
Patterns for Learning with Side Information

Rico Jonschkowski¹, Sebastian Höfer¹, Oliver Brock

{RICO.JONSKOWSKI, SEBASTIAN.HOEFER, OLIVER.BROCK}@TU-BERLIN.DE

Robotics and Biology Laboratory, Technische Universität Berlin, Berlin, Germany

Abstract

Supervised, semi-supervised, and unsupervised learning estimate a function given input/output samples. Generalization of the learned function to unseen data can be improved by incorporating *side information* into learning. Side information are data that are neither from the input space nor from the output space of the function, but include useful information for learning it. In this paper we show that learning with side information subsumes a variety of related approaches, e.g. multi-task learning, multi-view learning and learning using privileged information. Our main contributions are (i) a new perspective that connects these previously isolated approaches, (ii) insights about how these methods incorporate different types of prior knowledge, and hence implement different *patterns*, (iii) facilitating the application of these methods in novel tasks, as well as (iv) a systematic experimental evaluation of these patterns in two supervised learning tasks.

1. Introduction

An important branch of machine learning research focuses on supervised learning, estimating functions based on input/output samples with the goal of predicting the correct output for new inputs. However, generalization to unseen samples always requires prior knowledge (which we refer to as *priors*) about the target function (Mitchell, 1980; Schaffer, 1994; Wolpert, 1996). By incorporating stronger priors, we can learn from less input/output samples or solve more challenging problems. But discovering useful priors that generalize over a wide range of tasks is difficult, especially if we only consider to define such priors over the target function, its input, and its output.

For many problems, there are additional data \mathbf{z} available that are neither the input \mathbf{x} nor the output \mathbf{y} of function f but that carry valuable information about how f maps \mathbf{x} to \mathbf{y} , as illustrated in Fig. 1.

We refer to this kind of data as *side information* (Chen et al., 2012), also known as privileged information (Vapnik & Vashist, 2009). Examples for side information are (i) intermediate results computed by the true underlying f , (ii) output of a related function (with input \mathbf{x}) that shares computations with f , (iii) input of a related function (with output \mathbf{y}) that shares computations with f , or (iv) relations between inputs \mathbf{x}_i and \mathbf{x}_j or between outputs \mathbf{y}_i and \mathbf{y}_j .

Example: Suppose we want to estimate a function from the input/output samples: $3 \mapsto 14$, $5 \mapsto 30$, and $2 \mapsto 9$. From looking at these data, it is not immediately obvious what the true underlying function is. However, if we provide side information and the prior that they correspond to intermediate values that f computes, in this case $3 \mapsto \mathbf{9} \mapsto 14$, $5 \mapsto \mathbf{25} \mapsto 30$, and $2 \mapsto \mathbf{4} \mapsto 9$, we see that the function first squares its input and then adds five to the intermediate result, $f(\mathbf{x}) = \mathbf{x}^2 + 5$. Side information together with a prior about how they relate to f reveal the underlying function.

Incorporating priors about how \mathbf{z} relates to f is what we call *learning with side information*. By enforcing consistency with these priors, we regularize learning which improves generalization. Note that we use side information *only during training, not for prediction*. There are a number of approaches in the literature that (often implicitly) follow the paradigm of learning with side information and demonstrate impressive results. This paper connects these lines of work, makes the underlying paradigm explicit, and attributes the improved generalization to the use of priors enabled by side information.

1.1. Prior Knowledge in Machine Learning

As mentioned before, machine learning incorporates priors about the target function f to generalize beyond observed data. Although not always stated explicitly, priors about f are reflected in every component of a machine learning approach: in the *hypothesis space* (e.g. by defining features, kernels, neural network structure), in the *generation of training data* (e.g. by data augmentation), in the *learn-*

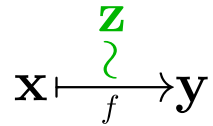


Figure 1. Side information \mathbf{z} is related to function $f(\mathbf{x}) = \mathbf{y}$.

¹The first two authors contributed equally to this work.

ing procedure (e.g. by following a curriculum or decaying the learning rate), and in the *learning objective* (e.g. by including a regularization loss).

Learning with side information provides an effective way to incorporate priors into the *learning objective* by exploiting data that are neither input nor output data of the target function f and are only required during training time. Note the difference to unsupervised and semi-supervised learning which only consider additional *input* data.

1.2. Contribution

This paper, for the first time, systematically analyzes how to exploit side information for improving generalization. It makes four main contributions:

The first contribution is a new perspective on machine learning problems. This perspective connects approaches from the literature such as *multi-task learning*, *multi-view learning*, *slow feature analysis*, *learning using privileged information*, as well as several recent works in deep learning. By connecting these lines of work, which previously did not reference each other, we enable a new exchange of ideas between them. To facilitate communication, we provide a unifying formalization of learning with side information (Sec. 2).

Our second contribution is a number of insights about these methods. First, they form a small set of *patterns* (Sec. 3) that correspond to different relationships between side information and target function (i-iv, second paragraph of the introduction). Second, the pattern’s effectiveness in generalization is a result of incorporating priors about these relationships. Since patterns incorporate different priors, their effectiveness must depend on whether the learning task and side information match the prior. We, therefore, hypothesize that different patterns work for different tasks.

As our third contribution, we demonstrate how our insights advance learning with side information. First, we use the presented patterns to systematically compare different ways to use side information. Second, we present a new pattern that has not been studied in the literature (Sec. 3.3). Third, we facilitate the practical application of learning with side information by giving a broad overview of successful applications in the literature (Appendix² C) and by making our implementation publicly available³.

The fourth contribution is a systematic experimental evaluation methods for learning with side information (Sec. 4). Our experiments confirm results from the literature by showing that learning with side information greatly improves

generalization. Moreover, the results support our hypothesis from contribution two, showing that a pattern’s performance strongly depends on the given task and the available side information.

2. Learning with Side Information

In *learning with side information*, we estimate a function $f : \mathbf{x} \rightarrow \mathbf{y}$ and optionally an auxiliary function β by minimizing two objective functions, the *main objective* \mathcal{L}_f and the *side objective* \mathcal{L}_z :

$$\begin{aligned} \operatorname{argmin}_f \mathcal{L}_f(f \mid \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N), \\ \operatorname{argmin}_{f, \beta} \mathcal{L}_z(f, \beta \mid \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_j\}_{j=1}^M). \end{aligned}$$

To define \mathcal{L}_f , we assume a supervised learning setting, in which the goal is to estimate a function $f : \mathbf{x} \rightarrow \mathbf{y}$ from a set of N input/output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. Then, \mathcal{L}_f corresponds to a standard supervised learning objective, e.g. mean-squared error for regression, and hinge loss for classification.

The side objective is captured by \mathcal{L}_z , which depends on *side information* \mathbf{z} and can include the auxiliary function β . The exact form of \mathcal{L}_z , \mathbf{z} and β depends on the *pattern* applied (Sec. 3). For all patterns, \mathbf{z} are data that are neither from the input space nor from the output space of f but carry valuable information about f , and are only needed for learning, not for prediction. Hence, the training data include M side information samples in addition to the N input/output pairs, $D = (\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_j\}_{j=1}^M)$. Each of the side information samples relates to one or more input/output samples, commonly $M = N$ or $M = N^2$.

To exploit \mathbf{z} for learning f , we formulate priors about how \mathbf{z} relates to f in the side objective \mathcal{L}_z . To express \mathcal{L}_z , many patterns require f to be split into two functions, ϕ and ψ , where ϕ maps \mathbf{x} to an intermediate representation \mathbf{s} , and ψ predicts \mathbf{y} based on \mathbf{s} , hence $\mathbf{y} = f(\mathbf{x}) = \psi(\phi(\mathbf{x})) = \psi(\mathbf{s})$. This split exposes the representation \mathbf{s} and facilitates the formulation of \mathcal{L}_z by relating \mathbf{s} and \mathbf{z} , possibly using β . Often it allows us to omit ψ and \mathbf{y} from \mathcal{L}_z , i.e. to define $\mathcal{L}_z(\phi, \beta \mid \{\mathbf{x}\}, \{\mathbf{z}\})$. For example, in the multi-task pattern (Sec. 3.2) the intermediate representation \mathbf{s} is shared amongst the main task of predicting \mathbf{y} with function $\psi(\mathbf{s})$ and an auxiliary task of predicting \mathbf{z} with $\beta(\mathbf{s})$. The auxiliary task regularizes the shared function ϕ and improves generalization for the main task.

