Understanding, Exploiting and Improving Inter-view Relationships

Sibi Venkatesan

May 6, 2022

CMU-RI-TR-22-19

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:
Artur Dubrawski, *Chair*
Jeff Schneider
Srinivasa Narasimhan
Junier Oliva, *UNC, Chapel Hill*

Submitted in partial fulfillment of the requirements
for the degree of Doctorate of Philosophy in Robotics.

Abstract

Multi-view machine learning has garnered substantial attention in various applications over recent years. Many such applications involve learning on data obtained from multiple heterogeneous sources of information, for example, in multi-sensor systems such as self-driving cars, or monitoring intensive care patient vital signs at their bed-side. Learning models for such applications can often benefit from leveraging not only the information from individual sources, but also the interactions and relationships between these sources.

In our research, we look at multi-view learning approaches which try to model these inter-view interactions explicitly. Here, we define interactions and relationships between views in terms of the information which is shared across them, including corroboration and redundancy of information. We distinguish between global relationships, which are shared across all views, and local relationships, which are only shared between a subset. For example, in a multi-camera system, we can think of global relationships to be defined over the part of a scene which is visible to all cameras, while local relationships would be defined by the intersection of the fields of view of only some of the cameras.

We consider three main aspects of modeling such relationships. First, we develop and study a framework for discovering and understanding them within multi-view data. We describe different approaches to uncover and model these global and local relationships. We look at simple multi-view extensions of auto-encoders, and then move onto more sophisticated generative models.

Second, we explore the benefits of this understanding of inter-view relationships to solve down-stream modeling tasks, exploiting the structure that multi-view data avails us. Here, we adapt our models to tackle different applications, and demonstrate the utility and effectiveness of explicitly modeling these relationships. We first look at incorporating the downstream loss function into the representation learning framework to cater to the task-specific problem. We then consider the applications in the domains of image data and temporal data to evaluate the adaptability of our methods.

Third, we investigate a methodology for improving these relationships directly by facilitating favorable interactions between views. We first look at how one can re-interpret individual views as data points, allowing us to apply traditional machine learning approaches to modeling inter-view relationships. Using this re-interpretation, we look at view-selection where we directly select views which manifest favorable relationships, and propose Scalable Active Search as a candidate for this. Active Search allows us to interactively search for informative views, given an initial set of views and a measure of similarity between them.

Acknowledgements

My eight years at CMU were slow at the best of times and a struggle at the worst. So many people contributed to pushing me over the finish line, and I know I would not have done it if I were missing the support from even a single one of them.

First and foremost, I have to thank my advisor, Prof. Artur Dubrawski whose boundless wisdom and patience were invaluable to me throughout my PhD. He has humor and wit in no short supply, which often helped me see the lighter side of life. Dr. Kyle Miller has been a close mentor to me, and has always been there to lend a hand and an ear whenever I found myself lost. Nick Gisolfi and Ben Boecking were the best office-mates I could have asked for; they made me look forward to coming in every day, even if they poked fun at my tastes in TV shows and coffee. My colleagues and friends at the AutonLab were always there to help, and the tight-knit culture always made it feel like home.

I would like to thank my thesis committee, Prof. Jeff Schneider, Prof. Srinivasa Narasimhan and Prof. Junier Oliva for all their valuable insight and feedback. Thank you to Prof. Oliva also for the many conversations we had when I was still putting the pieces together well before even my thesis proposal.

The RI, and CMU as a whole, have always made me feel welcome, and grad school has its way of helping you find kindred spirits to share in the many delights and despairs you face along the way. In no particular order, I want to thank: Tejas Mathai and Srinivas Somasundaram for being great housemates and even better friends; Venkatraman Rajagopalan and Wen Sun for the much needed coffee breaks and chit-chats; Vishwanath Saragadam for laughing at my jokes, even though I was forced to laugh at his in return; Satwik Kottur for dragging me to the gym despite my best efforts to hide; Praveen Venkatesh who has been a friend of mine for so long, I don't remember a time when he wasn't. I would also like to thank Aditya Sinha and Prahlad Venkatesh for their support and friendship from outside CMU, all the way back from highschool.

My family has always been there for me through thick and thin; my father whose sharpness and thoughtfulness kept me moving forward, my mother whose love and concern could be felt from across the globe, and my brother who always had my back

and put up with my dumb sense of humor without (much) complaint.

I also found solace in niche pockets of the internet, especially in the online communities of TagPro (you should check out the game!) and NGU Idle's Chat Room 5. It's odd how strangers on the internet can become such a strong source of support, even when you have never seen their faces or heard their voices.

Last, but definitely not the least, I want to thank my partner and best friend, Kelsey Wong, for her unconditional love and support. We were on different sides of the country for the entirety of my PhD but not a day went by when she wasn't there for me. She has held my hand through the darkest of times and celebrated with me through the happiest, and I love her that much more for it all.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data. Bio data. Image data. User data. Meta data. *Big data.* Today, we have access to an incredible amount of data to analyze, synthesize and utilize. But this data often does not present itself to straightforward processing; a lot of it is noisy, unstructured, or even missing. And, data can come from multiple, often heterogeneous sources. For example, multi-sensor systems such as bedside patient monitors used in intensive care and self-driving cars produce and depend on data from widely diverse modalities, such as image data and wave-form data. **Multi-view machine learning**, or multi-modal machine learning, involves machine learning on such data which comes from multiple different, but related, sources. Each of these sources gives us some, but not all, information contained in the system we are observing; e.g., ECGs and EEGs observe different aspects of the functioning of the human body, but ailments can potentially manifest in both of them as correlated deviations from "healthy" measurements.

Often, the *type* of data from these sources differs. For instance, in self-driving cars (e.g., Figure 1.1), LIDAR and other depth sensors can provide a more reliable 3D map of the surroundings than cameras, while the latter is necessary for interpreting road signs and traffic signals. Further, the information they can both provide, like the location of nearby pedestrians, is important as well. The corroboration of these sensors in such cases is imperative for callibration, de-noising and more. Similarly, in bed-side patient monitoring systems (e.g., Figure 1.2), physiological readings can come in a variety of sources – the patient's heart, brain, or respiration functions, blood flow in vessels, etc., and modalities – waveform time series, images, videos, all providing valuable information. The financial domain also has pertinent applications for multi-view learning, including credit monitoring and fraud detection. Any given person is likely to have access to multiple different financial accounts and credit sources, such as bank accounts, credit cards, home loans, retirement funds, etc., simultaneously reflecting their financial position and credit worthiness, as well as their banking habits and shopping patterns. We would be remiss to not also mention robots, such as CHIMP in Figure 1.3, equipped with their various sensors to help aid manipulation, movement, mapping, etc.

Figure 1.1: Multi-sensor system: Self-driving cars.

It is unsurprising, then, that the applications of multi-view machine learning are numerous. Making sense of different sources of information as well as the way they interact with each other is of vital importance.

In this thesis, we explore the topic of multi-view machine learning from three directions. We begin by studying and *understanding* inter-view relationships (IVRs), laying the groundwork for building our learning models. Here, we look at identifying redundancy, i.e. overlap and agreement, of information across views as a basis for these relationships. In a simple sense, the manner in which different views agree, or disagree, with each other tells us about how they may be related.

We then look at how to *exploit* this understanding to tackle real-world application domains. Here, we study how these relationships can be used to build domain specific models, as well as how the domain itself can inform how we model these relationsips. We consider a variety of applications for our models, including healthcare data, images and text.

Finally, we investigate ways in which to *improve* these relationships, where we directly manipulate them to bolster favorable interactions. This could be done in multiple ways, including system design (e.g., building our own robot) or context-

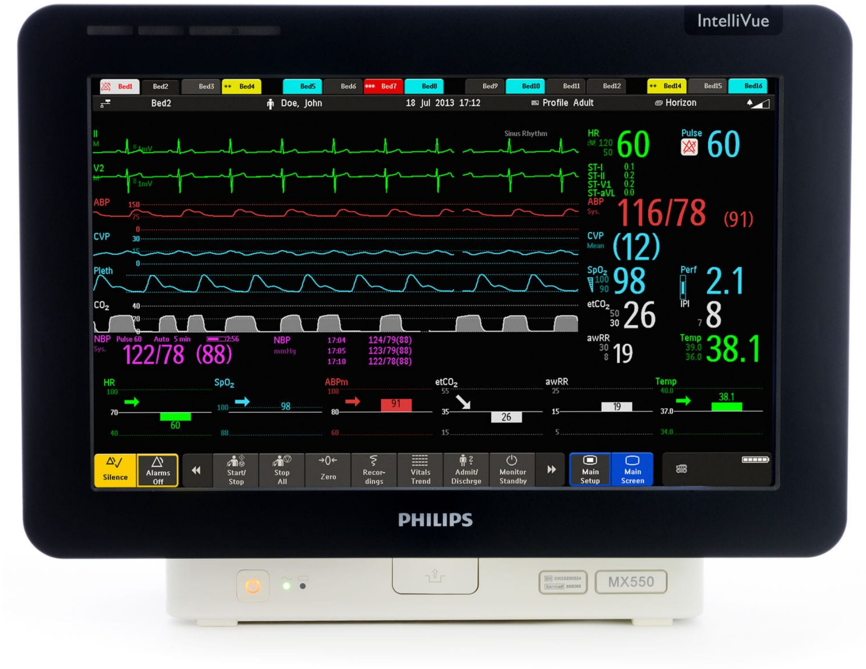


Figure 1.2: Multi-sensor system: Bed-side monitoring of patient vital signs.

centric data processing (e.g., corroboration between cameras can be established more reliably during the day, while LIDAR may see more relevance at night). In our work, we consider the idea of “view-selection,” where we directly select views which manifest favorable relationships.

Central to our exploration is the concept of **Representation Learning**. This refers to the learning of a transformation of data into an intermediate feature space which can better exploited for a given task. Raw pixel data is rarely directly consumed by computer-vision models; but is instead featurized using a plethora of well-studied techniques to extract useful patterns which are easier for models to interpret. Similarly, we will consider Representation Learning as the means to interpret and digest multi-view data so that it may be conducive to further modeling.

In the next sections, we establish some relevant terminology and notation, and give an outline of the different directions we will pursue in this thesis.

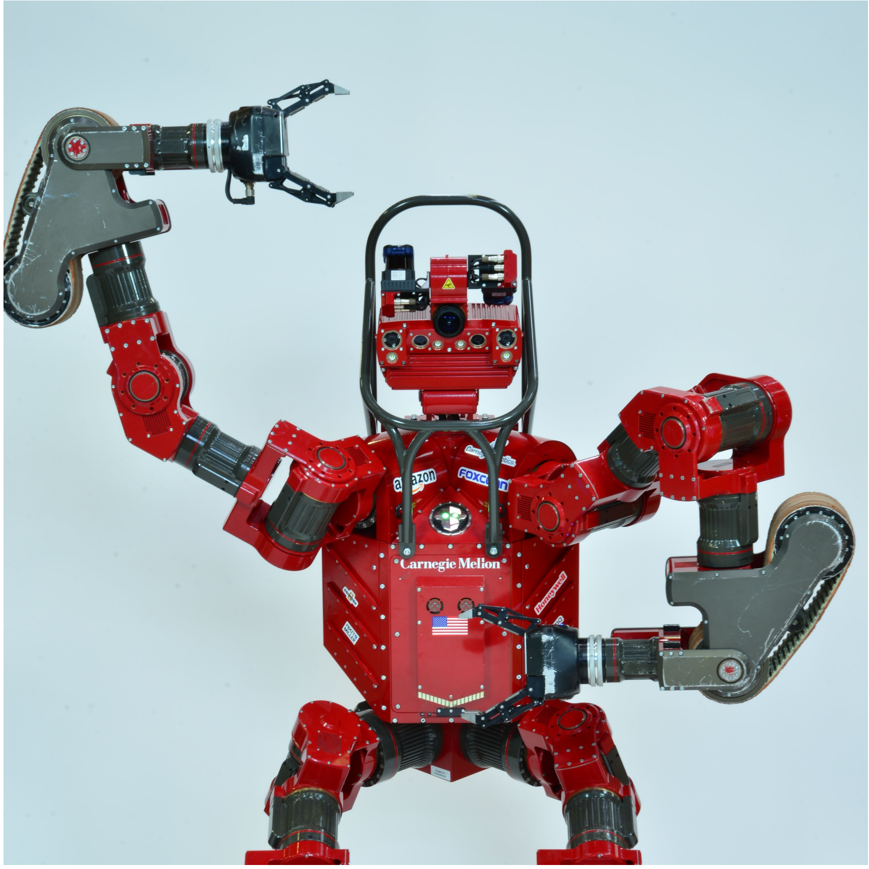


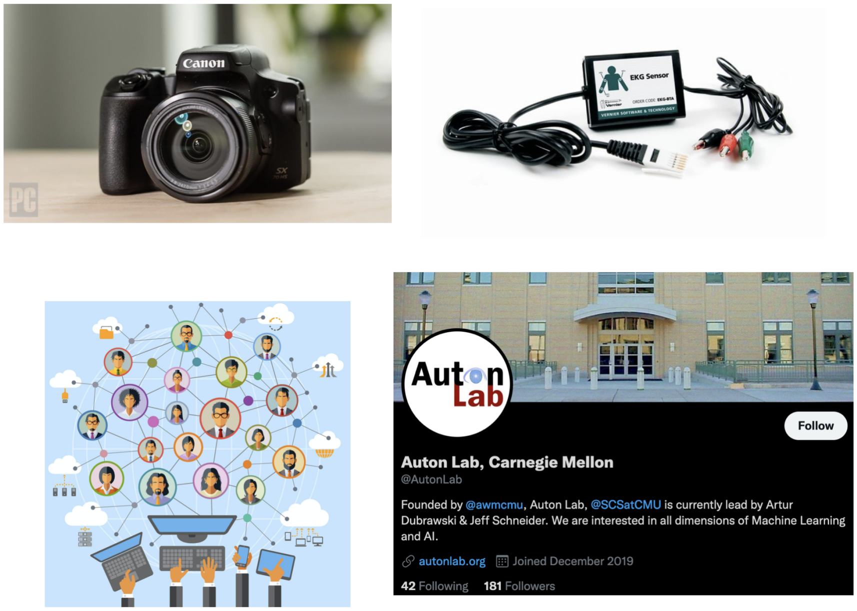
Figure 1.3: Multi-sensor system: CHIMP - one of CMU's many in-house robots.

1.1 Terminology and Notation

Throughout this thesis, we will frequently use words such as **modalities**, **sources** and **views**. While they are often used interchangeably, here are some useful distinctions to help establish context where it matters.

A **modality** refers to the *type* of data available. For example, image, text, audio, and various types of waveform data are all different modalities which may be available to us. A **source** refers to the device or entity producing the data, such as a camera for image data or an ECG for heart activity waveform data. And possibly the most abstract, a **view** refers to information which can be interpreted, and often gathered, independently. The stream of images we get from a video recorded from a camera can be considered a view, for instance.

To help disambiguate, here are some simple examples of how these terms relate. In a multi-camera system, we have a single **modality** being produced by multiple cameras, which are the **sources**. And each of these sources produces its own **view**,



Auton Lab, Carnegie Mellon
@AutonLab

Founded by @awmcmu, Auton Lab, @SCSatCMU is currently lead by Artur Dubrawski & Jeff Schneider. We are interested in all dimensions of Machine Learning and AI.

autonlab.org Joined December 2019

42 Following 181 Followers

Figure 1.4: What exactly are views? Cameras and ECGs are **sources** which provide us with information in the **modalities** of image and waveform data respectively. These streams of images and waveform readings which we interpret are what we consider as **views**. A **system** is the entity which produces *all* the data in all the views. It encompasses all its constituent modalities, sources and views. For e.g., a social media network could be considered as a system, with individual users in it as sources producing posts as views.

i.e. the corresponding stream of images. If we have a single camera, we can even consider different featurizations of the same data-stream. In this case, we have a single **modality** and **source** but multiple **views** corresponding to different featurizations of raw images or videos. Figure 1.4 walks through these concepts as well.

It is important to note that our models consume individual **views**. In technical terms, these are the different sets of features which will be fed into our models. We may use the terms interchangeably when we deal in abstracts, but these distinctions will help keep the context clear when it matters.

We also often use the term **system** to refer to the data-generating process. This may be the sensorized human body for physiological data, the environment surrounding the self-driving car or robot, or the individual whose credit is being evaluated. Further, this **system** produces "underlying" data-points, which represent the information (features) we receive as views. For example, a data-point could be a time-window during which a human body is observed. Featurized representations of vital signs waveforms measured in such a time-window could be taken as views. We use the term "underlying" to refer to the ground-truth data, and each view provides limited information about this data. This underlying data-point is never assumed to be fully characterized, even with all the views. It is just the entity which establishes a correspondence within the available views.

And finally, we will often use the term **relationship** (and less often, interaction), to refer to useful connections we can establish between two or more views from the information provided by them. One of the main focuses of our work will be to explore this idea in more detail. So while the concept may remain nebulous now, it will become more concrete as we move forward. In the simplest form, though, it can be considered as a function over the information shared between a subset of views. We will use the terms "multi-view" **relationship**, "inter-view" **relationship** or simply **relationship** interchangeably throughout this document.

1.1.1 Notation

Here is the notation we will be adopting throughout this document. Each view of the multi-view data will be represented as X_i where i is the index of the view, and K is the number of views. The data-points (samples) within each view will be indexed by a superscript: the j^{th} sample of view i is given by x_i^j. In general, we assume correspondences for a given sample index across different views. For example, x_1^j and x_2^j both correspond to the same underlying data-point observed via two different views. We take the total number of underlying data-points to be N.

In the case of missing views, we take $\mathcal{K}_\alpha$ to be the set of view indices which are available; this depends on the context but is usually associated with a single sample. If the j^{th} sample of view i is missing, we do not have a value for x_i^j, and $i \notin \mathcal{K}_\alpha^j$. In the case of an explicit latent representations, we take L_i to be the latent representation of view i and L_{all} as the shared latent representation across all views.

If there are special cases where we depart from this notation, we will describe any changes where they occur.

1.2 Modeling Inter-view Relationships

In this section, we will lay the foundation for much of the work we discuss in the rest of this thesis. We start by introducing the hypothesis that is central to the theme of our research:

> *Multi-view data does not just provide us with multiple sets of features through different views, it also provides structural information through the interactions between them.*

In other words, multi-view data can be considered a gestalt, where the information provided is more than just the set of all features. To make this clearer, let us revisit the example of self-driving cars. Depth sensors and cameras both provide us with useful information about the surrounding environment. But the *agreement* between them on the information they both provide can be vital. For instance, the information they each provide on the locations of nearby pedestrians can allow estimates of higher confidence than any single sensor. Here are a few, more general, utilities of looking at these interactions:

- Leveraging redundancies to bolster models built over the data, e.g., denoising.
- Reconstructing missing or corrupted views, e.g., data imputation.
- Quantifying usefulness of a given view through its relationships.

But what about simply **concatenating** or stacking together all the individual views into one, big single-view dataset? This definitely provides us with all the information available to us, and agreements and overlap between views just translates to that over feature-subsets. So, what do we lose if we do this? In theory, with enough complexity, a single-concatenated-view model should be able to implicitly recover and exploit any useful structure in the data. In practice, this isn't as straightforward [Liu et al., 2016].

We maintain that explicitly considering the distinctions between the views and modeling their interactions is beneficial in building efficient multi-view models. Guiding our models on *where to look* and *what to look for* allows us to make more effective use of the structure inherent to the data. To this end, our first step is to try to unravel what these relationships are, and how we can interpret and model them.

1.2.1 What is an Inter-view Relationship?

This is one of the primary questions we will be trying to answer with our work. To begin with, we distinguish between what we call *global* and *local* relationships.

- **Global** relationships are those which are shared across all views.
- **Local** relationships would be those which manifest only within a subset of views.

A simple example to distinguish between the two is that of a multi-camera system. A **global** relationship would be defined over the region of the observed space which is visible to all the cameras, while a **local** one would be formed over any region which is visible to some subset of cameras. While this particular example lends itself to geometric and computer vision based characterizations, e.g., calibration, the more general problem might require more abstract characterizations. We need to keep the following in mind while building our models: By definition, global relationships consider the most constrained region of the data-space: the *intersection* of information available from all the views. In contrast, local relationships can be defined anywhere in the *union* of this information. Going back to what we said earlier, this distinction helps define *where to look* for IVRs.

Existing modeling approachs largely tend to favor *global* relationships. The classic example is the formulation of Canonical Correlation Analysis (CCA) problem and multi-view extensions such as [Rupnik and Shawe-Taylor, 2010]; here, a single mapping is learned for all views into common embedding space which maximizes a measure of a joint correlation between views.

We are interested in the general setting: learning local and global relationships over general, source-agnostic, data. The nature and structure of these relationships are not always known within the data a priori. Further, the distinction between these two is not always obvious either. Unless there is very clear structural information we can exploit, this ambiguity will always be present. In short, this ambiguity in *what to look for* is the price we pay for generality.

So how do we go about dealing with this ambiguity? It is important to note that the practical utility of these relationships, and even how we might define them in a particular context, is largely dependent on the specific learning problems we are interested in. Information which is useful for one problem might not be relevant for another problem, even within the same data. For example, we could reasonably assume that multi-view image classification and missing-view image imputation would not derive the same utility from any specific model of these relationships. Ideally, the models we build will be able to gracefully incorporate domain-specific information into the learning process.

For this, we consider two avenues to pursue our modeling, namely, *task-agnostic* and *task-adaptive*. For both these cases, we will primarily model these relationships through **Multi-view Representation Learning** (MVRL). Again, representation learning is generally performed as an intermediate or meta step to extract information from data and convert it into a meaningful form which is more conducive to learning and modeling for downstream tasks. To elaborate on the distinction between *task-agnostic* and *task-adaptive*, the former uses MVRL truly as a meta-problem where we assume we do not know what task our learned representations will be used for. The latter considers the scenario where the MVRL is tied to a specific application or problem which we can use as scaffolding for our learning. In simple terms, these can be considered as *unsupervised* and *supervised* modeling approaches respectively.

In the case of *task-agnostic* modeling, we will introduce a simple proxy task which we can optimize towards. This typically is either reconstruction of missing views, or likelihood maximization. These two tasks can always be considered, even in the case of unsupervised data. For *task-adaptive* problems, we assume we have access to the down-stream loss function we care about; we can rope in this loss into our MVRL approaches to guide the learning process towards down-stream utility.

We also introduce the idea of **robustness** as a desirable quality of our models. For the learning problems we consider, we take **robustness** to mean resilience to missing information; i.e., we would like our model to produce reasonable outputs, even when some of the views go missing in our input data. In our *task-agnostic* setting, this would mean that we would like to produce reasonable imputations (or realistic samples) of missing data from the available information. Further, we want this *robustness* to persist regardless of which views go missing. In other words, we would like our models to be resilient under arbitrary subsets of missing views in our input data.

