Knowledge-Based Regularization in Generative Modeling

Naoya Takeishi[1]* and **Yoshinobu Kawahara**[2,1]

[1]RIKEN Center for Advanced Intelligence Project
[2]Institute of Mathematics for Industry, Kyushu University
naoya.takeishi@riken.jp, kawahara@imi.kyushu-u.ac.jp

Abstract

Prior domain knowledge can greatly help to learn generative models. However, it is often too costly to hard-code prior knowledge as a specific model architecture, so we often have to use general-purpose models. In this paper, we propose a method to incorporate prior knowledge of feature relations into the learning of general-purpose generative models. To this end, we formulate a regularizer that makes the marginals of a generative model to follow prescribed relative dependence of features. It can be incorporated into off-the-shelf learning methods of many generative models, including variational autoencoders and generative adversarial networks, as its gradients can be computed using standard backpropagation techniques. We show the effectiveness of the proposed method with experiments on multiple types of datasets and generative models.

1 Introduction

Generative modeling plays a key role in many scientific and engineering applications. It has often been discussed with the notion of graphical models [Koller and Friedman, 2009], which are built upon knowledge of conditional independence of features. We have also seen the advances in neural network techniques for generative modeling, such as variational autoencoders (VAEs) [Kingma and Welling, 2014] and generative adversarial networks (GANs) [Goodfellow *et al.*, 2014].

For efficient generative modeling, a model should be designed following *prior knowledge* of target phenomena. If one knows the conditional independence of features, it can be encoded as a graphical model. If there are some insights on the physics of the phenomena, a model can be built based on a known form of differential equations. Otherwise, a special neural network architecture may be created in accordance with the knowledge. However, although some tools (e.g., [Koller and Pfeffer, 1997]) have been suggested, it is often labor-intensive and possibly even infeasible to design a generative model meticulously so that the prior knowledge specific to each problem instance is hard-coded in the model.

On the other hand, we may employ general-purpose generative models, such as kernel-based models and neural networks with common architectures (multi-layer perceptrons, convolutional nets, and recurrent nets, etc.). Meanwhile, a general-purpose model, if used as is, is less efficient than specially-designed ones in terms of sample complexity because of large hypothesis space. Therefore, we want to incorporate as much prior knowledge as possible into a model, somehow avoiding the direct model design.

A promising way to incorporate prior knowledge into general-purpose models is via regularization. For example, structured sparsity regularization [Huang *et al.*, 2011] is known as a method for regularizing linear models with a rigorous theoretical background. In a related context, posterior regularization [Ganchev *et al.*, 2010; Zhu *et al.*, 2014] has been discussed as a methodology to impose expectation constraints on learned distributions. However, the applicability of the existing regularization methods is still limited in terms of target types of prior knowledge and base models.

In generative modeling, we often have prior knowledge of *relationship between features*. For instance, consider generative modeling of sensor data, which is useful for tasks such as control and fault detection. In many instances, we know *to some extent* how units and sensors in a plant are connected (Figure 1a), from which the dependence of a part of features (here, sensor readings) can be anticipated. In the example of Figure 1a, features x_1 and x_2 would be more or equally dependent than x_1 and x_3 because of external disturbances in the processes between the units. Another example is when we know the pairwise relationship of features (Figure 1b), which can be derived from some side information.

The point is that we rarely know the full data structure, which is necessary for building a graphical model. Instead, we know *a part of* the data structure or pairwise relationship of *some* features, and there remain unknown parts that should be modeled using general-purpose models. Most existing methods cannot deal with such partial knowledge straightforwardly. In this paper, we propose a regularizer to incorporate such knowledge into general-purpose generative models based on the idea that statistical dependence of features can be (partly) anticipated from the relationship of features. The use of this type of knowledge has been actively discussed in several contexts of machine learning, but there have been surprisingly few studies in the context of generative modeling.
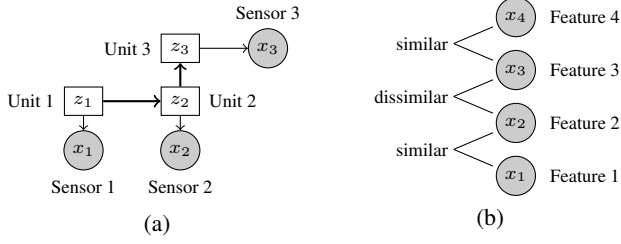
---

*Contact Author

Figure 1: Examples of knowledge of feature dependence.