Note that we intentionally kept this formalization narrow to improve readability. It is straightforward to extend the ideas presented here to a reinforcement learning setting, to multiple types of side information, to multiple intermediate representations, and to more than one side objective.

²The appendix can be found in the supplementary material.

³Our code for learning with side information is available at <https://github.com/tu-rbo/concarne>

2.1. Training Procedures

Since learning with side information requires us to optimize multiple learning objectives affected by different subsets of training data and functions, we need appropriate training procedures. We have identified three common training procedures that differ with respect to the order in which they (i) optimize the two objectives and (ii) modify the functions f and β :

Simultaneous learning jointly trains f and β by optimizing a weighted sum of the two learning objectives \mathcal{L}_f and \mathcal{L}_z (Weston et al., 2012). This procedure introduces the need to find a good weighting of the different learning objectives, which might be difficult if the gradients of the objectives differ by orders of magnitude and vary during learning.

If we split f into ϕ and ψ , as described in the previous section, we can choose among two additional procedures. In the **decoupled procedure**, we first optimize the side objective $\mathcal{L}_z(\phi, \beta \mid \{\mathbf{x}\}, \{\mathbf{z}\})$, while adapting ϕ and β to learn the intermediate representation \mathbf{s} . Then, we optimize the main objective $\mathcal{L}_f(\phi, \psi \mid \{\mathbf{x}, \mathbf{y}\})$, while keeping ϕ (and β) fixed. This simple procedure is only applicable if the side objective provides enough guidance to learn a task-relevant representation \mathbf{s} , whereas the simultaneous procedure is also applicable for “weak” side objectives \mathcal{L}_z . To alleviate this problem, the **pre-train and finetune procedure** first applies the decoupled procedure, but then optimizes $\mathcal{L}_f(\phi, \psi \mid \{\mathbf{x}, \mathbf{y}\})$ while adapting ϕ , too, in order to fine-tune \mathbf{s} for the task. This strategy is popular in deep learning as unsupervised pre-training (Erhan et al., 2010) and can be applied analogously for learning with side information. For this procedure to have an effect, \mathcal{L}_f must not be convex (otherwise, the pre-training step would be unlearned).

3. Patterns for Learning with Side Information

We will now present different approaches for learning with side information, which we have grouped into patterns. We describe for each pattern the general idea, the underlying prior, the side information \mathbf{z} , the side objective \mathcal{L}_z , and the auxiliary function β . We point to successful applications of each pattern (summarized in Appendix C) and visualize the patterns with schemas as in Fig. 2.

How to read the schemas: The schemas represent computation flow graphs where functions (drawn as arrows) connect variables (represented as nodes), both of which follow the definitions from Section 2. Predictions of variables are indicated by $\hat{\cdot}$. The target function is depicted in black. Additional elements that are only required at training time and can be omitted during prediction are shown in gray,

except for side information and the corresponding learning objectives, which are highlighted in green. Learning objectives are visualized by connecting variables with \sim to denote that the objective enforces similarity between these variables. The $=$ -sign (see Fig. 7) indicates that a function is replicated (e.g. by weight sharing).

Note that these graphs are *not* probabilistic graphical models (PGMs). We provide PGMs as a complementary visualization of causal dependencies in Appendix A. In contrast, the computation flow graphs are advantageous for the purpose of this paper since (i) they discriminate between variables and functions, (ii) they expose the sequence of computation, (iii) they visualize the learning objectives, and thus (iv) are easily converted into neural networks, which are employed by most of the related works reviewed in this paper.

3.1. Direct Pattern

The direct pattern leverages known, intermediate results of the computation performed by f . Given these intermediate results as side information \mathbf{z} , we can learn a function ϕ that transforms \mathbf{x} into the representation \mathbf{s} such that $\mathbf{s} \sim \mathbf{z}$, as shown in Fig. 2. No auxiliary function β is required. The pattern is only applicable if \mathbf{z} makes it easier to predict \mathbf{y} , and if \mathbf{x} contains enough information to predict \mathbf{z} . The example in Section 1 is an instance of this pattern.

To formalize this pattern, we use a suitable supervised learning side objective $\mathcal{L}_z = \mathcal{L}_{\text{direct}}(\phi \mid \{\mathbf{x}, \mathbf{z}\})$ that enforces the representation \mathbf{s} to be equal to the side information \mathbf{z} .

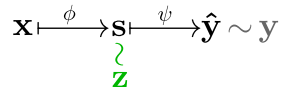


Figure 2. Direct pattern

Applications: Machine learning approaches in computational biology frequently use this pattern to combine understanding from biology research with learning. For example, in contact prediction, the goal is to predict which parts of a folded protein are close to each other based on the DNA sequence that describes the protein. Virtually all learning-based approaches to this problem first predict intermediate representations \mathbf{s} , such as secondary structures (local 3D structure categories), and then use \mathbf{s} to predict contacts (Cheng & Baldi, 2007). The representation \mathbf{s} can be reliably estimated which greatly facilitates learning ϕ .

Knowledge transfer (Vapnik & Izmailov, 2015) uses this pattern, but includes an additional step of extracting features $\beta(\mathbf{z})$ from the side information. Function ϕ is then learned by regression, such that $\mathbf{s} \sim \beta(\mathbf{z})$. They also suggest augmenting \mathbf{s} with the original input \mathbf{x} . Similarly, Chen et al. (2012) suggest to reconstruct only highly predictive features of \mathbf{z} using a modified version of AdaBoost.

3.2. Multi-Task Pattern

This pattern applies when the side information \mathbf{z} are outputs of a related function (with input \mathbf{x}) that shares computations with the function we want to estimate. As illustrated in Fig. 3, the pattern assumes that the target function $f = \psi \circ \phi$ and the related function $\beta \circ \phi$ share ϕ and therefore have the same intermediate representation $\mathbf{s} = \phi(\mathbf{x})$. By training the representation to predict both \mathbf{y} using ψ , and \mathbf{z} using the auxiliary learnable function $\beta : \mathbf{s} \rightarrow \mathbf{z}$, we incorporate the prior that *related tasks share intermediate representations*. This pattern corresponds to multi-task learning (Caruana, 1997), a type of transfer learning (Pan & Yang, 2010).

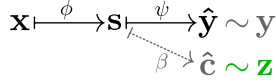


Figure 3. Multi-task pattern

To apply the multi-task pattern, we can use any suitable learning objective from supervised learning in order to learn to predict \mathbf{z} from \mathbf{x} , i.e. $\mathcal{L}_{\mathbf{z}} = \mathcal{L}_{\text{multi-task}}(\phi, \beta | \{\mathbf{x}, \mathbf{z}\})$.

Applications: Multi-task learning has been successfully applied in a wide variety of tasks (Caruana, 1997; Pan & Yang, 2010). Recently, Zhao & Itti (2015) proposed to use object pose information to improve object recognition in a convolutional deep neural network. Similarly, Levine et al. (2015) use image classification and pose prediction as side information to teach a robot remarkable vision-based manipulation skills, such as stacking lego blocks or screwing caps onto bottles.

3.2.1. IRRELEVANCE PATTERN

A special case of the multi-task patterns exploits knowledge about unrelated tasks, by enforcing the prediction of the side information to be orthogonal to the main task (Romera-Paredes et al., 2012). This idea is formalized by forcing ψ to be orthogonal to the auxiliary prediction function β (see Fig. 4), which allows to use knowledge about irrelevant distractors present in the input data. However, it is unclear how to efficiently formulate the orthogonality constraints between ψ and β for the non-linear case.

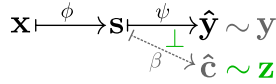


Figure 4. Exploiting irrelevant side information.

3.3. Multi-View Pattern

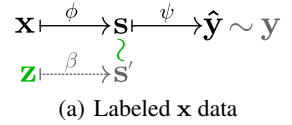
The multi-view pattern is complementary to the multi-task pattern, treating side information as input instead of output. It applies when \mathbf{z} are inputs of a related function (with output \mathbf{y}) that share computations with f . This pattern corresponds to multi-view learning (Sun, 2013).

