ALEXANDRU IOAN CUZA UNIVERSITY of IAȘI

“ALEXANDRU IOAN CUZA” UNIVERSITY OF IAȘI

Contributions in Probabilistic Machine Learning

Summary

by

Sebastian-Adrian Ciobanu

Supervisor: Professor Dr. Henri Luchian

A thesis submitted in fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Computer Science

Iași, 2022

"ALEXANDRU IOAN CUZA" UNIVERSITY OF IAȘI

Abstract

Faculty of Computer Science

Doctor of Philosophy

Contributions in Probabilistic Machine Learning

by Sebastian-Adrian Ciobanu

This thesis is concerned with the investigation of a portion of the theory in probabilistic machine learning in order to create new models/algorithms. In this process, we established some building blocks for composing novelties: probabilistic distributions, probabilistic models/tasks, and relationships between the models, e.g. turning on/off the linear/non-linear character of a model. Picking a distribution and a model/task can result in a new research idea that can be augmented with extra directions: to use the model in a different context than the one in which it is usually used or to pick a relationship through which you turn the chosen model into another one. By doing so, we have obtained five applications, all having common properties, i.e. each of them creates a new model/algorithm, uses maximum likelihood estimation for training/learning, and has an experimentation part. If the first application refers to a traditional approach to density estimation in the context of semantic image inpainting, the second one creates the supervised counterpart of factor analysis—a pre-existing unsupervised machine learning model—, links it to linear regression, and creates semisupervised and missing-data versions of this counterpart. Two other applications involve adding non-linearities to the Gaussian mixture model via neural networks directly or by using a normalizing flow model. Another application regards the autoencoder as a discriminative model and uses distributions different from the normal distribution. Regarding the implementation which is public, four applications are written in Python using TensorFlow (and implicitly automatic differentiation) and TensorFlow Probability, and the application related to factor analysis is written in R and does not use automatic differentiation. As for the results, we do not necessarily beat the state-of-the-art performances, but we consider their presentation relevant for knowing how our models behave in comparison to other pieces of related work in terms of metrics and advantages/disadvantages.

Keywords: Probabilistic Machine Learning, Discriminative/Generative Models, Probabilistic Distributions, Linear/Non-linear Models, Maximum Likelihood Estimation, Systematization

Contents

Contributions and Thesis Structure

Each chapter of the thesis has a corresponding homonymous chapter in this summary.

When referring to the contributions of this thesis, we are compelled to mention two keywords: systematization and MLE. The systematization of different concepts in (probabilistic) machine learning was our basis for constructing the paths to our research. MLE stands for maximum likelihood estimation [15, sec. 4.2], and each model we created in our research was trained/learned using this principle. Another possible title for our thesis would have been "*Applications of Maximum Likelihood Estimation*" which again stresses the importance of the MLE pillar in our research.

The contributions presented in this thesis revolve around the probabilistic machine learning field, as expected from the title. The structure of this thesis is as follows.

- The next chapter (chapter 1) is an introduction to probabilistic machine learning.
- Each chapter from chapter 2 to chapter 6 corresponds to a contribution, as listed in Table 1.
- The last chapter (chapter 7) concludes this thesis.

Table 1 shows a high-level view of the contributions. Each contribution is considered to be a combination of three aspects:

I. Original probabilistic model/task: an existing probabilistic model/task which is the starting point of the research in that chapter

II. Distribution: we set a distribution or a mixture of distributions to use in our research

III. Novelty: different axes on which we can test the novelty of our research.

Each chapter of the thesis contains granular contributions: experiments (organization, tables, figures), links to code (experimentation code—chapter 2, chapter 4, chapter 5, chapter 6—or library—chapter 3), and theoretical propositions (chapter 3).

Furthermore, the organization in Table 1 can also be interpreted as a framework to create new research ideas: one needs to set I and II and to have at least a check mark in the novelty part (III).

Contribution	I. Original probabilistic...		II. Distribution		III. Novelty			
	... model	... task	Name	Number of distributions in the mixture (1=there is no mixture)	The combination model/task x distribution (x methodology) is novel?	Uses the original model/task in a novel context?	Converts the original model/task to be supervised / unsupervised?	Converts the original model/task to be linear / non-linear?
Chapter 2	×	density estimation	PixelCNN++ (an autoregressive model)	1	×	✓ **semantic image inpainting**	×	×
			Multivariate normal	2	×			
			Matrix normal		×			
Chapter 3	FA Probabilistic PCA	dimensionality reduction	The distribution(s) in the original model		×	×	✓ **supervised** (dimensionality reduction turns into regression)	×
Chapter 4	GMM (k-means)	clustering	Multivariate normal	the (already known) number of clusters in the dataset	×	×	×	✓ **non-linear** (beyond quadratic) via neural networks
			Multivariate Cauchy		✓			
Chapter 5	Autoencoder (viewed as a discriminative probabilistic model)	dimensionality reduction	Multivariate normal	1 10	×	×	×	×
			Multivariate t		✓			
			MAF (an NF model)		✓			
			Matrix normal		✓			
			Matrix t		✓			
			(we compared the model also to a VAE)		×			
Chapter 6	(GMM)	**density estimation** and clustering	Multivariate normal	the (already known) number of clusters in the dataset	×	×	×	✓ **non-linear** (beyond quadratic) via NFs (implicitly via neural networks)
			MAF (an NF model)		✓			

TABLE 1: Contributions and thesis structure

List of Publications

1. Sebastian Ciobanu. "Fixed Representatives and Variable Features: From Regularized Deep Clustering to Classic Clustering". In: *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE. 2021, pp. 1–6. URL: https://ieeexplore.ieee.org/document/9548500. **Conference Paper (regular/long), C Category (2 points)**.

2. Sebastian Ciobanu. "Mixtures of Normalizing Flows". In: *Proceedings of ISCA 34th International Conference on Computer Applications in Industry and Engineering*. Vol. 79. 2021, pp. 82–90. URL: https://easychair.org/publications/paper/Scnv. **Conference Paper (regular/long), C Category (2 points)**.

3. Sebastian Ciobanu. "Vanilla Probabilistic Autoencoder". In: *Proceedings of ISCA 34th International Conference on Computer Applications in Industry and Engineering*. Vol. 79. 2021, pp. 71–81. URL: https://easychair.org/publications/paper/R2l1. **Conference Paper (regular/long), C Category (2 points)**.

4. Sebastian Ciobanu and Liviu Ciortuz. "A Factor Analysis Perspective on Linear Regression in the 'More Predictors than Samples' Case". In: *Entropy* 23.8 (2021). URL: https://www.mdpi.com/1099-4300/23/8/1012. **Journal Paper (regular/long), B Category (4 points)**.

5. Sebastian Ciobanu and Liviu Ciortuz. "Semantic Image Inpainting via Maximum Likelihood". In: *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE. 2020, pp. 153–160. URL: https://ieeexplore.ieee.org/document/9357079. **Conference Paper**

(regular/long), D Category (1 point); 1 citation on 05/31/2022 which is worth 1 point[1]:

- Su, S., Yang, M., He, L., Shao, X., Zuo, Y., Qiang, Z. (2022). A Survey of Face Image Inpainting Based on Deep Learning. In: Khosravi, M.R., He, Q., Dai, H. (eds) Cloud Computing. CloudComp 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 430. Springer, Cham. https://doi.org/10.1007/978-3-030-99191-3_7

.

[1] "Perspectiva c): Impactul rezultatelor" at https://www.uaic.ro/wp-content/uploads/2013/08/1-ORDIN-nr.6129-din-2016-aprobare-standarde-minime-pentru-conferirea-titlurilor-didactice.pdf; visited on 05/31/2022

Chapter 1

Introduction to Probabilistic Machine Learning

Probabilistic machine learning encompasses all the machine learning algorithms that use probabilities. Although a wide range of algorithms does so, our focus will be on probability distributions (classic—e.g. Gaussian distribution [13, sec. 2.4.1]—and modern) and their use in machine learning algorithms. When we say *modern distributions*, we refer to distributions that arose from the deep learning community. These distributions are from the category called *deep generative models* to which the following belong: autoregressive models [2, lecture 3], variational autoencoders (VAE) [8], normalizing flow (NF) models [2, lecture 7], and generative adversarial networks (GAN) [4].

Keywords: Probabilistic Machine Learning, Distributions, Gaussian, Autoregressive, VAE, Normalizing Flow, GAN

Machine learning algorithms can be classified as probabilistic and non-probabilistic, depending on whether they do or do not use probabilities. Although the book *Machine Learning: A Probabilistic Perspective* [13], by Kevin Murphy, tackles this subject in an extensive manner, after 2012 (the year that the book was published) new probabilistic machine learning algorithms appeared.[1] Those are often referred to as deep generative models. As a result, there is a multitude of probabilistic machine learning models, but we will try to treat in the thesis just the subjects with which our research is concerned:

- probability basics,
- classic models,

[1] This is an important reason why the author, Kevin Murphy, has recently updated and extended his original book, *Machine Learning: A Probabilistic Perspective*, into two new ones: [15] and [14].