In the next section, we give a brief outline of the rest of the thesis.

1.3 Outline

This thesis is composed of three parts:

1. **Understanding** and studying IVRs, where we look at MVRL approaches for modeling them.
2. **Exploiting** these relationships for learning problems, where we adapt our learning approaches to specific applications.
3. **Improving** and expanding upon the relationships themselves, where look at how we can facilitate favorable properties in the relationships themselves.

When relevant, we describe *where we look* and *what we look for* with our models.

1.3.1 Part I: Understanding Inter-view Relationships

In this part, we focus on *task-agnostic* MVRL approaches.

Chapter 2: Robust Multi-view Representation Learning

In this chapter, we look at discriminative approaches for MVRL. We first propose a multi-view CCA based model to directly extract IVRs: Multi-view One-vs-Rest Embedding Learning (MOREL). For each view, we learn an embedding which tries to explicitly characterize local relationships this view is involved in. The purpose of this model is to serve as a simple validation of our hypothesis in Section 1.2.
Where to look: Each view explicitly, to characterize specific local relationships.
What to look for: Correlation-based (or similar) individual embeddings, given a view.

We then move to an extension of multi-view auto-encoders, which uses view-drop-out during MVRL: Robust Multi-view Auto-Encoder (RMAE). The idea here is that, by randomly dropping subsets of views during training while optimizing for the proxy-task of reconstructing all views, we encourage model *robustness*(i.e. resilience to missing data).
Where to look: Subsets of views implicitly, to characterize general local relationsips.
What to look for: Reconstruction-based shared embeddings, given any subset.
Take-away: This chapter demonstrates the utility of modeling local IVRs within the data, either implicitly or explicitly, especially when training our models to be *robust*.

Chapter 3: Generative Models for Multi-view Representation Learning

In this chapter, we look at generative modeling for MVRL, allowing us to estimate and sample from probability densities. Here, we follow the same general framework and view-drop-out training of RMAE, but switch to the proxy-task of likelihood maximization. For this, we propose a Flow-based approach: Multi-view AC-Flow (MACF).
Where to look: All subsets of views implicitly, as in RMAE.
What to look for: Likelihood-maximization-based shared embeddings, given any subset.
Take-away: This chapter shows the added benefits from considering generative models such as sampling missing views, in addition the previously established utility of implicitly modeling IVRs.

1.3.2 Part II: Exploiting Inter-view Relationships

This part moves on to *task-adaptive* representation learning approaches.

Chapter 4: Incorporating Down-Stream Tasks into MVRL

In this chapter, we look at incorporating down-stream task information into the MACF model through propagating domain/application-specific losses upwards into the training procedure. We consider applications in image and medical data to evaluate the utility of adapting the MACF.
Where to look: All subsets of views implicitly.
What to look for: Task-augmented likelihood-maximization-based shared embeddings, given any subset.
Take-away: This chapter introduces *task-adaptivity* into the MACF learning pipeline, as well as the application of these approaches to real-world learning problems. Through this, we learn that *task-adaptiveness* be constructive or detrimental depending on context and the data used.

1.3.3 Part III: Improving Inter-view Relationships

This part steps away from MVRL to look directly at how we can facilitate favorable IVRs directly in the data.

Chapter 5: View Selection and Scalable Active Search

In this chapter, we describe the concept of view-duality, i.e., a reinterpretation of views as data points and vice-versa. Using this reinterpretation, we can then improve IVRs through view-selection. In this context, view-selection involves dynamically choosing views which manifest properties in IVRs which are desirable. There is an analogy to the active learning paradigm that chooses most useful data-points to adjudicate, but we wish to choose the most useful views. As a candidate for this, we look at a scalable extension of Active Search: here, we are given an explicitly quantified local relationship between views as a pair-wise similarity function.

We first show the connection between Active Search and improving IVRs: This similarity function may not be task-aligned, so our goal is to select views which are more relevant to the task, with input from an oracle on each selection we make. E.g., we have a social network inducing a follow/friendship based similarity function on users, but we want to gauge the general sentiment on a topic which only a subset of users care about.
Where to look: The set of all available views.
What to look for: Views improving task-specific performance.

We then separately look at a scalable extension to deal with large amounts of data: If we specifically consider linear similarity functions, we can derive an equivalent formulation to the original which can scale several orders of magnitudes. This formulation prescribes alternate initialization and update steps which reduces the quadratic and cubic dependence on number of data-points respectively to linear in both, in lieu of a new dependence on dimensionality features.
Take-away: This chapter introduces a reinterpretation of multi-view data which relates views and data-points, allowing us to use traditionally single-view machine learning methods for view-selection. It describes Active Search as a candidate method for the view-selection task, and a highly scalable reimplementation of the same.

1.4 Related Work

Multi-modal machine learning has seen an impetus of recent work, owing to the increasingly easy access to a vast amount of multi-view data. For instance, video data, language-to-language translation, and even on-board sensors on self-driving cars give multiple modalities of viewing the same dataset. It is then imperative to interpret and analyze these multiple channels holistically for models to make the best use of the increased amount of information.

Learning over multiple modalities is often difficult, due to heterogeneous sources of data, different levels of noise or missing data in some views. This makes it imperative to extract meaningful information from the different views in a robust fashion. Representation learning is thus one of the core directions of multi-modal machine learning research. It is common as an intermediate step before learning over a down-stream task.

Many such learning methods are tailored to certain domains, wherein they exploit the structure available specific to the data. For example, Audio-Video Speech Recognition (AVSR) has been the subject of research for many years now. Traditionally, deep neural networks are used to handle visual, textual and acoustic data [Ngiam et al., 2011], [Ouyang et al., 2014], [Wang et al., 2015] where the model projects the modalities into a joint space [Antol et al., 2015], [Mroueh et al., 2015], [Ouyang et al., 2014], [Wu et al., 2014]. This representation is then used for the relevant learning task. It is common to pretrain such networks using Auto-Encoders on unsupervised data [Hinton and Zemel, 1994].

Multi-view Auto-Encoders are also extended to learn latent representations over multi-modal data. [Ngiam et al., 2011] learn modality specific AEs and then fuse together the latent states into a final shared representation. [Silberer and Lapata, 2014] use auto-encoders for semantic concept grounding, with the addition of a loss-term for object-label prediction. [Wang et al., 2015] fine-tunes the representation learned by the generic AEs on a given task.

Sequential data often needs additional care while learning latent representations; the data is not always fixed-length (sentences, video, etc.). RNN/LSTM based encoder-decoder networks are often used for this where the hidden state at the end of a variable-length sequence is used as its fixed-length representation [Bahdanau et al., 2014], [Venugopalan et al., 2014]. RNNs are often used for ASVR [Cosi et al., 1994], audio-visual affect recognition [Chen and Jin, 2015], [Nicolaou et al., 2011], etc.

Neural networks have their advantages; given domain-specific architectures and the potential for pre-training, they often show superior performance on certain tasks. But they need a lot of data and are not always able to gracefully handle missing data from modalities.

Another popular multi-modal representation learning approach is based on graphical models. Unsupervised methods such as Deep Boltzman Machines (DBN) have been extended to multi-modal datasets. [Srivastava and Salakhutdinov, 2012a] introduce multi-modal Deep Belief Networks for representation learning. Multi-modal DBNs are also used for audio-visual emotion recognition [Kim et al., 2013], AVSR [Huang and Kingsbury, 2013], gesture recognition [Wu and Shao, 2014], and other applications [Ouyang et al., 2014], [Suk et al., 2014].

Graphical model based approaches deal with missing data well, and can often be trained in an unsupervised manner. But they tend to be difficult to train with high computational costs, often needing approximate variational training methods [Srivastava and Salakhutdinov, 2012b].

Another class of models are coordinated-representation models, where the modalities are not projected to a joint space but instead are coordinated through similarity constraints. Some of these approaches are based on similarity models [Weston et al., 2010], [Weston et al., 2011], [Frome et al., 2013], [Kiros et al., 2014], [Socher et al., 2014], [Pan et al., 2016], [Xu et al., 2015]. Others are based on correlated embedding spaces such as CCA-like techniques. CCA models are common for cross-modal retrieval problems [Hardoon et al., 2004], [Klein et al., 2014], [Rasiwasia et al., 2010] and AV signal analysis [Sargin et al., 2007], [Slaney and Covell, 2001]. Nonlinear and deep extensions of CCA such as Kernel CCA [Lai and Fyfe, 2000], Deep CCA [Andrew et al., 2013], Correspondence AEs [Feng et al., 2014], etc. have also been proposed.

Coordinated-representation models are usually limited to two-modality problems, though there have been some extensions into three or more views. Multi-view CCA [Rupnik and Shawe-Taylor, 2010] learns a projection for each view to maximize some measure of "global" correlation, but these don't share the simple eigen/singular-value decomposition based solutions which 2-view CCA problems often have.

1.5 Contributions

Where does current work fall short?

Here, we describe what we believe to be gaps in the literature, and how we try to address them:

The main gap in the existing literature for multi-view machine learning is in how relationships between views are modeled. The majority of the research does not consider or model these relationships explicitly. That is to not say they are ignored; they are usually modeled implicitly, through domain or application knowledge. As a consequence, these methods are typically tailored to the specifics of the data or the application considered; the interactions between the views are then learned as a by-product of the choice of models, frameworks and architecture. Existing approaches which *are* domain/application agnostic are almost always for the two-view case.

Further, in most cases, the nature of these relationships is rarely discussed. Similarly, the nuances within these relationships such as the distinction between *global* and *local*, are not explored. They are considered implicit to the problem itself, such as linguistic connections between languages for machine translation, or geometric calibration between multiple cameras and sensors. Thus, theoretical or statistical investigations of these relationships are rarely done outside of the narrow contexts of the applications considered, if they are done at all.

Lastly, our notion of *robustness* is rarely considered i.e., models typically make assumptions of how and which views are available. For example, in machine translation, the learning models almost always know which language is being generated and which one is given.

To summarize, the related work fails to adequately address the following gaps:

- Explicitly considering and modeling nuanced IVRs, as well as considering *robustness* of the learned models.
- Domain and application agnostic modeling of multi-view data, especially in the case of three or more views.
- A principled theoretical characterization of IVRs.

What does this thesis address?

This thesis tries to primarily address the first two concerns listed above, leaving the context-agnostic theoretical investigations as a potential next step. Our work tries to bridge these gaps by proposing MVRL models which, *explicitly* or *implicitly*, characterize IVRs. As mentioned before, the nature of these relationships is not always easy to categorize and certainly do depend on the context of the problem involved. We explore this by considering both *task-agnostic* and *task-adaptive* models, and weigh their costs and benefits. In short, contributions of this dissertation are as follows:

1. Developing new *task-agnostic* and *task-adaptive* representation learning methods for multi-view data by considering IVRs.
2. Building models which place no explicit restrictions on number of views.
3. Introducing and modeling for the notion of *robustness*, the resilience to arbitrary subsets of missing views.

Part I

Understanding Inter-view Relationships

Chapter 2

Robust Multi-view Representation Learning

2.1 Introduction

With this chapter, we begin our foray into modeling IVRs. As we stated before, the core hypothesis in our work is that multi-view data provides us with more information than just simply multiple streams of data. *Understanding* how these data-streams interact with each other gives us valuable insight into the system. We just need to know *where to look* and *what to look for.* For this, we look at Multi-view Representation Learning (MVRL).

Our goal is to extract useful and robust latent representations by leveraging the these IVRs. Unlike typical existing approaches, we model *local* relationships. In other words, we do not want to impose stringent modeling restrictions on which views we consider and how they interact with each other. We propose two methods to approach this problem:

- Multi-view One-vs-Rest Embedding Learning (**MOREL**)

  MOREL is an extension of CCA where we consider multiple one-vs-rest CCA problems, and learn embeddings separately for each view. This is akin to searching for local relationships for each view. To encourage searching for more nuanced interactions in the data, we add group-sparse regularization where the groups correspond to the views.

- Robust Multi-view Auto-Encoder (**RMAE**)

  RMAE is an extension of multi-view auto-encoders, where we move our focus towards the property of *robustness.* We achieve this through applying the idea of drop-out to views: By dropping random subsets of the views while still reconstructing all views during training, we force the model to uncover useful local redundancies and overlap.

For this chapter and the next, we will be looking at *task-agnostic* modeling. We demonstrate the utility of directing our attention to the *where* and *what* of the IVR through synthetic experiments. We also evaluate our approaches on real-world experiments on multi-view text-based and image datasets.

2.2 Approaches

2.2.1 Multi-view One-vs-Rest Embedding Learning

2.2.2 Background: CCA

Canonical Correlation Analysis (CCA) fits in the intersection of dimensionality reduction and multi-view representation learning. CCA tries to find a projections for multiple (typically 2) feature sets into a common space where the correlations between them are maximized. The typical loss function is as follows:

$$\max(X_1 w_1)^T (X_2 w_2)$$

Typically, extensions to the multi-view (3+ views) setting [Kakade and Foster, 2007], [Rupnik and Shawe-Taylor, 2010] involve finding a single set of projections for each view which maximize some overall correlation objective across all views. For example, [Rupnik and Shawe-Taylor, 2010] formulate their objective as follows:

$$\max_{w_i} \sum_{i<j} (X_i w_i)^T (X_j w_j) \tag{2.1}$$

This formulation is usually more suited to extract *global* relationships in the data, unable to capture nuanced *local* relationships since all the projections are into a single shared space. To naively extend this to model all possible local relationships, we would have to do a combinatorial number of multi-view CCA computations, one for each subset of views.

2.2.3 MOREL

Our proposed method exploits a simple fact: any local relationship contains at least one view. MOREL solves subproblems for each view, to try to characterize interactions which involve that view. Through this, we learn **directed** relationships between a given view and the remaining. The term **directed** here refers to how each subproblem is within the context of a particular view: we look at the contribution to the correlation from other views to the given view. Below, we describe the approach we propose in [Venkatesan et al., 2020].

We cast the problem of latent representation learning into K one-vs-rest CCA problems, one for each view. Given a single view, we learn the combined projections for each remaining view to maximize the correlation with the given one. Here, we are essentially performing a 2-view CCA computation where the remaining views together form the second "view". This way, we hope to uncover a dependency structure for correlation/reconstruction where we can model the "contributions" of any other view to the given view. Of course, as we have discussed before, a simple concatenation of views ignores the multi-view structure.

To remedy this, we make use of group-sparse regularization to encourage learning projections which respect this structure. The groups here correspond to the individual views within the aggregation of the remaining views. With this, we can define our objective. For now, we assume that all our projections are linear; we discuss how to move away from this a little later. For each view, we minimize the following:

$$\min_{P_{ij}} f(X_i P_{ii}, \sum_{j \neq i} X_j P_{ji}) + \lambda \sum_{j \neq i} R_G(P_{ij}) + \gamma R_{all}(P_{i:}) \tag{2.2}$$

where $f(A, B) = -A^T B$, R_G is the group-sparsity regularizer, R_{all} is a global regularizer and $P_{i:}$ refers to the concatenation of all projections P_{ij} where $j \neq i$. We typically take the group-sparse regularizer to be the L_∞ norm and the global regularizer to be the L_1 norm. For CCA, we add the associated orthogonality constraints as well:

$$(X_i P_{ii})^T (X_i P_{ii}) = \left(\sum_{j \neq i} X_j P_{ji} \right)^T \left(\sum_{j \neq i} X_j P_{ji} \right) = \mathbf{I}$$

Since this is basically a 2-view CCA problem for each view, we optimize this using linearized ADMM following [Suo et al., 2017].

While this approach has been described from the context of CCA, the primary points to note are the (i) one-vs-rest reduction to uncover local relationships and the (ii) group sparsity regularization to respect the view-structure of the data. With this in mind, the function f in Equation 2.2 (and relevant constraints) can be replaced by other appropriate loss functions which optimize similar criteria. A simple replacement is the squared L_2 reconstruction error: $f(A, B) = ||A - B||_2^2$. Without orthogonality constraints, this reduces to a straightforward convex optimization problem. To guard against the trivial solution here, we can fix P_{ii} to be the identity. Here, we define a new term to represent the combination of all the projection matrices:

Definition 2.2.1 (Redundancy Matrix)**.** The Redundancy Matrix $\mathbf{M}$ is the block matrix where the block in i^{th} row and j^{th} column is to the projection P_{ij} from view

i into view j. I.e,

$$\mathbf{M} = \begin{bmatrix} -I_{d_1} & \cdots & P_{1K} \\ \vdots & \vdots & \vdots \\ P_{K1} & \cdots & -I_{d_K} \end{bmatrix} \tag{2.3}$$

With this, we can rewrite the problem as a linear system solved with least squares:

$$||[X_1 \cdots X_K]\,\mathbf{M}||_2^2 \tag{2.4}$$

This formulation shows that we are essentially trying to construct an $\mathbf{M}$ which has the data distribution as its null space. Group-sparse regularization here translates to block-sparse regularization of $\mathbf{M}$. We expect that, by optimizing this loss, the sparsity structure of $\mathbf{M}$ reveals information about the directed relationship between different views.

So far, we have assumed linear projections P_{ij} but we can also pick classes of projection functions $P_{ij}(X_{ij})$ which we can optimize over. For example, deep neural networks are straightforward to incorporate into this formulation; we can apply the group-sparsity penalty over the parameters of each projection network.

To summarize: we go back to the *where* and *what* of IVR: MOREL looks at each view individually, and it looks for relationships in the form of embeddings which maximize one-to-rest correlation. It is important to keep in mind that MOREL does not uncover *all possible* local relationships. But the formulation guides us toward recovering the more apparent and prominent ones.

We note that propose MOREL primarily as a means to validate our hypothesis of the gestalt nature of multi-view data. Our evaluations for this are on simple synthetic data where we can induce the types of IVR we want to investigate.

2.2.4 Robust Multi-view Auto-Encoder

We now move onto a more practical approach based on Multi-view Auto-Encoders (MVAE), better tailored for real-world applicability. We will also turn our focus to the notion of *robustness*.

The typical strategies for training an MVAE have the same concerns outlined in the previous section; they try to directly learn a shared embedding space which best reconstructs all views ([Ye et al., 2016], [Wang et al., 2015]). This often means learning a single bottle neck representation shared across all views. Just as before, this implicitly constrains the model to only look at the intersection of all views, i.e. *global* relationships.

In our proposed framework, Robust Multi-view Auto-Encoder (RMAE) [Venkatesan et al., 2020], we tackle this problem through two ideas. The first is the structure of the framework itself. The RMAE is composed of two tiers; one is at the view

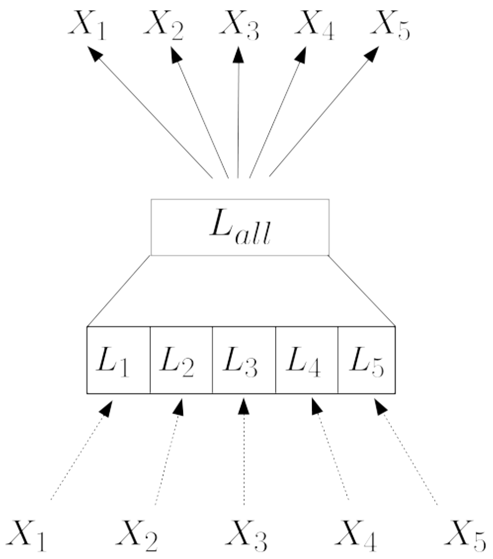


Figure 2.1: This is the outline of the Robust Multi-view Auto-Encoder for 5 views. Each bottom arrow represents the encoder for its respective view, and each upper arrow represents the decoder. L_i is the encoding for view i and L_S is the global latent representation. The bottom arrows are dotted to represent potential drop-out during training time.

level, where every view has its own individual encoder network, and the second is common to all views. The individual encoders produce the latent embeddings L_i for their respective views, which are then concatenated and fed through an additional meta-encoder to produce the final global latent representation L_{all}. This framework is depicted in Figure 2.1.

The second idea lies in how we introduce *robustness* into our framework. In this context, we take *robustness* of a representation as the ability to faithfully reconstruct all views given an arbitrary subset of available views. For this, we borrow from the idea of dropout; every batch, we drop a different, random subset of views while forcing the reconstruction of all views. In this way, the training encourages the latent representation to exploit redundancy of information across different views. We can also sample "available" views during training using prior knowledge; e.g., we can incorporate knowledge that a specific view will never go missing by always having it as input in training.

To emulate dropout more appropriately, we perform a relative scaling of the input to the encoders based on number available views. In dropout with probability p, the output of the used units are scaled by $\frac{1}{p}$ to compensate for the missing unites. Similarly, we scale the available views by $\frac{K}{|\mathcal{K}_\alpha|}$. However, unlike in unit-level dropout, we also do this during test time when we have missing views.

We can control where the view-dropout takes place; we can either zero out the input X_i or we can zero out the latent encoding L_i. The former method is similar to encouraging every individual view encoder to output an informative "mean" embedding which works well in lieu of missing data. The latter localizes the *robustness* of the encoding to the meta-encoder level. In general, we go with the former, since it allows the view-specific encoders to participate in the training procedure even when the views are taken to be missing.

This is similar to the Variational Auto-Encoder with Arbitrary Conditioning (VAEAC) [Ivanov et al., 2019], which is a generative model for estimating arbitrary missing feature values in data (e.g., in-painting). While theirs is a single-view approach, similar to RMAE, they also consider sampling "dropped" features from some prior distribution. However, our approach allows us to incorporate view-specific encoders, since we can exploit the structure in our data. These view-specific encoders can be learned while training, or can be pre-trained or otherwise fixed (e.g., through domain knowledge).

To summarize, let us look at the *where* and *what* again. RMAE implicitly searches over different subsets of views; we say implicit since we do not learn parameters specific to any given subset of views. It looks for reconstruction-based characterizations of information overlap a given subset of views.

Alternative MVAE approaches

We also consider the following MVAE-based competitors against RMAE:

- **Intersection MVAE (IntersectMAE):** This version of the MVAE considers a single shared bottleneck representation.
- **Concatenation MVAE (ConcatMAE):** This version of the MVAE just pre-concatenates the views into one before feeding them into a standard uni-modal auto-encoder.

The framework for these two versions of the MVAE are shown in Figure 2.2.