When we treat \mathbf{z} as auxiliary input, we can use it in two different ways: explicitly by *correlating* it with the original

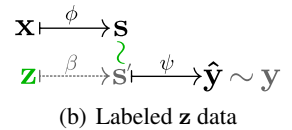
input \mathbf{x} (Fig. 5), or implicitly by *predicting* the target output (Fig. 6). In both cases, we learn functions $\phi : \mathbf{x} \mapsto \mathbf{s}$ and $\beta : \mathbf{z} \mapsto \mathbf{s}'$, such that $\mathbf{s} \sim \mathbf{s}'$.

The **multi-view (correlation) pattern** assumes that *correlated representations computed from related inputs are a useful intermediate representation for predicting the target output*. It can be formalized with a learning objective that enforces the correlation between $\phi(\mathbf{x})$ and $\beta(\mathbf{z})$, e.g. the mean squared error $\mathcal{L}_{\mathbf{z}} = \mathcal{L}_{\text{multi-view}}(\phi, \beta | \{\mathbf{x}, \mathbf{z}\}) = \sum_i \|\phi(\mathbf{x}_i) - \beta(\mathbf{z}_i)\|^2$. If we apply the decoupled training procedure, i.e. only optimize the objective, we have to add constraints, e.g. unit variance, to $\mathcal{L}_{\text{multi-view}}$ in order to avoid the trivial solution of having a constant intermediate representation. In case ϕ and β are linear, $\mathcal{L}_{\text{multi-view}}$ with unit variance corresponds to Canonical Correlation Analysis (CCA).

Applications: The pattern is often employed in multi-modal scenarios (Sun, 2013). Chen et al. (2014) show how to enhance object recognition from RGB-only images by leveraging depth data as side information during training. In computational neuroscience, the pattern is widely used to learn from multiple modalities (e.g., EEG and fMRI) or across subjects (Dähne et al., 2014). The pattern can also be applied for clustering (Feyereisl & Aicke-



(a) Labeled \mathbf{x} data



(b) Labeled \mathbf{z} data

Figure 5. Multi-view (correlation) pattern

lin, 2012). The idea is to repeatedly cluster on both $\{\mathbf{x}_i\}$ and $\{\mathbf{z}_j\}$ and then return the clustering of \mathbf{x} with the highest agreement with \mathbf{z} . In a recent article, Wang et al. (2015) suggest and compare deep architectures that combine multi-task and multi-view learning, and show that a deep canonically correlated auto-encoder gives superior results for visual, speech, and language learning.

The **multi-view prediction pattern** is based on the prior that *predicting the target output from related inputs requires similar intermediate representations*. It trains the functions $\phi : \mathbf{x} \mapsto \mathbf{s}$ and

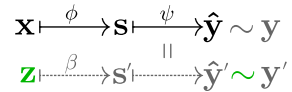


Figure 6. Multi-view prediction pattern

$\beta : \mathbf{z} \mapsto \mathbf{s}'$ such that both \mathbf{s} and \mathbf{s}' map to the target output using the same prediction function ψ , e.g. using weight sharing. Since \mathbf{s} and \mathbf{s}' are coupled to \mathbf{y} via the main objective, we do not only regularize ϕ , but also ψ .

Despite their similarities, we are not aware of any systematic comparison of multi-view and multi-task learning. Neither have we found applications of the prediction pattern in the literature. Our experiments provide a first empirical

comparison of these patterns (Sec. 4).

3.4. Pairwise Patterns

Pairwise patterns use side information \mathbf{z}_{ij} that carry information about the relationship between samples i and j to shape the intermediate representation, e.g. the difference between their intermediate representations (Fig. 7, the = indicates weight sharing).

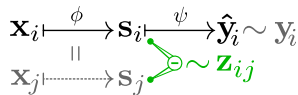


Figure 7. Pairwise pattern

3.4.1. PAIRWISE SIMILARITY/DISSIMILARITY PATTERN

If the side information gives information about similarity of samples with respect to the task, we can impose the prior that *samples that are similar (dissimilar) according to their side information should have similar (dissimilar) intermediate representations*. Such side information is often available as information about local neighborhoods of samples (Tenenbaum et al., 2000). Another powerful source of similarity information are time sequences, since temporally subsequent samples often have similar task-relevant properties, as exploited by slow feature analysis (SFA) and temporal coherence (Wiskott & Sejnowski, 2002; Weston et al., 2012). Additionally, an intelligent teacher can provide information about which samples are similar (Vapnik & Izmailov, 2015).

Similarity can be enforced with a squared loss on the distance between similar samples:

$$\mathcal{L}_{\text{sim}}(\phi | \{\mathbf{x}, \mathbf{z}\}) = \sum_{i,j} \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 \mathbb{1}(\mathbf{z}_{ij} = \text{sim.}),$$

where $\mathbb{1}$ denotes the indicator function. Solely using this objective might lead to trivial solutions where all samples are mapped to a constant. We can resolve this problem by imposing additional balancing constraints on \mathbf{s} (Weston et al., 2012) or selectively push samples apart that are dissimilar according to the side information (or optionally according to the labels \mathbf{y}):

$$\mathcal{L}_{\text{dis}}(\phi | \{\mathbf{x}, \mathbf{z}\}) = \sum_{i,j} \sigma(\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|) \mathbb{1}(\mathbf{z}_{ij} = \text{dis.}),$$

where σ is a function that measures the proximity of dissimilar samples in representation \mathbf{s} . Candidates for σ are the margin-based $\sigma(d) = \max(0, m - d^2)$ for some predefined margin m (Hadsell et al., 2006), the exponential of the negative distance $\sigma(d) = e^{-d}$ (Jonschkowski & Brock, 2014), or the Gaussian function $\sigma(d) = e^{-d^2}$ (Jonschkowski & Brock, 2015). Another way to avoid trivial solutions is to impose an input-reconstruction objective, e.g. by using an auto-encoder (Watter et al., 2015).

Vapnik & Izmailov (2015) incorporate similarity information into support vector machines by replacing the free slack variables with a function of \mathbf{z} . This method incorporates the prior that slack variables should be similar for samples with similar side information.

Applications: This pattern has been shown to successfully guide the learner in identifying task-relevant properties of \mathbf{x} . Hadsell et al. (2006) show how to learn a lighting invariant pose representation of objects in the NORB dataset. Weston et al. (2012) show that regularizing a convolutional network with a temporal coherence objective outperforms pure supervised object classification in the COIL-100 dataset by 20% in terms of recognition accuracy.

Recent works show how to apply this pattern to reinforcement learning settings. Watter et al. (2015) exploit the time sequence to jointly learn a state representation and the world dynamics from raw observations for a variety of standard tasks, such as cart-pole balancing. Jonschkowski & Brock (2015) apply the pattern in a robot navigation task, and show how leveraging temporal and robot action information enable the robot to learn a state representation from raw observations, despite the presence of visual distractors.

Note that this pattern only preserves *local* similarities between samples. If the side information provides a global distance metric, Weston et al. (2012) propose to formulate side objectives for learning a distance-preserving mapping of \mathbf{x} to \mathbf{z} , e.g. based on multi-dimensional scaling (Kruskal, 1964). Alternatively, the distance metric can be learned using side information (Fouad et al., 2013).

3.4.2. PAIRWISE TRANSFORMATION PATTERN

Instead of exploiting only binary similarity information between samples, the pairwise transformation pattern exploits continuous information about the relative transformations between samples, to make *the internal representation (or parts of it) consistent or equivariant with the known relative transformations*. Such side information is often available in robot and reinforcement learning settings.