- modern models.

The reasons why we started to study probabilistic (interpretations of) machine learning models—as opposed to non-probabilistic ones—are multiple. These models can be used in (after [13, sec. 8.6.1] and [2, lecture 3]):

- density estimation
  - plugging this (new) density/mass (probability density function or probability mass function) in various situations:
    - clustering
    - output of a classic neural network
    - etc.
  - anomaly detection, e.g. fraud detection
  - handling missing data
  - handling unlabeled training data: semisupervised learning
- sampling (generating new data)
  - data augmentation
- unsupervised representation learning
  - dimensionality reduction
  - latent space interpolation, e.g. image morphing from X to Y, from smiling person X to non-smiling person X, etc.

Observation: not all these advantages apply to a single model! Some models have some advantages, some models have others.

In the thesis, we presented the concepts we investigated in order to create a better view of the probabilistic machine learning field. Since the field is very wide, the concepts we chose serve just as an introduction. Nevertheless, this limited view helped us to organize our ideas and to create paths of research in probabilistic machine learning. As a result, we can systematize the probabilistic distributions in Table 1.1, the probabilistic tasks in Figure 1.2, and the probabilistic models in Figures 1.3 and 1.4. These tables/figures do not contain all the concepts presented in the thesis in this chapter, but only those we considered crucial for this new system we composed. Regarding the last two figures mentioned, for each model we indicated two algorithmic types:

<table>
<tr><th rowspan="3"></th><th colspan="2">discrete</th><th colspan="4">continuous</th></tr>
<tr><th colspan="2">parametric</th><th colspan="2">parametric</th><th colspan="2">non-parametric</th></tr>
<tr><th>with 2 possible values (binary)</th><th>with 2 or more possible values</th><th>1-dimensional</th><th>multi-dimensional</th><th>1-dimensional</th><th>multi-dimensional</th></tr>
<tr><td rowspan="3">classic</td><td rowspan="3">Bernoulli</td><td rowspan="3">Categorical</td><td>Normal</td><td>Multivariate Normal</td><td colspan="2" rowspan="2">Gaussian Process
(needs inputs; outputs are 1-dimensional; however, its hyperparameters are learnable via ML-II)</td></tr>
<tr><td rowspan="2">others, not discussed in this chapter: e.g. t, Gamma, etc.</td><td rowspan="2">others, not discussed in this chapter: e.g. multivariate t, matrix normal, etc.</td></tr>
<tr><td colspan="2">Kernel Density Estimation (KDE)</td></tr>
<tr><td>modern</td><td>-</td><td>-</td><td>-</td><td>Autoregressive models: NADE, RNADE, etc.</td><td colspan="2" rowspan="2">-</td></tr>
<tr><td></td><td></td><td></td><td>Normalizing Flows (NFs): NICE, MAF, etc.</td></tr>
</table>

TABLE 1.1: List of probabilistic distributions. Abbreviations: NADE—Neural Autoregressive Density Estimation, RNADE—Real-valued Neural Autoregressive Density Estimator, NICE—Non-linear Independent Components Estimation, MAF—Masked Autoregressive Flow

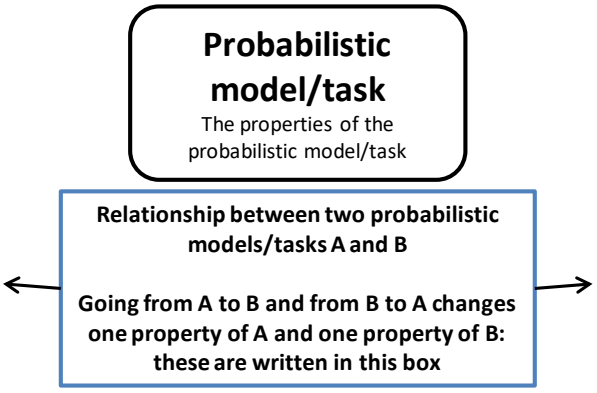

FIGURE 1.1: Interactions between probabilistic models/tasks: legend

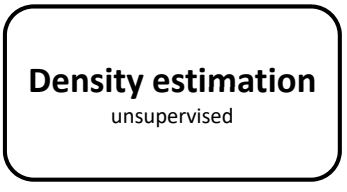

FIGURE 1.2: Probabilistic tasks; see Figure 1.1 for the legend

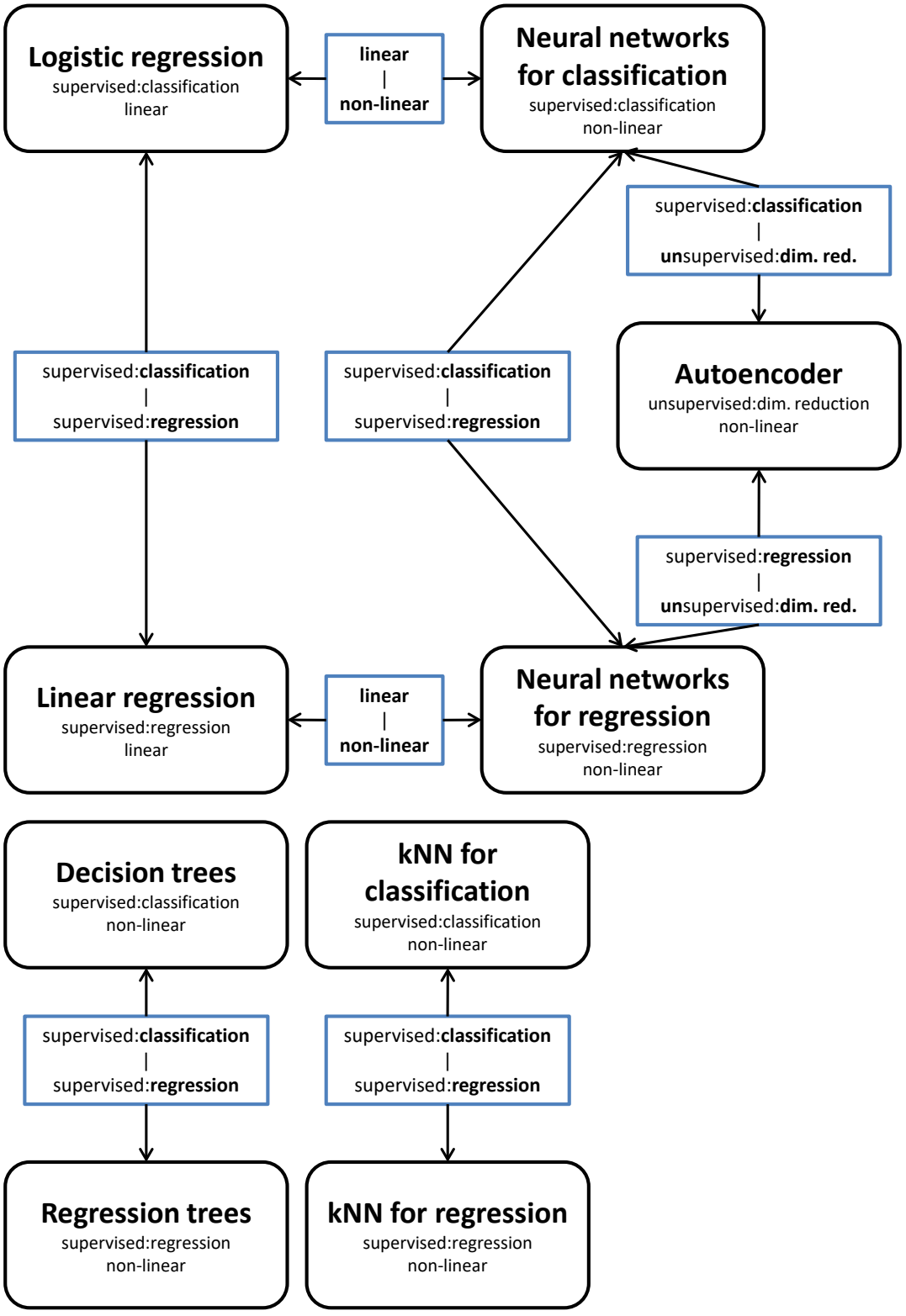

FIGURE 1.3: Interactions between the discriminative [13, sec. 8.6] models; see Figure 1.1 for the legend. Abbreviations: kNN—k-Nearest Neighbors