2.3 Synthetic Experiments

2.3.1 Synthetic Data

Here, we describe a simple approach for the construction of a multi-view dataset with redundant IVRs. Figure 2.3 shows a Venn diagram for the case with three views

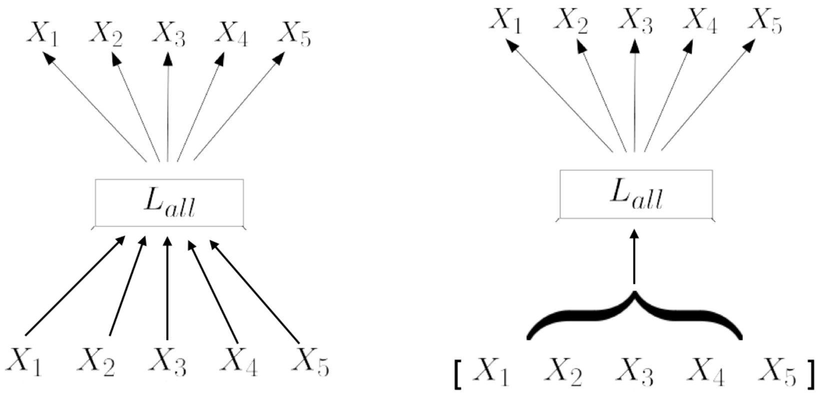


Figure 2.2: This figure shows the frameworks of the IntersectMAE (left) and ConcatMAE (right) versions of the MVAE.

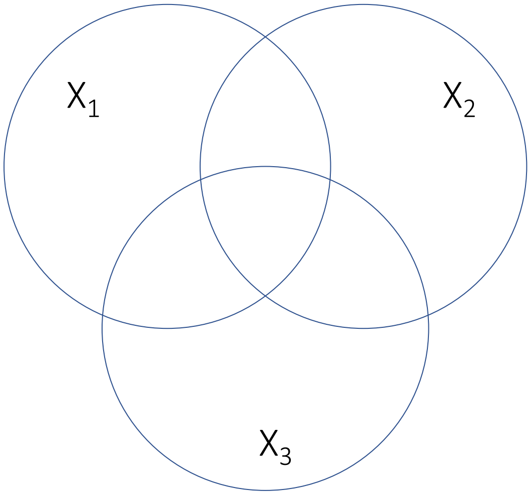


Figure 2.3: Synthetic muti-view dataset construction with inter-view redundancy.

X_1, X_2 and X_3.

We can treat each individual partition as having an independent data distribution which can contribute to the overall latent space of the data. Any partition which lies within the circle of a given view is accessible by it; thus, view X_1's latent space contains the contributions of the four partitions contained within its circle. For example, the three views could be cameras with the circles corresponding to their fields of view, and the individual partitions are regions which are accessible to a specific subset of the cameras.

To construct a redundant multi-view dataset, each of these partitions can be turned "on" or "off" to allow or restrict their contribution to the latent space. For example, if we only turn on $X_1 \bigcap X_2$, $X_2 \bigcap X_3$ and $X_3 \bigcap X_1$, then, this would give us a dataset where any two views are enough to reconstruct the entire latent space. This allows us to customize the redundancy-relationships offered by the different views, and study our methods under different levels/forms of this redundancy.

2.3.2 Evaluating MOREL

4-view problem with simple redundancy Here, we consider the simple case where any two views are enough to reconstruct all other. We extend the three view case in the previous description as follows: We consider four underlying independent feature partitions A, B, C, D. The four views are created as $X_1 = [BCD]$, $X_2 = [ACD]$, $X_3 = [ABD]$ and $X_4 = [ABC]$, with some noise added. E.g., each of A, B, C, D are individual vital signs of a patient (e.g., heartbeat, blood pressure). Any device X_i can potentially measure multiple readings at once, so different devices can measure the same readings.

For MOREL, we used the reconstruction loss, without the CCA constraints. Figure 2.4 shows the redundancy matrix M learned using only least-squares while 2.5 shows M learned with the addition of group-sparse regularization. The rows represent individual one-vs-rest CCA subproblems. Each non-diagonal block on each row represents the projection from the column-view to the row-view. Reconstruction loss is close to 0 for both cases.

Here, blocks getting zeroed out is favorable; it shows that the inclusion of group-sparsity allows the model to ignore some views and find more exclusive IVRs.

2.3.3 Evaluating RMAE

For the RMAE, we ran a similar experiment but with the goal of understanding how reconstruction error of all views varies with number of available input views. Here, we looked at experiments with varying number of total views.

Here we have a plot of a curve for average error of reconstruction given number of available views for a 5-view and 6-view system respectively. Figures 2.6, 2.8 shows the trend we would expect; the average reconstruction error of all output views goes down the more input we have. “Relative error” refers to error relative to the largest single-view reconstruction error across all methods. In general, the trend we notice is that the error of reconstruction of an output view decreases with more input views which are locally redundant with it. Again, this is something which we would expect.

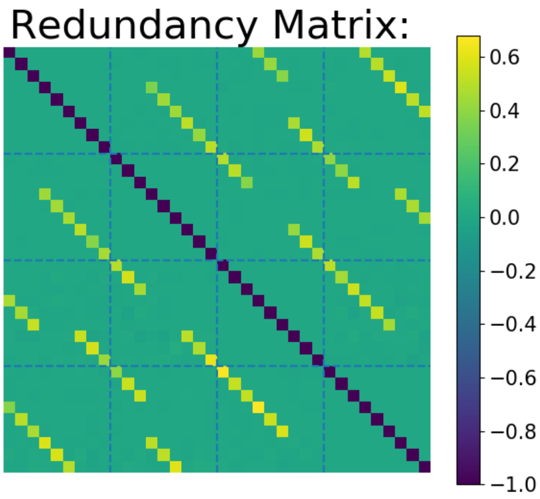


Figure 2.4: Redundancy matrix after optimizing using unregularized least squares. The learned projections do not exploit the redundancies.

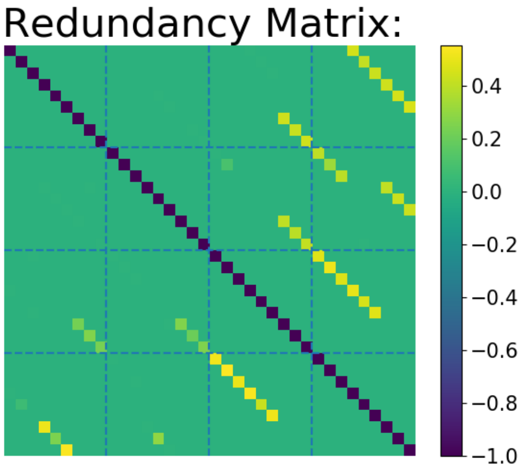


Figure 2.5: Redundancy matrix after optimizing using group-sparse regularization. The learned projections largely ignore some redundant views in the reconstruction.

Performance under different levels of redundancy

Next, we consider the problem of reconstruction using the learned robust latent representation of the RMAE. The synthetic dataset family we look at are those where every numbered view intersects in information with only the ones next to it (indexed 1 less and 1 more in this case). The algorithms we compare here are the RMAE, the

ConcatMAE and the IntersectMAE. We show training and test-set errors for average reconstruction of all views as a function of available input views.

We also try to show an empirical estimation of "usefulness" of a single view under each algorithm by plotting the one-to-one reconstruction error for each view to each other view. These are given in Figures 2.7 and 2.9. We would hope that the methods are able to recognize which views have local intersections of information and are able to reconstruct those views better.

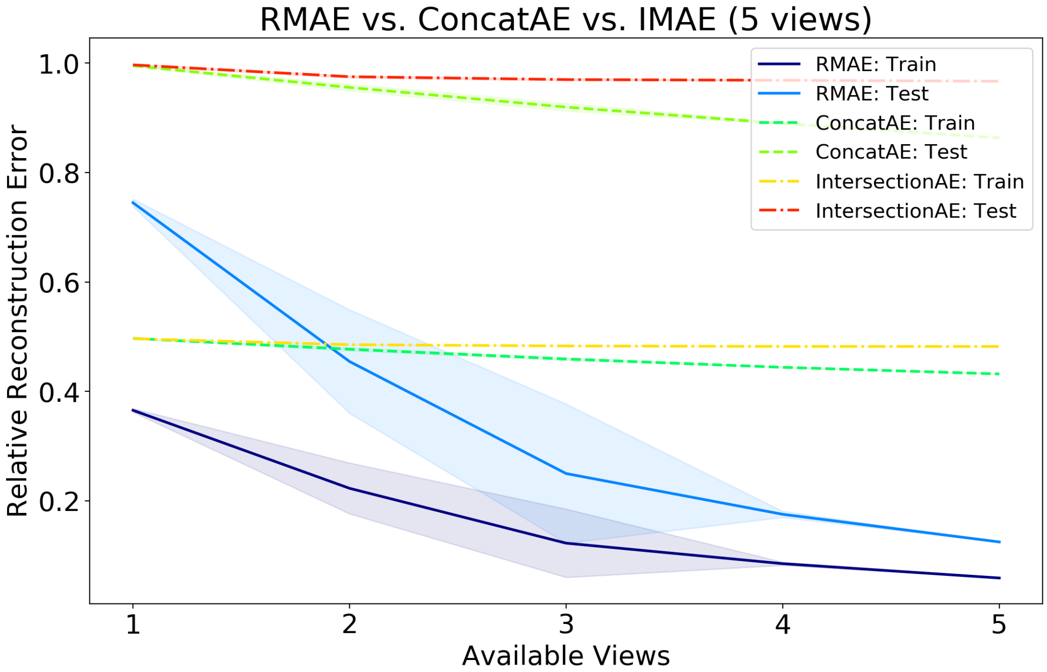


Figure 2.6: [5-view problem] Train and test reconstruction error vs. number of views for different AE competitors. This plot also shows the 1-sigma confidence interval for different choices of available views.

The overall performance of the different methods is as expected. Since IntersectMAE directly learns a bottleneck representation which is common to all views, the training implicitly tries to learn the intersection of information across the different views, as opposed to the union. This is evident from the training vs. test error curves as well as the single-view error matrices. The pattern is the same for a different number of views.

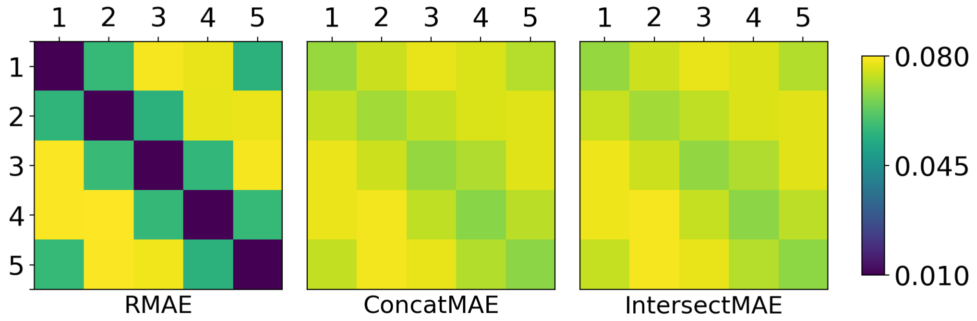


Figure 2.7: [5-view problem] One-to-one single-view reconstruction for different AE competitors.

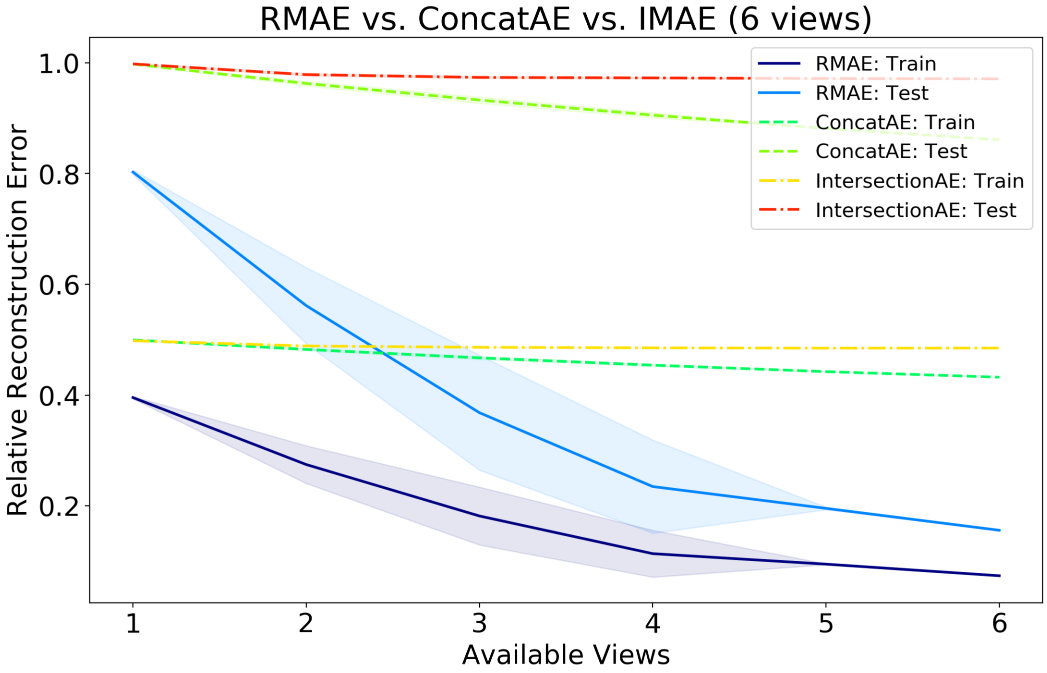


Figure 2.8: [6-view problem] Train and test reconstruction error vs. number of views for different AE competitors. This plot also shows the 1-sigma confidence interval for different choices of available views.

2.4 Real-world Experiments

2.4.1 Datasets

Here, we briefly describe the datasets we consider:

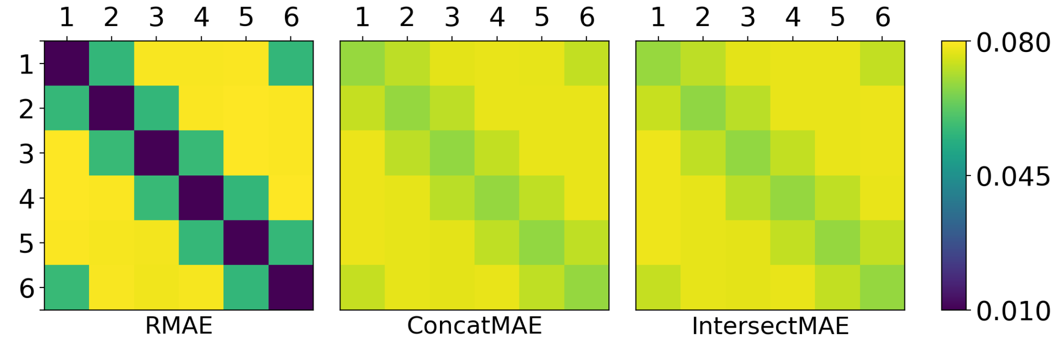


Figure 2.9: [6-view problem] One-to-one single-view reconstruction for different AE competitors.

- *3 Sources News Dataset*[1] This dataset consists of news articles as collected from three well-known news sources: BBC, Reuters and The Guardian. Each view news-source is considered a view, and the articles are samples from them. Each article corresponds to some real world event; every news source that publishes an article on the event has a sample for that data point. Not all sources write articles for every event, so we have missing data from some views.

- *NUS-Wide-Lite* [Chua et al., 2009][2] This dataset consists of various images and 81 categories that they may contain. These categories are objects, people, types of landscape, etc. so multiple such categories can be present in each image. We take a subsampled version of this dataset with 550 training and test images, and look at detecting three different categories: lakes, people and sunsets. Subsampling is done to preserve category prevalance from the main dataset. The views are five different image featurizations: (i) color histogram (CH), (ii) color correlogram (CORR), (iii) edge direction histogram (EDH), (iv) wavelet texture (WT) and (v) block-wise color moments (CM).

- *N-MNIST* [Basu et al., 2017][3] This dataset consists of three noisy versions of the original MNIST dataset. The noise functions are: (i) additive white gaussian noise (AWGN), (ii) motion blur (MB) and (iii) reduced contrast with additive white gaussian noise (RCAWGN). We take a sub-sampled version of the dataset with 6000 training and 600 test points, while preserving digit prevalences.

[1] http://mlg.ucd.ie/datasets/3sources.html

[2] http://mlg.ucd.ie/datasets/3sources.html

[3] https://csc.lsu.edu/ saikat/n-mnist/

2.4.2 Evaluation

Down-stream tasks

Here, we look at the performance of the RMAE as a representation learning method for down-stream tasks. We compare the representation learned by RMAE with those learned by CAT, IMAE and CAE in classification/regression performance.

2.4.3 Results

Here, we show plots for training and testing error curves using the different methods. We notice over the different experiments that there is no clear winning method among the ones tested. But RMAE is always either the best or the second best among the testing errors, when all views are available. RMAE's consistent performance shows that it is able to extract meaningful information about views and their interactions.

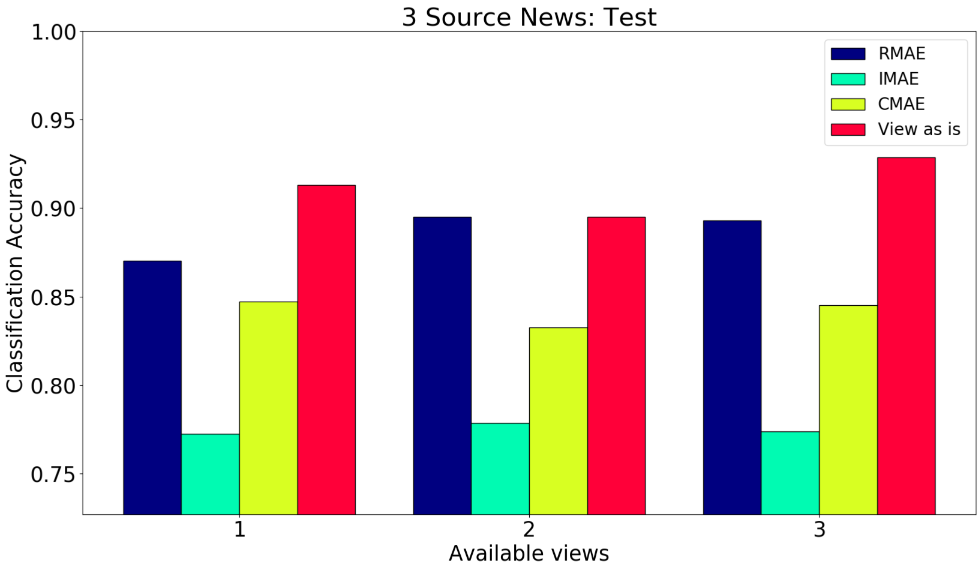


Figure 2.10: [3-Source News] Plots for accuracy vs. number of available views for the different approaches.

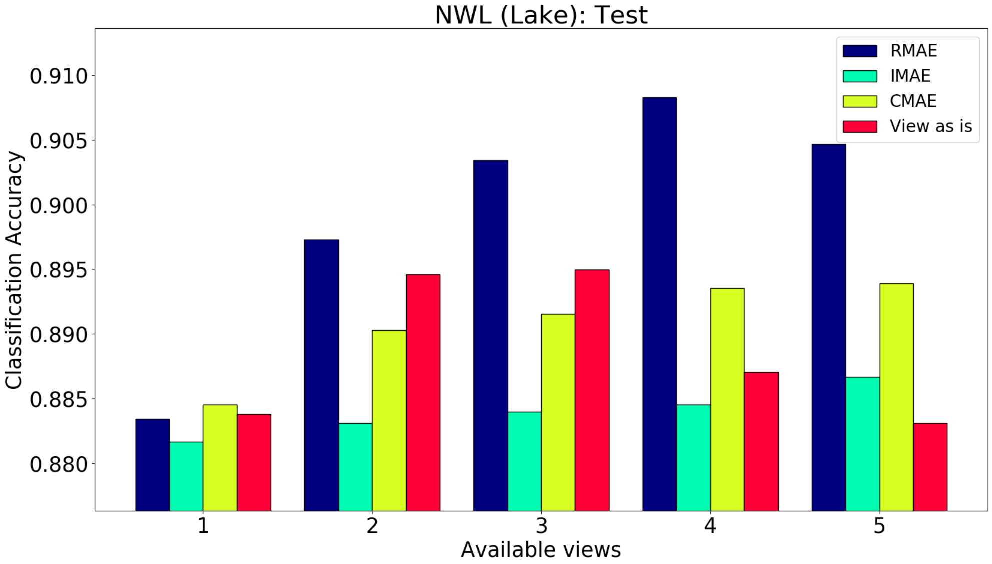


Figure 2.11: [NUS-WIDE-Lite: Lake] Plots for accuracy vs. number of available views for the different approaches.

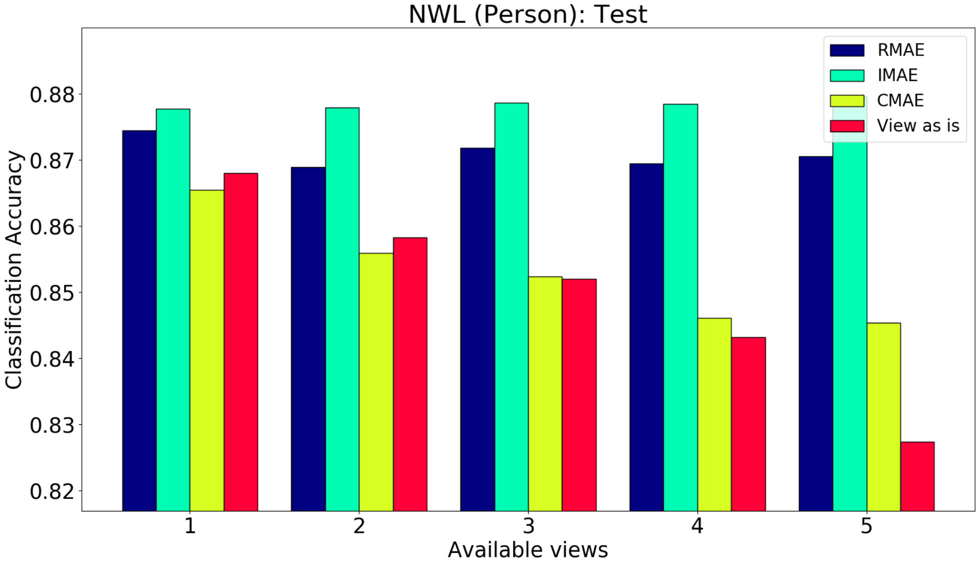


Figure 2.12: [NUS-WIDE-Lite: Person] Plots for accuracy vs. number of available views for the different approaches.