Consistency with the transformations \mathbf{z} can be enforced in different ways: (a) Hinton et al. (2011) require the transformation \mathbf{z} to affect \mathbf{s} in a known way, and suggest the *transforming autoencoder model* shown in Fig. 8(a) to learn such an \mathbf{s} . The idea is to learn to reconstruct the transformed input from the original input and the known transformation. (b) If the transformations in \mathbf{s} are unknown, Jayaraman & Grauman (2015) suggest to learn these transformations as an auxiliary task using the pattern depicted in Fig. 8(b). (c) We can also turn this approach around and try to predict the transformation based on the original and the transformed representation (Agrawal et al., 2015) as depicted in Fig. 8(c). All three variants (a)-(c) enforce equivariance

of s with respect to the relative transformations, and can be trained using supervised side objectives. (d) Instead of optimizing for equivariance, we can also enforce that the same transformation has the same effect, when applied to different samples (Fig. 8(d)). When transformations are discrete, we formalize this by penalizing the squared difference of the change in internal representation after applying the same transformation:

$$\mathcal{L}_{\text{transf.}}(\phi | \{\mathbf{x}, \mathbf{z}\}) = \sum_{i,j} \|\Delta\phi(\mathbf{x}_i) - \Delta\phi(\mathbf{x}_j)\|^2 \mathbb{1}(\mathbf{z}_i = \mathbf{z}_j),$$

where Δ denotes the change caused by the transformation, i.e. $\Delta\phi(\mathbf{x}_i) = \phi(\mathbf{x}_{i+1}) - \phi(\mathbf{x}_i)$ for sequential data. This objective can be extended to continuous transformations by replacing the indicator function with a similarity function $\sigma(\mathbf{z}_i - \mathbf{z}_j)$ from Section 3.4.1. Variants of this pattern allow to enforce only locally consistent transformations, by multiplying $\sigma(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))$, or to enforce only consistent magnitudes of change by comparing norms $\|\Delta\phi(\mathbf{x}_i)\|$ (Jonschkowski & Brock, 2015).

Applications: Many results in the literature demonstrate the usefulness of the pairwise transformation pattern. Agrawal et al. (2015) report that using relative pose information as side information can reduce the error rate on MNIST by half with respect to pure supervised learning. They also demonstrate the approach for scene recognition on the SUN dataset, and show that pre-training using limited of relative pose side information is almost as good class-based supervision. Jayaraman & Grauman (2015) demonstrate a recognition accuracy of $\approx 50\%$ on the KITTI

dataset, outperforming pure supervised learning (41.81% accuracy) and SFA (47.04%). Interestingly, both works enforce learning a pose *equivariant* representation, although the classification task they address requires *invariance*. It is still unclear why equivariant representations help in such tasks (Lenc & Vedaldi, 2014).

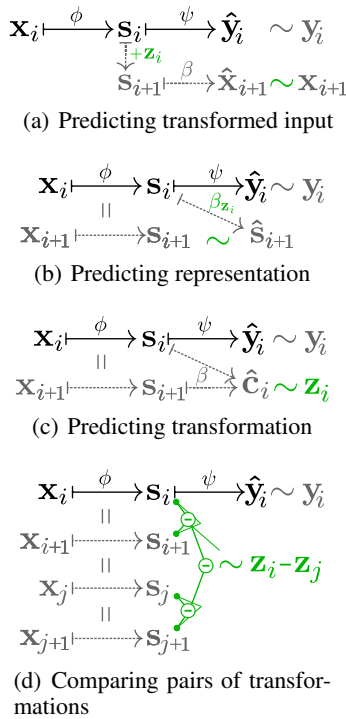


Figure 8. Pairwise transf. patterns

3.4.3. LABEL DISTANCE PATTERN

The label distance pattern is a special case of the pairwise pattern, where the side information defines distances between labels, not samples (see Fig. 9).

An instance of this pattern, often used in structured prediction, is hierarchical multi-class learning

$$\mathbf{x} \xrightarrow{f} \hat{\mathbf{y}} \approx \mathbf{y}$$

Figure 9. Label distances

(Silla Jr & Freitas, 2010), where a hierarchy is imposed on the labels to penalize misclassifications between samples with similar classes less severely.

4. Experiments

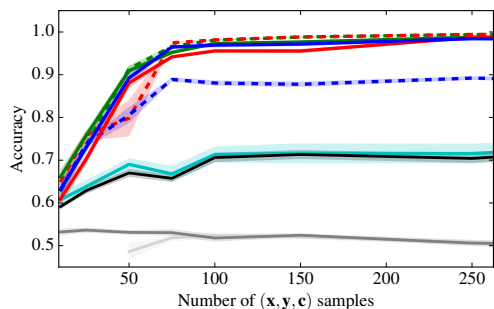
The related work, discussed in the previous section, demonstrated that learning with side information greatly improves generalization. In our experiments, we provide, for the first time, a systematic comparison of the different patterns in two supervised learning tasks. We outline the experimental rationale and results here, and refer to Appendix B for details.

4.1. Synthetic Task

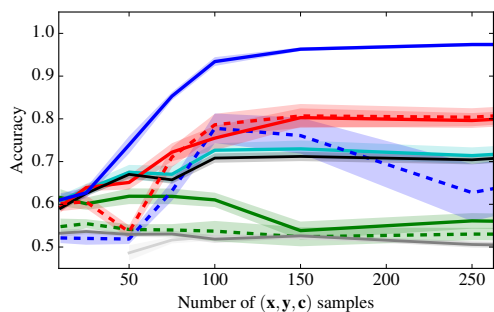
In the first, synthetic, experiment the goal is to predict the position of a randomly moving agent in a 1-dimensional state space s . The learner cannot perceive s directly, but gets an observation \mathbf{x} , which embeds s and a set of distractor signals in a high-dimensional space. The learned functions are linear (ϕ and β), and logistic functions (ψ), respectively. We study the effect of different combinations of (i) side information, (ii) patterns, and (iii) training procedures on prediction accuracy. The side information are either a noisy variant of the real state (*direct* side information, Fig. 10(a)), a second noisy high-dimensional observation (*embedded*, Fig. 10(b)), or a noisy variant of the agent’s actions, i.e. the agent’s relative motion (*pairwise*, Fig. 10(c)). We apply the direct, multi-view, multi-task, and pairwise transformation patterns, and perform training using the decoupled and simultaneous procedure (pre-training is futile with linear functions). We compare to supervised and semi-supervised baselines.

Results: While none of the baselines are able to solve the task with the given amount of training data, for each form of side information, at least one pattern achieves close to optimal performance. For the simple direct side information (Fig. 10(a)) all patterns except the multi-view prediction pattern are applicable; the reason is that the direct data correspond to the real state, and thus make solving the task almost trivial. This does not hold true for the embedded side information (Fig. 10(b)). Here, the simultaneously trained multi-view correlation pattern clearly outperforms all other methods. The reason is that the embedded side in-

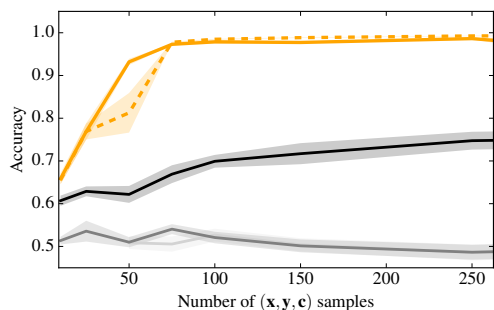
formation exactly matches the prior of the multi-view pattern. The pairwise transformation pattern, when applied to pairwise side information (Fig. 10(c)), is as effective as learning from direct side information. Overall, the experiments confirm our hypothesis that the effectiveness of each pattern strongly depends on the type of side information.



(a) Direct side information



(b) Embedded side information



(c) Pairwise side information

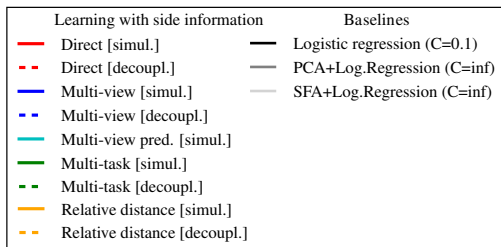


Figure 10. Results for the synthetic task, averaged over 10 runs. The most suitable patterns outperform supervised and semi-supervised baselines and generalize better with much less data.

4.2. Handwritten Character Recognition

In this experiment, we test learning with side information for handwritten character recognition in images (see Fig. 11), where we use the pen trajectory as side information. As in the previous experiment, we vary (i) the representation of the side information, either as continuous vectors or discrete categories; (ii) the pattern: direct, multi-task, or multi-view; and (iii) the training procedure: decoupled, pre-train and finetune, or simultaneous. In all our experiments, we keep the number of unlabeled data and side information fixed and examine how the accuracy of the main task is affected by changing the number of labeled data. All approaches use the same convolutional neural network architecture. As baselines we use purely supervised learning and unsupervised pre-training (deep auto-encoder).