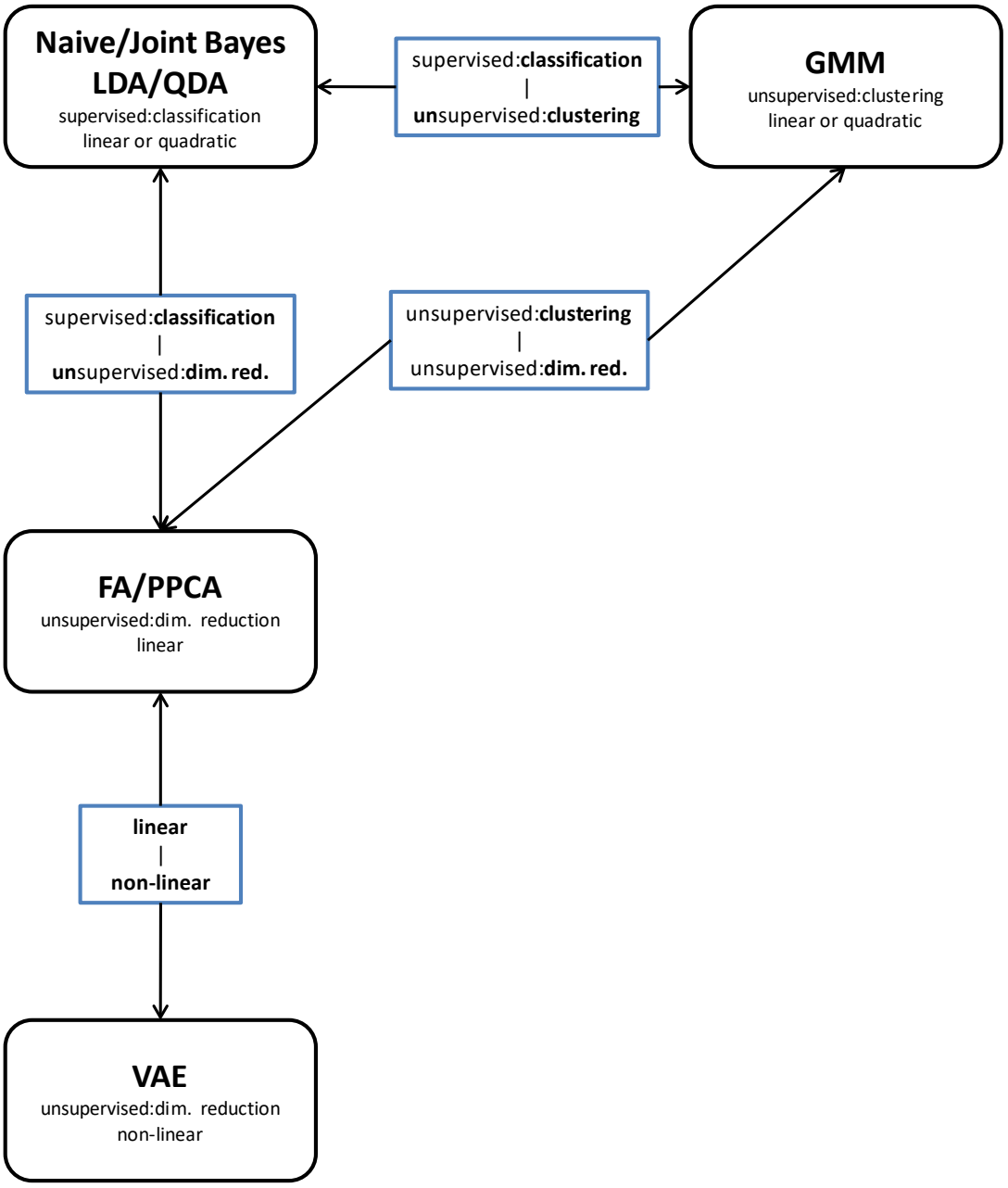

FIGURE 1.4: Interactions between the generative [13, sec. 8.6] models; see Figure 1.1 for the legend. Abbreviations: LDA—Linear Discriminant Analysis, QDA—Quadratic Discriminant Analysis, GMM—Gaussian Mixture Model, FA—Factor Analysis, PPCA—Probabilistic Principal Component Analysis, VAE—Variational AutoEncoder

1. supervised:classification, supervised:regression, unsupervised:clustering, unsupervised:dimensionality reduction

2. linear, non-linear.

For example (see Figure 1.4), FA (factor analysis) [16] [13, p. 381] is the counterpart of Naive/Joint Bayes/LDA/QDA (naive Bayes [13, sec. 3.5]/joint Bayes/linear discriminant analysis [13, sec. 4.2.2]/quadratic discriminant analysis [13, sec. 4.2.1]) from the classification/dimensionality reduction point of view and the counterpart of GMM (Gaussian Mixture Model) [13, sec. 11.2.1] from the clustering/dimensionality reduction point of view.

Each one of the next chapters excluding the last one will consist of combining a list of distributions with a probabilistic model/task and optionally of a new context (a new task, transforming a model from one type to another, etc.) in which one can test this combination.

In the final chapter we will revise Figures 1.1, 1.2, 1.3, and 1.4 to see what we managed to add throughout the thesis.

The next chapter applies generative algorithms (or the density estimation task) to semantic image inpainting; we compare the classic approach to the modern approach and use, in both cases, the same meta-algorithm for image inpainting.

Chapter 2

Semantic Image Inpainting via Maximum Likelihood

In this chapter, we proposed a meta-algorithm for semantic image inpainting. It involves setting a probabilistic distribution on images, learning the parameters of the distribution from a training dataset through maximum likelihood, and, for an image with holes, determining the optimal values of pixels that maximize the density/mass of the image. We employed classic distributions (multivariate normal, matrix normal), modern ones (PixelCNN++ [21]), and mixtures of distributions. We compared our results to one representative in the literature for semantic image inpainting (Partial Convolution [10]) and to another one for non-semantic image inpainting (PatchMatch [1]). At least two major results should be mentioned:

- generating new samples vs. inpainting: after the learning step, PixelCNN++ works well at sampling but poorly at inpainting in the manner we proposed; for multivariate normal and matrix normal, this happens the other way around: they work poorly at sampling but well at inpainting in the manner we proposed;
- faster experimentation vs. better visual results: we can experiment faster than Partial Convolution with our meta-algorithm, but Partial Convolution returns the best results in terms of visual quality.

When it comes to the **advantages** of our model compared to other inpainting algorithms, we will highlight the following:

- our model is generic, i.e. we proposed a meta-algorithm that can be instantiated by setting a parametric probabilistic distribution;

- the training phase is straightforward, i.e. likelihood maximization;
- our model can be used to see in a timely manner if, for a given dataset, an inpainting task can be learned because the training time is shorter than the time necessary for training a Partial Convolution model.

The next chapter creates the supervised version of factor analysis (FA), simple-supervised factor analysis (S2.FA), by using a classic approach, links this model to linear regression (LR), and then extends S2.FA to semisupervised learning (S3.FA) and learning with missing data (MS3.FA).

Chapter 3

A Factor Analysis Perspective on Linear Regression in the 'More Predictors Than Samples' Case

In this chapter, we created the supervised counterpart (S2.FA) of factor analysis (FA) [16] [13, p. 381]—a generative probabilistic machine learning model usually used in dimensionality reduction—and we employed it as a solution to the problem of applying linear regression (LR) [13, ch. 7] when "$D >> n$"[1]. This was possible by proving a link between S2.UncFA (an unconstrained version of S2.FA) and LR. Furthermore, since the factor analysis model is generative, we managed to create versions supporting semisupervised learning (S3.FA) and learning with missing data in the input (MS3.FA). In fact, compared to other solutions (Moore-Penrose inverse and L2 regularization), this is the **advantage** of our approach (S2.FA), i.e. that it is easily adaptable to all the following scenarios: multi-output, semisupervised, and missing-data. The "S2.FA", "S3.FA", and "MS3.FA" variants of factor analysis were compared with other algorithms (from the literature) applicable in that specific context—regression (Moore-Penrose inverse, ridge regression), semisupervised regression (label propagation [24]), and missing-data imputation (mean imputation). Results on multi-output regression were reported as well. We mark below whether the results suggest or not that a specific algorithm should be taken into consideration as a candidate when one compares multiple algorithms on a dataset in order to find the best model; "✓" means "positive suggestion, i.e. you should take it into consideration" and "✗" means "negative suggestion, i.e. you should not take it into consideration":

[1] "$D >> n$" means "the number of input columns is greater than the number of rows"

- for regression: S2.FA `X`;
- for semisupervised regression: S3.FA `7`;
- for missing-data imputation: MS3.FA `X`.

The next chapter creates the non-linear version of GMM (Gaussian Mixture Model) [13, sec. 11.2.1]/k-means [11] by adding neural networks.