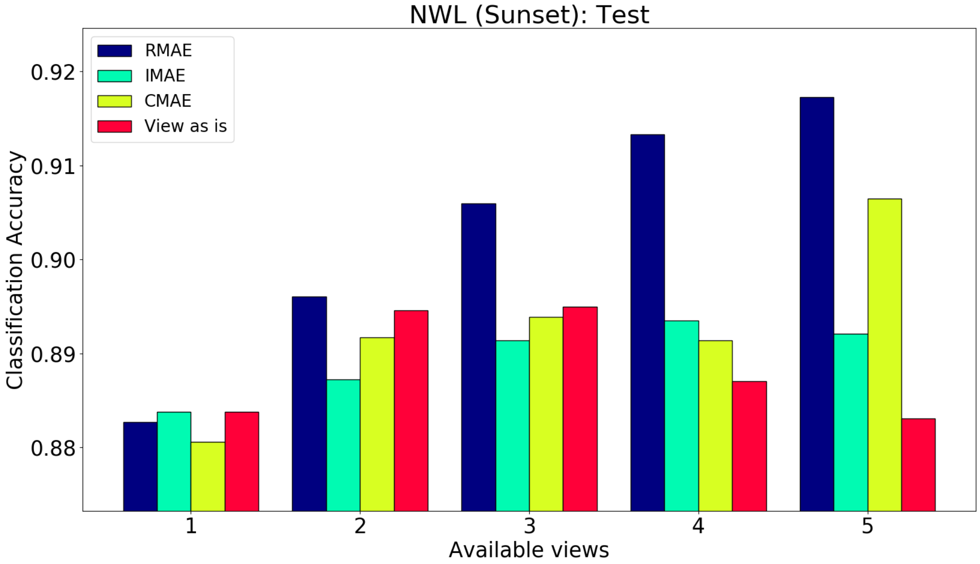


Figure 2.13: [NUS-WIDE-Lite: Sunset] Plots for accuracy vs. number of available views for the different approaches.

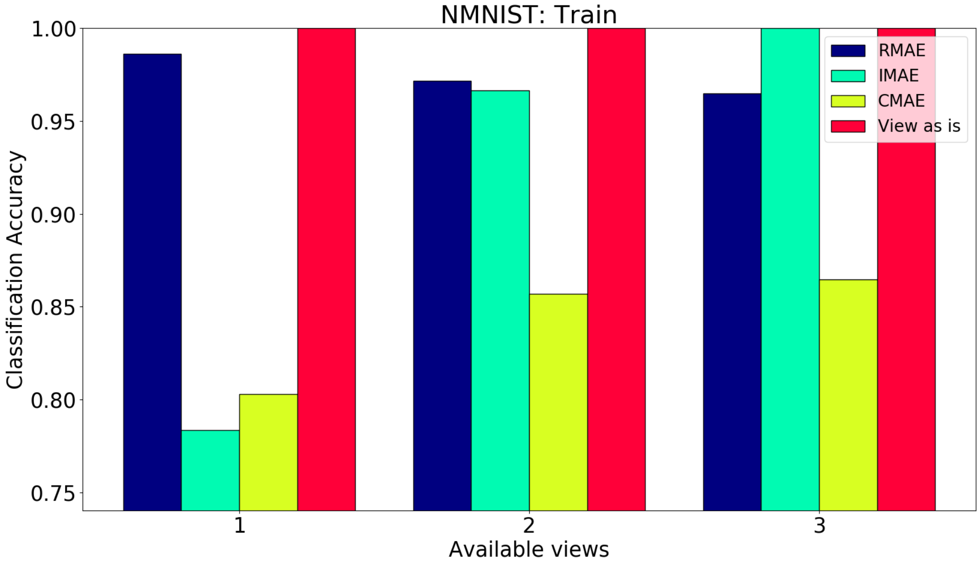


Figure 2.14: [N-MNIST] Plots for accuracy vs. number of available views for the different approaches.

2.5 Conclusion

We have described two approaches for MVRL: (i) Multi-view One-vs-Rest Embedding Learning with group sparse penalization and (ii) Robust Multi-view Auto-Encoder with randomized view-dropout. In both these approaches, we demonstrate the utility of modeling local IVRs; if we have an idea of *where to look* and *what to look for*, we can effectively model IVRs.

MOREL is a simple testament to the validity of our hypothesis, and RMAE is a more practical approach for MVRL which focuses on *robustness* as an important property of our model. Results on synthetic and real world experiments reaffirm the relevance and efficacy of our models.

Chapter 3

Generative Models for Multi-view Representation Learning

3.1 Introduction

So far, we have looked at understanding IVR through discriminative modeling for MVRL. In this chapter, we look at introducing generative modeling into the framework we have been working with. This opens our methods to broader applicability, e.g., sampling missing views, likelihood estimation, etc.

Generative modeling provides us an intuitive perspective to understand multi-view data, i.e. an underlying system with an associated data-generation process. Each view is an observation model which provides a limited window into this process. These observation models can be assumed to be independent of each other given the process; of course, we can never truly observe this. Thus, our goal is to use these observation models to learn the distribution of a proxy for the underlying process, akin to learning a joint distribution over all views.

This is philosophically different from our original approach where we optimized for *reconstruction* of absent view. This indirectly assumes that, in the ideal case, all the views are available; the model is then trained to emulate this ideal case. But this isn't always true – we can have *asynchronous* systems where we can no longer assume that, for any subset of views, there exists a data point containing all of them. For example, health records of patients in different geographical locations are unlikely to have gone to the same hospital, but still may have the same underlying medical concerns. The data from such patients might not have much clear overlap, depending on how different hospitals operate and what equipment they have available. But the underlying data distribution can still be closely related. In such cases, optimizing for "reconstruction" of unavailable views may not the right objective, but learning the underlying data distribution is still relevant.

For our work, we consider flow-based generative modeling. The next sections give a primer on flow-based modeling, followed by its application to our models for MVRL.

3.2 Background: Flow-based Generative Modeling

Flow-based models are based on the philosophy that a *good representation* of data is one in which the data distribution is simple and easily modeled. These approaches ([Dinh et al., 2014], [Dinh et al., 2016]) learn a sequence of invertible transform over the input data to a latent state where the base distribution has a simple form. Samples in the latent space and then be transformed into the data space, as long as we know the formulaic relationship between the two.

For the more technical details, we consider the case of single-view data in this section. Given an invertible transform f and a data point x, the data distribution is related to the distribution in the latent space L using the chain rule as follows:

$$p_X(x) = \left| \mathtt{det} \frac{\partial f(x)}{\partial x} \right| p_L(f(x)) \tag{3.1}$$

Here, $\frac{\partial f(x)}{\partial x}$ is the Jacobian matrix of the transform f, and `det` is the determinant function.

The transform f is typically chosen such that the determinant of the Jacobian is simple to compute; for example, the determinant of an upper triangular Jacobian is just the product of the diagonal values. Even with such restrictions on the transforms, we can still use powerful and expressive function classes (including neural networks) to represent these invertible transforms. Using the fact that the Jacobian of a composition of functions is the product of the Jacobians of the individual functions, f can be represented as a cascade of such transforms, allowing us to build more complicated transforms while maintaining the simplicity of the overall computation.

Given a transform f represented by a composition of T transforms $f^{(1)}$ to $f^{(T)}$, and a base distribution parameterized by θ, the overall loss function then is given by the negative log-likelihood:

$$\mathcal{L}_{nll}(X) = -\sum_{t=1}^{T} \log \left| \mathtt{det} \frac{\partial f^{(t)}}{\partial f^{(t-1)}} \right| - \log p(f(X); \theta) \tag{3.2}$$

Sampling from these models is very straight-forward as long as we can efficiently sample from the base distribution. We just apply the inverse of the learned transform to samples from the base distribution to produce samples in the data distribution. Flow-based approaches model and learn these transforms ([Dinh et al., 2014], [Dinh et al., 2016]), and sometimes both the transforms and the base distributions [Oliva et al., 2018].

3.2.1 Flow-Transforms

In this section, we will give an overview of some of the common flow-transforms [Dinh et al., 2014], [Dinh et al., 2016] and [Oliva et al., 2018], which we will adapt for our methods.

- *Fixed Linear Transform*
  This is a transform in the form of an invertible square matrix M. Inverting an arbitrary matrix is not straightforward, so M is not represented directly. Instead, it can represented by its LU decomposition, i.e. $M = LU$. Inverting M then involves just solving a triangular system of equations. The determinant of the Jacobian is also easily computed for triangular matrices.

- *Coupling Transforms*
  Coupling transforms split the input covariates into two subsets, and transform one of the subsets using a function parameterized by the other subset [Dinh et al., 2014], [Dinh et al., 2016]. Given a D dimensional covariate x, without loss of generality, the input can be split as $x = [x_{1:d}; x_{d+1:D}]$. Then, the transform can be represented as follows:

$$\begin{aligned} z_{1:d} &= x_{1:d} \\ z_{d+1:D} &= f(x_{d+1:D}; x_{1:d}) \end{aligned} \tag{3.3}$$

  With f being invertible, computing the inverse of the coupling is easy. Further, the Jacobian is block triangular with one of the diagonal blocks being the identity, and the other corresponding to the Jacobian of f.

  In our models, we mainly consider an affine scale-and-shift coupling transform:

$$\begin{aligned} z_{1:d} &= x_{1:d} \\ z_{d+1:D} &= x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}) \end{aligned} \tag{3.4}$$

  where $\odot$ is the element-wise product s and t are functions which map from $\mathcal{R}^d \to \mathcal{R}^{D-d}$. These can be parameterized by neural networks.

- *Misc. transforms*
  We also consider a few simple invertible transforms without learnable parameters like LeakyReLU, the reverse transform (just inverting the indices of the input), elementwise sigmoid and logistic functions. The inverses and determinants of the Jacobians of these transforms are simple to compute.

3.2.2 Base Distributions

These are distributions of the latent space; the choices are usually those which are simple to model and efficient to sample, e.g., multi-variate Gaussians (with diagonal covariance) They can also be learned along with the transforms, e.g., Auto-Regressive Gaussian mixture models [Oliva et al., 2018]. The parameters for the mixture models are usually given by fully connected neural networks over the transformed latent variables.

3.3 Flow-based Multi-view Representation Learning

Now, we move onto the application of flow-based models to multi-view data. We adopt the same two-tier framework as in RMAE from Chapter 2 (Figure 3.1), where we have view-specific as well as shared encoders. We will start with the simple case where we have all views available at train time, and then move on the more general case where we have missing views.

3.3.1 Case 1: All views available

To start with, we set the notion of *robustness* aside and assume all views are available at train time. Our goal here would then be to just learn a joint distribution over these views. As we described before, flow-based models learn a cascading sequence of invertible transforms into a latent space with a “simple” distribution. The pipeline for this just replaces the shared encoder from Figure 3.1 with flow based transforms. The view-specific encoders may or may not be flow-based.

3.3.2 Case 2: Missing views at training/test time

Here, we consider the more general case where we can have an arbitrary subset of missing views. For this, we propose a multi-view extension of ACFlow [Li et al., 2019], which looks at the single-view case of flow-based data imputation when an arbitrary

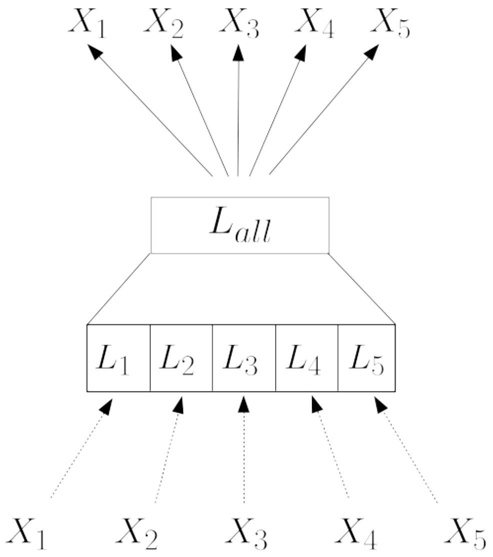


Figure 3.1: Two-tiered structure for the RMAE from Chapter 2.

subset of covariates are missing. The idea is to learn *conditional* flow-transforms where the parameters of the transforms are functions of the available covariates.

In our case, we have the advantage of having some structure in the missing covariates: we assume that views go missing as a whole. I.e. if a view is available, all of its covariates are available and if a view is missing, none of the covariates are available.

Conditional Flow-Transforms

We follow [Li et al., 2019] and propose conditional variants of the transforms described in the previous section. Conditioning here involves parameterizing the transforms using the available views. Let us represent the available/observed views as X_o, and the missing/unobserved views as X_u The flow-based transforms themselves are the same as before, but the parameters which define them depend on the conditioning variables. The fixed linear transform is a straightforward implementation of this; the values of the L and U matrices are now outputs of a neural network instead of directly being optimized over.

The coupling transforms are more involved. Consider a transform operating over a single view x_v, conditioned on X_o. Following Equation 3.3, we split x_v into two subsets $x_v = [x_{1:d}; x_{d+1:D}]$. Then, can represent the coupling transforms in two ways.

- *Meta parameterization:* We take $z_{d+1:D} = f(x_{d+1:D}; g(x_{1:d}; X_o))$, where g produces "meta" parameters of the coupling function f given the observed views. This involves two steps, (i) computing the parameters for and applying g, and (ii) the same for f.

- *Direct parameterization:* We take $z_{d+1:D} = f(x_{d+1:D}; g(x_{1:d}, X_o))$, where g takes both X_o and the first split and produces the parameters for the coupling function f. In this case, the two steps are conflated into one, since g is directly applied on both the initial split and X_o.

The former is more general and has more representation power, but is computationally very expensive. If f and g are represented by neural networks, the "meta"-network g would need to be computed for every single data point, since X_o would not be the same for different data points. This would mean that an individual neural network must be created an applied for each individual data point.

The latter is a lot less expensive since it is an extension of the non-conditional version where the input is a simple function of both X_o and $x_{1:d}$ (e.g., concatenation). This reduces the flexibility of the learned transforms but eases the computational load significantly.

Base Distributions

Here, we have similar choices for our base distribution. We can use a simple parameterless distribution, or a parameterized AR mixture model. The parameters of the mixture model can depend on both X_o and the latent variable z_L.

Pipeline

Again, we facilitate *robustness* through arbitrary conditioning, by using the same training strategy as in RMAE – we randomly drop subsets of views during training while still trying to model the entire joint distribution. Now, the dimensionality of X_o depends on the views which are missing, which is not known ahead of time. [Li et al., 2019] deal with this by zero-imputing the missing data, as well as adding a bit-flag of the same size as the data that represents whether any given covariate is available or not. This produces a fixed dimensional featurization which can be used as inputs to the function which produces the parameters for the flow transform. The flow-transform is then constructed from the subset of these produced parameters which correspond to the unavailable covariates.

In our case, we can exploit the view-structure of data to circumvent the arbitrary dimensionality of X_u. Since we assume that any given view is either entirely available or entirely missing, any view i is either in X_o or X_u, but not both. This allows us to learn conditional transforms for each view within X_u individually. The conditioning over X_o still requires us to deal with the arbitrary dimensionality of the observed views. We use the same zero-imputation and bit-flag introduction strategy as in ACFlow.

Our overall pipeline then consists of individual view-based transforms/encoders, followed by individual conditional transforms for each view. For our conditional coupling transforms, we use the *direct parameterization* approach. We call our proposed method Multi-view AC Flow (MACF).

Characterizing IVRs

Once again, we summarize MACF as a method for MVRL by considering the *where* and *what* of IVR. Similar to RMAE, MACF implicitly searches over different subsets of views. But here, it looks for characterizating joint likelihood estimates through the information overlap a given subset of views.

Note: The requirement of invertible transforms implies that the latent space and the data have the same dimension. For multi-view data, we can use the view-specific encodings to perform dimensionality reduction if needed.

3.3.3 Experiments

For evaluating MACF, we use a multi-view version of the original MNIST dataset: we split each image into four quadrants to represent the views. Our dataset consists of a small subset of MNIST digits, with about 1500 training digits, stratified over the different digit classes.

We encode each view by first reducing its dimension using a truncated SVD: we keep as the dimensions with singular value over 5% of the largest singular value. This comes up to about 50 dimensions per view. For the common flow-based transform, we use three scale-shift coupling segments, interleaved by reversal transforms. A *segment* here corresponds to four individual scale-shift coupling transforms in a row, where we alternate the fixed and transformed covariates in each. We use a standard multi-variate normal for our base distribution.

Digit Completion

We look at digit completion/sampling under different numbers of missing views for our model. In Figures 3.2, 3.4 and 3.6, we show samples of test digits given various numbers and combinations of available views, along with the true digit at the top. The availability pattern is depicted to the left of each row, where green means available.

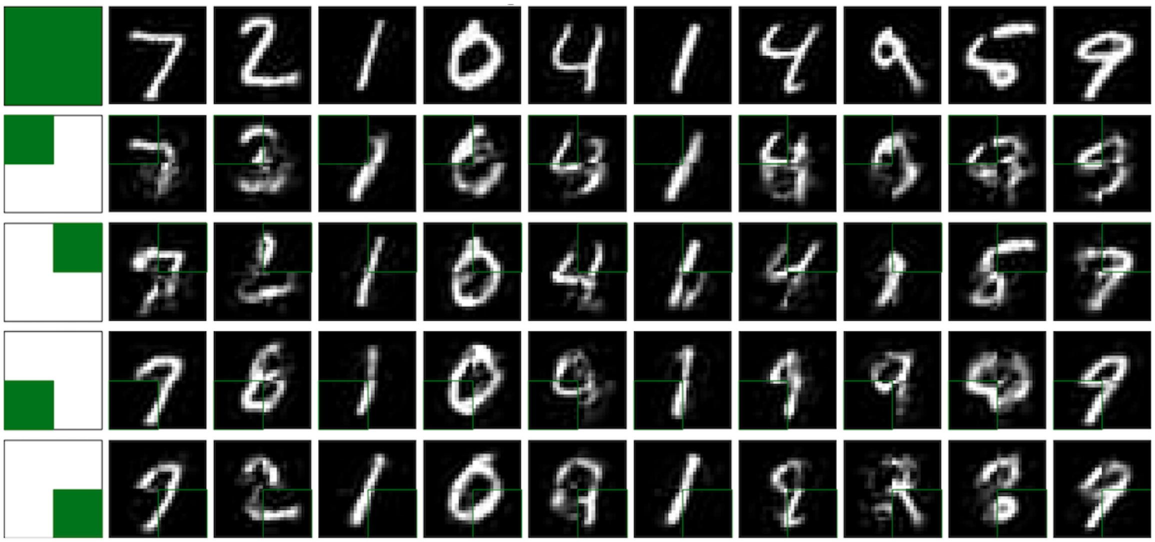


Figure 3.2: Sampling whole digits given one view.

We also evaluate our method using the McNemar test against the following: (i) the partial input digits with missing views zero-imputed (baseline), and (ii) the ground-truth completed digit. We used four views for this experiment. The Figures 3.3, 3.5, and 3.7 show the results of the test across different sets of available views. Column 1 corresponds to points that the reference model correctly classifies. Column 2 corresponds to points that the reference model gets wrong. Similarly, Row 1 and Row 2 correspond to our model correctly and incorrectly classifying the point, respectively. The blue (top left of every box) corresponds to the ground truth model, and the red corresponds to the baseline. Each 2x2 matrix thus shows McNemar Test results for both the reference models.

In general, as the number of views increases, the digits sampled become more reliable in their completion of the true digit. Failure cases are often graceful where the sampled digit looks reasonable, even if different from the true digit.

Sampling complete digits from a single instance

In this experiment, we took a single input digit (with missing views) and sampled multiple complete digits from the learned distribution. For this, we considered 3-view horizontal splits of the MNIST digits, and sampled the middle given the top and the bottom third. The remaining setup is the same as before, i.e. the same strategy for view encoders, the shared flow-transform, etc.

The inputs were two randomly chosen digits from the test set – 0 and 7. The samples are shown in Figures 3.8 and 3.9. Here, we often observe what we call a

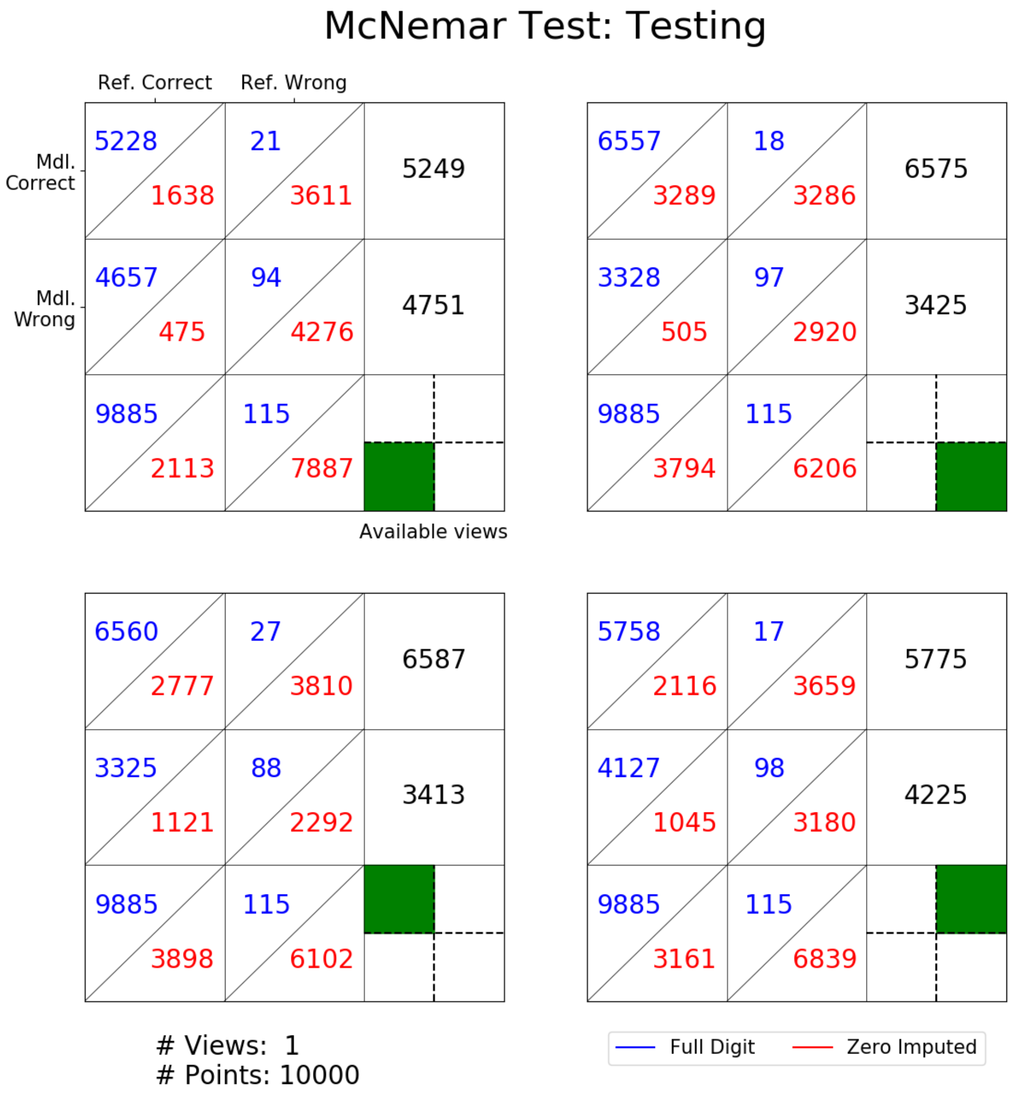


Figure 3.3: McNemar evaluation of benchmark MNIST classifier on Test vs. ground truth and zero imputed with 1-view available.

graceful failure, where the completed digit is not the same as the input, but is a reasonable completion given the limited available views; e.g., 9 for 7, 8/6 for 0.