Figure 11. Sample input for each of the 20 single-stroke characters in this task: a, b, c, d, e, g, h, l, m, n, o, p, q, r, s, u, v, w, y, z. Variations like the added random lines make this task challenging.

Results: First, we see that learning with side information can dramatically improve generalization, achieving the same performance using 5 labels per class as the baselines achieve with 100 (see Fig. 12(b)). Second, for this task, only the multi-task pattern exhibits this significant increase in performance. However, we also note that the effectiveness of learning with side information does not only depend on the pattern, but also on the representation of the side information (compare Figs. 12(a) and 12(b)): When we use the vector of pen coordinates as side information, the direct pattern provides some improvement over the baselines, but the multi-task and multi-view patterns do not improve the performance by large amounts (see Fig. 12(a)). However, when discretizing the trajectories into a small number of categories and applying the multi-task pattern for predicting these categories, we drastically reduce the number of labels required to solve this task (see Fig. 12(b)). This is not the case if we use other patterns. Moreover, we see that multi-task learning applied to the discretized side information is not significantly influenced by the training procedure (compare to direct pattern). This shows how—in this task—the multi-task pattern is able to find a good intermediate representation independently of the main objective (Fig. 12(c)). These results confirm our hypothesis that the available side information must match the prior imposed by the applied pattern in order to improve generalization.

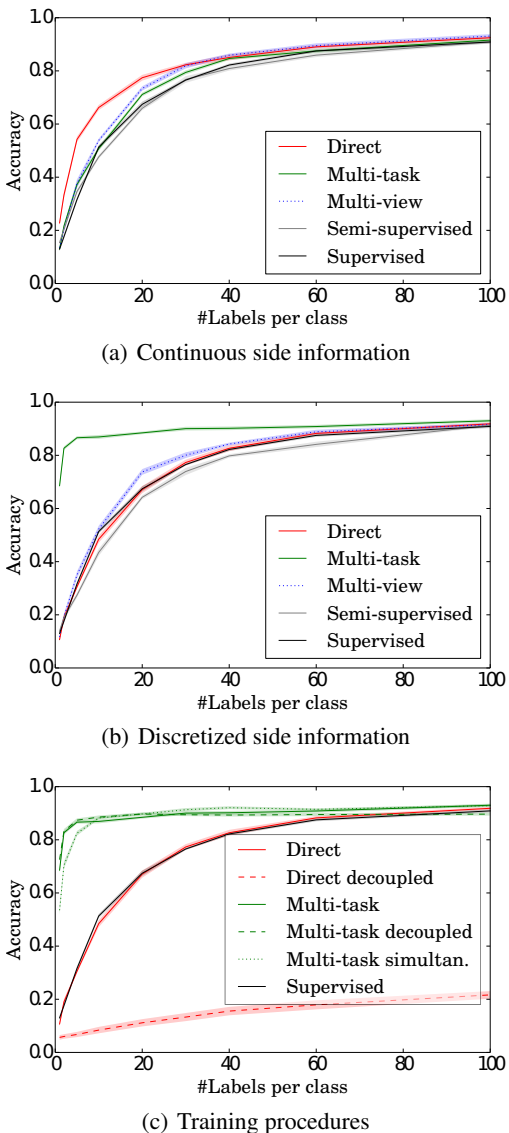


Figure 12. Results for handwritten character recognition task. (a) uses continuous side information. (b) and (c) use the discretized side information. Line styles denote the training procedure (solid = pretrain/finetune, dashed = decoupled training, dotted lines = simultaneous training).

5. Conclusion and Discussion

In this paper, we show how learning with side information provides a new perspective on machine learning, and complements existing paradigms such as supervised learning, representation learning, and deep learning. This new perspective allows us to connect various methods in the literature that (implicitly) use side information. It also enables us to systematically analyze these methods and extract patterns from them that show how they incorporate different priors about how side information relates to the target function. Since different priors coincide with different learning

tasks, we hypothesize that the performance of these patterns will vary strongly depending on the applicability of the corresponding prior. Our experiments confirm this hypothesis and also show that learning with side information can substantially improve generalization.

Our perspective of learning with side information can be helpful in a number of ways. First of all, the patterns that we have presented allow researchers and practitioners to exploit side information in novel tasks in a systematic fashion. Applying the patterns in novel tasks is facilitated by our publicly available implementation and a broad overview of existing methods and applications in this paper. Our literature review provides a formalization that unifies different lines of research, which currently seem to be unaware of the strong similarities among them. This common view will allow researchers to exchange ideas more easily, to find novel patterns for using side information, and to effectively combine patterns to exploit multiple sources of side information.

Moreover, we expect learning with side information to be very effective beyond supervised learning settings. In particular, reinforcement learning can benefit significantly because of the strong relationship between the different data sources (observations, actions, and rewards over time). By formulating priors over their relationships, we can exploit this rich side information in order to learn better state, action, and policy representations. This stands in contrast to most datasets available in machine learning, which are shaped according to the supervised learning paradigm and thus only consist of input/output samples. We, therefore, suggest to construct and augment datasets with relevant side information.

Although the utility of the view that we have presented can ultimately only be estimated in hindsight, we strongly believe that unifying ideas and providing new perspectives is vital to scientific progress, as exemplified by Bengio et al. (2013). We hope that the presented perspective of learning with side information triggers further research in that direction that generates new insights in our field.

ACKNOWLEDGMENTS

We gratefully acknowledge the funding provided by the European Commission (SOMA project, H2020-ICT-645599), the German Research Foundation (Exploration Challenge, BR 2248/3-1), and the Alexander von Humboldt foundation (funded by the German Federal Ministry of Education and Research). We would like to thank Marc Toussaint and the University of Stuttgart for granting us access to their GPU cluster, and Sven Dähne, George Konidaris, Johannes Kulick, Tobias Lang, Robert Lieck, Ingmar Posner, and Michael Schneider for fruitful discussions and comments on this manuscript.

References

- Agrawal, Pulkit, Carreira, Joao, and Malik, Jitendra. Learning to See by Moving. *arXiv:1505.01596 [cs]*, May 2015.
- Ando, Rie Kubota and Zhang, Tong. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, 6:1817–1853, November 2005.
- Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian, Bergeron, Arnaud, Bouchard, Nicolas, Warde-Farley, David, and Bengio, Yoshua. Theano: new features and speed improvements. In *NIPS 2012 deep learning workshop*, 2012.
- Baxter, Jonathan. A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR)*, 12:149–198, 2000.
- Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. Representation Learning: A Review and New Perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.
- Blum, Avrim and Mitchell, Tom. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100. ACM, 1998.
- Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.
- Chen, Jixu, Liu, Xiaoming, and Lyu, Siwei. Boosting with Side Information. In Lee, Kyoung Mu, Matsushita, Yasuyuki, Rehg, James M., and Hu, Zhanyi (eds.), *Computer Vision - ACCV 2012*, number 7724 in Lecture Notes in Computer Science, pp. 563–577. Springer Berlin Heidelberg, November 2012.
- Chen, Lin, Li, Wen, and Xu, Dong. Recognizing RGB Images by Learning from RGB-D Data. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1418–1425, June 2014.
- Cheng, Jianlin and Baldi, Pierre. Improved residue contact prediction using support vector machines and a large feature set. *BMC Bioinformatics*, 8(1):113, April 2007.
- Dähne, Sven, Nikulin, Vadim V., Ramírez, David, Schreier, Peter J., Müller, Klaus-Robert, and Haufe, Stefan. Finding brain oscillations with power dependencies in neuroimaging data. *NeuroImage*, 96:334–348, August 2014.
- Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron, Manzagol, Pierre-Antoine, Vincent, Pascal, and Bengio, Samy. Why Does Unsupervised Pre-training Help Deep Learning? *J. Mach. Learn. Res.*, 11:625–660, March 2010.
- Evgeniou, Theodoros and Pontil, Massimiliano. Regularized Multi-task Learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pp. 109–117, New York, NY, USA, 2004. ACM.
- Farquhar, Jason, Hardoon, David, Meng, Hongying, Shawe-taylor, John S., and Szedmak, Sandor. Two view learning: SVM-2k, theory and practice. In *Advances in neural information processing systems*, pp. 355–362, 2005.
- Feyereisl, Jan and Aickelin, Uwe. Privileged information for data clustering. *Information Sciences*, 194:4–23, July 2012.
- Fouad, S., Tino, P., Raychaudhury, S., and Schneider, P. Incorporating Privileged Information Through Metric Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 24(7):1086–1098, July 2013.
- Hadsell, Raia, Chopra, Sumit, and LeCun, Yann. Dimensionality Reduction by Learning an Invariant Mapping. In *Proc. Computer Vision and Pattern Recognition Conference (CVPR'06)*. IEEE Press, 2006.
- Hinton, Geoffrey E., Krizhevsky, Alex, and Wang, Sida D. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning-ICANN 2011*, pp. 44–51. Springer, 2011.
- Jayaraman, Dinesh and Grauman, Kristen. Learning image representations equivariant to ego-motion. *arXiv:1505.02206 [cs, stat]*, May 2015.
- Jonschkowski, Rico and Brock, Oliver. State Representation Learning in Robotics: Using Prior Knowledge about Physical Interaction. In *Robotics Science and Systems (RSS) X*, Berkeley, USA, 2014.
- Jonschkowski, Rico and Brock, Oliver. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, July 2015.
- Kingma, Diederik P. and Welling, Max. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.
- Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.
- Legenstein, Robert, Wilbert, Niko, and Wiskott, Laurenz. Reinforcement Learning on Slow Features of High-Dimensional Input Streams. *PLoS Comput Biol*, 6(8): e1000894, 2010.