The proposed regularizer is defined using a kernel-based criterion of dependence [Gretton *et al.*, 2005], which is advantageous because its gradients can be computed using standard backpropagation without additional iterative procedures. Consequently, it can be incorporated into off-the-shelf gradient-based learning methods of many general-purpose generative models, such as latent variable models (e.g., factor analysis, topic models, etc.), VAEs, and GANs. We conducted experiments using multiple datasets and generative models, and the results showcase that a model regularized using prior knowledge of feature relations achieves better generalization. The proposed method can provide a trade-off between performance and workload for model design.

2 Background

In this section, we briefly review two technical building blocks: generative modeling and dependence criteria.

2.1 Generative Modeling

We use the term "learning generative models" in the sense that we are to learn $p_\theta(\boldsymbol{x})$ explicitly or implicitly. Here, $\boldsymbol{x}$ denotes the observed variable, and θ is the set of parameters to be estimated. Many popular generative models are built as Bayesian networks and Markov random fields [Koller and Friedman, 2009]. Also, the advances in deep learning technique include VAEs [Kingma and Welling, 2014], GANs (e.g., [Goodfellow *et al.*, 2014]), autoregressive models (e.g., [van den Oord *et al.*, 2016]), and normalizing flows (e.g., [Dinh *et al.*, 2018]).

The learning strategies for generative models are usually based on minimization of some loss function $L(\theta)$:

$$\underset{\theta}{\text{minimize}}\ L(\theta). \quad (1)$$

A typical loss function is the negative log-likelihood or its approximation (e.g., ELBO in variational Bayes) [Koller and Friedman, 2009; Kingma and Welling, 2014]. Another class of loss functions is those designed to perform the comparison of model- and data-distributions, which has been studied recently often in the context of learning implicit generative models that have no explicit expression of likelihood. For example, GANs [Goodfellow *et al.*, 2014] are learned via a two-player game between a discriminator and a generator.

2.2 Dependence Criteria

Among several measures of statistical dependence of random variables, we adopt a kernel-based method, namely

Hilbert–Schmidt independence criterion (HSIC) [Gretton *et al.*, 2005]. The advantage of HSIC is discussed later in Section 3.3. Below we review the basic concepts.

HSIC is defined and computed as follows [Gretton *et al.*, 2005]. Let p_{xy} be a joint measure over $(\mathcal{X} \times \mathcal{Y}, \Gamma \times \Lambda)$, where $\mathcal{X}$ and $\mathcal{Y}$ are separable spaces, Γ and Λ are Borel sets on $\mathcal{X}$ and $\mathcal{Y}$, and $(\mathcal{X}, \Gamma)$ and $(\mathcal{Y}, \Lambda)$ are furnished with probability measure p_x and p_y, respectively. Given reproducing kernel Hilbert spaces (RKHSs) $\mathcal{F}$ and $\mathcal{G}$ on $\mathcal{X}$ and $\mathcal{Y}$, respectively, HSIC is defined as the squared Hilbert–Schmidt norm of a cross-covariance operator C_{xy}, i.e., $\text{HSIC}(\mathcal{F}, \mathcal{G}, p_{xy}) := \|C_{xy}\|_{\text{HS}}^2$. When bounded kernels k and l are uniquely associated with the RKHSs, $\mathcal{F}$ and $\mathcal{G}$, respectively,

$$\begin{aligned}
\text{HSIC}(\mathcal{F}, \mathcal{G}, p_{xy}) &= \mathbb{E}_{x,x',y,y'}[k(x,x')l(y,y')] \\
&+ \mathbb{E}_{x,x'}[k(x,x')]\mathbb{E}_{yy'}[l(y,y')] \\
&- 2\mathbb{E}_{x,y}\big[\mathbb{E}_{x'}[k(x,x')]\mathbb{E}_{y'}[l(y,y')]\big] \geq 0. \quad (2)
\end{aligned}$$