3.4 Conclusion

In this chapter, we looked at a generative modeling extension of our RMAE using flow-based methods: Multi-view AC Flow (MACF). We evaluated MACF on MNIST digits, split into multiple views, on digit classification and digit completion.

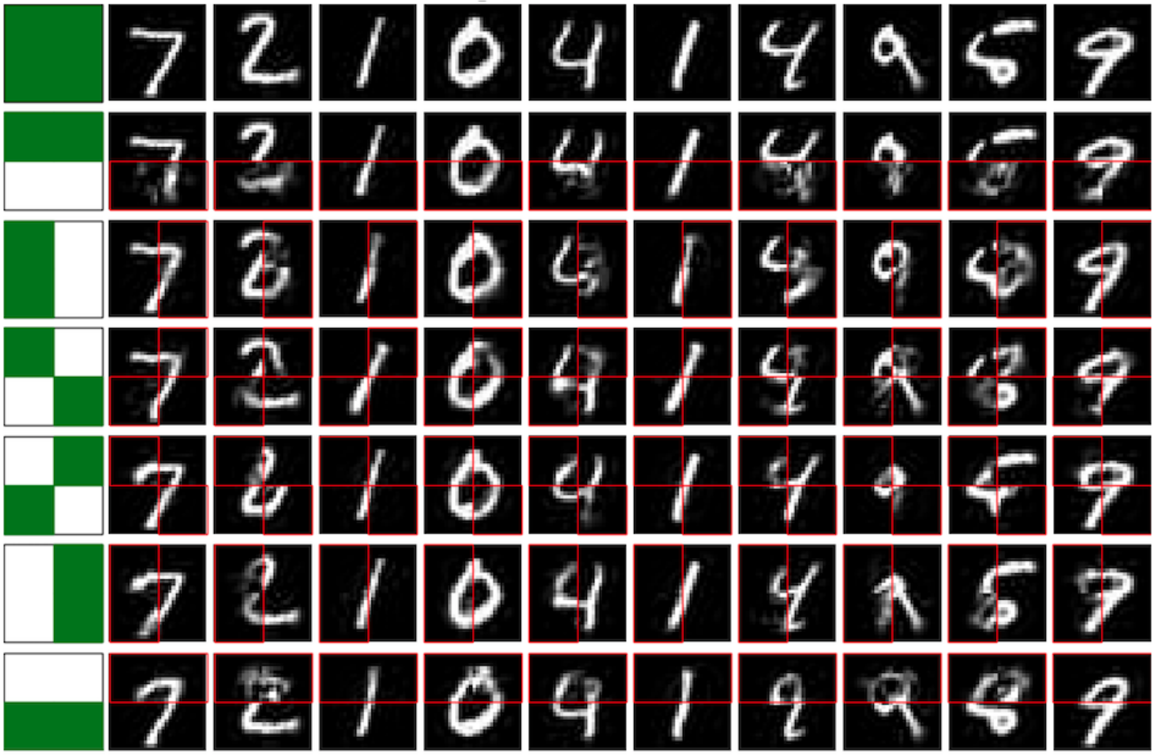


Figure 3.4: Sampling whole digits given two views.

Our results show the representative power of the MACF; producing reasonable sampled completions for missing views, and often failing gracefully when the completion doesn't match the original digit.

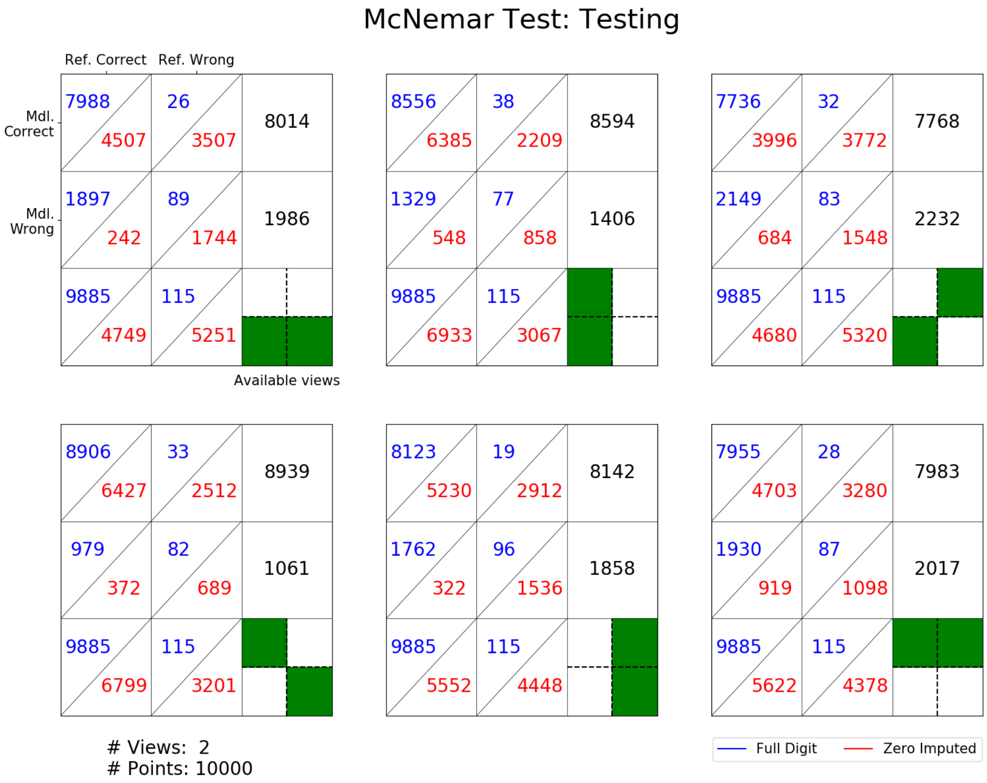


Figure 3.5: McNemar evaluation of benchmark MNIST classifier on Test vs. ground truth and zero imputed with 2-views available.

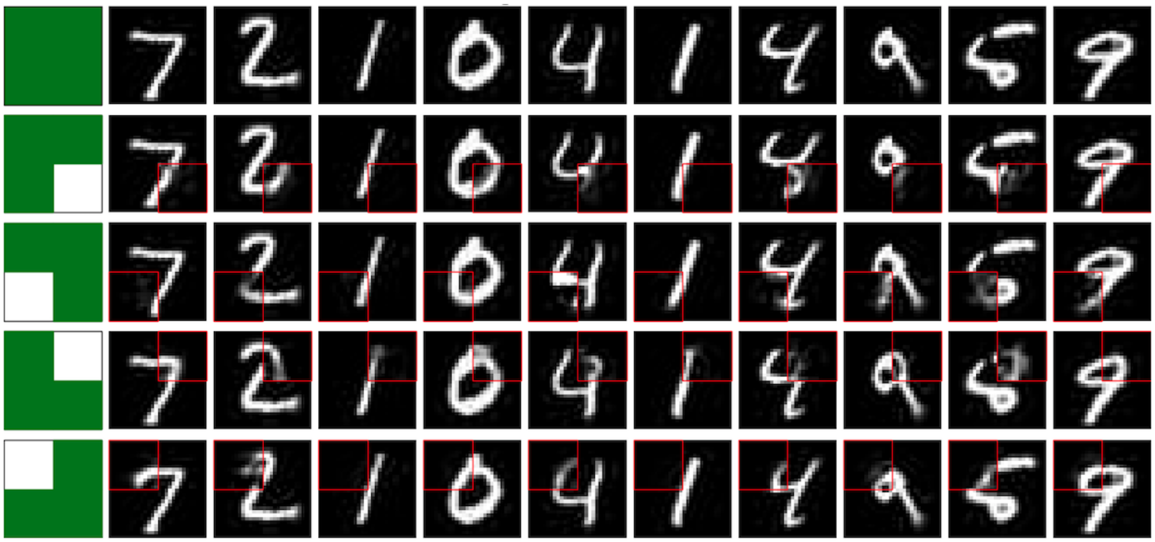


Figure 3.6: Sampling whole digits given three views.

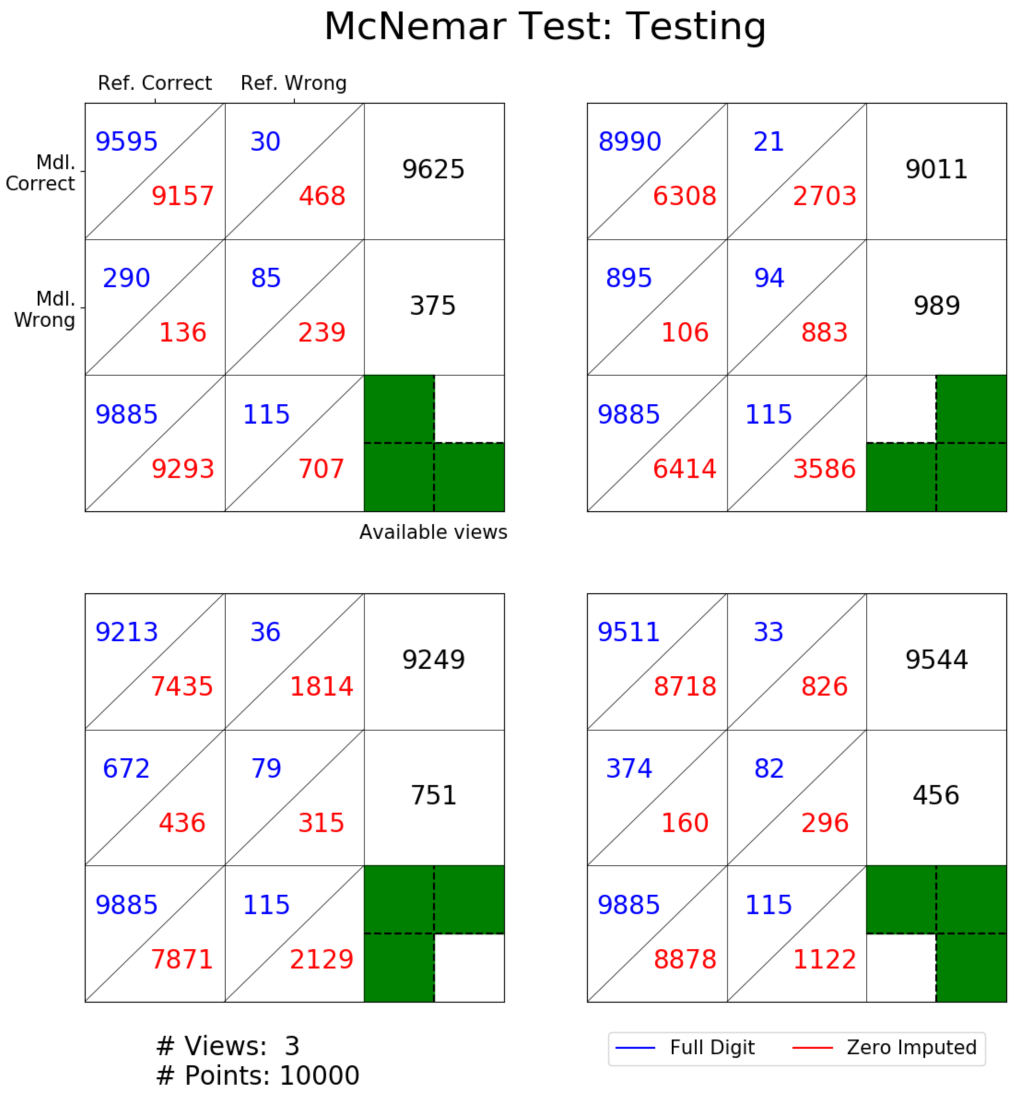


Figure 3.7: McNemar evaluation of benchmark MNIST classifier on Test vs. ground truth and zero imputed with 3-views available.

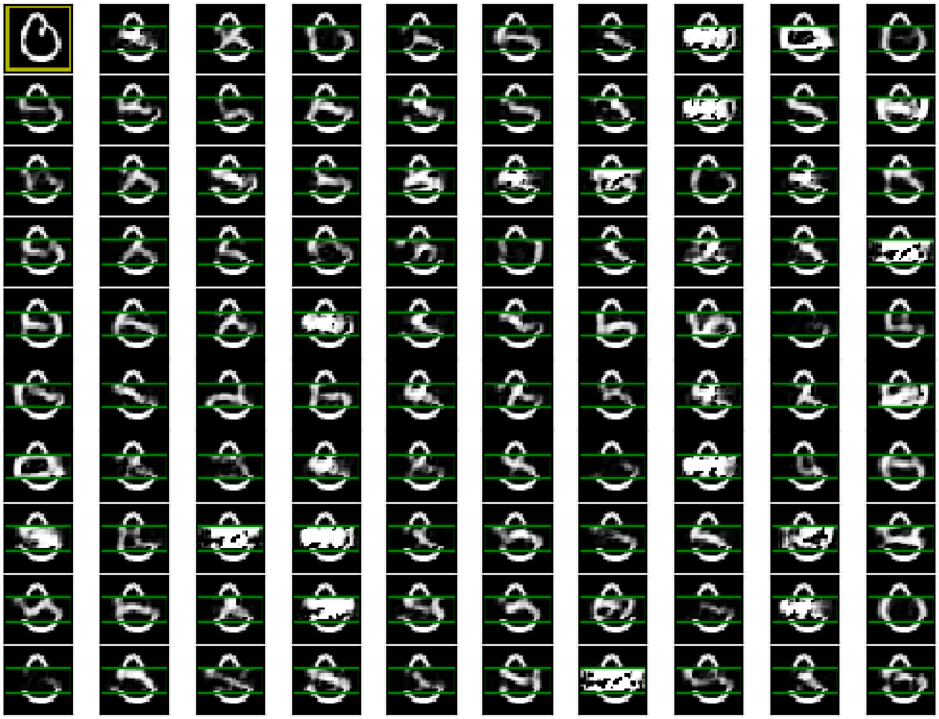


Figure 3.8: Multiple samples of the middle third given the same digit 0 as input. Top left digit is the original.

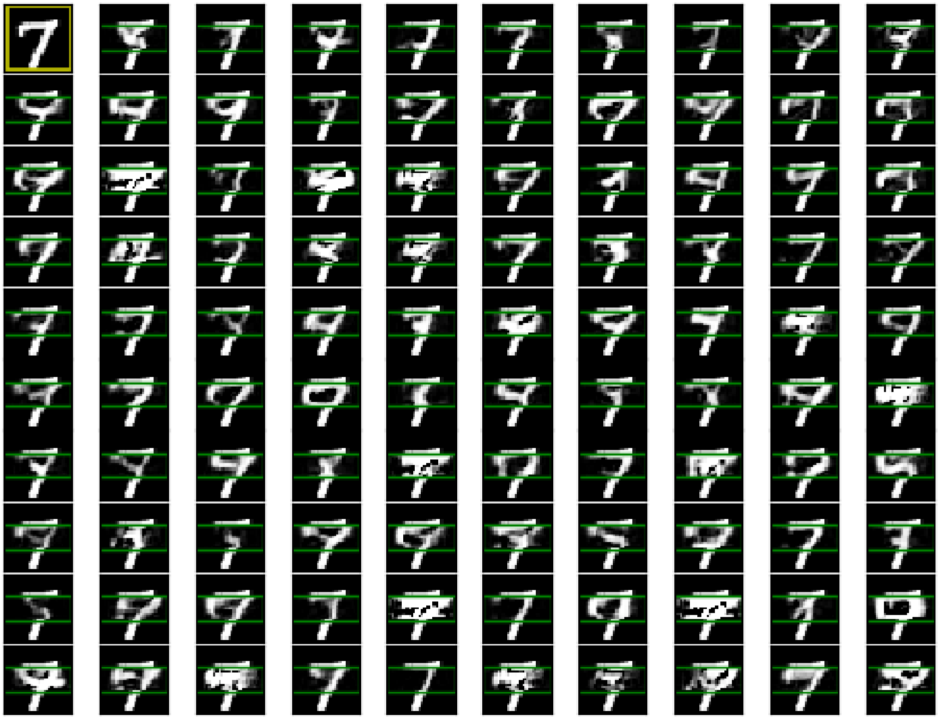


Figure 3.9: Multiple samples of the middle third given the same digit 7 as input. Top left digit is the original.

Part II

Exploiting Inter-view Relationships

Chapter 4

MVRL for Down-stream Tasks and Application Domains

4.1 Introduction

So far, our proposed MVRL methods have been agnostic to any down-stream task that could leverage their learned representations. Since our *task-agnostic* MVRL methods operate on unsupervised proxy-tasks such as reconstruction and likelihood maximization, they can always be applied to general multi-view data-sets. Indeed, they can be used as meta-learning approaches, which are modular and straight-forward to use right off the shelf. This flexibility comes at the cost of the application-context-oriented specializability. Their application to a given task carries the implicit assumption that the representations they have already learned are relevant for the given downstream task.

In this chapter, we look at extensions of our MVRL modeling which introduce context-awareness in a given application or domain. In other words, we take a look at the other side of the flexibility-vs-specificity trade-off. We can consider two avenues to approach this:

- **Task-adaptive**

  One concept is to tailor our models to the given down-stream application, e.g., classification, density estimation, etc. Here, we essentially have information in the form of a down-stream loss function to contend with. This would allow us to directly influence the representation learning process, guiding it towards a latent space more conducive to tackling that specific down-stream task. As a consequence, our MVRL approaches will no longer qualify as meta-learning approaches.

- **Data-adaptive**

> The other concept is to adapt our models to the domain of data. For instance, image or text data have a plethora of available encoding methods which we can plug-in directly into our models. As such, our models already allow us to gracefully introduce this form of adaptivity, since the choice of individual view encoders is not constrained.
>
> We might also have to change the framework itself for special cases such as *asynchronous* data, where we assume little-to-no correspondences between some sets of views. For example, consider the case of A/B testing for an update to a website. Any given user is only exposed to either A or B, not both; if we try to featurize browsing patterns on a given version (A or B) as views, there will be little to no correspondence in the data for the views. The idea of view-dropout cannot be directly applied to this case, which has been a crucial cog in many of our methods, so this would call for a redesign of the training framework.

In our research, we look at *task-adaptive* models. As noted before, our models tend to be implicitly *data-adaptive*, insofar as we maintain our original framework. We first look at augmenting the MVRL loss function with the task-specific loss. Following this, we consider applications with image and temporal data to evaluate our task-specific learning approach.

4.2 Task-Adaptative MVRL

Task-adaptation essentially introduces supervision into representation learning by coupling it with task-specific information. The most straightforward way to connect an application to our learning models is to explicitly incorporate the down-stream loss (DSL) into the MVRL optimization.

We do this by simply adding the DSL to the negative-log-likelihood (NLL) loss of our MACF procedure, with a tradeoff parameter to control its influence in training. The only constraint is that we must be able to acquire gradients from such a model, so we can apply standard optimization procedures such as Stochastic Gradient Descent (SGD). Augmenting the loss in Equation 3.2, our new loss becomes the following:

$$\mathcal{L}(X,Y) = \mathcal{L}_{nll}(X) + \gamma\mathcal{L}_{dsl}(X,Y) \tag{4.1}$$

where $\mathcal{L}_{dsl}$ is the DSL, Y is additional information (e.g., labels) for the task and γ is the trade-off parameter.

The DSL could involve a separate pre-trained model as in the case of our digit detection, or a trainable model which is optimized simultaneously with the representation learning. Our evaluation only uses pre-trained or benchmark down-stream models for the DSL; the latter is more prone to overfitting on the training data, unless careful protective measures are put in place.

In the following section, we consider the case where we have the labels available at training time but not at test time.

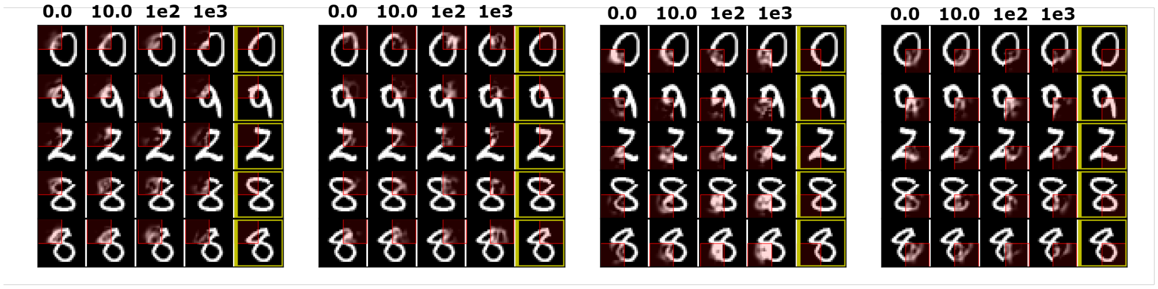


Figure 4.1: 1-view available: Digit completion with 1-view available, given different DSL trade-off coefficients.

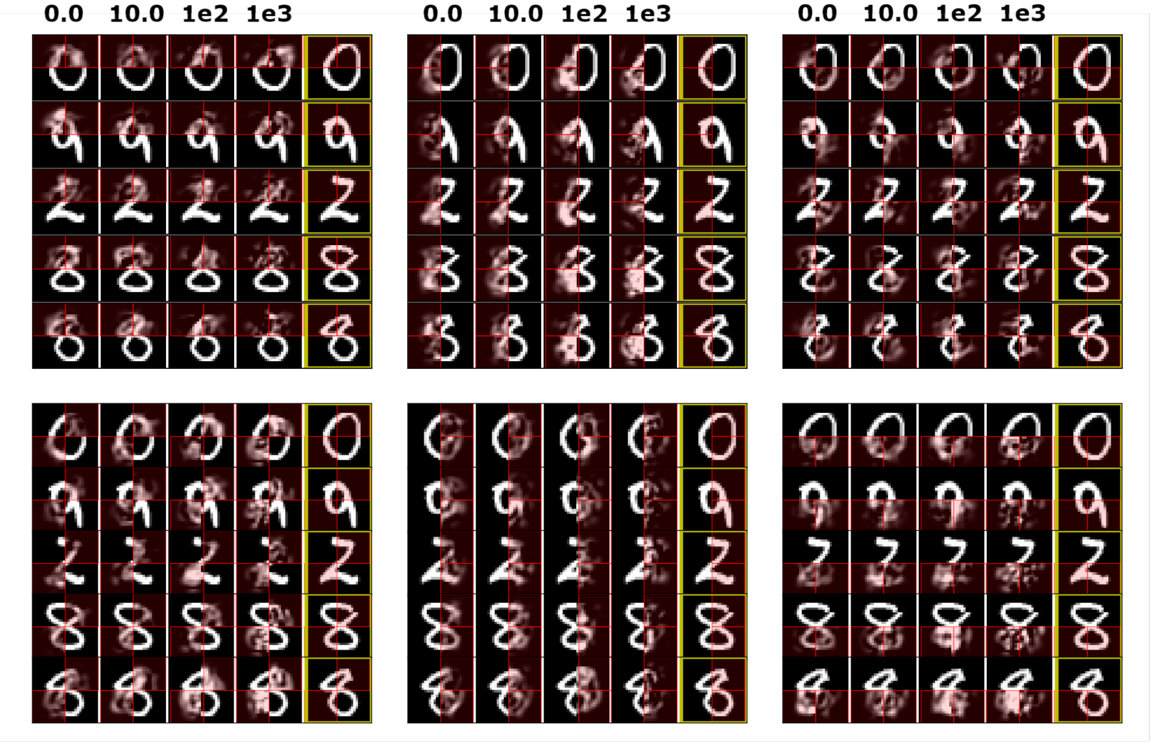


Figure 4.2: 2-views available: Digit completion with 2-views available, given different DSL trade-off coefficients.