Chapter 4

Fixed Representatives and Variable Features: From Regularized Deep Clustering to Classic Clustering

In this chapter, we created a new clustering algorithm. We started from an approach involving autoencoder-based deep clustering and ended with a simpler approach involving vanilla-deep-neural-network-based deep clustering (although the neural network architecture may be considered shallow). In order to obtain the final model, we added regularization to the initial model by dropping components, by fixing the cluster representatives. and by increasing the variance of the features in the new/mapped space. In fact, compared to state-of-the-art deep clustering models, this is the **advantage** of our approach, i.e. the regularization schemes we proposed are simple, new, and capable of avoiding trivial/unsatisfactory solutions. Since our final model has many hyperparameters, we have chosen four configurations (km1, km50, gmm1, cmm1) with which we have experimented on four datasets. The results suggest that km50 and gmm1:

- are the best among these four configurations,
- are poorer than DKM (deep k-means) [3],
- have their clustering metrics close to the values obtained by k-means [11] and EM/GMM[1] [15, sec. 8.7.3] and surpass them in some cases.

The next chapter takes the autoencoder [15, sec. 20.3] model, views it as a discriminative model, and uses probabilistic distributions other than the normal one.

[1] EM—Expectation Maximization; GMM—Gaussian Mixture Model

Chapter 5

Vanilla Probabilistic Autoencoder

We proposed in this chapter a new version of the autoencoder [15, sec. 20.3]. It involves a probabilistic perspective on the autoencoder:

$$X_{\text{reconstructed}}|x \sim \text{Distribution}(\text{parameters} = \text{neural\_network}(x)).$$

If "Distribution" is the normal distribution [13, sec. 2.4.1] with constant variance or with a constant covariance matrix (its value is I—the identity matrix), then maximizing the likelihood is the same as minimizing the MSE loss. If "Distribution" is the categorical distribution [13, sec. 2.3.2], then maximizing the likelihood is the same as minimizing the cross-entropy (CH) loss [22, 23]. We explored in this chapter multiple distributions: other normal distributions, t-distributions, MAF (masked autoregressive flow) [18], matrix-based distributions, and mixtures [13, sec. 11.2]. The CNN (convolutional neural network [15, ch. 14]) case does not provide suitable visual results. Compared to the raw autoencoder, our model has the following **advantages** (for the MLP[1] case):

- our 10-component mixture models can be considered for augmenting the dataset;
- a mixture of raw autoencoders obtains better visual results than just a single raw autoencoder;
- for some distributions, the reconstructed image is returned along with a variance image which is a visualization from the scale/covariance matrices and which can be used in interpreting if a reconstructed pixel is reliable.

Two last takeaways would be the following:

[1] MLP—multilayer perceptron [15, sec. 13.2]

- if "Distribution" has a covariance/scale matrix that is neither constant nor diagonal, then the model requires more parameters, which translates into requirements for more memory and long running time;
- the 10-component normal mixtures produce decent visual results even with constant scale/covariance matrices.

The next chapter presents a non-linear version of GMM (Gaussian Mixture Model) [13, sec. 11.2.1] by creating a mixture of NFs (normalizing flows [2, lecture 7]).

Chapter 6

Mixtures of Normalizing Flows

In this chapter, we proposed a new model by creating a mixture of normalizing flows[1]. The fitting algorithm involves the maximization of the likelihood. Clustering has been the machine learning problem investigated in the experiments. We used bidimensional and image datasets and just one type of normalizing flow (the "MAF"[2] model [18]). The results were both visual and numeric. From the bidimensional datasets, the bottom line is that our model returns more flexible probability distributions than a simple mixture of normal distributions, which is an **advantage** of our model. From the image datasets, our model surpasses or resembles EM/GMM[3] [15, sec. 8.7.3] in some situations, but it is not the best when it is compared to both k-means [11] and EM/GMM. Compared to the variational mixture of normalizing flows [19], we may say we obtain similar results on the MNIST5 dataset, but our **advantage** is that the learning phase does not involve complex techniques, i.e. variational inference, but simple ones, i.e. direct log-likelihood maximization.

The next chapter concludes the thesis.

[1] NFs—normalizing flows [2, lecture 7]
[2] MAF—Masked Autoregressive Flow
[3] EM—Expectation Maximization; GMM—Gaussian Mixture Model

Chapter 7

Conclusions and Future Work

7.1 Conclusions

As we stated in the introductory chapters, we will review here some diagrams to highlight our contributions: Figures 1.1, 1.2, 1.3, and 1.4 now become Figures 7.1, 7.2, 7.3, and 7.4 respectively. The legend in Figure 7.1 tells us that the red color stands for a model/task/relationship chosen by us and that the blue color stands for a new model that we created. There are five blue nodes which correspond to our five contributions, and for each we dedicated a chapter.

Figure 7.2 illustrates that we chose "Density estimation" twice as the original probabilistic task: firstly, in order to create **a semantic image inpainting model** (chapter 2) and, secondly, to create **a mixture of normalizing flows** (chapter 6).

When it comes to our meta-algorithm for semantic image inpainting, the normal distribution gives better visual results than the PixelCNN++ (an autoregressive model) distribution and than PatchMatch (a state-of-the-art model in non-semantic image inpainting). Compared to Partial Convolution (a state-of-the-art model in semantic image inpainting), our inpainting via a normal distribution is faster at learning but returns poorer visual results. An advantage of our model is that the training phase is straightforward: just maximize the likelihood of the data. Moreover, since the training time is shorter than the time required for Partial Convolution, one can use our approach in order to test if the inpainting task is learnable on a specific dataset. Furthermore, our model is generic since it is a schema whose (hyper)parameter, given by the parametric probabilistic distribution, dictates almost entirely its behavior.

When it comes to our mixture of normalizing flows, on bidimensional datasets we can see that the shapes of densities expand beyond some ellipses, i.e. beyond the capabilities

of a mixture of normal distributions, which is a sign of better expressiveness than the one provided by a classic Gaussian mixture model. When our algorithm is applied on images, this flexibility is not reflected in the computed metrics which are (just sometimes) similar to those obtained by EM/GMM. However, compared to the variational mixture of normalizing flows (a candidate model from the literature), we obtain arguably similar results on the MNIST5 dataset, but our training approach has the advantage of being simpler since we directly maximized the log-likelihood of the data without using variational inference.

Figure 7.3 illustrates that we chose "Autoencoder" as the original probabilistic model in order to create the **vanilla probabilistic autoencoder** (chapter 5). Compared to the raw autoencoder, our model has some advantages. It is able to return not only the reconstructed input but also the variances associated to each reconstructed dimension if a distribution with a covariance/scale matrix is used. Then, the model can be employed as a data augmentation technique if a mixture of distributions is used. Moreover, a mixture of raw autoencoders obtains better visual results than just a single raw autoencoder.

Figure 7.4 illustrates that we chose "FA/PPCA" with the "supervised:regression|unsupervised:dim.red." relationship (chapter 3) and "GMM" with the "linear|non-linear" relationship (chapter 4 and chapter 6). Firstly, "**S2.FA**" is created as the supervised (regression) counterpart of "FA/PPCA"; moreover, it can also be interpreted as the counterpart of "Naive/Joint Bayes/LDA/QDA" from the classification/regression point of view and as the counterpart of "GMM" from the clustering/regression point of view. We proved that "S2.UncFA" (which is close to "S2.FA") and linear regression were mathematically equivalent from the prediction function perspective. We used "S2.FA" as a new solution to the problem of adapting linear regression to the case when the number of input columns is greater than the number of rows. Compared to other solutions (Moore-Penrose inverse and L2 regularization), our approach has the advantage that it is easily adaptable to all the following scenarios: multi-output, semisupervised, and missing-data.

Secondly, "**Fixed representatives...**" is created as the non-linear counterpart of the Gaussian mixture model. It can also be interpreted as a clustering counterpart of "VAE" from the clustering/dimensionality reduction point of view. Here, we used neural networks to obtain the non-linear feature and started from a general deep clustering framework. The results show that there are situations where we surpass the results obtained by classic clustering algorithms. Compared to state-of-the-art deep clustering models, although we obtain poorer results, the advantage of our model is that the regularization schemes we proposed are simple, new, and capable of avoiding trivial/unsatisfactory

solutions: fix the cluster representatives and maximize the variance of the features in the mapped space.

Optionally, our **mixture of normalizing flows** (presented earlier in the context of Figure 7.2) can be interpreted as a non-linear counterpart of "GMM".

