A Study of Semi-Supervised Generative Ensembles

Manuela Zanda, Gavin Brown

School of Computer Science, University of Manchester, UK
{zandam,gbrown}@cs.man.ac.uk

Abstract. Machine Learning can be divided into two schools of thought: generative model learning and discriminative model learning. While the MCS community has been focused mainly on the latter, our paper is concerned with questions that arise from ensembles of generative models. Generative models provide us with neat ways of thinking about two interesting learning issues: model selection and semi-supervised learning. Preliminary results show that for semi-supervised low-variance generative models, traditional MCS techniques like Bagging and Random Subspace Method (RSM) do not outperform the single classifier approach. However, RSM introduces diversity between base classifiers. This starting point suggests that diversity between base components has to lie within the *structure* of the base classifier, and not in the dataset, and it highlights the need for novel generative ensemble learning techniques.

1 Introduction

In the past few years, the MCS community has mainly focused on *supervised* problems, that is, learning scenarios where classifiers are trained on *labelled* examples. Nevertheless, many real applications are nowadays characterised by two *contrasting factors*, namely *the need for large quantities of labelled data* to design supervised classifiers with high accuracy, and *the difficulty and cost of collecting such data*.

A possible answer to this accuracy/labelling dilemma is to consider *semi-supervised* algorithms, that is, techniques which are able to learn from a small amount of labelled data together with a large amount of unlabelled data [**?**]. The majority of the work done so far has been concerned with ensembles of semi-supervised *discriminative* models, where some external procedure is responsible for labelling the unlabelled data before base classifiers can learn from them [**?**,**?**,**?**,**?**,**?**].

Generative models are algorithms that can learn from labelled and unlabelled data [**?**]. There are very few examples of semi-supervised generative ensembles [**?**,**?**], and so far there is still no common understanding of the way unlabelled data affects ensembles of generative models.

This paper is an attempt to further investigate semi-supervised ensembles of generative models. Generative and discriminative approaches are two ways of solving the same problem (Sec. 2). A comparison of ensemble techniques shows

that the way they make use of unlabelled data is different (Sec. 3); moreover they provide different levels of understanding in terms of model mismatch (Sec. 4). As generative models have not been explored in the MCS community, we present a preliminary experimental analysis (Sec.5). Our results show that for semi-supervised low-variance generative models, diversity between base classifiers has to be structurally imposed.

2 Discriminative or Generative Models?

The generative/discriminative dilemma seems to divide Machine Learning into two separate communities.

In a statistical approach a classification problem is modelled by the joint distribution $\mathrm{p}(X, Y)$, where X and Y denote the data and the class random variables, respectively. Because we want to solve a classification problem, our goal is to find the optimal estimate of the class posterior $\mathrm{p}(Y|X)$. This can be determined via Bayes' rule:

$$\mathrm{p}(Y|X) = \frac{\mathrm{p}(X|Y)\mathrm{p}(Y)}{\mathrm{p}(X)}.$$

Discriminative classifiers directly model the class posterior distribution $\mathrm{p}(Y|X)$. In practical terms, this corresponds to modelling our problem as decision regions between classes. Typical example of discriminative models are neural networks, where we try to learn decision boundaries by minimising some error function. In a generative approach we make explicit assumptions about the form of the class conditional distributions $\mathrm{p}(X|Y)$ and class priors $\mathrm{p}(Y)$. Therefore, a generative model in practice models the data distribution rather than the decision regions. An example of a generative model is a Naïve Bayes network, which is based on the assumption that all the features are conditionally independent given the class:

$$\mathrm{p}(\boldsymbol{X}|Y) = \prod_{f=1}^{\mathrm{D}} \mathrm{p}(X_f|Y), \tag{1}$$

as depicted in Fig. 1. If all features are discrete, we can estimate Eq. (1) by frequency counts.

In terms of *problem applicability*, generative models can naturally incorporate unlabelled data because they learn the way data is distributed. On the other hand, discriminative models have no knowledge at all about the data distributions and therefore, their major drawback is that they cannot naturally handle unlabelled data. This implies that in semi-supervised problems there must exist an external mechanism that labels the unlabelled data before a discriminative classifier can incorporate them into the learning process, as in Co-training [**?**] and Tri-Training [**?**].

