalability is achieved by randomly sampling columns with an appropriate distribution. For instance (Boutsidis et al., 2009) develop a column sampling algorithm, where the distribution depends on the singular subspace of input matrix. Large scale non-linear methods such as deep neural networks also benefit heavily from distributed training algorithms, such as distributed stochastic gradient descent (SGD) (Zinkevich et al., 2010; Dean et al., 2012).

While the computational bottleneck for linear feature extraction is often the number of features, the bottleneck for kernelized non-linear methods is the number of instances. Since most nonlinear projections involve decomposition of an m-by-m kernel matrix, where m is the number of instances, and implies that only datasets of up to tens of thousand of points are feasible for manifold learning on a single machine. A breakthrough that has allowed kernel methods to scale are approximate kernel feature maps (Rahimi and Recht, 2007), which map the input data to a randomized feature space, where the inner product approximates the kernel function. This allows one to switch from solving the dual problem to solving the primal problem and overcome the number of instances bottleneck as well as use more easily distributed algorithms such as SGD. A number of efficient approximate maps have been proposed for different types of kernels, with particular attention to polynomial kernels (Pham and Pagh, 2013; Hamid et al., 2013). However a useful approximation generally requires generating a large number of random features (a factor larger than the input dimension), which creates another bottleneck. One novel work that attacks this problem is Pennington et al.

(2015), where they show that a simple normalization of input data to a unit l_2 sphere and the use of spherical random Fourier features achieves an accurate kernel approximation with many fewer random features. Moreover, the use of Monte Carlo methods for shift invariant kernels has shown strong results (Avron et al., 2014).

The latest research in approximate kernel methods is asking an even deeper question than how to make kernel methods scalable. The question is: how can we learn approximate nonlinear maps in a supervised manner jointly with a classifier? A recent work (Yu et al., 2015) attempts to answer that question by supervised learning of nonlinear approximate map parameters instead of random generation, which results in a more compact model and competitive performance. Additionally, joint learning of approximate feature maps can be represented as a shallow neural network, which brings us to the open problem described in the next section: connection between convex and non-convex feature extraction.

3.3 Global and local minima in feature extraction

As discussed in previous paragraphs, representation learning is a novel and promising feature extraction technique that relies on neural networks. However, despite compelling empirical performance, neural networks lack well grounded theoretical guarantees in part due to their non-convexity (although there has been some recent progress (Arora et al., 2014; Livni et al., 2014; Sedghi and Anandkumar, 2014)) and multiple local minima. On the other hand, we have well studied dimensionality reduction methods with global minima and as showed in previous paragraphs, most of them can be described with kernels. This raises a significant open question that links distinct parts of feature extraction, inspired by the developments in both kernel methods and deep learning: what is the connection between kernels and deep neural networks as means of feature extraction? While deep nets suffer from non-convexity and the lack of theoretical guarantees, kernel machines are convex and well studied mathematically. Thus, it is extremely tempting for us to resort to kernels (in addition to any other available tools) to help better understanding neural nets. Answering these question could help bridge the gap between convex and non-convex feature extraction techniques.

There has been some progress to that end. Particularly, sequences of deep kernels have been used to study the layer-wise transformation of neural nets input. The works of (Montavon et al., 2011; Bach et al., 2015) analyzed deep network with the idea of *relevance decomposition* - that is, determining which inputs/pixels are important for an image to be classified as what type of objects. The importance of pixels is explained by a heatmap that highlights pixels that are responsible for the predicted class membership. This sheds light on the internal decision logic of a deep network.

A major impulse towards understanding the connection between kernels and deep networks comes from the development of scalable approximate kernel maps. Achieving scalability is a necessary first step in comparison of kernel methods versus deep nets, because until recently the data sets for which deep nets perform best are not even computationally feasible for kernels. A successful attempt to solve that issue comes from the works that use distributed approximate feature maps (Lu et al., 2014; Huang et al., 2014), which allows them to build large scale kernel architectures that match deep networks in accuracy. As discussed in Yu et al. (2015), the problem of joint learning of approximate map and a classifier can be described as training a shallow neural network, which is a great start for understanding the connection. If the gap can be bridged for shallow nets, can this be done for deep ones?

Given that for some applications shallow networks can achieve the performance of deep ones, there is a natural question: "Do deep nets really need to be deep?" (Ba and Caruana, 2014). More specifically, the question addressed by Ba and Caruana (2014) as well as by Vincent et al. (2010); Dauphin and Bengio (2013); Seide et al. (2011) is whether the increase in accuracy of deep nets over the shallow nets is explained by the inherent ability of deep nets to learn more complex representations or merely by better training. It has been shown in Ba and Caruana (2014) that if a shallow network is trained on the output of a more complex deep network to mimic it, then the former can be as accurate as the latter, in some cases with the same number of parameters. Such empirical evidence suggests that, possibly, the outstanding performance of deep networks arises from "a good match between deep architectures and the current training procedures".