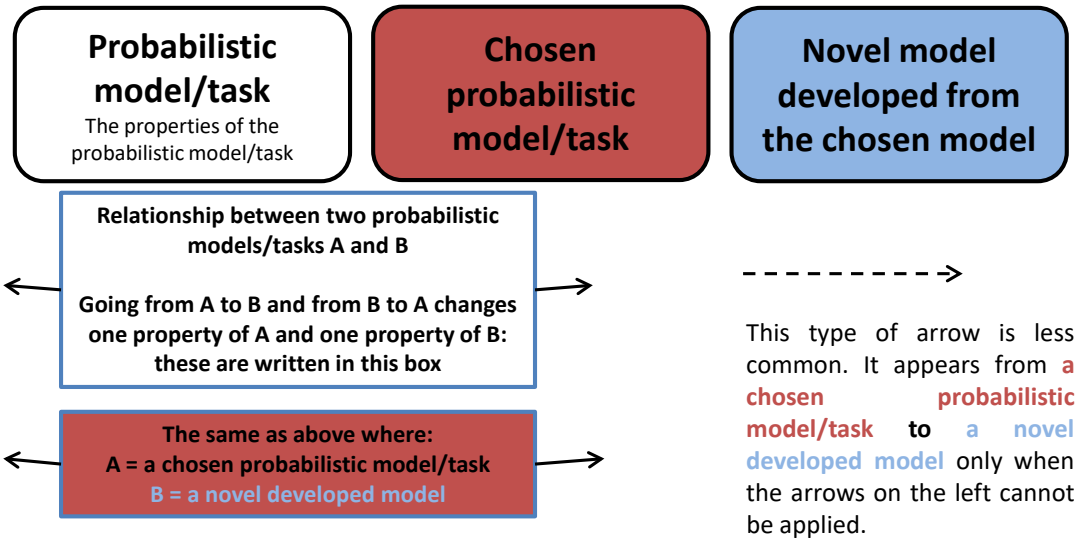

FIGURE 7.1: Interactions between probabilistic models/tasks—with our contributions: legend

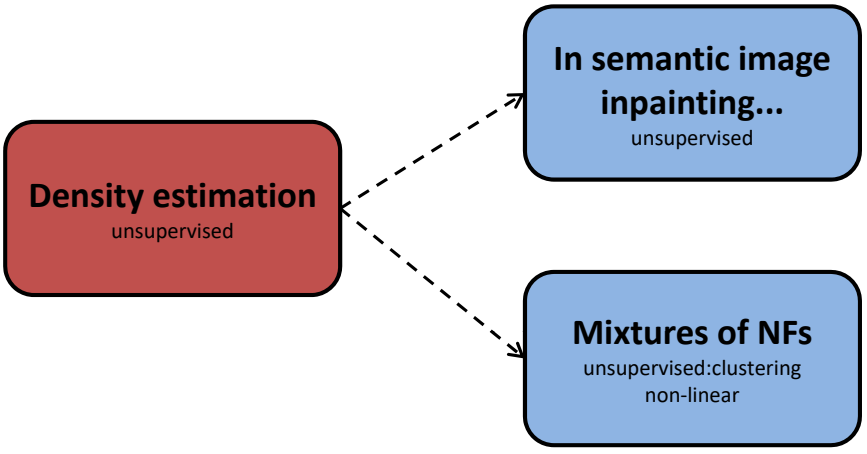

FIGURE 7.2: Interactions between probabilistic models/tasks—with our contributions; see Figure 7.1 for the legend. Abbreviations: NFs—Normalizing Flows

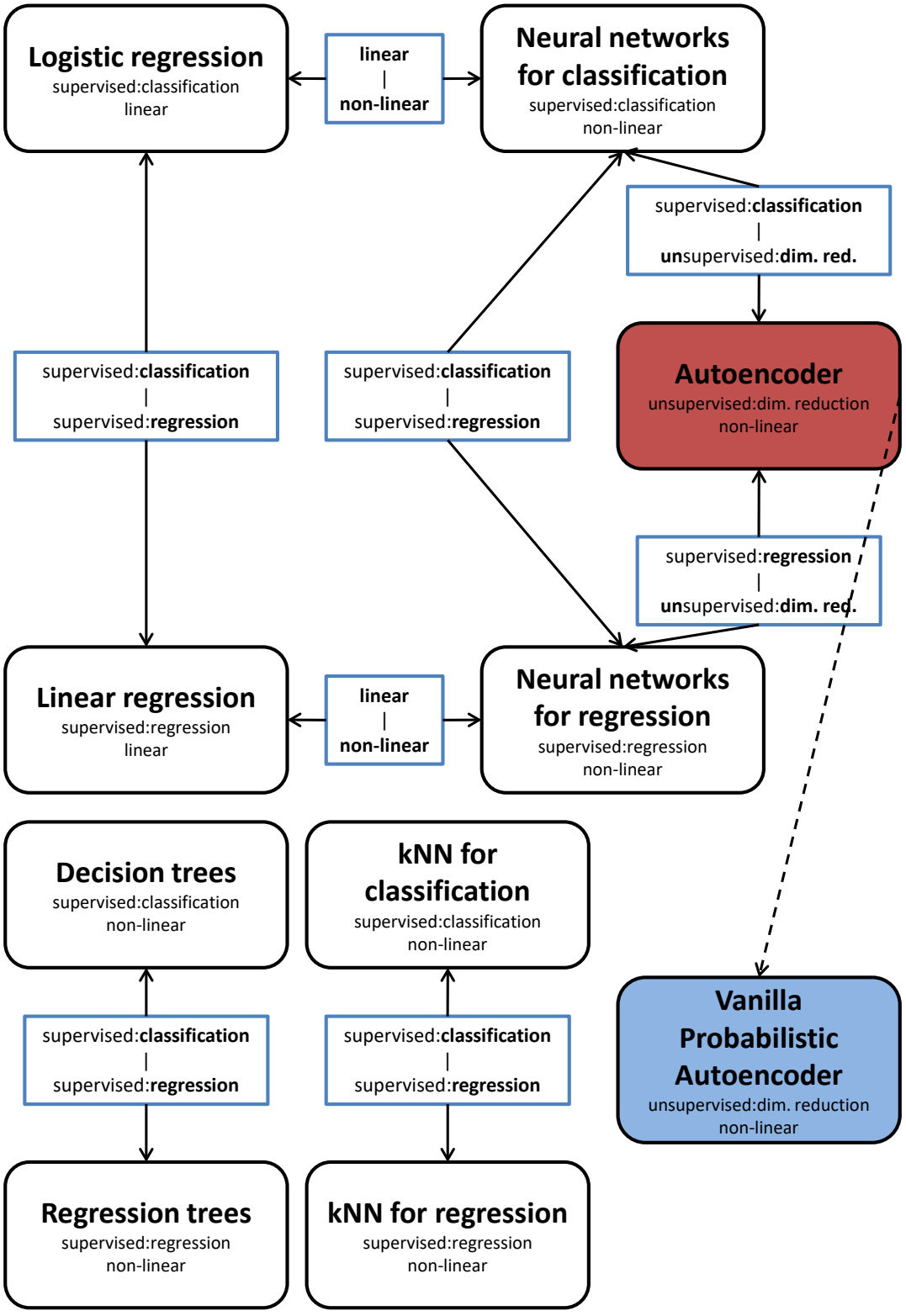

FIGURE 7.3: Interactions between the discriminative [13, sec. 8.6] models—with our contributions; see Figure 7.1 for the legend. Abbreviations: kNN—k-Nearest Neighbors