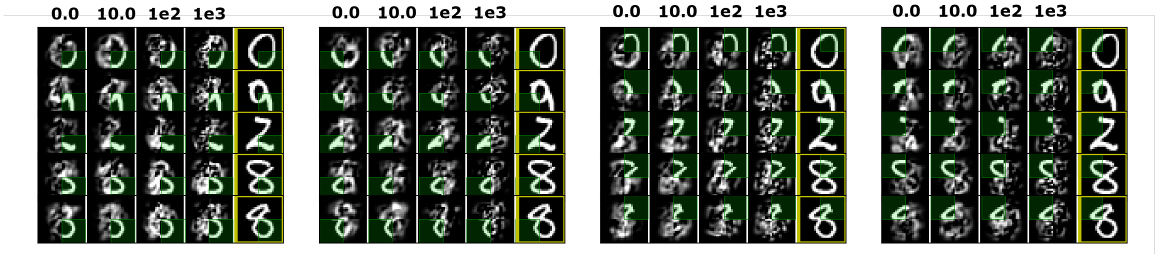


Figure 4.3: 3-views available: Digit completion with 3-view available, given different DSL trade-off coefficients.

4.2.1 Experiment: MNIST Classification

In this section, we look at including the classification loss from an off-the-shelf MNIST classifier into our overall loss function. The classifier we use here is the same from the ONNX zoo we used in the previous chapters. Since the classifier itself has been optimized for MNIST, we freeze the parameters of the model, and just use the gradients produced by the classifier for the parameterization of our original framework. Every batch, we sample and complete the digits, and feed them to the benchmark classifier, which gives us logits as output. Given these logits, we use the DSL as the cross-entropy loss for classification. We evaluate performance on the down-stream task against different trade-off parameters: $\gamma = 0.0, 10.0, 100.0, 10^3, 10^4$.

In Figures 4.1, 4.2 and 4.3, we look at digit completion of the same digits over all the values for γ. We notice that as γ increases, the digit gets progressively more garbled. In general, $\gamma = 0.0$ and 10.0 seem to produce the most visually reasonable digits for various numbers of views available. But, we see from Figures 4.4 and 4.5, that benchmark classification accuracy goes *up* with γ, even though the sampled digits appear to look more garbled to a human observer. But, the appearance of the digits reconstructed from partial views is not as relevant in this context as the observed boost of the downstream digit recognition performance.

Discussion

Often, we observe an inconsistency between a visual evaluation of the digits and the benchmark classifier accuracy. This shows that including the DSL in our training procedure indeed helps the downstream task of digit recognition, but in some application contexts it might not actually be what we want. Our method, however, certainly is doing what it is supposed to do: optimize for the DSL as a part of the overall training procedure.

Alas, results of using DSL might not always be conducive of what constitutes a "good" reconstructed digit to a human observer. This is especially evident when

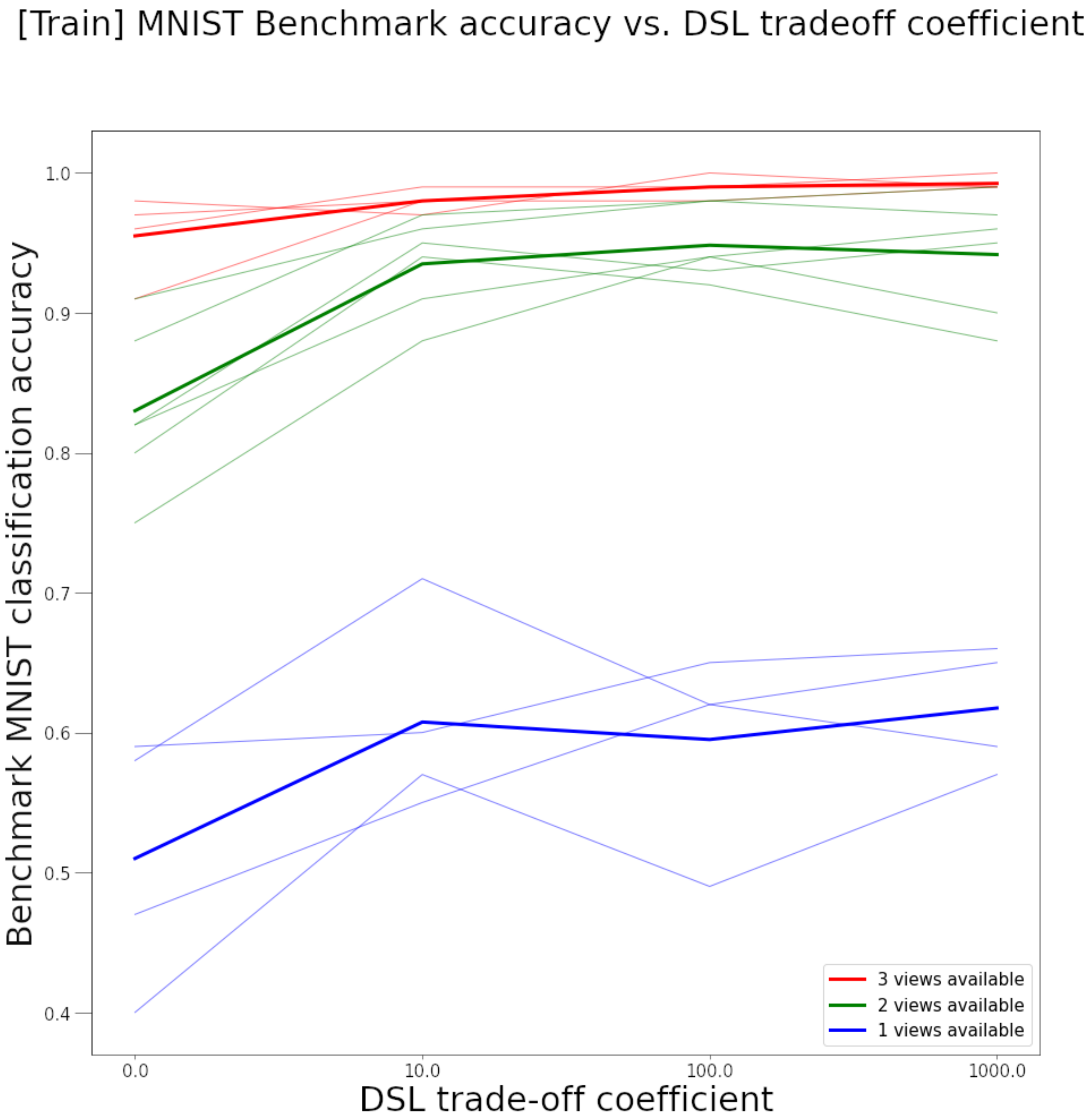


Figure 4.4: [Train] Benchmark classification error vs. DSL trade-off coefficients. The light lines are plots for different view-subsets of a given number of views. The bold lines reflect averages over a given number of views.

we have very few views available to attempt the reconstruction. With very limited information, we can have multiple, reasonable digit completions which make sense to us as humans; even if the produced digit does not visually match the intended label, these “failures” can be graceful. But the inclusion of the DSL essentially pigeonholes us into a specific digit, even when a single quadrant of a digit barely constrains the viable completion possibilities. This way, the MVRL approach does not necessarily

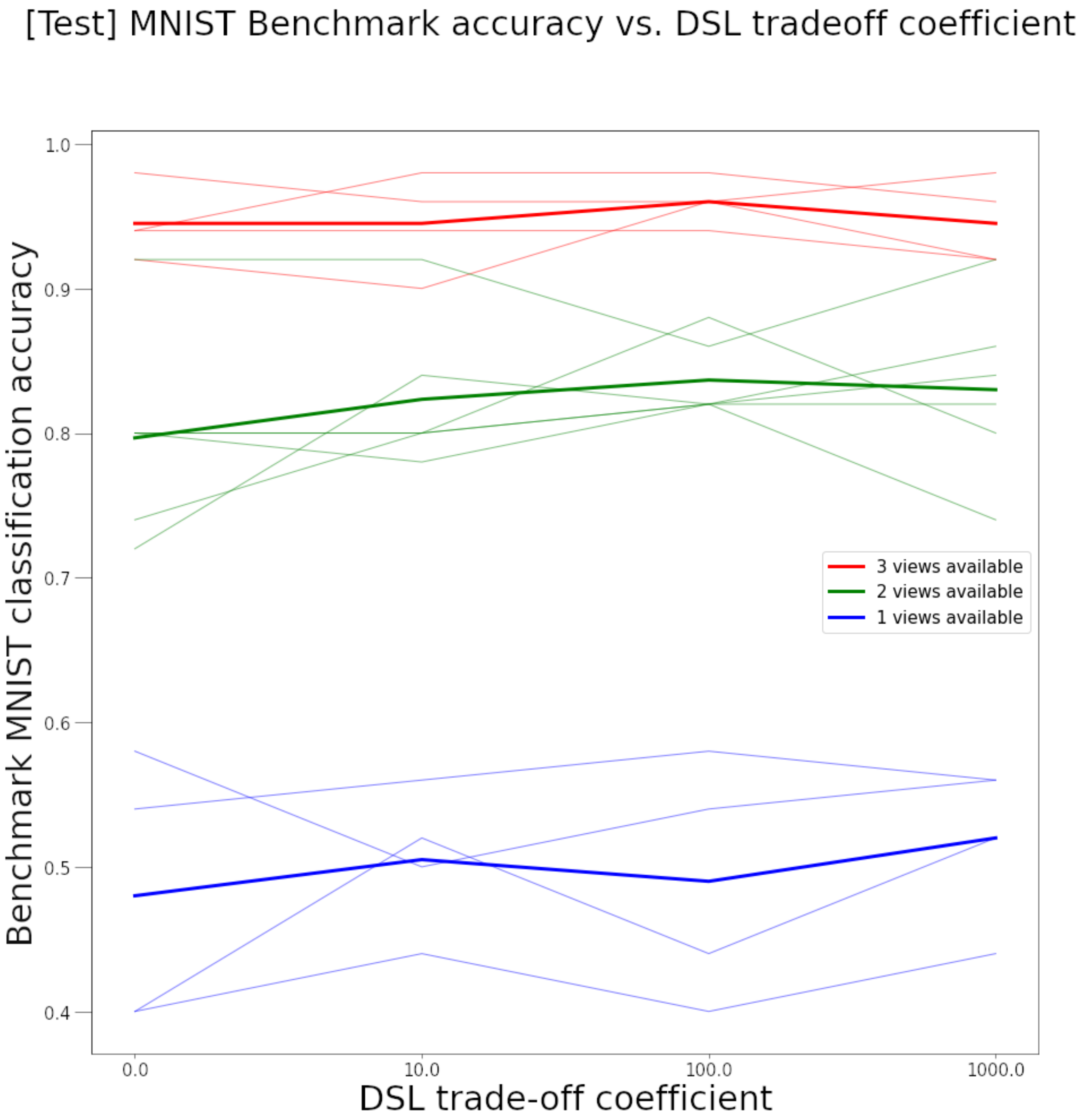


Figure 4.5: [Test] Benchmark classification error vs. DSL trade-off coefficients. The light lines are plots for different view-subsets of a given number of views. The bold lines reflect averages over a given number of views.

learn what a good digit looks like. Rather, it learns how to support the benchmark classifier, by considering the classifier's evaluation criterion.

The lesson learned here is that we need to be wary of how exactly we want to influence the representation learned. We can possibly aid the particular case above by first estimating likelihoods of different digits given the available views, and then choosing one such digit as the candidate for digit completion. Of course, this solution is specific to the problem at hand; domain expertise might be imperative in formulating a good DSL as well as an augmented training procedure for a given task.

4.3 Application Domain: Medical Temporal Data

In this section, we take a look at applying our methods to a specific application with temporal data. For this, we look at the following medical dataset:

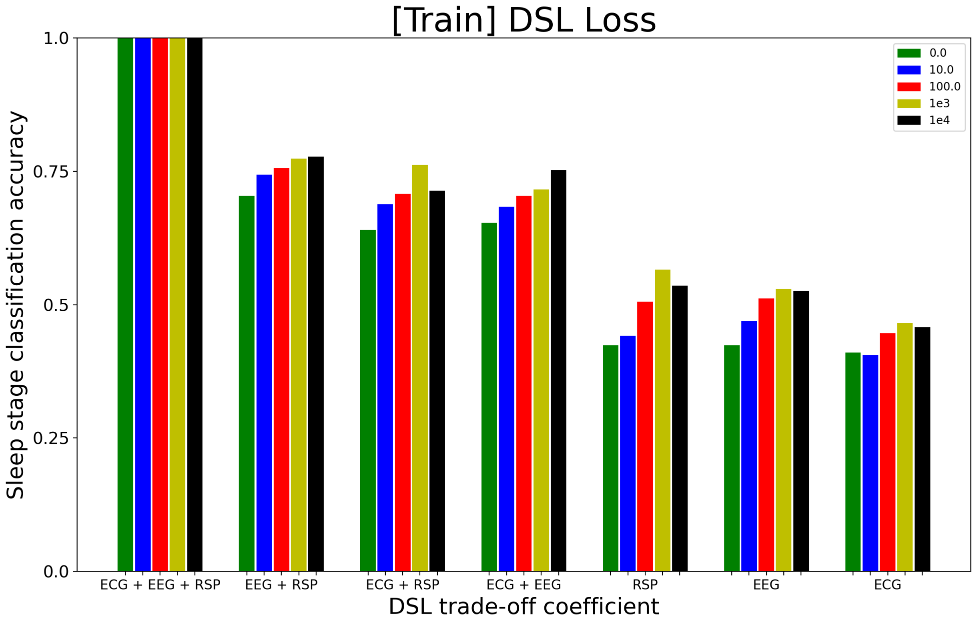


Figure 4.6: [Train] Sleep stage classification vs. available views using pre-trained DSL, given different DSL trade-off coefficients.

MIT-BIH Polysomnographic Database This dataset consists of the recording of multiple physiological signals from individuals during sleep. Over 80 hours of physiological data has been collected from over 16 male subjects, between 32 and 56 years of age. These signals include ECG, EEG, blood pressure and respiratory signals among others, sampled at 250Hz. Each record is annotated with labels denoting stages of sleep and apnea, once every 30s through the duration of recording.

We take the three signals which are available for all subjects as our three views: View 0: EEG, View 1: ECG, View 2: Resp (nasal). For each of these signals, we

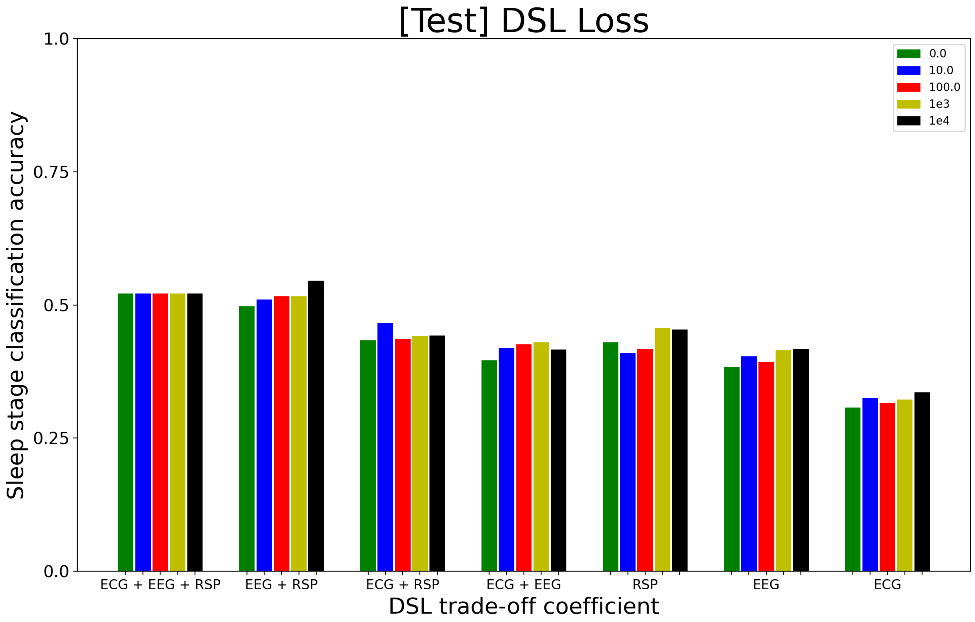


Figure 4.7: [Test] Sleep stage classification vs. available views using pre-trained DSL, given different DSL trade-off coefficients.

extract a wide range of time-series features using the `tsfresh` Python library, followed by dimensionality reduction to go down to 30 dimensions per view.

There are six sleep-related labels we consider in this data-set: Awake, Sleep stages 1-4 and REM sleep. The annotations provided for the subjects are recorded every 30s, so we have taken 30s time-windows to represent individual data-points from the different views. We use a train-test split of approximately 85%-15%, as split over the subjects. Similar to the MNIST experiments, we evaluate classification performance with the MACF+DSL trained over different trade-off coefficients: $\gamma = 0.0, 10.0, 100.0, 10^3, 10^4$.

Unlike the MNIST data we looked at before, this dataset does not have a publicly available benchmark classifier to use for computing DSL gradients. So, we pre-train a fully connected neural network for sleep-classification as the DSL, and freeze its parameters for the MVRL training loop.

We evaluate the classification performance using three classifiers: a separate multi-layer perceptron (MLP) different from the DSL, a random forest (RF) classifier and the original DSL itself. The first two classifiers are trained using grid search over tunable parameters (layer units for MLP, max depth and max features for RF) + 5-fold cross validation stratified by subject. Figures 4.6 and 4.7 show the train and test classification performance for the DSL classifier over different available views and

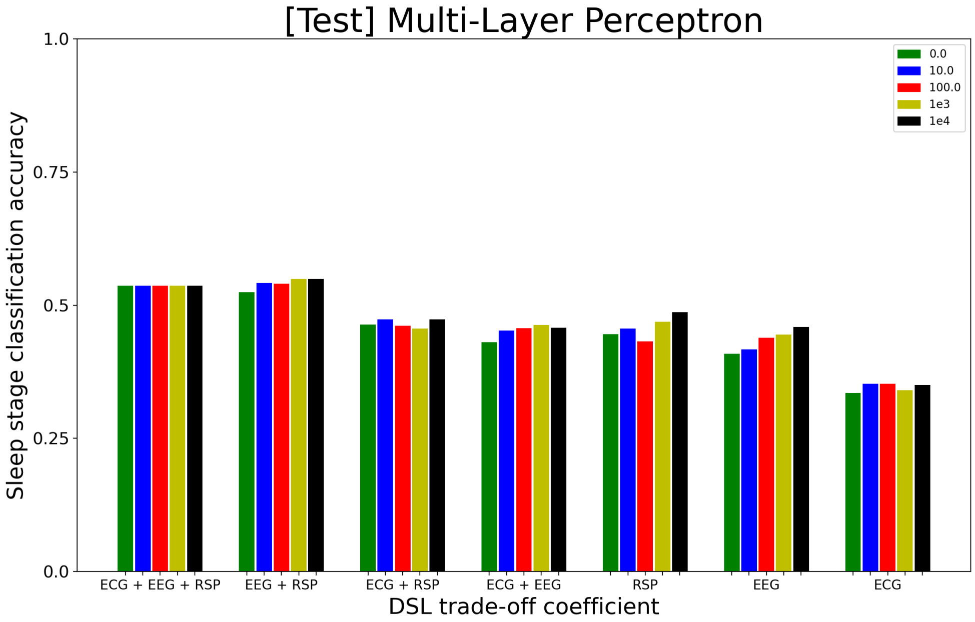


Figure 4.8: [Test] Sleep stage classification vs. available views using Deep Net (MLP), given different DSL trade-off coefficients.

values of γ. Similarly, figures 4.8 and 4.9 show the test classification performance of the MLP and RF.

4.3.1 Discussion

For the polysomnographic data-set, the inclusion of the DSL produces an intuitive trend across the various classifiers. Evaluating on the DSL loss itself, we see the trend of improved down-stream classification performance with increasing γ. This is evident in both the train and test set evaluations.

For the external classifiers, again, the test set classification performance exhibits the same patterns, i.e., improved classification with increased influence of the DSL loss. This shows that the MVRL procedure is able to extract generalizable task-specific information that can be utilized by different down-stream models, even if they do not explicitly participate in the training procedure. The effect of γ on test data is understandably more mild for all three classifiers, but the trend is still clearly observable.

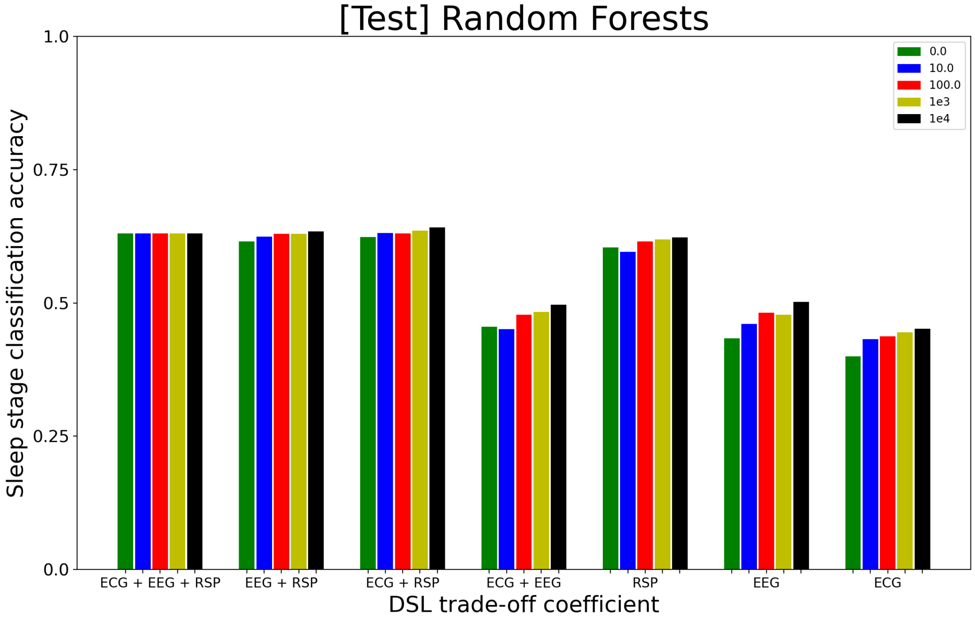


Figure 4.9: [Test] Sleep stage classification vs. available views using Random Forests, given different DSL trade-off coefficients.