- Lenc, Karel and Vedaldi, Andrea. Understanding image representations by measuring their equivariance and equivalence. *arXiv:1411.5908 [cs]*, November 2014.
- Levine, Sergey, Finn, Chelsea, Darrell, Trevor, and Abbeel, Pieter. End-to-End Training of Deep Visuomotor Policies. *arXiv:1504.00702 [cs]*, April 2015.
- Maurer, Andreas. Bounds for Linear Multi-Task Learning. *J. Mach. Learn. Res.*, 7:117–139, December 2006.
- Mitchell, Tom M. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ., 1980.
- Ngiam, Jiquan, Khosla, Aditya, Kim, Mingyu, Nam, Juhan, Lee, Honglak, and Ng, Andrew Y. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.
- Pan, Sinno Jialin and Yang, Qiang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.
- Romera-Paredes, Bernardino, Argyriou, Andreas, Berthouze, Nadia, and Pontil, Massimiliano. Exploiting unrelated tasks in multi-task learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 951–959, 2012.
- Schaffer, C. A conservation law for generalization performance. In *Proceedings of the Eighth International Machine Learning Conference*, pp. 259–265. Morgan Kaufmann, 1994.
- Silla Jr, Carlos N. and Freitas, Alex A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2): 31–72, April 2010.
- Sun, Shiliang. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, February 2013.
- Tenenbaum, Joshua B., Silva, Vin de, and Langford, John C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.
- Vapnik, Vladimir and Izmailov, Rauf. Learning Using Privileged Information: Similarity Control and Knowledge Transfer. *Journal of Machine Learning Research*, 16: 2023–2049, 2015.
- Vapnik, Vladimir and Vashist, Akshay. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, July 2009.
- Wang, Wei and Zhou, Zhi-Hua. A new analysis of co-training. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.
- Wang, Weiran, Arora, Raman, Livescu, Karen, and Bilmes, Jeff. On Deep Multi-View Representation Learning. 2015.
- Watter, Manuel, Springenberg, Jost, Boedecker, Joschka, and Riedmiller, Martin. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, pp. 2728–2736, 2015.
- Weston, Jason, Ratle, Frédéric, Mobahi, Hossein, and Collobert, Ronan. Deep Learning via Semi-supervised Embedding. In Montavon, Grégoire, Orr, Genevive B., and Müller, Klaus-Robert (eds.), *Neural Networks: Tricks of the Trade*, number 7700 in Lecture Notes in Computer Science, pp. 639–655. Springer Berlin Heidelberg, 2012.
- Williams, Ben H., Toussaint, Marc, and Storkey, Amos J. A Primitive Based Generative Model to Infer Timing Information in Unpartitioned Handwriting Data. In *IJCAI*, pp. 1119–1124, 2007.
- Wiskott, Laurenz and Sejnowski, Terrence J. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, April 2002.
- Wolpert, David H. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7): 1341–1390, October 1996.
- Zhao, Jiaping and Itti, Laurent. Improved Deep Learning of Object Category using Pose Information. 2015.

A. Patterns as Probabilistic Graphical Models

To complement the computation flow schemas of the patterns used throughout the paper, we provide an interpretation of the main patterns as probabilistic graphical models (PGMs). These models treat the variables and functions introduced in Sec. 2 as random variables, represented as nodes. Arrows between these random variables indicate causal relationships. Gray nodes indicate observable, and white nodes latent random variables. The latent functions can be learned by performing inference in these models.

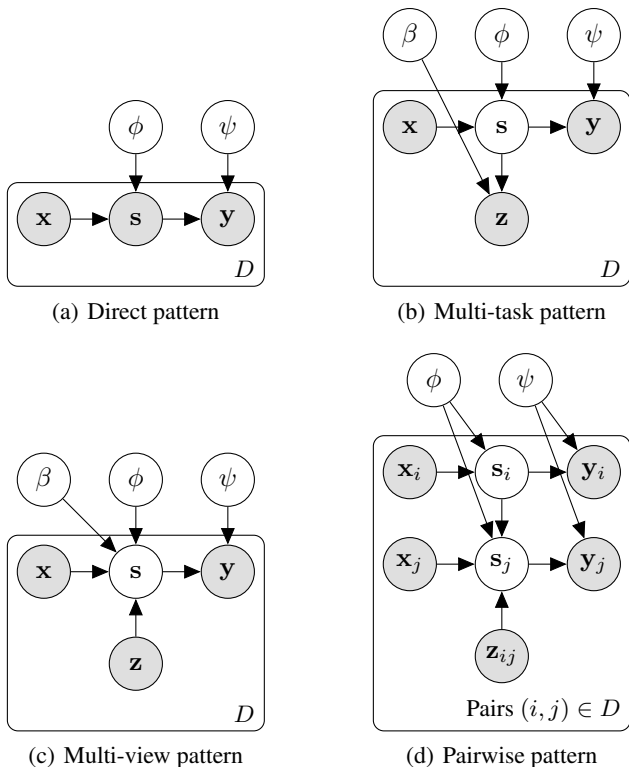


Figure 13. Probabilistic graphical models for patterns

The PGMs for the four main patterns are shown in Fig. 13. The variables \mathbf{x} , \mathbf{s} , \mathbf{z} and \mathbf{y} are observable random variables and are part of the training data D . The functions ϕ , ψ and β have become latent random variables, which the observed variables are conditioned on. We now discuss aspects of individual patterns.

The *direct pattern* is shown in Fig. 13(a). In comparison to its computation flow graph (Fig. 2), the side information \mathbf{z} is considered as drawn from the distribution over \mathbf{s} , and therefore \mathbf{z} does not appear in the graphical model of the direct pattern.

Fig. 13(b) and Fig. 13(c) show the PGM for the *multi-task* and *multi-view pattern*, respectively (computation flow graphs: Fig. 3 and Fig. 5). We see that they are struc-

turally similar and only differ on whether \mathbf{z} depends on \mathbf{s} and β or whether \mathbf{s} depends on \mathbf{z} and β . In this regard, they are equivalent to their corresponding computation flow schemas apart from the fact that the PGM for the multi-view pattern conceals how the variables \mathbf{x} and \mathbf{z} belong to the functions ϕ and β .

Finally, the prototypical *pairwise pattern* is shown in Fig. 13(a) (computation flow graph: Fig. 7). Notice that here \mathbf{s}_j is conditioned on $\mathbf{z}_{i,j}$ and \mathbf{s}_i , reflecting the fact that $\mathbf{z}_{i,j}$ is information about how \mathbf{s}_j relates to \mathbf{s}_i .