4. Conclusion

In this paper we have surveyed, compared and contrasted several different methods for feature extraction. We have also described some high level research directions and challenges in this field, including supervised versus unsupervised methods, scalability and convex versus non-convex models. We hope this serves as a useful resource across the related fields.

References

Selim Aksoy and Robert M Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern recognition letters*, 22(5):563–582, 2001.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, pages 337–404, 1950.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *Proceedings of The 31st International Conference on Machine Learning*, pages 584–592, 2014.

Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael Mahoney. Quasi-monte carlo feature maps for shift-invariant kernels. *arXiv preprint arXiv:1412.8293*, 2014.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014.

Francis R. Bach, Gert R.G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.

Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. Learning to rank with (a lot of) word features. *Information retrieval*, 13(3):291–314, 2010.

Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *ICML*, volume 3, pages 11–18, 2003.

Claude Barras and Jean-Luc Gauvain. Feature and score normalization for speaker verification of cellular data. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 2, pages II–49. IEEE, 2003.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Yoshua Bengio and Aaron C. Courville. Deep learning of representations. *Handbook on Neural Information Processing*, 49, 2013.

Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards AI. *Large-scale kernel machines*, 34(5), 2007.

Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

Gilles Blanchard and Laurent Zwald. Finite-dimensional projection for classification and statistical learning. *Information Theory, IEEE Transactions on*, 54(9):4169–4182, 2008.

Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294, 2007.

Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1):245–271, 1997.

Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.

Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977. Society for Industrial and Applied Mathematics, 2009.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Yi-Wei Chen and Chih-Jen Lin. Combining SVMs with various feature selection strategies. In *Feature extraction*, pages 315–324. Springer, 2006.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L 2 regularization for learning kernels. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 109–116, 2009.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 247–254, 2010a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 247–254, 2010b.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *International Conference on Artificial Intelligence and Statistics*, pages 113–120, 2010c.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Corinna Cortes, Marius Kloft, and Mehryar Mohri. Learning kernels using local rademacher complexity. In *Advances in Neural Information Processing Systems*, pages 2760–2768, 2013.

John P Cunningham and Zoubin Ghahramani. Unifying linear dimensionality reduction. *arXiv preprint arXiv:1406.0873*, 2014.

John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 2015.

Yann N Dauphin and Yoshua Bengio. Big neural networks waste capacity. *arXiv preprint arXiv:1301.3583*, 2013.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

Dick de Ridder, Olga Kouropteva, Oleg Okun, Matti Pietikäinen, and Robert PW Duin. Supervised locally linear embedding. In *Artificial Neural Networks and Neural Information ProcessingICANN/ICONIP 2003*, pages 333–341. Springer, 2003.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

Richard O Duda, Peter E Hart, and David G Stork. Unsupervised learning and clustering. *Pattern classification*, pages 519–598, 2001.

Hazim Kemal Ekenel and Rainer Stiefelhagen. Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 34–34. IEEE, 2006.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Ronald A Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8(4):376–386, 1938.

Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *The Journal of Machine Learning Research*, 5:73–99, 2004.

Xing Geng, De-Chuan Zhan, and Zhi-Hua Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35 (6):1098–1107, 2005.

Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. Feature selection for ranking. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 407–414. ACM, 2007.

Amir Globerson and Sam T Roweis. Metric learning by collapsing classes. In *Advances in neural information processing systems*, pages 451–458, 2005.

Mehmet Gönen. Coupled dimensionality reduction and classification for supervised and semi-supervised multilabel learning. *Pattern recognition letters*, 38:132–141, 2014.

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *Advances in neural information processing systems*, pages 646–654, 2009.

Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

Luis Guerra, Laura M McGarry, Víctor Robles, Concha Bielza, Pedro Larranaga, and Rafael Yuste. Comparison between supervised and unsupervised classifications of neuronal cell types: a case study. *Developmental neurobiology*, 71(1):71–82, 2011.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47. ACM, 2004.

Raffay Hamid, Ying Xiao, Alex Gittens, and Dennis DeCoste. Compact random feature maps. *arXiv preprint arXiv:1312.4626*, 2013.

Trevor Hastie and Rolbert Tibshirani. Discriminant adaptive nearest neighbor classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(6):607–616, 1996.

Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2005.

Chinmay Hegde, Michael Wakin, and Richard Baraniuk. Random projections for manifold learning. In *Advances in neural information processing systems*, pages 641–648, 2008.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Steven CH Hoi, Wei Liu, and Shih-Fu Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

Po-Sen Huang, Haim Avron, Tara N Sainath, Vikas Sindhwani, and Bhuvana Ramabhadran. Kernel methods match deep neural networks on timit. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 205–209. IEEE, 2014.

Faten Hussein, Nawwaf Kharma, and Rabab Ward. Genetic algorithms for feature selection and weighting, a review and study. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 1240–1244. IEEE, 2001.

Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

Olga Kouropteva, Oleg Okun, and Matti Pietikäinen. Supervised locally linear embedding algorithm for pattern recognition. In *Pattern Recognition and Image Analysis*, pages 386–394. Springer, 2003.

Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

Martin H.C. Law and Anil K. Jain. Incremental nonlinear dimensionality reduction by manifold learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):377–391, 2006.

ND Lawrence. Gaussian process models for visualisation of high dimensional data, 2003. In *Neural Information Processing Systems (NIPS)*, 2004.

Neil D Lawrence. Learning for larger datasets with the gaussian process latent variable model. In *International Conference on Artificial Intelligence and Statistics*, pages 243–250, 2007.

Neil D Lawrence. A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *The Journal of Machine Learning Research*, 13(1):1609–1638, 2012.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009a.

Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009b.

John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

Chun-Guang Li and Jun Guo. Supervised isomap with explicit mapping. In *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on*, volume 3, pages 345–348. IEEE, 2006.

Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media, 1998.

Huan Liu and Hiroshi Motoda. *Computational methods of feature selection*. CRC Press, 2007.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

Zhiyun Lu, Avner May, Kuan Liu, Alireza Bagheri Garakani, Dong Guo, Aurélien Bellet, Linxi Fan, Michael Collins, Brian Kingsbury, Michael Picheny, et al. How to scale up kernel methods to be as good as deep neural nets. *arXiv preprint arXiv:1411.4000*, 2014.

Mehryar Mohri, Afshin Rostamizadeh, and Dmitry Storcheus. Foundations of coupled nonlinear dimensionality reduction. *arXiv preprint arXiv:1509.08880v2*, 2015.

Grégoire Montavon, Mikio L Braun, and Klaus-Robert Müller. Kernel analysis of deep networks. *The Journal of Machine Learning Research*, 12:2563–2581, 2011.

Patrenahalli M Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *Computers, IEEE Transactions on*, 100(9):917–922, 1977.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Jeffrey Pennington, X Yu Felix, and Sanjiv Kumar. Spherical random features for polynomial kernels. *NIPS*, 2015.

Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–247. ACM, 2013.

Christopher Poultney, Sumit Chopra, Yann L Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2006.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

Alain Rakotomamonjy. Variable selection using svm based criteria. *The Journal of Machine Learning Research*, 3:1357–1370, 2003.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Lawrence K. Saul, Kilian Q. Weinberger, Jihun H. Ham, Fei Sha, and Daniel D. Lee. Spectral methods for dimensionality reduction. *Semisupervised learning*, pages 293–308, 2006.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural NetworksICANN'97*, pages 583–588. Springer, 1997.

D Sculley, G Holt, D Golovin, E Davydov, T Phillips, D Ebner, V Chaudhary, and M Young. Machine learning: The high interest credit card of technical debt. In *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*, 2014.

Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. *arXiv preprint arXiv:1412.2693*, 2014.

Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Interspeech*, pages 437–440, 2011.

Andreas Stolcke, Sachin Kajarekar, and Luciana Ferrer. Nonparametric feature normalization for svm-based speaker verification. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 1577–1580. IEEE, 2008.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

Joel A Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 978–986. Society for Industrial and Applied Mathematics, 2009.

Laurens JP van der Maaten, Eric O Postma, and H Jaap van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

Jarkko Venna et al. *Dimensionality reduction for visual exploration of similarity structures*. Helsinki University of Technology, 2007.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

Meihong Wang, Fei Sha, and Michael I. Jordan. Unsupervised kernel dimension reduction. In *Advances in Neural Information Processing Systems*, pages 2379–2387, 2010.

Kilian Q. Weinberger and Lawrence K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*, volume 6, pages 1683–1686, 2006.

Kilian Q. Weinberger, Fei Sha, and Lawrence K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, page 106. ACM, 2004.

Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

Dietrich Wettschereck, David W Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):273–314, 1997.

Eric P. Xing, Michael I. Jordan, Stuart Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512, 2002.

Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, 2, 2006.

Liu Yang, Rong Jin, and Rahul Sukthankar. Bayesian active distance metric learning. *arXiv preprint arXiv:1206.5283*, 2012a.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012b.

Xin Yang, Haoying Fu, Hongyuan Zha, and Jesse Barlow. Semi-supervised nonlinear dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 1065–1072. ACM, 2006.

Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

Felix X Yu, Sanjiv Kumar, Henry Rowley, and Shih-Fu Chang. Compact nonlinear maps and circulant extensions. *arXiv preprint arXiv:1503.03893*, 2015.

Qijun Zhao, David Zhang, and Hongtao Lu. Supervised lle in ica space for facial expression recognition. In *Neural Networks and Brain, 2005. ICNN&B'05. International Conference on*, volume 3, pages 1970–1975. IEEE, 2005.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, pages 2595–2603, 2010.