In terms of *performance*, when only few labelled data are available, there is strong evidence [**?**] that generative models outperform discriminative models.

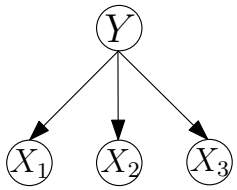


Fig. 1. A Naïve Bayes network. Each arc represent a inter-variable dependency, while the absence of an arc is an indication of independence between random variables.

Moreover, while discriminative models can achieve better performance, generative models have a faster speed of convergence. The main reason discriminative models are usually *preferred* to generative models is that the latter rely on strong assumptions. The choice of the *right* assumption in generative models is *crucial*, for studies have shown that when there is *model mismatch*, unlabelled data can degrade classification performance [**?**].

3 Semi-Supervised Learning

3.1 Discriminative Ensembles for Semi-Supervised Problems

The MCS community has traditionally focused on *discriminative base classifiers*, and most of the work done so far about semi-supervised learning has been concerned with techniques that let discriminative classifiers exploit unlabelled data [**?**,**?**,**?**,**?**]. As a *discriminative model cannot make use of data without labels, an external mechanism has to assign "pseudo-labels" to the unlabelled data before a classifier can effectively process them.* We now describe the main principles of how discriminative models can learn from unlabelled data. A full review of semi-supervised ensemble techniques is out of the scope of this paper; the reader might refer to [**?**,**?**] for a more extensive survey.

Decision-directed ensemble approaches [**?**] are based on the idea that a classifier can iteratively self-teach itself. Within this framework a single classifier is initially trained on the labelled data. Afterwards, the same classifier is used to classify unlabelled data and the most confident predicted patterns are selected and added along with their "pseudo label" to the labelled patterns; the process iterates until a stopping criterion is reached. In an ensemble approach the ensemble prediction could be used to assign pseudo labels to the unlabelled data. *Co-Training* [**?**] can be considered the first attempt to apply ensemble learning to semi-supervised problems. This approach is based on the assumption that the feature space can be split into two disjoint subsets called *views*, and that each one of these is sufficient for correct classification. Therefore, a single classifier is trained on each of these views. Initially, both classifiers are trained only on labelled data. Each classifier is then asked to classify a small amount of unlabelled data. The most confident predictions are added to the labelled training set of the other classifier; then the process re-iterates for a given amount of

times. The basic idea behind co-training is that whenever classifiers disagree, the mistaken one can be "taught" by the other one, for each view is sufficient to make a correct prediction. In other words, co-training is an ensemble method that enforces *agreement on unlabelled data* [**?**]. An interesting extension is given by *tri-training* [**?**], where three classifiers use majority voting to label unlabelled data. If two classifiers agree, then the unlabelled pattern is labelled accordingly.

These "pseudo labels" are used only with discriminative classifiers. Although this mechanism might succeed, it seems somewhat ad-hoc. In contrast, generative models can lead to more elegant ensemble approaches.

3.2 Generative Ensembles for Semi-Supervised Problems

The reason why we should be interested in generative models is that unlabelled data can be incorporated into their learning process without need of "pseudo labels". Once we have made our assumptions about the *form* of our joint distribution $\mathrm{p}(X, Y) = \mathrm{p}(X, Y, \theta)$, the learning process consists of finding the parameters θ that *most likely* fit our data[1].

For instance, let us consider a C class problem in a D dimensional space. In a semi-supervised problem, our data can be split into a finite set of labelled patterns $\mathcal{D}_{\mathrm{L}} = \{\boldsymbol{\mathcal{X}}_{\mathrm{L}}, \mathcal{Y}_{\mathrm{L}}\} = \{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, \mathrm{N}\}$ and a finite set of unlabelled data $\mathcal{D}_{\mathrm{U}} = \{\boldsymbol{\mathcal{X}}_{\mathrm{U}}\} = \{(\boldsymbol{x}_j) \mid j = \mathrm{N}+1, \ldots, \mathrm{M}\}$, $\mathcal{D} = \{\mathcal{D}_{\mathrm{L}}, \mathcal{D}_{\mathrm{U}}\}$. We assume that labelled and unlabelled patterns are independent and identically distributed samples drawn from the same joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$. A semi-supervised Maximum Likelihood approach seeks to find the set of parameters θ that maximize the log-likelihood $\log \mathrm{p}(\mathcal{D}|\theta) = \log \mathrm{p}(\boldsymbol{\mathcal{X}}_L, \mathcal{Y}_L, \boldsymbol{\mathcal{X}}_U|\theta)$:

$$\begin{aligned} \log\ \mathrm{p}(\boldsymbol{\mathcal{X}}_L, &\mathcal{Y}_L, \boldsymbol{\mathcal{X}}_U|\theta) = \\ &= \log\ \mathrm{p}(\boldsymbol{\mathcal{X}}_L, \mathcal{Y}_L|\theta) + \log\ \mathrm{p}(\boldsymbol{\mathcal{X}}_U|\theta) \\ &= \sum_{i=1}^{N} \log \mathrm{p}(y_i|\theta) p(x_i|y_i, \theta) + \sum_{j=N+1}^{M} \log \sum_{k=1}^{C} \mathrm{p}(y_k|\theta) p(x_j|y_k, \theta) \end{aligned} \tag{2}$$

From (2) it is easy to observe that the log-likelihood is made of two terms, the first one depending on the labelled data, and the second one depending on the unlabelled data. It follows that in a generative ensemble approach each base classifier can learn from labelled and unlabelled data, and in addition ensemble techniques could be used to improve classification accuracy.

4 Model Selection

Model selection is the process of choosing a specific class of models according to our knowledge of the problem. Whenever a generative model does not match the problem data distribution, we call this *model mismatch*.

[1] Alternatively we can use a full Bayesian Learning approach, which consists of integrating out parameters via approximation methods such as Variational Inference.

In generative models, we make assumptions about the *form* of probability distributions and about the *inter-variable dependencies* within these distributions. For instance we might assume that our data is Normally distributed, and we might also assume a Naïve Bayes approach, by making any feature conditioned on the class label statistically independent from any other, as depicted in Fig. 1. A model mismatch indicates our model does not represent the problem correctly because these independence assumption are violated in practice.

Similarly to generative models, discriminative classifiers are based on model assumptions, and therefore they are not always able to model boundaries between decision regions. A "model mismatch" in this case would correspond to selecting a linear perceptron to solve an XOR problem, or not using enough hidden nodes in our neural network.

The main difference between generative and discriminative approaches is that a mismatch is explicit for generative models, whereas it is hidden and more subtle for discriminative models: can the correspondence between the number of hidden nodes and decision boundaries be quantified in terms of model mismatch?

At a more abstract level, any learning algorithm can be thought of as a search in the space of *representable* models $\mathcal{H}$. The model mismatch problem then corresponds to asking the question: *What happens when the true model f does not belong to this search space $\mathcal{H}$?* This situation, which is depicted in Fig. 2, is known as the "representational problem" [**?**] in the MCS community.

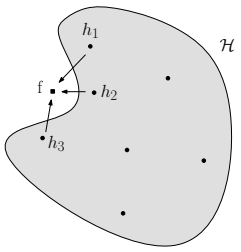


Fig. 2. An ensemble approach can deal with a representational problem by approximating the true hypothesis with a combination of wrong ones [**?**].

Discriminative ensemble learning tries to overcome this model limitation by replacing the single classifier approach with a combination of *accurate* and *different* models: if enough data are available, a combination of different models $h_1, h_2, \ldots, h_M$ in the search space can lead to a better approximation of the true model f even if this does not belong to $\mathcal{H}$ [**?**].

In theory the same ensemble principle could be applied to generative models. Moreover, we could exploit the property of generative models of *explicitly selecting* the model bias to *define* the boundaries of the hypothesis space. If the search space is then large enough, it might possible to combine *diverse* generative base models to achieve better performance than the single base classifier, and solve

not only the representational problem but also the semi-supervised problem. We now illustrate some studies we have carried out on generative model ensembles.