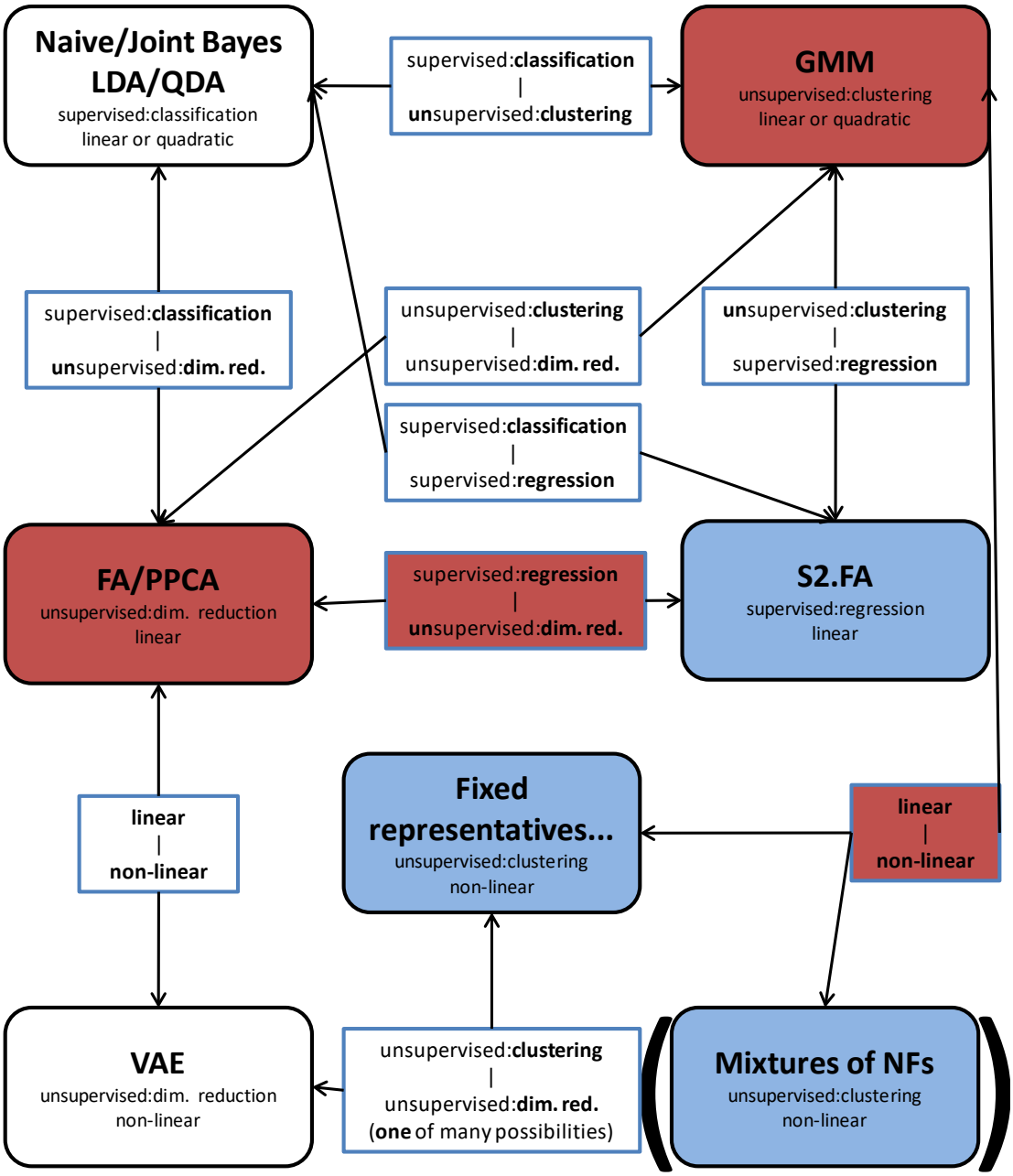

FIGURE 7.4: Interactions between the generative [13, sec. 8.6] models—with our contributions; see Figure 7.1 for the legend. Abbreviations: LDA—Linear Discriminant Analysis, QDA—Quadratic Discriminant Analysis, GMM—Gaussian Mixture Model, FA—Factor Analysis, PPCA—Probabilistic Principal Component Analysis, VAE—Variational AutoEncoder, NFs—Normalizing Flows, S2.FA—Simple-Supervised Factor Analysis

7.2 Future Work

As for future work, each chapter from the thesis contains work that can be done on that specific model. Here we present ideas that apply to the probabilistic machine learning field in general:

- In order to create new models, one can pick other starting models/tasks, not those we chose and marked with red.
- The list of distributions can be extended with other distributions, e.g. new normalizing flow models as they arise in the literature or a vector of 1-dimensional independent distributions (e.g. Gamma).
- The list of probabilistic models/tasks can be extended, e.g. with an HMM (Hidden Markov Model) [13, sec. 10.2.2], an ICA (Independent Component Analysis) [17] model, or energy-based models [14, ch. 23] which were not discussed in the thesis.
- One can add other algorithmic types in Figures 7.3 and 7.4, e.g. if you consider adding the HMM to your list of models, then a new suitable algorithmic type would be sequential/non-sequential and there would be a relationship, between GMM and HMM, which turns on/off the sequential property.
- One can use the models in different contexts, other than image inpainting and switching on/off the properties of pre-existing machine learning algorithms.
- One can go from pure MLE (Maximum Likelihood Estimation) [15, sec. 4.2] to a Bayesian approach [13, ch. 5], starting with the MAP (Maximum A Posteriori) [13, sec. 5.2] estimate.
- One can create autoregressive models that have multivariate conditional probabilities.
- One can combine deep generative models as in (after [2, lecture 16]): PixelVAE [6], autoregressive flows [7], VAE+NF [20, 9], FlowGAN [5] (losses combination: NF+GAN), adversarial autoencoder [12] (losses combination: VAE+GAN).

Bibliography

[1] Connelly Barnes et al. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28.3 (Aug. 2009).

[2] *Deep Generative Models course, Stanford University.* URL: https://deepgenerativemodels.github.io/syllabus.html (visited on 05/13/2022).

[3] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. "Deep k-means: Jointly clustering with k-means and learning representations". In: *Pattern Recognition Letters* 138 (2020), pp. 185–192.

[4] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[5] Aditya Grover, Manik Dhar, and Stefano Ermon. "Flow-GAN: Bridging implicit and prescribed learning in generative models". In: *arXiv preprint arXiv: 1705.08868* 1 (2017).

[6] Ishaan Gulrajani et al. "PixelVAE: A latent variable model for natural images". In: *arXiv preprint arXiv: 1611.05013* (2016).

[7] Chin-Wei Huang et al. "Neural autoregressive flows". In: *arXiv preprint arXiv: 1804.00779* (2018).

[8] Diederik P Kingma and Max Welling. "Auto-encoding variational Bayes". In: *arXiv preprint arXiv: 1312.6114* (2013).

[9] Durk P Kingma et al. "Improved variational inference with inverse autoregressive flow". In: *Advances in neural information processing systems*. 2016, pp. 4743–4751.

[10] Guilin Liu et al. "Image inpainting for irregular holes using partial convolutions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 85–100.

[11] Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[12] Alireza Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv: 1511.05644* (2015).

[13] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[14] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL: probml.ai.

[15] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2021. URL: http://probml.ai.

[16] Andrew Ng. *Machine Learning course, lecture notes, part X*. URL: http://cs229.stanford.edu/notes2020spring/cs229-notes9.pdf (visited on 05/13/2022).

[17] Andrew Ng. *Machine Learning course, lecture notes, Part XII, Independent Component Analysis*. URL: http://cs229.stanford.edu/notes2020spring/cs229-notes11.pdf (visited on 05/13/2022).

[18] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked autoregressive flow for density estimation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2338–2347.

[19] Guilherme GP Pires and Mário AT Figueiredo. "Variational mixture of normalizing flows". In: *arXiv preprint arXiv: 2009.00585* (2020).

[20] Danilo Jimenez Rezende and Shakir Mohamed. "Variational inference with normalizing flows". In: *arXiv preprint arXiv: 1505.05770* (2015).

[21] Tim Salimans et al. "PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications". In: *arXiv preprint arXiv: 1701.05517* (2017).

[22] *TensorFlow Losses*. URL: https://www.tensorflow.org/api_docs/python/tf/keras/losses (visited on 05/13/2022).

[23] Kilian Weinberger. *Machine Learning for Intelligent Systems course, lecture notes, lecture 10, Empirical Risk Minimization*. URL: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote10.html (visited on 05/13/2022).

[24] Zhu Xiaojin and Ghahramani Zoubin. "Learning from labeled and unlabeled data with label propagation". In: *Technical Report CMU-CALD-02–107, Carnegie Mellon University* (2002).

Bibliography of the Thesis Uncited in This Summary

1. Martín Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv: 1603.04467* (2016).
2. Manya V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems". In: *IEEE Transactions on Image Processing* 20.3 (2010), pp. 681–695.
3. Elie Aljalbout et al. "Clustering with deep learning: Taxonomy and new methods". In: *arXiv preprint arXiv: 1801.07648* (2018).
4. Coloma Ballester et al. "Filling-in by joint interpolation of vector fields and gray levels". In: *IEEE transactions on image processing* 10.8 (2001), pp. 1200–1211.
5. Ziv Bar-Joseph and Eric Xing. *Machine Learning*. course, midterm exam, pr. 7. fall 2015. URL: https://bit.ly/3N6jte3 (visited on 05/13/2022).
6. David Barber and Christopher M Bishop. "Ensemble learning in Bayesian neural networks". In: *Nato ASI Series F Computer and Systems Sciences* 168 (1998), pp. 215–238.
7. Connelly Barnes et al. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28.3 (Aug. 2009).
8. *Bayesian Methods in Machine Learning course, Coursera*. URL: https://www.coursera.org/learn/bayesian-methods-in-machine-learning (visited on 07/13/2020).
9. Marcelo Bertalmio et al. *Image inpainting in Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000.
10. Christopher M Bishop. *Mixture density networks*. Tech. rep. 1994.
11. Christopher M Bishop. *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006.
12. Charles Blundell et al. "Weight uncertainty in neural networks". In: *arXiv preprint arXiv: 1505.05424* (2015).
13. Vanessa Böhm and Uroš Seljak. "Probabilistic auto-encoder". In: *arXiv preprint arXiv: 2006.05479* (2020).
14. Sebastian Ciobanu. "Fixed Representatives and Variable Features: From Regularized Deep Clustering to Classic Clustering". In: *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2021, pp. 1–6. DOI: 10.1109/INISTA52262.2021.9548500.