4.4 Conclusion

This chapter investigated *task-adaptive* MACF, where we directly incorporated the down-stream loss (DSL) into our representation learning training procedure. We evaluated this approach on MNIST digit completion, as well as the MITBIH Polysomnographic dataset for classifying sleep-stages.

Our experiments show that *task-adaptivity* does indeed help improve down-stream model performance. For the sleep-stage classification experiment, the inclusion of the DSL generally shows an improvement for different types of classifiers, as well as different available views.

But we must be careful as to how we utilize the DSL in our training procedure; optimizing for the DSL might not always be ideal in the case of missing views. For instance, MNIST digit completion seems to benefit empirically from the addition of DSL. But visually, the digits often look garbled if the trade-off coefficient skews too much towards DSL, even though the down-stream accuracy increases on the benchmark classifier. The training procedure essentially exploits the benchmark classifier's limitations instead of producing digits which may look appealing to human observers.

Part III

Improving Inter-view Relationships

Chapter 5

View Selection and Scalable Active Search

5.1 Introduction

In the previous chapters, we take the multi-view data as a fixed entity. I.e., the data itself is immutable, we can just study, understand and exploit the structures we find. In this chapter, we look at IVRs from another perspective. Given the building blocks we have put together so far, can we improve the IVRs directly? What control do we actually have over the data?

The most obvious option is where we are the architect of the system itself. We can build the robot and its assortment of sensors, or design the probes we poke our poor patients with. The system is constructed to our specifications, so we can have it manifest the specific relationships and qualities that we care about. But sometimes, all we have is the data we get, collected months ago, along with all its, quirks, bugs and mistakes. Short of system design, we still have some degree of control over IVRs.

For one, we can simply select relevant views from all those available, e.g., choosing which probes or sensor readings to consider. Another is to reorient the context within which we view the data, e.g., different featurizations of the same image can be relevant for different problems.

For both those formulations, it would be beneficial to quantify the relationships themselves. Here, we introduce the concept of *duality* between views and data points, i.e., reinterpreting views as data points, and vice versa. This reinterpration allows us to apply more traditional single-view machine learning techniques on multi-view data. We can characterize IVR as function over input views (seen as data-points), giving a quantifiable metric we can use or optimize over.

In our work, we will consider the problem of view-selection using this duality. Active Search is an excellent candidate for this, which searches over a graph for views while efficiently querying an oracle on the utility of each view selected. In this chapter, we describe Active Search and propose a scalable alternative to it, which uses linear pairwise similarity functions between views.

5.2 View Duality

Here, we provide a possible formalization of the notion of view-duality.First, we will distinguish between the views themselves and the samples from the views. Let us revisit our notation as described in Chapter 1; we will use X_i to denote the view i itself, and x_i^j to denote the j^{th} sample from view i. The multi-view dataset contains N points, with corresponding samples in each view. We assume that no samples are missing in any view. This does not significantly affect most of our discussion, since we mainly look at statistics over the samples.

5.2.1 Views as data-points

We can consider modeling views through statistics as calculated over the samples of the views. Writing these as functions over the samples, we would have the following form for view i:

$$f(X_i) = \phi(\{x_i^1, \cdots, x_i^N\}) \tag{5.1}$$

for some feature function ϕ over sets. We can also model the interactions between views similarly through such sample statistics. Considering the pairwise case, for view i and view j, we can model their interaction as:

$$g(X_i, X_j) = \mathcal{K}\left(\psi(\{x_i^1, \cdots, x_i^N\}), \psi(\{x_j^1, \cdots, x_j^N\})\right) \tag{5.2}$$

where ψ is some feature function over sets, and $\mathcal{K}$ is a pairwise function such as a kernel function or some other similarity metric. This formulation allows us to models views and their relationships through statistics of their samples. In general, we can look at approaches for learning over sets and distributions as candidate models.

Modeling over Sets and Distributions

Learning over distributions has received a lot of attention in recent years. Sutherland et al. [Sutherland et al., 2016] discuss approximate kernel embeddings which can be computed in linear-time; this is applicable to Scalable Active Search as mentioned before. Maundet et al. [Muandet et al., 2016] review various approaches for producing kernel mean embeddings. For modeling sets, Zaheer et al. [Zaheer et al., 2017] discuss necessary and sufficient conditions required for functions over sets and propose deep models which satisfy these conditions

Learnable Relationships

Given tools to model sets, and their interactions, we have what we need to model learnable IVRs. We can apply kernel learning as a metric learning problem [Kulis et al., 2012], or look at multi-kernel learning [Gönen and Alpaydın, 2011], [Sonnenburg et al., 2006] techniques to model relevant relationships between views. These approaches also allow natural integration of prior knowledge, for example, in the form of triplet constraints. We can also use the Deep Sets framework [Zaheer et al., 2017] to have trainable set-functions which can optimized given the context of the problem.

Another paradigm to analyze and model the relationship between views is through causal discovery and causality-based modeling. Glymour et al. [Glymour et al., 2019] review various methods for causal discovery and causality-based modeling based on graphical models. Huang et al. [Huang et al., 2019] even propose an approach to causal discovery over non-stationary data distrbutions, which is often the case with dynamical systems.

5.2.2 Data-points as views

Here, we will briefly look at the other side to this duality, namely, interpreting data points as views. The idea here is to look at the extreme case where any given view only ever produces a single data point. In this sense, every data point can be considered a view which only produces one sample. And the ground-truth correspondence between these views is the data distribution itself; i.e. each point is a view into the data distribution. This interpretation provides other avenues to visualize and model multi-view data. For example, we can consider the problem of *coreset construction* which in essence is a summarization of a large data-set using relatively few data points as representative examples.

In this thesis, we mainly focus on the other direction of the duality; i.e. the interpretation of views as data points.

5.3 View Selection

Now, we shift our attention specifically to *View Selection* as a means to improve IVR. When we might not always be afforded as much control over the system producing the data, we can always choose which views we consider. This would be useful in applications with a large number of views, where we would want to select only a small subset of views which are relevant to us. These views could be selected to optimize for coverage over the underlying data space, or redundancy across views for reducing noise and dealing with missing view-data.

An example would be social media networks, like Twitter, which can be considered a large multi-view system. As described before, if we consider every user of Twitter as a view, we can take their tweets as samples from the view. Locating influential nodes in social networks and understanding the spread of information are well studied problems [Guille et al., 2013]. But these problems usually consider the relationship structure between as induced directly by the social network, while we want to leverage the *data* itself as produced by the users to construct this structure.

Again, we can take real-world events as the true underlying data points. Twitter already provides an easy way to associate tweets to real-world events, namely through hashtags. Given a set of hashtags connected to an event, we can take tweets using those hashtags as corresponding samples from different user-views. With this structure in mind, we can consider view selection as a means to tackle different problems. For one, we can select a subset users to study and contrast IVR with other characteristics (e.g., geographical, occupational, etc.) they may have. We can also aggregate users who share sentiments on different events into a bigger view, possibly to deal with noise or to augment the data from any one user.

As we mentioned before, an explicit quantification of IVRs would be very beneficial for view selection. If we can parameterize IVR, we can build models which use this parameterization. Of course, this is contingent on us being able to give an explicit mathematical definition of an IVR. Using the concept of *view-duality*, an IVR could be defined as a function over input views (seen as data-points) producing a numerical quantification of the relationship. We can then use this quantification as a basis of our IVR improvement methods; e.g., as a metric for view selection, or a parametric formulation to then optimize.

The simplest way we can quantify an IVR is by looking at pairwise functions. These may be similarity functions, directed/undirected network connectivity, etc. and necessarily look at *local* relationships in the data.

Defining a pairwise function over a set of data points induces a graph over them, with the adjacency matrix represented by the values of the function. Such a formulation is conducive for view-selection, e.g., the set-cover problem selecting a small number of views while maximizing the information we get, de-noising a given view based on related/similar views, etc. Graph-based search approaches are good candidates for such a view-selection strategy. In case of social networks, we might not want to use the social network itself for our graph structure, but rather uncover such a structure in a data-driven fashion.

5.4 Scalable Active Search for View Selection

Here, we consider Active Search [Wang et al., 2013] as a candidate algorithm for view-selection. Active Search comes under the Active Learning framework; the algorithm interactively recovers relevant samples, given a small initial set of target samples.

Background: Active Search on Graphs [ASG]

We briefly describe the algorithm introduced by Wang et al. [Wang et al., 2013]. They interpret the data as a graph where the edge-weights between points is given by the similarity $\mathcal{K}$. Their method then uses a harmonic function f to estimate the label of data points, inspired by the work done by Zhu et al. [Zhu et al., 2003]. This is done by minimizing the energy:

$$E(f) = \sum_{i \in \mathcal{L}} (y_i - f_i)^2 D_{ii} + \lambda \left(w_0 \sum_{i \in \mathcal{U}} (f_i - \pi)^2 D_{ii} + \sum_{i,j} (f_i - f_j)^2 A_{ij} \right) \quad (5.3)$$

where $A_{ij} = \mathcal{K}(x_i, x_j)$, $D_{ii} = \sum_j \mathcal{K}(x_i, x_j)$, and the regularizing constants λ and w_0 depend on transition probabilities into pseudo-nodes (see [Wang et al., 2013] for details). Explicitly, if η and ν are transition probabilities into labeled and unlabeled pseudo-nodes respectively, then $\lambda = \frac{1-\eta}{\eta}$ and $w_0 = \nu$. The minimizer is:

$$f^* = (I - BD^{-1}A)^{-1}(I - B)y', \quad (5.4)$$

$$B = \begin{bmatrix} \frac{\lambda}{1+\lambda} I_{\mathcal{L}} & 0 \\ 0 & \frac{1}{1+w_0} I_{\mathcal{U}} \end{bmatrix}, \quad y' = \begin{bmatrix} y_{\mathcal{L}} \\ \pi \end{bmatrix}$$

For simplicity of notation, f^* will simply be denoted by f moving forward.

To pick points for label queries, ASG uses a heuristic called the Impact Factor which looks at the change of f values if a given unlabeled point was labeled as positive.

$$IM_i = f_i \sum_{j \in \{U \setminus i\}} (f_j^+ - f_j)$$

The final selection criterion is $\arg\max_i f_i + \alpha IM_i$. With this, ASG iteratively queries labels and updates f and IM. ASG has an $O(n^3)$ time initialization and $O(n^2)$ time per-iteration.

5.4.1 ASG for View Selection

In this section, we propose a simple experiment to evaluate the utility of Active Search for view selection over a large number of views. We formulate the experiment as follows: we have an underlying classification problem, which different views have different amounts of information about. For example, identifying truth/falsehood in Twitter posts of a given user, would potentially be easier for someone who follows the account, as opposed to someone who doesn't.

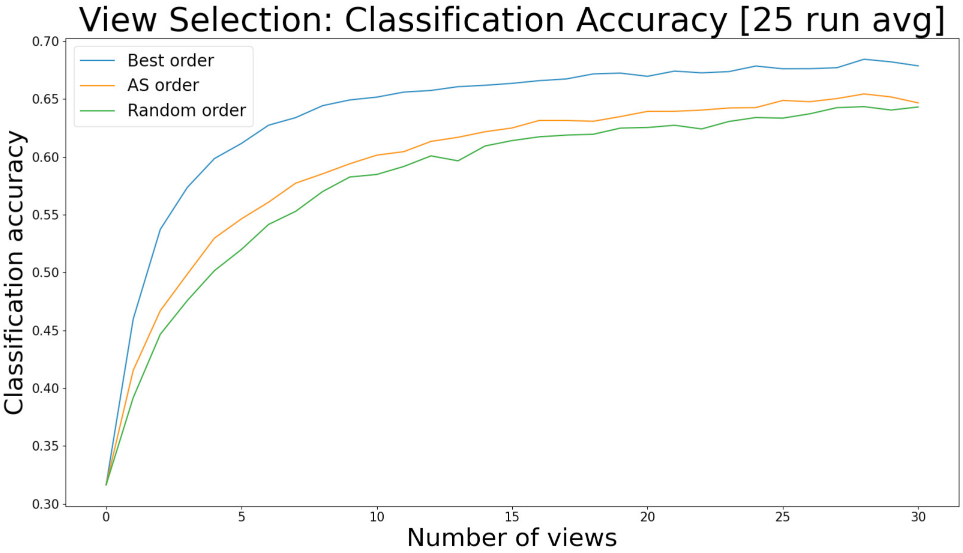


Figure 5.1: Active Search for View Selection: Comparing ASG vs. random choice on toy data

In our case, we assume there is a hidden view which provides the *true* features relevant for the classification task. The existing views have access to some noisy, limited version of these features, based on the similarity function. We want to select additional views to help improve classification performance, but we have a budget on how many views we can consider.

We compare the utility of ASG vs. random choice on view selection for classification. The oracle we use is based on the validation-set performance of a classifier trained on all the views we have choosen so far. Figure 5.1 demonstrates this utility; systematic view-selection using ASG allows us to get better classification performance with any given budget of views.

The immediate concern is that social networks are usually prohibitively large to create our own inter-user relationship structure. Most graph-based methods succumb to large amounts of data, given that they usually have a quadratic memory and time dependency on number of points N. ASG has an additional cubic dependency on N for initialization. In the next section, we look at an extension of AS which scales several orders of magnitude, and can be used on very large datasets.

5.4.2 Scalable Active Search

Here, we describe Linearized Active Search (LAS), a highly scalable extension of ASG. Similar to ASG, we look at the graph over data-points which is induced by a similarity metric $\mathcal{K}$. For our method to scale, we require that this metric is linear in the input. In other words, we require feature vectors for our points, and the dot-product as the similarity function. This requirement is often not too restrictive; in fact, some popular kernels can be approximated using a linear embedding into some feature space. For example, the RBF kernel can be approximated by Random Fourier Features [Rahimi and Recht, 2007]. For simplicity, let x_i itself represent the feature vector. The similarity between two points is then $\mathcal{K}(x_i, x_j) = x_i^T x_j$.

Note: As mentioned, ASG requires purely graphical data as input, i.e. the graph adjacency matrix. LAS works with a different class of data, which lives in some multi-dimensional feature space. A graph is induced over the data by the similarity function. If the input to ASG and LAS is the same, the results will be identical. By "the same", we mean the adjacency matrix for ASG is the same as the one of the induced graph for LAS. In this case, f, IM and the point queried will be identical at every iteration.

Algorithm 1 LAS: Linearized Active Search

```
Input: $X, \mathcal{L}_0, w_0, \lambda, \pi, \alpha, T$
  $\mathcal{U} \leftarrow \{x_1, \cdots, x_n\} \backslash \mathcal{L}_0$
  Initialize $K^{-1}$, $f$, $IM$
  for $i = 1 \rightarrow T$ do
      Query: $x_i \leftarrow argmax_{\mathcal{U}}(f + \alpha IM)$
      Update $K^{-1}$, $f$, $IM$ with $x_i, y_i$
      Remove $x_i$ from $\mathcal{U}$
  end for
```

The pseudo-code is given in Algorithm 1. We now discuss how a linear similarity function helps us update f efficiently.

Initialization The adjacency matrix is $A = X^T X$ where $X = [x_1 \cdots x_n]$, with n points and r features. Then, $D = diag(X^T X \mathbb{1})$. This gives us:

$$f = (I - RX^T X)^{-1} q$$

$$R = BD^{-1}, \quad q = (I - B)y'$$

Using the matrix inversion lemma, we get:

$$f = q + RX^T K^{-1} X q \tag{5.5}$$

$$K = I - XRX^T \tag{5.6}$$

This converts an $O(n^3)$ time matrix inverse in ASG into the $O(r^3)$ time inverse of K. For large datasets, we can expect $r \ll n$. Below, we show that we only need to invert K once; its inverse can be efficiently updated every iteration.

The initialization runs in $O(nr^2 + r^3)$ time for computing K^{-1} and $O(nr^2)$ for computing f. Next, we describe our efficient updates to K and f given a new label.

Updates to f on receiving a new label We have $K^{-1} = (I - XRX^T)^{-1}$ at the previous iteration. Only one element in R changes each iteration. Take superscript $^+$ to mean the updated value of a variable. We have:

$$R^+ = R - \gamma e_i e_i^T$$

where $\gamma = -\left(\frac{\lambda}{1+\lambda} - \frac{1}{1+w0}\right) D_{ii}^{-1}$ and e_i is the i^{th} standard basis vector. Using the matrix inversion lemma:

$$(K^+)^{-1} = K^{-1} - \frac{\gamma(K^{-1}x_i)(K^{-1}x_i)^T}{1 + \gamma x_i^T K^{-1} x_i} \tag{5.7}$$

Only one element in q changes: $q_i^+ = y_i \dfrac{1}{1+\lambda}$. Thus, the update to f can be calculated as:

$$f^+ = q^+ + R^+ X^T (K^+)^{-1} X q^+$$

This takes $O(r^2 + rn)$ time per-iteration as it just involves cascading matrix-vector multiplications.

Impact Factor LAS also includes appropriate modifications for the initialization and updates of the Impact Factor which adhere to the improved running time. We do not describe these here as they are much more involved than those above, while not being fundamentally complicated. We also slightly changed the Impact Factor from ASG: we scaled IM so that it has the same mean as the f vector. This allows us to tune α without worrying about the magnitude of values in IM, which varies based on the dataset.

Note: We have omitted some derivations and proofs for the sake of presentation. If the reader is interested, we direct them to [Venkatesan et al., 2017] for further exposition.

Weighted Neighbor Active Search [WNAS]

Here, we briefly describe a simple and intuitive alternate approach for query selection which also scales well with large amounts of data. This approach is similar to the

Nadaraya-Watson kernel regressor:

$$f_i = \frac{\sum_{j \in \mathcal{L}} y_i \cdot \mathcal{K}(x_i, x_j)}{\sum_{j \in \mathcal{L}} |\mathcal{K}(x_i, x_j)|}$$

The updates for f for this approach are simple. We keep track of the numerator and denominator individually for each unlabeled point. Each time we get a new labeled point x_i, we can compute its similarity to all other unlabeled points efficiently as the following vector:

$$\mathcal{K}(X_{\mathcal{U}}, x_i) = X_{\mathcal{U}}^{\mathrm{T}} x_i$$

We can then update the numerator and denominator of all unlabeled points directly from this vector. The numerators would be updated by adding $y_i \mathcal{K}(X_{\mathcal{U}}, x_i)$ and the denominators would be updated by adding $|\mathcal{K}(X_{\mathcal{U}}, x_i)|$. These computations require $O(nr)$ time for initialization and iteration.

5.4.3 Experiments

We performed experiments on the following datasets:

- The Covertype dataset contains multi-class data for different forest cover types. There are around 581,000 points with 54-dimensional features. We take the class with the lowest prevalence of 0.47% as positive. The data is unit normalized across features and a bias feature is appended to give 55 in total. Then, we project these onto a 550-dimensional space using Random Fourier Features [Rahimi and Recht, 2007] to approximate an RBF Kernel.

- The Adult dataset consists of census data with the task of predicting whether a person makes over $50k a year or not. It contains 14 features which are categorical or continuous. The continuous features are made categorical by discretization. Each feature is converted into a one-hot representation with m binary features for m categories. The features are then unit normalized. The positives are those making more than $50k a year. We modified the dataset size to make the target prevalence 5%. The final dataset has a 39,000 points.

- For the MNIST dataset, we combine the training, validation and testing sets into one. The 28x28 pixel images give us 784 features which are then unit normalized. We take the positive class to be the digit 1, and modified its prevalence to be 1%. The final dataset has around 63,500 points.

We compare LAS and WNAS to **Anchor Graph Regularization** with Local Anchor Embedding [AGR] as described in [Liu et al., 2010][1]. Their approach creates

[1] This was re-implemented in Python for our experiments.

a proxy graph called the Anchor Graph which approximates the larger dataset; the labels given to points are then a weighted combination of the labels of the anchor points. Since this is a semi-supervised classification approach, we retrain it every iteration with all the data and known labels. We then use the confidence values for each unlabeled point to be positive as the f value. This algorithm requires anchors to be computed beforehand. For this, we generated k-means over the transformed data points, with $k = 500$ for each dataset.

Our main experiment measured recall (number of positives found) over a fixed number of iterations for each dataset. For each dataset, 10 runs were performed starting with one randomly chosen positive as initialization. For LAS, we took α (the coefficient for the Impact Factor) to be the best from empirical evaluations. This was 10^{-6} for CoverType and Adult, and 0 for MNIST. π was taken as the true positives prevalence.

We also carried out smaller experiments over each dataset where we studied the predictive performance of LAS vs. WNAS immediately after initialization. Here, we randomly sampled 100 pairs of one positive and one negative point to initialize. Then, we reported the number of positives in the top 100 unlabeled points according to their f-values. These 100 pairs did **not** include "bad" initializations, where neither approach found any positives.

Note: We did not compare our approach vs. purely graph based methods as in the [Wang et al., 2013] Since our results are identical to ASG given the "same" data as described before, we only considered data with feature vectors.

	CoverType		MNIST	
	250	500	200	400
LAS	**198.7 ± 32.0**	**377.8 ± 55.7**	**199.5 ± 1.0**	**386.4 ± 4.9**
WNAS	188.8 ± 21.5	375.7 ± 37.9	193.7 ± 3.0	379.8 ± 7.2
AGR	27.2 ± 11.2	43.5 ± 11.8	192.8 ± 3.1	380.1 ± 4.0

	Adult	
	100	200
LAS	**53.7 ± 11.7**	**116.7 ± 13.3**
WNAS	46.1 ± 16.3	99.4 ± 26.4
AGR	23.1 ± 18.5	57.1 ± 39.2

Table 5.1: This table shows mean recall ± standard deviation at the middle and last iteration for each algorithm and dataset.