5 Empirical Analysis

Very few experiments have been carried out on semi-supervised ensembles of generative models [**?**,**?**]. The aim of this study is to further investigate how unlabelled data can affect ensemble learning when we combine generative base classifiers.

The base model we chose for this analysis was a Gaussian Naïve Bayes network, i.e. a Naïve Bayes with Normally distributed continuous features and identity covariance matrix $\mathcal{N}(\boldsymbol{x}|\mu, \mathrm{I})$. We adopted a MAP approach and we used a scaled conjugate gradient descent algorithm to learn our model parameters. We applied three different ensemble methods: RSM [**?**] and two different variants of Bagging [**?**]: BaggingL– which samples with replacement from labelled data, and BaggingLU– which samples with replacement from labelled and unlabelled data. We used simple mean as a combination rule. Each technique has been evaluated according to a 5 times 2 statistical test. We tested our model on three different datasets, two of which were artificial datasets, the other was a real dataset:

Ringnorm Artificial dataset that implements Breiman's ringnorm example. It is a 2 class problem with 20 features and it has 7400 patterns. This dataset is a *model mismatch* for our model, as one class has not been generated by $\mathcal{N}(\boldsymbol{x}|\mu, \mathrm{I})$.

Uniringnorm Artificial dataset that represents a 2 class problem with 20 features and it has 1000 patterns. This dataset is a *model match* for our model, being the data generated from $\mathcal{N}(0, \mathrm{I})$ or $\mathcal{N}(\mu_2, \mathrm{I})$, where $\mu_2 = (a, a, ..., a)$, with $a = \frac{2}{\sqrt{20}}$.

Feltwell We also applied our model on a real dataset by selecting 5124 patterns from Feltwell dataset. This is a 5 class problem with 15 features.

Following some experiments in [**?**], we studied how supervised and semi-supervised ensembles of generative models perform in comparison with the respective single classifier counterparts, as we increase the amount of labelled data. Our aim was to *identify any specific situation where semi-supervised ensemble learning is more beneficial than the semi-supervised single approach and the supervised ensemble*. Our results can be summarised as follow:

- *Data acts as a variance reducing factor*. Both semi-supervised ensembles and semi-supervised single classifiers show less variance than the supervised counterparts. This is unsurprising, as Naïve Bayes are low variance classifiers.
- BaggingLU
  - *Model match:* the semi-supervised ensemble performs exactly like the semi-supervised single classifier, and it always outperforms the supervised counterpart for any amount of labelled data.

    - *Model mismatch:* semi-supervised BaggingLU performs slightly worse than the semi-supervised single classifier, and in general semi-supervised learning outperforms the supervised one only when few labelled data are available (i.e. less than 40 labelled patterns).
- BaggingL
    - There is no difference between the semi-supervised ensemble and the semi-supervised single base classifier accuracy. *This implies that bagging the unlabelled data is effectively worsening the ensemble classification performance.*
- RSM
    - *Model match:* semi-supervised learning usually outperforms supervised learning for any amount of labelled data.
    - *Model mismatch:* In general semi-supervised learning outperforms supervised learning only when few labelled data are available (i.e. less than 50 labelled patterns). However, the ensemble techniques perform slightly and much (nearly 6%) worse, respectively, than the single counterparts for both supervised and semi-supervised learning.

We found similar results for Feltwell, where the semi-supervised ensemble techniques achieves almost the same accuracy as the semi-supervised respective single classifiers.

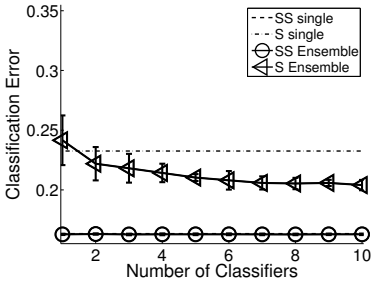
We conclude that Bagging and RSM techniques do not work well with semi-supervised low variance generative models, as data resampling or data random projections do not seem to increase the ensemble accuracy over the single classifier.