15. Sebastian Ciobanu. "Mixtures of Normalizing Flows". In: *Proceedings of ISCA 34th International Conference on Computer Applications in Industry and Engineering*. Ed. by Yan Shi et al. Vol. 79. EPiC Series in Computing. EasyChair, 2021, pp. 82–90. DOI: 10.29007/nq4f. URL: https://easychair.org/publications/paper/Scnv.

16. Sebastian Ciobanu. "Vanilla Probabilistic Autoencoder". In: *Proceedings of ISCA 34th International Conference on Computer Applications in Industry and Engineering*. Ed. by Yan Shi et al. Vol. 79. EPiC Series in Computing. EasyChair, 2021, pp. 71–81. DOI: 10.29007/s1mx. URL: https://easychair.org/publications/paper/R2l1.

17. Sebastian Ciobanu and Liviu Ciortuz. "A Factor Analysis Perspective on Linear Regression in the 'More Predictors than Samples' Case". In: *Entropy* 23.8 (2021). ISSN: 1099-4300. DOI: 10.3390/e23081012. URL: https://www.mdpi.com/1099-4300/23/8/1012.

18. Sebastian Ciobanu and Liviu Ciortuz. "Semantic Image Inpainting via Maximum Likelihood". In: *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. 2020, pp. 153–160. DOI: 10.1109/SYNASC51798.2020.00034.

19. Liviu Ciortuz, Alina Munteanu, and Elena Bădărău. *Machine Learning exercise book (in Romanian)*. Alexandru Ioan Cuza University of Iași, Romania, 2019. URL: https://bit.ly/320ZuIk (visited on 05/13/2022).

20. Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

21. *Deep Generative Models course, Stanford University*. URL: https://deepgenerativemodels.github.io/syllabus.html (visited on 05/13/2022).

22. Olivier Delalleau, Aaron Courville, and Yoshua Bengio. "Efficient EM training of Gaussian mixtures with missing data". In: *arXiv preprint arXiv: 1209.0521* (2012).

23. Marco Di Zio, Ugo Guarnera, and Orietta Luzi. "Imputation through finite Gaussian mixture models". In: *Computational Statistics & Data Analysis* 51.11 (2007), pp. 5305–5316.

24. Joshua V Dillon et al. "TensorFlow distributions". In: *arXiv preprint arXiv: 1711.10604* (2017).

25. Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear independent components estimation". In: *arXiv preprint arXiv: 1410.8516* (2014).

26. Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. “Density estimation using Real NVP”. In: *arXiv preprint arXiv: 1605.08803* (2016).

27. Laurent Dinh et al. “A RAD approach to deep mixture models”. In: *arXiv preprint arXiv: 1903.07714* (2019).

28. Jarek Duda. “Gaussian AutoEncoder”. In: *arXiv preprint arXiv: 1811.04751* (2018).

29. Emilien Dupont and Suhas Suresha. “Probabilistic semantic inpainting with pixel constrained CNNs”. In: *arXiv preprint arXiv: 1810.03728* (2018).

30. Pierre Dutilleul. “The MLE algorithm for the matrix normal distribution”. In: *Journal of statistical computation and simulation* 64.2 (1999), pp. 105–123.

31. Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

32. Pablo A Estévez et al. “Normalized mutual information feature selection”. In: *IEEE Transactions on neural networks* 20.2 (2009), pp. 189–201.

33. Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. “Deep k-means: Jointly clustering with k-means and learning representations”. In: *Pattern Recognition Letters* 138 (2020), pp. 185–192.

34. *Generative adversarial networks (within [21])*. URL: https://deepgenerativemodels.github.io/notes/gan/ (visited on 05/13/2022).

35. Mathieu Germain et al. “MADE: Masked autoencoder for distribution estimation”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 881–889.

36. Samuel Gershman and Noah Goodman. “Amortized inference in probabilistic reasoning”. In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 36. 36. 2014.

37. Zoubin Ghahramani and Thomas L Griffiths. “Infinite latent feature models and the Indian buffet process”. In: *Advances in neural information processing systems*. 2006, pp. 475–482.

38. Zoubin Ghahramani, Geoffrey E Hinton, et al. *The EM algorithm for mixtures of factor analyzers*. Tech. rep. http://mlg.eng.cam.ac.uk/zoubin/papers/tr-96-1.pdf. CRG-TR-96-1, University of Toronto, 1996.

39. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

40. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

41. Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

42. Aditya Grover, Manik Dhar, and Stefano Ermon. "Flow-GAN: Bridging implicit and prescribed learning in generative models". In: *arXiv preprint arXiv: 1705.08868* 1 (2017).

43. Ishaan Gulrajani et al. "PixelVAE: A latent variable model for natural images". In: *arXiv preprint arXiv: 1611.05013* (2016).

44. Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. Vol. 104. Portions availible online at https://bit.ly/3xGjMGC (visited on 05/13/2022). CRC Press, 2018.

45. John R Hershey et al. "Deep clustering: Discriminative embeddings for segmentation and separation". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 31–35.

46. Martin Heusel et al. "GANs trained by a two time-scale update rule converge to a local Nash equilibrium". In: *Advances in Neural Information Processing Systems* 30 (2017).

47. Chih-Chung Hsu and Chia-Wen Lin. "CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data". In: *IEEE Transactions on Multimedia* 20.2 (2017), pp. 421–429.

48. Chin-Wei Huang et al. "Neural autoregressive flows". In: *arXiv preprint arXiv: 1804.00779* (2018).

49. Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218.

50. J. J. Hull. "A database for handwritten text recognition research". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.5 (1994), pp. 550–554. DOI: 10.1109/34.291440.

51. Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion". In: *ACM Transactions on Graphics (ToG)* 36.4 (2017), pp. 1–14.

52. Zhuxi Jiang et al. “Variational deep embedding: An unsupervised and generative approach to clustering”. In: *arXiv preprint arXiv: 1611.05148* (2016).

53. Tero Karras et al. “Progressive Growing of GANs for Improved Quality, Stability, and Variation”. In: *CoRR* abs/1710.10196 (2017). arXiv: 1710.10196. URL: http://arxiv.org/abs/1710.10196.

54. Kian Katanforoosh. *Deep Learning course, lecture Attacking Networks with Adversarial Examples - Generative Adversarial Networks*. URL: http://cs230.stanford.edu/files/lecture-notes-4.pdf (visited on 05/13/2022).

55. Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv: 1412.6980* (2014).

56. Diederik P Kingma and Max Welling. “Auto-encoding variational Bayes”. In: *arXiv preprint arXiv: 1312.6114* (2013).

57. Durk P Kingma and Prafulla Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: *Advances in neural information processing systems*. 2018, pp. 10215–10224.

58. Durk P Kingma et al. “Improved variational inference with inverse autoregressive flow”. In: *Advances in neural information processing systems*. 2016, pp. 4743–4751.

59. *Knowledge distillation*. URL: https://en.wikipedia.org/wiki/Knowledge_distillation (visited on 05/13/2022).

60. Teuvo Kohonen. “The self-organizing map”. In: *Proceedings of the IEEE* 78.9 (1990), pp. 1464–1480.

61. Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

62. Tomasz Kuśmierczyk and Arto Klami. “Reliable Categorical Variational Inference with Mixture of Discrete Normalizing Flows”. In: *arXiv preprint arXiv: 2006.15568* (2020).

63. Hugo Larochelle and Iain Murray. “The neural autoregressive distribution estimator”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 29–37.

64. Neil D Lawrence. “Gaussian process latent variable models for visualisation of high dimensional data”. In: *Advances in neural information processing systems*. https://papers.nips.cc/paper/2540-gaussian-process-latent-variable-models-for-visualisation-of-high-dimensional-data.pdf. 2004, pp. 329–336.

65. *Lazy learning*. URL: https://en.wikipedia.org/wiki/Lazy_learning (visited on 05/13/2022).

66. Yann LeCun, Corinna Cortes, and CJ Burges. *MNIST handwritten digit database*. 2010.

67. Hung-Shin Lee et al. "Discriminative autoencoders for speaker verification". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 5375–5379.

68. Jinghua Li et al. "Matrix-variate variational auto-encoder with applications to image process". In: *Journal of Visual Communication and Image Representation* 67 (2020), p. 102750.

69. Guilin Liu et al. "Image inpainting for irregular holes using partial convolutions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 85–100.

70. Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

71. Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. ICML*. Vol. 30. 1. Citeseer. 2013, p. 3.

72. Alireza Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv: 1511.05644* (2015).

73. Leland McInnes, John Healy, and James Melville. "UMAP: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv: 1802.03426* (2018).

74. Erxue Min et al. "A survey of clustering with deep learning: From the perspective of network architecture". In: *IEEE Access* 6 (2018), pp. 39501–39514.

75. Tom Mitchell. "Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression". In: *Machine Learning*. draft chapter intended for inclusion in [76]. McGraw-Hill Science/Engineering/Math; (March 1, 1997), 2017. URL: https://bit.ly/39Ueb4o (visited on 05/13/2022).