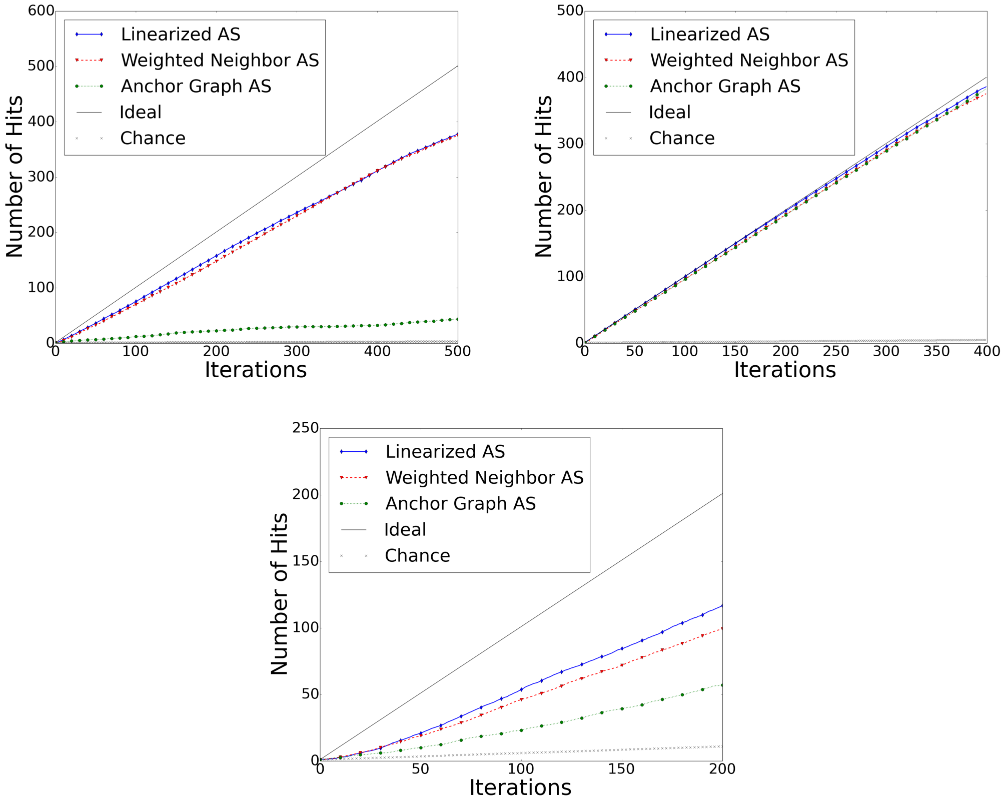


Figure 5.2: These plots show recall vs. iteration averaged across 10 runs for LAS, WNAS and AGR, along with ideal and random recall. The left image is for CoverType, the middle image is for MNIST and the right image is for Adult.

Dataset (pos%)	LAS	WNAS
Covertype (0.47%)	**4.19**	1.66
MNIST (1.00%)	**94.25**	60.68
Adult (5.00%)	**27.25**	17.29

Table 5.2: This table shows the average positives in the top 100 unlabeled points from the f-values of LAS and WNAS.

Results

Figure 5.2 shows plots of the recall per iteration of LAS, WNAS and AGR for the different datasets. Table 5.1 shows mean recall and standard deviation of these experiments in the mid and final iteration. LAS and WNAS both have good performance in all three experiments. The CoverType dataset has high variance in estimates, likely because the data has many scattered positives which are not very informative during initialization. The algorithms would then take longer to discover the remaining positives. The MNIST data-set showed particularly good performance across the different approaches; all three approaches have near ideal recall. This is likely because the targets are tightly clustered together in the feature-space. The performance of AGR in the CoverType, though much better than random choice, is poorer than the other approaches. This is because AGR incurs significant overhead in the initialization of the algorithm. Computing k-means, followed by the weights and the reduced Laplacian of the Anchor Graph takes a few hours for CoverType. Furthermore, any change in the feature function used between the data points requires recomputation of the Anchor Graph. Due to this, we only used 500 Anchors even though it is a larger data-set. This poorer approximation of the data likely led to worse performance.

Table 5.2 shows the comparison between LAS and WNAS given a single positive and negative point for initialization.

Note: We also conducted experiments on much larger datasets from the UCI Repository: the HIGGS dataset (5.5 million points) and the SUSY dataset (2.5 million points). We have not reported these results. These experiments were not any more informative than those above; they just served as a demonstration of scale.

5.5 Conclusion

In this chapter, we looked at improving IVR through quantifying the interactions of views directly. We first introduce the notion of view-data duality where we argue that we can interpret views as data points and vice-verse, which allows us to use different algorithms in our analysis of multi-view data. Such a formulation is conducive to applying metric learning and other such methods to form relationships between views.

We then proposed Scalable Active Search as a candidate for view-selection. Given a similarity metric between featurizations of views, Active Search allows us to search effectively for relevant and similar views which can corroborate and de-noise a given initial selection of views. LAS can scale this search by several orders of magnitude given a linear similarity metric between views, as demonstrated by our experimental evaluations.

Chapter 6

Conclusion

6.1 In this thesis...

We primarily investigated the following hypothesis to address shortcomings in the existing multi-view machine learning literature:

> *Multi-view data does not just provide us with multiple sets of features through the views, it also provides structural information through the interactions and relationship between them.*

Existing work rarely considers IVRs explicitly in their approaches, typically modeling them implicitly through the application or problem domain. Subsequently, most approaches are application or domain specific and an exploration of the nuances within these IVRs is absorbed into the nature of the problem itself.

To this end, we looked at developing new and adapting existing algorithms for *task-agnostic* and *task-adaptive* learning over multi-view data. We paid special attention to **Multi-view Representation Learning** as an intermediate learning step, and to the use of the learned representations to tackle down-stream applications, illustrating the approach using missing view reconstruction and classification tasks. Through our work, we show the utility of knowing *where to look* and *what to look for* while modeling IVRs.

Here, we will revisit the overall contributions of this thesis and look at how they come together. Within each, we will take a revisit to the relevant chapters, and discuss the main takeaways from them.

Contribution 1: Developing new *task-agnostic* representation learning methods to model multi-view data while respecting IVR.
[Chapter 2]

- We first introduced a direct approach, Multi-view One-vs-Rest Embedding Learning (MOREL), to verify the validity of our principal hypothesis. Here, we departed from the typical CCA-based framework where a single shared embedding space is learned for all sets of features. We instead optimized for each view individually vs. the rest, where we used group-sparsity to uncover *local* relationships in the data.

  We evaluated MOREL through synthetic experiments where we found that we were able to uncover relationships between views which were otherwise obfuscated when applying the more traditional CCA variants. The purpose of this demonstration was to show the utility of explicitly considering IVRs in our models.

- Next, we proposed the Robust Multi-view Auto-Encoder (RMAE), as a more widely applicable MVRL approach. For the RMAE, we directly introduced the notion of *robustness* into our training procedure by applying dropout to the views, i.e. randomly dropping views during training. In our experiments, we evaluated the performance of RMAE vs. other multi-view auto-encoder alternatives in how they modeled relationships between views: (i) Intersection MAE which had a single bottle-neck representation for all views (i.e. targeting purely *global* relationships) and (ii) Concatenation MAE which ignored all view-based structure and simply concatenated the views before the encoding. All three of the approaches were trained with view-dropout.

  Our synthetic experiments showed RMAE's clear superiority in reconstructing missing views from limited information. In our real-world experiments in text-based and image-based datasets with associated downstream classification tasks, we noticed that there was not a single clear winner. But in every experiment, RMAE was either the best or second best at generalizing with limited information which was not true for the others. We attribute the individual differences in performance to the contexts of the downstream problems. Often, the intersection of bottleneck representations is what one would want, especially in cases where any given view can perform reasonably on the task on its own. Regardless, RMAE is able to generalize better in the task-agnostic sense. Even when uninformed about the context, it is able to produce a representation that is *robust* and leverage IVRs.

[Chapter 3]

- Then, we looked at a generative modeling approach for multi-view data using on flow-based models. Specifically, we developed a multi-view extension of the AC Flow [Li et al., 2019], MACF. Building generative models of multi-view data, allows us to sample and estimate likelihoods of data, which is not possible with RMAE. MACF follows the same framework as before, where we have a shared encoder and individual view encoders, as well as view-dropout in the training procedure for *robustness*. Given the nature of our data, we are able to exploit view-based structures to learn conditional MACF models for each view. When we pre-train or fix view encoders, we can gracefully decompose the problem into a set of independent and parallelizable subproblems for each view.

Our primary evaluations involved digit sampling and digit classification of the benchmark handwritten character recognition (MNIST) data, where we took the four quadrants of every image as four views. We compared our classification performance against two baselines – ground truth and zero imputation. We show significant improvement over zero imputation for classification, especially when we have very few views available. Further, as more views become available, the performance of our method approaches the performance of ground-truth classification. In our digit sampling evaluations with missing views, our method very often produces reasonable digit completions. And often, in cases where the sampled digit differs from the ground-truth, the completions become graceful errors, i.e. they resolve into a different digit which is reasonable given the information available to the algorithm.

The main drawback of the MACF approach, especially in our endeavour to remain *task-agnostic*, is the slow run-time and limited scalability with large datasets. For this reason, most of our experiments were run on scaled down (and stratified) versions of the MNIST data. Parallelization into separable subproblems (in the case of fixed view-encoders) does alleviate this issue to some extent, but nonetheless, it remains a problem to be addressed in future work .

Contribution 2: Applying and adapting MVRL methods to real world problems and application, i.e. *task-adaptive* representation learning
[Chapter 4]

- We used context from the down-stream problem to inform the training of our MACF. Namely, we included the down-stream loss function into our training procedure to guide the representation learning toward task-specific utility. This introduced trade-offs – additional model parameters and therefore increased complexity of the overall learning task, as well as the overhead of the problem not being separable into views anymore – adding to the run-time of the method and increasing its propensity for overfitting. In turn, the augmented loss function can provide relevant information for the problem, and it can also introduce consistency between the views by explicitly relating the likelihood-based losses for the individual views, while contextualizing that to the downstream task needs. To investigate this, we conducted experiments on image and temporal datasets, where we varied the trade-off coefficient γ of the DSL term in the overall MVRL objective.

From our experiments on digit sampling using the MNIST database, we found that the shift to *task-adaptive* representation learning was not a straightforward win. While the performance on the down-stream evaluation criterion, i.e., the benchmark MNIST classifier, improved with increasing γ, the visual appearance of the sampled digits themselves did not. They proceeded to become more garbled as the influence from the DSL component became larger. We believe this to be the case because the DSL objective is not necessarily aligned with the goal of producing realistic samples. When we have limited information like only a quadrant of the entire digit, the range of possible digit completions is much larger. But the inclusion of the DSL loss pigeonholes the training procedure into one digit, since we are associating a label with every training instance.

In our experiments on the Polysomnographic MITBIH data-set, our evaluations provide more intuitive results. In this case, we had no benchmark classifier for the sleep-classification data, so we pre-trained a fully connected deep-neural network and froze its parameters, and used it as is in our MVRL training procedure. We notice that the inclusion of the DSL loss improves down-stream classification of sleep stages with the expected trend: classification accuracy increases as we increase γ. This holds true for all the classification models we used: The original frozen DSL model itself, a separately trained Multi-layer Perceptron (MLP) and a separately trained Random Forest (RF). In these experiments, we don't have additional evaluation criteria as before, i.e. we don't have the equivalent of an assessment of realistic samples produced by our model. Since we only evaluate through classification performance using different down-stream models, the benefit of using DSL is more straightforward to observe.

Another conclusion we can draw, which is shared across all the chapters so far, is in relation to *data-agnosticity*. This involves how we tune our encoders and our framework depending on the type or modality of the data involved (such as CNNs for images or RNNs for sequences). Now, we are rarely truly data-agnostic, since we often tune our view and shared encoders based on the data we consider. That said, our models do have more general applicability, and the flexibility of being easier to apply to various multi-view problems relatively easily, regardless of domain.

But we do notice the following: In our attempt to build more generalizable models, the flexibility gained from wider applicability comes at the cost of performance on specific tasks and applications. This is true in terms of run-time (e.g., we do not make any assumptions on how or when views go missing), and performance (e.g., we often do not use off-the-shelf architectures built specifically for our application domains).

Contribution 3: Improving IVRs directly through view-selection using Active Search, and proposing a scalable extension of it for large amounts of data.
[Chapter 5]

- We introduced the idea of *view-duality*, i.e. a reinterpretation of views as data points. We showed through a straightforward experiment that this reinterpretation allows us to consider traditionally single-view models for multi-view learning and used this as a bridge to improve IVRs. Namely, we considered view-selection, where we directly choose views which manifest favorable relationships between them. We looked at how we can quantify and parameterize such relationships, and how we can build models which use such a quantification.

  Given this, we can use Active Search as a candidate for view-selection. We demonstrate through a simple experiment where pairwise similarity function, that Active Search is useful for view-selection to denoise and improve downstream classification performance.

  We then unpacked Active Search on Graphs, and looked at how the switch to linearized similarity functions and the associated algorithmic changes, gives us a significant speed-boost to performance. The scalability of our new approach, Linearized Active Search, shows an improvement of about 2-3 orders of magnitude as compared to the original method. Further, the need for linear similarity functions is not as restrictive as it may initially seem; using techniques such as random Fourier features, we can consider linearized forms of non-linear similarity functions which we can then include in our approach scalably.

6.2 Future Work

As before, we divide these into three categories using the same overarching themes.

6.2.1 Understanding IVR

In our work so far, we have primarily considered one notion of *robustness* for our representation learning models, i.e. resilience to missing information. We can explore other options for this, including de-noising capabilities, resilience to anomalies and adversarial attacks. Further, we can look at optimizing for notions other than robustness; for example, low-bandwidth settings could call for learning an ordering over *utility* of views given a specific task.

We can also look at other generative modeling techniques to apply to our MVRL framework. Variational Auto-Encoders with Arbitrary Conditioning (VAEAC) [Ivanov et al., 2019] is one such approach which naturally fits into our proposed framework.

Finally, a principled statistical characterization of views and their relationships is missing in the literature. This would be important to address next as well.

6.2.2 Exploiting IVR

We looked at task-based model adaptation for exploiting our MVRL models. We can also look at more extensive data-based adaptations; in our work, we have only discussed the specific case of temporal data, specific to the medical domain. We can look at better adapting our framework, including view-specific and shared encoders/decoders, to specific domains and data types. RNN and CNN based encoders/decoders would be useful for temporal and image based data. We can look at similar considerations for text-based data and other domains; we can also go in the opposite direction where we look at unstructured annotations and miscellaneous meta-information of the data which can be treated as additional views. Asynchronous data is another application which would require special consideration, where we don't assume that correspondences between views always exist.

6.2.3 Improving IVR

We looked at the duality between views and data points, and proposed view-selection as an avenue to use this connection to improve multi-view relationships. We can take this a step further by building more general parametric models of IVRs themselves; the similarity metrics we described for SAS is one such option but we can look at other options which do not need to defined over pairs of views. For example, we can use general distribution statistics for featurizing views, and look at coordinated/shared space representations for multi-view data.

Building such a meta-framework, i.e. an intermediate framework for defining and modeling view-relationships, we can open up different learning methods for improving IVRs directly. We can also consider online relationship-learning and view-selection built on top of this meta-framework, which would be especially useful in dynamic environments where the multi-view datadistribution is non-static (e.g., self-driving cars in various terrains).

Bibliography

[Andrew et al., 2013] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255.

[Antol et al., 2015] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Basu et al., 2017] Basu, S., Karki, M., Ganguly, S., DiBiano, R., Mukhopadhyay, S., Gayaka, S., Kannan, R., and Nemani, R. (2017). Learning sparse feature representations using probabilistic quadtrees and deep belief nets. *Neural Processing Letters*, 45(3):855–867.

[Chen and Jin, 2015] Chen, S. and Jin, Q. (2015). Multi-modal dimensional emotion recognition using recurrent neural networks. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 49–56.

[Chua et al., 2009] Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., and Zheng, Y.-T. (July 8-10, 2009). Nus-wide: A real-world web image database from national university of singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece.

[Cosi et al., 1994] Cosi, P., Caldognetto, E. M., Vagges, K., Mian, G. A., and Contolini, M. (1994). Bimodal recognition experiments with recurrent neural networks. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–553. IEEE.

[Dinh et al., 2014] Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: Non-linear independent components estimation.

[Dinh et al., 2016] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real nvp.

[Feng et al., 2014] Feng, F., Wang, X., and Li, R. (2014). Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 7–16.

[Frome et al., 2013] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129.

[Glymour et al., 2019] Glymour, C., Zhang, K., and Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524.

[Gönen and Alpaydın, 2011] Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268.

[Guille et al., 2013] Guille, A., Hacid, H., Favre, C., and Zighed, D. A. (2013). Information diffusion in online social networks: A survey. *ACM Sigmod Record*, 42(2):17–28.

[Hardoon et al., 2004] Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

[Hinton and Zemel, 1994] Hinton, G. E. and Zemel, R. S. (1994). Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10.

[Huang et al., 2019] Huang, B., Zhang, K., Zhang, J., Ramsey, J., Sanchez-Romero, R., Glymour, C., and Schölkopf, B. (2019). Causal discovery from heterogeneous/nonstationary data. *arXiv preprint arXiv:1903.01672*.

[Huang and Kingsbury, 2013] Huang, J. and Kingsbury, B. (2013). Audio-visual deep learning for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7596–7599. IEEE.

[Ivanov et al., 2019] Ivanov, O., Figurnov, M., and Vetrov, D. (2019). Variational autoencoder with arbitrary conditioning. In *International Conference on Learning Representations*.

[Kakade and Foster, 2007] Kakade, S. M. and Foster, D. P. (2007). Multi-view regression via canonical correlation analysis. In *International Conference on Computational Learning Theory*, pages 82–96. Springer.

[Kim et al., 2013] Kim, Y., Lee, H., and Provost, E. M. (2013). Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 3687–3691. IEEE.

[Kiros et al., 2014] Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

[Klein et al., 2014] Klein, B., Lev, G., Sadeh, G., and Wolf, L. (2014). Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. *arXiv preprint arXiv:1411.7399*.

[Kulis et al., 2012] Kulis, B. et al. (2012). Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364.

[Lai and Fyfe, 2000] Lai, P. L. and Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377.

[Li et al., 2019] Li, Y., Akbar, S., and Oliva, J. B. (2019). Flow models for arbitrary conditional likelihoods.

[Liu et al., 2016] Liu, H., Mao, H., and Fu, Y. (2016). Robust multi-view feature selection. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 281–290. IEEE.

[Liu et al., 2010] Liu, W., He, J., and Chang, S.-F. (2010). Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 679–686.

[Mroueh et al., 2015] Mroueh, Y., Marcheret, E., and Goel, V. (2015). Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2130–2134. IEEE.

[Muandet et al., 2016] Muandet, K., Fukumizu, K., Sriperumbudur, B., and Schölkopf, B. (2016). Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*.

[Ngiam et al., 2011] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning.

[Nicolaou et al., 2011] Nicolaou, M. A., Gunes, H., and Pantic, M. (2011). Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *IEEE Transactions on Affective Computing*, 2(2):92–105.

[Oliva et al., 2018] Oliva, J. B., Dubey, A., Zaheer, M., Póczos, B., Salakhutdinov, R., Xing, E. P., and Schneider, J. (2018). Transformation autoregressive networks.

[Ouyang et al., 2014] Ouyang, W., Chu, X., and Wang, X. (2014). Multi-source deep learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2329–2336.

[Pan et al., 2016] Pan, Y., Mei, T., Yao, T., Li, H., and Rui, Y. (2016). Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4594–4602.

[Rahimi and Recht, 2007] Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184.

[Rasiwasia et al., 2010] Rasiwasia, N., Costa Pereira, J., Coviello, E., Doyle, G., Lanckriet, G. R., Levy, R., and Vasconcelos, N. (2010). A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 251–260.

[Rupnik and Shawe-Taylor, 2010] Rupnik, J. and Shawe-Taylor, J. (2010). Multi-view canonical correlation analysis. In *Conference on Data Mining and Data Warehouses (SiKDD 2010)*, pages 1–4.

[Sargin et al., 2007] Sargin, M. E., Yemez, Y., Erzin, E., and Tekalp, A. M. (2007). Audiovisual synchronization and fusion using canonical correlation analysis. *IEEE transactions on Multimedia*, 9(7):1396–1403.

[Silberer and Lapata, 2014] Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732.

[Slaney and Covell, 2001] Slaney, M. and Covell, M. (2001). Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In *Advances in Neural Information Processing Systems*, pages 814–820.

[Socher et al., 2014] Socher, R., Karpathy, A., Le, Q. V., Manning, C. D., and Ng, A. Y. (2014). Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

[Sonnenburg et al., 2006] Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7(Jul):1531–1565.

[Srivastava and Salakhutdinov, 2012a] Srivastava, N. and Salakhutdinov, R. (2012a). Learning representations for multimodal data with deep belief nets. In *International conference on machine learning workshop*, volume 79.

[Srivastava and Salakhutdinov, 2012b] Srivastava, N. and Salakhutdinov, R. R. (2012b). Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230.

[Suk et al., 2014] Suk, H.-I., Lee, S.-W., Shen, D., Initiative, A. D. N., et al. (2014). Hierarchical feature representation and multimodal fusion with deep learning for ad/mci diagnosis. *NeuroImage*, 101:569–582.

[Suo et al., 2017] Suo, X., Minden, V., Nelson, B., Tibshirani, R., and Saunders, M. (2017). Sparse canonical correlation analysis. *arXiv preprint arXiv:1705.10865*.

[Sutherland et al., 2016] Sutherland, D. J., Oliva, J. B., Póczos, B., and Schneider, J. (2016). Linear-time learning on distributions with approximate kernel embeddings. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[Venkatesan et al., 2020] Venkatesan, S., Miller, J. K., and Dubrawski, A. (2020). Robust multi-view representation learning (student abstract). In *AAAI*, pages 13939–13940.

[Venkatesan et al., 2017] Venkatesan, S., Miller, J. K., Schneider, J., and Dubrawski, A. (2017). Scaling active search using linear similarity functions. *arXiv preprint arXiv:1705.00334*.

[Venugopalan et al., 2014] Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., and Saenko, K. (2014). Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.

[Wang et al., 2015] Wang, D., Cui, P., Ou, M., and Zhu, W. (2015). Deep multimodal hashing with orthogonal regularization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[Wang et al., 2013] Wang, X., Garnett, R., and Schneider, J. (2013). Active search on graphs. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13.

[Weston et al., 2010] Weston, J., Bengio, S., and Usunier, N. (2010). Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35.

[Weston et al., 2011] Weston, J., Bengio, S., and Usunier, N. (2011). Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[Wu and Shao, 2014] Wu, D. and Shao, L. (2014). Multimodal dynamic networks for gesture recognition. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 945–948.

[Wu et al., 2014] Wu, Z., Jiang, Y.-G., Wang, J., Pu, J., and Xue, X. (2014). Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 167–176.

[Xu et al., 2015] Xu, R., Xiong, C., Chen, W., and Corso, J. J. (2015). Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[Ye et al., 2016] Ye, T., Wang, T., McGuinness, K., Guo, Y., and Gurrin, C. (2016). Learning multiple views with orthogonal denoising autoencoders. In *International Conference on Multimedia Modeling*, pages 313–324. Springer.

[Zaheer et al., 2017] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In *Advances in neural information processing systems*, pages 3391–3401.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*.