6 Discussion

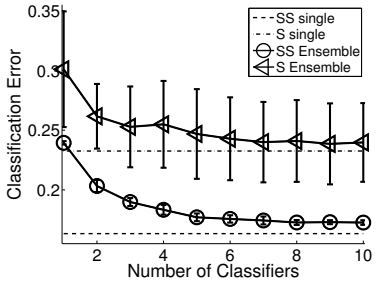
Both semi-supervised Bagging and RSM ensemble techniques seem not to improve classification accuracy over the single classifier approach – *but why?*

Let us focus on a typical semi-supervised scenario, where a large amount of unlabelled data and only few labelled data are available. We fix the amount of labelled data to be 30 patterns and we look at the ensemble behavior as we increase the number of base classifiers from 1 to 10. Results are shown in Figures 3 and 4 for a match problem and a mismatch problem, respectively. If we look at the leftmost part of both figures, we can observe the ensemble behavior of BaggingL as we increase the number of components in the ensemble. In both a model match and mismatch the semi-supervised ensemble error does not change as we increase the number of base classifiers, but at the same time this ensemble performs exactly like the semi-supervised single classifier. This is true for any amount of labelled data, and not only for 30 labelled patterns. A similar behavior has been observed for BaggingLU. It seems that when enough data are available, data resampling does not infer any kind of diversity on low variance generative base classifiers.

The rightmost part of both figures shows the ensembles created according to RSM. Whereas semi-supervised RSM fails for a model mismatch, Fig. 3 shows an

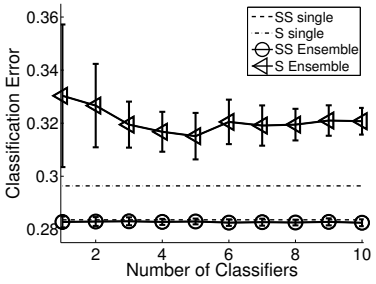


Model Match, BaggingL

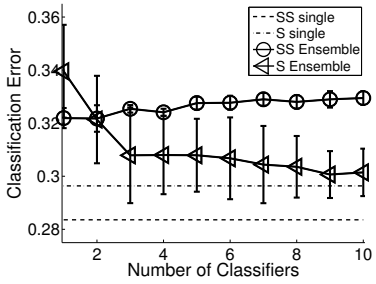


Model Match, RSM

Fig. 3. Classification error for a model match (Uniringnorm) with 30 labelled data. semi-supervised BaggingL does not create different base classifiers, whereas semi-supervised RSM does.



Model Mismatch, BaggingL



Model Mismatch, RSM

Fig. 4. Classification error for a model mismatch (Ringnorm) with 30 labelled data. semi-supervised BaggingL does not create different base classifiers, but the model mismatch cannot be model by the semi-supervised RSM.

unexpected behavior: *the semi-supervised ensemble error has very low variance and the error decreases as we increase the number of classifiers*. In other words, base classifiers are *diverse*. Increasing the amount of labelled data does not alter this behavior. However this semi-supervised ensemble does not perform better than the semi-supervised single classifier. A possible reason for this could be that each base classifier is a projection of the feature space, and therefore it is missing information about the full data distribution, whereas the single classifier can model the data completely. From a representational problem perspective, this corresponds to the space of hypotheses being so small that it is not possible to find hypotheses that, if combined, can lead to a good approximation of the true function that represents our problem.

To sum up, our analysis of Bagging and RSM techniques shows how generative models cannot be learnt like their discriminative siblings:

– Resampling techniques like Bagging do not work well with low variance generative models, as the amount of training data acts like a variance reduction factor.

– RSM techniques introduce some diversity between base classifier components, but they do not outperform the single classifier. A reason for that might be that the search space of the base classifier is not powerful enough to solve a representational problem.

Nevertheless this pattern of behavior looks promising and might indicate the need for novel generative ensemble techniques.