76. Tom Mitchell. *Machine learning*. McGraw-Hill New York, 1997.

77. Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

78. Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL: probml.ai.

79. Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2021. URL: http://probml.ai.

80. Radford M Neal. "Connectionist learning of belief networks". In: *Artificial intelligence* 56.1 (1992), pp. 71–113.

81. Andrew Ng. *Machine Learning course, lecture notes, Generalized Linear Models*. URL: https://cs229.stanford.edu/notes2020spring/cs229-notes1.pdf (visited on 05/13/2022).

82. Andrew Ng. *Machine Learning course, lecture notes, Mixtures of Gaussians and the EM algorithm*. URL: http://cs229.stanford.edu/notes2020spring/cs229-notes7b.pdf (visited on 05/13/2022).

83. Andrew Ng. *Machine Learning course, lecture notes, part X*. URL: http://cs229.stanford.edu/notes2020spring/cs229-notes9.pdf (visited on 05/13/2022).

84. Andrew Ng. *Machine Learning course, lecture notes, part XI*. URL: http://cs229.stanford.edu/notes2020spring/cs229-notes10.pdf (visited on 05/13/2022).

85. Andrew Ng. *Machine Learning course, lecture notes, Part XII, Independent Component Analysis*. URL: http://cs229.stanford.edu/notes2020spring/cs229-notes11.pdf (visited on 05/13/2022).

86. *Normalizing flow models (within [21])*. URL: https://deepgenerativemodels.github.io/notes/flow/ (visited on 05/13/2022).

87. *Novelty and Outlier Detection*. URL: https://scikit-learn.org/stable/modules/outlier_detection.html (visited on 05/13/2022).

88. Avital Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms". In: *arXiv preprint arXiv: 1804.09170* (2018).

89. Aaron Oord et al. "Parallel WaveNet: Fast high-fidelity speech synthesis". In: *International conference on machine learning*. 2018, pp. 3918–3926.

90. Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel Recurrent Neural Networks". In: *arXiv preprint arXiv: 1601.06759* (2016).

91. Aaron van den Oord et al. "WaveNet: A generative model for raw audio". In: *arXiv preprint arXiv: 1609.03499* (2016).

92. George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked autoregressive flow for density estimation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2338–2347.

93. Deepak Pathak et al. "Context encoders: Feature learning by inpainting". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2536–2544.

94. Angshuman Paul, Angshul Majumdar, and Dipti Prasad Mukherjee. "Discriminative autoencoder". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 3049–3053.

95. KB Petersen, MS Pedersen, et al. "The Matrix Cookbook, vol. 7". In: *Technical University of Denmark* 15 (2008).

96. Guilherme GP Pires and Mário AT Figueiredo. "Variational mixture of normalizing flows". In: *arXiv preprint arXiv: 2009.00585* (2020).

97. Janis Postels et al. "Go with the Flows: Mixtures of Normalizing Flows for Point Cloud Generation and Reconstruction". In: *arXiv preprint arXiv: 2106.03135* (2021).

98. *Probabilistic Graphical Models course, Stanford University, Sampling methods*. URL: https://ermongroup.github.io/cs228-notes/inference/sampling/ (visited on 05/13/2022).

99. Rajesh Ranganath, Sean Gerrish, and David Blei. "Black box variational inference". In: *Artificial Intelligence and Statistics*. 2014, pp. 814–822.

100. Sebastien Razakarivony and Frédéric Jurie. "Discriminative autoencoders for small targets detection". In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 3528–3533.

101. *Regression with Probabilistic Layers in TensorFlow Probability*. URL: https://blog.tensorflow.org/2019/03/regression-with-probabilistic-layers-in.html (visited on 05/13/2022).

102. Danilo Jimenez Rezende and Shakir Mohamed. "Variational inference with normalizing flows". In: *arXiv preprint arXiv: 1505.05770* (2015).

103. Tim Salimans et al. "PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications". In: *arXiv preprint arXiv: 1701.05517* (2017).

104. Jianhong Shen and Tony F Chan. "Mathematical models for local nontexture inpaintings". In: *SIAM Journal on Applied Mathematics* 62.3 (2002), pp. 1019–1043.

105. Aarti Singh. *Machine Learning course, homework 4, pr 1.1*. in [19, pag. 528]. CMU, fall 2010.

106. Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models". In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 3483–3491.

107. Yuhang Song et al. "Image inpainting using multi-scale feature image translation". In: *arXiv preprint arXiv: 1711.08590* 2 (2017), p. 1.

108. Jost Tobias Springenberg. "Unsupervised and semi-supervised learning with categorical generative adversarial networks". In: *arXiv preprint arXiv: 1511.06390* (2015).

109. E Spyromitros-Xioufis et al. "Drawing parallels between multi-label classification and multi-target regression". In: *arXiv preprint arXiv: 1211.6581 v2* (2014).

110. Alexandru Telea. "An image inpainting technique based on the fast marching method". In: *Journal of graphics tools* 9.1 (2004), pp. 23–34.

111. *TensorFlow Losses*. URL: https://www.tensorflow.org/api_docs/python/tf/keras/losses (visited on 05/13/2022).

112. Lucas Theis, Aäron van den Oord, and Matthias Bethge. "A note on the evaluation of generative models". In: *arXiv preprint arXiv: 1511.01844* (2015).

113. Louis C Tiao. "Building Probability Distributions with the TensorFlow Probability Bijector API". In: *tiao.io* (2018). URL: https://tiao.io/post/building-probability-distributions-with-tensorflow-probability-bijector-api/.

114. Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999). https://bit.ly/2PCxoRr, pp. 611–622.

115. Benigno Uria, Iain Murray, and Hugo Larochelle. "RNADE: The real-valued neural autoregressive density-estimator". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2175–2183.

116. Aaron Van den Oord et al. "Conditional image generation with PixelCNN decoders". In: *Advances in neural information processing systems*. 2016, pp. 4790–4798.

117. Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.11 (2008).

118. *Variational autoencoders (within [21])*. URL: https://deepgenerativemodels.github.io/notes/vae/ (visited on 05/13/2022).

119. Kert Viele and Barbara Tong. "Modeling with mixtures of linear regressions". In: *Statistics and Computing* 12.4 (2002), pp. 315–330.

120. Cinzia Viroli. "Finite mixtures of matrix normal distributions for classifying three-way data". In: *Statistics and Computing* 21.4 (2011), pp. 511–522.

121. Cinzia Viroli and Geoffrey J McLachlan. "Deep Gaussian mixture models". In: *Statistics and Computing* 29.1 (2019), pp. 43–51.

122. Junxiang Wang. *SSL: Semi-Supervised Learning*. R package version 0.1; https://CRAN.R-project.org/package=SSL. 2016.

123. Kilian Weinberger. *Machine Learning for Intelligent Systems course, lecture notes, lecture 10, Empirical Risk Minimization*. URL: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote10.html (visited on 05/13/2022).

124. Kilian Weinberger. *Machine Learning for Intelligent Systems course, lecture notes, lecture 15, Gaussian Processes*. URL: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html (visited on 05/13/2022).

125. Kilian Weinberger. *Machine Learning for Intelligent Systems course, lecture notes, lecture 17, Decision Trees*. URL: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote17.html (visited on 05/13/2022).

126. Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

127. Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv: 1708.07747* (2017).

128. Zhu Xiaojin and Ghahramani Zoubin. "Learning from labeled and unlabeled data with label propagation". In: *Technical Report CMU-CALD-02–107, Carnegie Mellon University* (2002).

129. Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis". In: *International conference on machine learning*. PMLR. 2016, pp. 478–487.

130. Bo Yang et al. "Towards k-means-friendly spaces: Simultaneous deep learning and clustering". In: *international conference on machine learning*. PMLR. 2017, pp. 3861–3870.

131. Jianwei Yang, Devi Parikh, and Dhruv Batra. "Joint unsupervised learning of deep representations and image clusters". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5147–5156.

132. Raymond A Yeh et al. "Semantic image inpainting with deep generative models". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5485–5493.

133. Jiahui Yu et al. "Generative image inpainting with contextual attention". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5505–5514.

134. Matei Zaharia et al. "Apache Spark: A Unified Engine for Big Data Processing". In: *Commun. ACM* 59.11 (Oct. 2016). http://doi.acm.org/10.1145/2934664, 56–65. ISSN: 0001-0782. DOI: 10.1145/2934664.

135. Xiuling Zhou et al. “Mixture of matrix normal distributions for color image inpainting”. In: *International Conference on Neural Information Processing*. Springer. 2017, pp. 95–104.

136. Andrey Ziyatdinov et al. “Bioinspired early detection through gas flow modulation in chemo-sensory systems”. In: *Sensors and Actuators B: Chemical* 206 (2015), pp. 538–547.

137. Bo Zong et al. “Deep autoencoding Gaussian mixture model for unsupervised anomaly detection”. In: *International Conference on Learning Representations*. 2018.