Several interesting research questions arise from the probabilistic view on learning with side information. The majority of the reviewed literature uses non-probabilistic loss functions, mostly for training neural networks. Translating them into probabilistic ones is an interesting, but non-trivial research question, as the recent work on variational auto-encoders (which are a probabilistic version of auto-encoders) shows (Kingma & Welling, 2014). A similar question arises on the relationship of the side objectives and prior probability distributions on ϕ , ψ , β and \mathbf{s} . It would be interesting to investigate whether certain side objectives can be shown to be equivalent to priors in the Bayesian sense, similar to the well-known fact that L2 regularization is equivalent to a Gaussian prior.

B. Experimental Methods

The implementation of our experiments is based on Theano (Bastien et al., 2012) and Lasagne⁴. We have made our code publicly available at: [url removed for double-blind review]

B.1. Synthetic Task

Task: The task consists of an agent moving randomly through a 1-dimensional state space $s_t \in \mathbb{R}$ where t denotes the time index. The space is split into two regions $O_1 = \{s \mid s > 0\}$ and $O_2 = \{s \mid s < 0\}$ and the goal of the learner is to determine in which of the two regions the agent is located. However, the learner cannot observe the state space directly, it only gets d -dimensional observations which are obfuscated by $d - 1$ distractors $u_t^{(i)}$, $i \in \{1, \dots, d - 1\}$: the observation is generated by applying a random rotation \mathbf{R} to the concatenated state and distractor vector: $\mathbf{x}_t = \mathbf{R} [s_t, u_t^{(1)}, \dots, u_t^{(d-1)}]$. In every time step, both the state as well as the distractor dimensions change randomly: $s_{t+1} = s_t + \varepsilon_t^{(s)}$, $u_{t+1}^{(i)} = u_t^{(i)} + \varepsilon_t^{(i)}$, where $\varepsilon_t^{(s)}, \varepsilon_t^{(i)} \sim \mathcal{N}(0, 1)$. In addition the agent receives a supervised signal \mathbf{y} which is 0 if the agent is in region O_1 and 1 in zero O_2 .

Baselines: We compare different variants of learning with

⁴<https://github.com/Lasagne/Lasagne>

side information to a supervised method (logistic regression mapping \mathbf{x} directly to \mathbf{y}) and to two semi-supervised methods. For the semi-supervised baselines we apply either PCA or SFA to learn a 1-dimensional \mathbf{s} , and then train a logistic regression mapping from \mathbf{s} to \mathbf{y} . For the logistic regression, we use L_2 regularization with $C \in \{0.001, 0.01, 0.1, 1.0, \infty\}$ and choose the result with the lowest test error. (We do not apply L_2 regularization for variants of learning with side information.)

Patterns: We implemented the four principal patterns from Section 3, using a linear function for $\phi(\mathbf{x}) = \mathbf{w}_\phi^T \mathbf{x} = \mathbf{s}$ and a logistic function for $\psi(\mathbf{s}) = \frac{1}{1+e^{-\mathbf{w}_\psi^T \mathbf{s}}}$. We apply Stochastic Gradient Descent with Nesterov momentum with value 0.9 to learn ϕ and ψ using a logistic regression loss in addition to the side objective. We train each pattern using the decoupled and simultaneous training procedures (pre-train/fine-tune is futile due since ϕ and ψ are linear, and both the target and the side objectives are convex; see Section 2.1).

For the **direct pattern**, we learn ϕ directly by performing a linear regression on \mathbf{z} , using the mean squared error loss. When training simultaneously, we weigh the main and the side objective equally.

We implement the **multi-task pattern** by using a linear function $\beta(\mathbf{s}) = \mathbf{w}_\beta^T \mathbf{s}$ for the auxiliary task and optimize it using linear regression. Again, we weigh the main and the side objective equally.

We implement two versions of the **multi-view pattern**: first, the **correlation** variant, using $\beta(\mathbf{z}) = \mathbf{w}_\beta^T \mathbf{z} = \mathbf{s}'$, optimizing for $\mathbf{s} \approx \mathbf{s}'$, secondly, the **prediction** variant. For the correlation pattern we have to give weight 0.99 to the supervised and 0.01 to the side objective, since the gradients from the side objective (MSE loss) and the supervised softmax loss differ by several orders of magnitude.

Finally, we implement the **pairwise pattern**, in form of the transformation pattern. We use a simplified version of the variant depicted in Fig. 8(c) with a fixed auxiliary function $\beta(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{s}_i - \mathbf{s}_j$. Again, we weigh the main and the side objective equally.

We evaluate subsets of the implemented patterns with three types of side information. In each experiment, we use different amounts of $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ triplets for learning, and test the prediction accuracy in the main task for a test set of size 50000. The dimensionality of \mathbf{x} is set to 50. We average the results over 10 independently generated training and test sets.

Direct Side Information: First, we provide the learner with very informative data in the form of a noisy variant of the real state: $\mathbf{z}_t^{(s)} = s_t + \varepsilon_t^{(s)}$, with $\varepsilon_t^{(s)} \sim \mathcal{N}(0, 0.05)$. Figure 10(a) shows that all variants of learning with side information except for the multi-view-prediction pattern generalize well, even given low amounts of data. The unsu-

pervised methods fail to extract a good state representation since the real hidden \mathbf{s} neither exhibits high variance, nor a slower trajectory than the distractors. Pure supervised logistic regression works better, but even when doubling the number of (\mathbf{x}, \mathbf{y}) pairs, it does not reach the performance of the methods that use side information. We believe that the multi-view-prediction pattern works badly because it does not propagate enough information from the learned ψ to regularize ϕ .

Embedded Side Information: In the second experiment, we provide side information corresponding to an additional, noisy sensor view by mapping the state into a different e -dimensional observation space, $\mathbf{z}_t^{(v)} = \mathbf{Q} [s_t + \varepsilon_t^{(v)}, v_t^{(1)}, \dots, v_t^{(e-1)}]$ with distractors $v_t^{(i)}$, random rotation matrix \mathbf{Q} and $\varepsilon_t^{(v)} \sim \mathcal{N}(0, 0.05)$. In the experiments, we set $e = \frac{d}{2}$. Figure 10(b) shows that most variants of learning with side information still outperform the supervised method, but need more data to generalize well due to the less informative side information. The multi-view method performs best, whereas the multi-task performs even worse than logistic regression. Moreover, we see that the simultaneous training procedures outperform the decoupled variants, most drastically in the multi-view pattern.

Relative Side Information: The last type of data corresponds to a noisy variant of the “actions”, $\mathbf{z}_t^{(a)} = s_t - s_{t-1} + \varepsilon_t^{(a)}$ with $\varepsilon_t^{(a)} \sim \mathcal{N}(0, 0.05)$. Fig. 10(c) shows clearly that this side information is highly useful, and allows to learn from few samples.

B.2. Handwritten Character Recognition

Dataset and task: The dataset for this experiment is based on the character trajectories dataset⁵, which consists of time series of pen velocities in x and y direction and pen tip force (more details in Williams et al. (2007)). The dataset includes 20 characters that can be written in a single stroke. Based on this dataset, we generate monochrome images of size 32×32 pixels. During image generation, we add different variations to make this task more challenging. We trace the character trajectories with varying pen width, we translate the characters randomly, and we overlay distracting lines that connect three random points in the image (see Fig. 11). These images form the input \mathbf{x} for this task. Additionally, we generate side information in the form of coordinates of 32 points along the character trajectory making up a 64D vector \mathbf{z} . Unlike the input images, these points are not translated. The task is to recognize which of the 20 characters is in the given image. The training data consist of 100 input/side information pairs per character, a random subset of which are labeled. In our experiment, we vary the

⁵<https://archive.ics.uci.edu/ml/datasets/Character+Trajectories>

number of labels per character from 1 to 100.