7 Conclusion and Future Work

How to Combine Generative Models?
While discriminative ensembles have the benefit of generating diverse base classifiers, base components require an external mechanism to make use of unlabelled data. *On the other hand, generative model ensembles naturally gain the ability of learning from unlabelled data but at the same time they lose in terms of diversity that can be generated by traditional ensemble techniques.*

Nevertheless, our results point towards the design of semi-supervised generative ensemble techniques that seek diversity in other ways than the traditional ones in MCS. It might be the case that generative model transparency can be exploited to build base classifiers that are *structurally diverse* and therefore extend the hypothesis search space. For instance, we could combine generative models that are characterised by different inter-model dependencies. An example is given by Super Parent One Dependency Estimator (SPODE) ensembles [**?**], where each base classifier feature depends not only on the class but also on another feature called superparent, as depicted in Fig. 5.

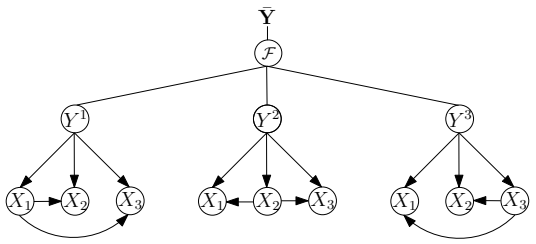


Fig. 5. An ensemble as a combination of all possible SPODEs.

Generative models are the only *systematic* way we can explore hybrid ensembles because we can actually choose the structural difference between models. This is not possible with discriminative models, where it is not clear which boundaries might arise by combining different classifiers (for instance SVMs with neural networks). Instead, with generative models not only can we systematically place models in the search space but we can also decide how big the search space is.

A natural way to quantify diversity between generative models is given by the KL divergence [**?**], a non-commutative measure of the difference between probability distributions. Multivariate Mutual Information measures this KL divergence for multidimensional probability distributions. In practical terms it gives us an indication of how correlated, i.e. how diverse, random variables are [**?**]. The focus of our future work will be to use Multivariate Mutual Information to rank and select diverse generative base classifiers from the hypothesis space. This might allow us to solve both the representational and the semi-supervised problems in an ensemble fashion.

References

1. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005)
2. Roli, F.: Semi-supervised multiple classifier systems: Background and research directions. In: 6th Int. Workshop on MCS. LNCS, Springer (2005) 1–11
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of the 11th conf. on COLT. (1998) 92–100
4. Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. IEEE Trans. on Knowledge and Data Engineering **17**(11) (2005) 1529–1541
5. Bennett, K.P., Demiriz, A., Maclin, R.: Exploiting unlabeled data in ensemble methods. In: Proc. of the 8th Int. Conf. on KDD. (2002) 289–296
6. Leskes, B.: The value of agreement, a new boosting algorithm. In Auer, P., Meir, R., eds.: Proc. of the 18th conference on COLT. LNCS, Springer (2005) 95–110
7. Miller, D., Uyar, S.: A mixture of experts classifier with learning based on both labelled and unlabelled data. Proc. of Advances in NIPS **9** (1997) 571–578
8. Buc, F., Grandvalet, Y., Ambroise, C.: Semi-supervised marginboost. Proc. of Advances in NIPS **14** (2002)
9. Ng, A., Jordan, M.: On generative vs. discriminative classifiers: A comparison of logistic regression and naive bayes. Proc. of Advances in NIPS **15** (2002)
10. Cozman, F.G., Cohen, I., Cirelo, M.C., et al.: Semi-supervised learning of mixture models. In: 20th International Conference on Machine Learning. (2003) 99–106
11. Martínez, C., Fuentes, O.: Face Recognition Using Unlabeled Data. Computación y Sistemas **7**(2) 123–129
12. Balcan, M., Blum, A.: An Augmented PAC Model for Semi-Supervised Learning. In Chapelle, O., al., eds.: Semi-Supervised Learning. The MIT Press (2006)
13. Dietterich, T.G.: Ensemble methods in machine learning. 1st Int. Workshop on MCS **1857** (2000) 1–15
14. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley Press (2004)
15. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning **39**(2/3) (2000)
16. Yang, Y., Webb, G., et al.: To select or to weigh: A comparative study of model selection and model weighing for spode ensembles. In: ECML. (2006)
17. Brown, G.: A new perspective on information theoretic feature selection. In: 12th Int. Conf. on Artificial Intelligence and Statistics (to appear). (2009)