Neural network structure and training: We use a convolutional neural network (CNN) with rectified linear units (ReLU). We first apply a convolution with 32 filters of size 5×5 followed by ReLU non-linearity and max-pooling. The same sequence is repeated, followed by 50% dropout and a fully connected layer of 32 ReLUs (the intermediate representation s). This is again followed by a 50% dropout and 20 softmax output units. The entire network has 52756 parameters. The supervised task is formulated using the categorical crossentropy loss and optimized using Nesterov momentum with learning rate 0.003, momentum 0.9, and batch size 20 for 100 epochs, followed by 10 epochs with learning rate 0.0003. All experiments are repeated 10 times.

Discretization: In the experiment, we test two different representations of the side information. The original continuous vector representation and a discretization into 32 classes, which we obtain with k-means clustering. The rationale behind the discretization is that the exact trajectory cannot be recovered from the image (because it is not clear from the image where the character trajectory starts). We tested the same discretization on the image in the semi-supervised baseline.

Applied patterns: We compare the direct pattern, the multi-task pattern, and the multi-view pattern. The direct pattern uses a mean-squared-error objective to enforce the intermediate representation to be equal to a 32D version of the pen trajectory or the one-hot-vector that corresponds to the discretized trajectory. The multi-task pattern incorporates an additional network layer to predict the side information, either a linear layer with mean-squared-error loss to predict the trajectory or a softmax layer with cross-entropy loss to predict its discretization. The multi-view pattern uses two ReLU-layers with 32 units and dropout to compute the intermediate representation s' which we tie to s with a mean-squared-error objective. Since this objective creates trivial solutions if trained independently, we optimize it simultaneously with the main objective (weighing them with 0.05 and 0.95, respectively). All other patterns are trained using the pretrain/finetune procedure unless indicated otherwise. For simultaneous training of the multi-task pattern, we used uniform weighting.

Baselines: We compare against supervised learning on the labeled data and semi-supervised baselines: In the continuous case, we reconstruct the image using a convolutional autoencoder that mirrors the structure of the convolutional network. In the discrete case we use a similar structure as for multi-task learning but predict the discretized image instead of the discretized trajectory.

C. Overview of Related Work

In the following table, we summarize related works that apply learning with side information. Since an exhaustive list of references for each pattern is beyond the scope of this paper, we include works that span a wide variety of instantiations of the proposed patterns and refer to survey articles if available.

Abbreviations: AE=auto-encoder, CCA=canonical correlation analysis, ED=eigen decomposition, GMLVQ=generalized matrix learning vector quantization, kNN=k-nearest-neighbors, LBP=locally binary pattern, MMD=maximum mean discrepancy, NN=neural network, RBM=restricted Boltzmann machine, RL=reinforcement learning, SGD=stochastic gradient descent, SL=supervised learning (classification unless stated otherwise), SVM=support vector machine, UL=unsupervised learning

Patterns for Learning with Side Information

Pattern	Side Objective	Articles	Method, Train. Procedure	Application: Task, Input, Dataset	Side Information
<i>Direct</i> (Fig. 2)	SVM loss	Cheng & Baldi (2007)	SVM (<i>decoupl.</i>)	SL: Contact prediction on sequences	Secondary (3D) structure categories
	Regression on highly predictive features of side information	Chen et al. (2012)	AdaBoost+ (<i>simul.</i>)	SL on images: Digit (Vapnik & Vashist, 2009), facial expression (Cohn-Kanade)	Holistic image descriptions, LBP features from high-res images
	Regression loss	Vapnik & Izmailov (2015)	SVM with knowledge transfer (<i>decoupl.</i>)	SL on images Theoretical analysis: learning using privileged information	Image sections
<i>Multi-task</i> (Fig. 3)	Various supervised: hinge, MSE, softmax	Caruana (1997)	NN (<i>simul.</i>)	SL: pneumonia detection	Hematocrit, white blood cell count, potassium
		Evgeniou & Pontil (2004)	SVM (<i>simul.</i>)	SL: exam score prediction	One task per school
		Levine et al. (2015)	Conv. NN (<i>decoupl.</i>)	RL on RGB-D: robot manipulation	Image class, object pose
		Zhao & Itti (2015)	Conv. NN (<i>simul.</i>)	SL on images (YYY-20M)	Object pose
		Pan & Yang (2010)	Survey	SL, UL	-
		Baxter (2000); Ando & Zhang (2005) Maurer (2006)	Theoretical analysis of multi-task learning	-	-
<i>Multi-view</i> (Fig. 5)	Kernel CCA+soft margin SVM hinge loss	Farquhar et al. (2005)	SVM-2K (<i>simul.</i>)	SL on images (PASCAL-VOC)	Keypoint features (SIFT)
	AE reconstruction error	Ngiam et al. (2011)	RBM / NN (<i>simul.</i>)	SL on video/audio (various, e.g. CUAVE, AVLetters)	Video/audio
	Adjusted rand index, mutual information	Feyereisl & Aickelin (2012)	k-means	UL on images: MNIST	Poetic descriptions
	Kernel CCA [+MMD for domain adaption]	Chen et al. (2014)	Kernel SVM on kernel descriptor features	SL on RGB: gender (EURECOM, LFW-a), object (RGB-D O.D., Catech-256)	RGB-D
	SPoC (non-linear CCA)	Dähne et al. (2014)	Linear/quadratic, ED (<i>decoupl.</i>)	SL: Mental state prediction	EEG diff. subject
	CCA+AE reconstruction error	Wang et al. (2015)	Deep Canonically Correlated AE and others (<i>simul.</i>)	SL on images (MNIST), speech (XRMB), word embedding (WMT2011)	Noisy images, articulations, 2nd language
	-	Sun (2013)	Survey	SL, UL	-
	-	Blum & Mitchell (1998) Wang & Zhou (2010)	Theoretical analysis of multi-view learning	-	-
<i>Pairwise Similarity</i> (Fig. 7) (Fig. 9)	Slowness (first equation in Sec. 3.4.1 with $\mathbf{z}_{ij} = \mathbb{1}\{j = i + 1\}$ and covariance constraints).	Wiskott & Sejnowski (2002); Legenstein et al. (2010)	Linear/quadratic, ED (<i>decoupl.</i>)	RL on images: Navigation (physical simulation)	Time index
	Equations in Sec. 3.4.1 with margin-based $\sigma(d)$ (see Section 3.4.1)	Hadsell et al. (2006)	Conv. NN, (<i>decoupl.</i>)	UL on images: dimensionality reduction (MNIST, NORB)	
		Weston et al. (2012)	Conv. NN (<i>simul.</i>)	SL on images (MNIST, COIL100)	
	State predictability +variational AE	Watter et al. (2015)	CNN (<i>simul.</i>)	RL: inverted pendulum, cart-pole, robot arm	
	Adapted SVM loss	Vapnik & Vashist (2009)	SVM with similarity control (<i>simul.</i>)	SL: protein classification, finance market prediction, digit recognition	3D protein structure, future events, textual description
	Distance metric learning	Fouad et al. (2013)	GMLVQ/kNN (<i>decoupl.</i>)	SL: images (MNIST); galaxy morphology	Poetic descriptions; spectral features
	Hierarchical multi-class loss	Silla Jr & Freitas (2010)	Survey on hierarchical classification (SL)	-	Label similarity
<i>Pairwise Transformation</i> (Fig. 8(a)) (Fig. 8(b)) (Fig. 8(c))	Softmax (?)	Hinton et al. (2011)	NN; transforming AE (<i>simul.</i>)	SL for pose prediction (MNIST, 3D simulation)	Relative pose
	See (Hadsell et al., 2006)	Jayaraman & Grauman (2015)	Siamese-style conv. NN (<i>simul.</i>)	SL on images (NORB, KITTI, SUN)	Relative pose (discretized; with k-means)
	Softmax	Agrawal et al. (2015)	Siamese-style conv. NN (<i>pre-train/fine-tune</i>)	SL on images (MNIST, SF, KITTI)	Relative pose (discretized; uniformly)
	Various	Jonschkowski & Brock (2015)	Linear, SGD (<i>decoupl.</i>)	RL: control, navigation,	Actions, rewards, time
<i>Irrelevance</i> (Fig. 4)	$\mathcal{L} \approx \ \psi^T \beta\ _F^2$, with ψ, β linear, $\ \cdot\ _F^2$ Frobenius norm.	Romera-Paredes et al. (2012)	Linear, orthogonal matrix factorization	SL on images: emotion detection (JAFPE)	Subject identity