HSIC works as a dependence measure because HSIC = 0 if and only if two random variables are statistically independent (Theorem 4, [Gretton *et al.*, 2005]).

HSIC can be empirically estimated using a dataset $\mathcal{D} = \{(x_1, y_1), \dots, (x_m, y_m)\}$ from (2). Gretton *et al.* [Gretton *et al.*, 2005] presented a biased estimator with $O(m^{-1})$ bias, and Song *et al.* [Song *et al.*, 2012] suggested an unbiased estimator. In what follows, we denote an empirical estimation of $\text{HSIC}(\mathcal{F}, \mathcal{G}, p_{xy})$ computed using a dataset $\mathcal{D}$ by $\widehat{\text{HSIC}}_{x,y}(\mathcal{D})$. While the computation of empirical HSICs requires $O(m^2)$ operations, it may be sped up by methods such as random Fourier features [Zhang *et al.*, 2018].

As later discussed in Section 3.2, we particularly exploit *relative* dependence of the features. Bounliphone *et al.* [Bounliphone *et al.*, 2015] discussed the use of HSIC for a test of relative dependence. Here, we introduce $\mathcal{H}$ as a separable RKHS on another separable space $\mathcal{Z}$. Then, the test of relative dependence (i.e., which pair is more dependent, (x, y) or (x, z)?) is formulated with the null and alternative hypotheses:

$$H_0\colon\ \rho_{x,y,z} \leq 0 \quad \text{and} \quad H_1\colon\ \rho_{x,y,z} > 0,$$

where $\rho_{x,y,z} := \text{HSIC}(\mathcal{F}, \mathcal{G}, p_{xy}) - \text{HSIC}(\mathcal{F}, \mathcal{H}, p_{xz})$. For this test, they define a statistic

$$\hat{\rho}_{x,y,z} := \widehat{\text{HSIC}}_{x,y}(\mathcal{D}) - \widehat{\text{HSIC}}_{x,z}(\mathcal{D}). \quad (3)$$

From the asymptotic distribution of empirical HSIC, it is known [Bounliphone *et al.*, 2015] that a conservative estimation of p-value of the test with $\hat{\rho}_{x,y,z}$ is obtained as

$$p \leq 1 - \Phi\big(\hat{\rho}_{x,y,z}(\sigma_{xy}^2 + \sigma_{xz}^2 - 2\sigma_{xyxz})^{-1/2}\big), \quad (4)$$

where Φ is the CDF of the standard normal, and σ's denote the standard deviations of the asymptotic distribution of HSIC. See [Bounliphone *et al.*, 2015] for the definition of $\sigma_{xy}^2, \sigma_{xz}^2$, and σ_{xyxz}, which can also be estimated empirically. Here, (4) means that under the null hypothesis H_0, the probability that $\rho_{x,y,z}$ is greater than or equal to $\hat{\rho}_{x,y,z}$ is bounded by this value. In the proposed method, we exploit this fact to formulate a regularization term.

3 Proposed Method

First, we manifest the type of generative models to which the proposed method applies. Then, we define what we expect to have as prior knowledge. Finally, we present the proposed regularizer and give its interpretation.

3.1 Target Type of Generative Models

Our regularization method is agnostic of the original loss function of generative modeling and the parametrization, as long as the following (informal) conditions are satisfied:

Assumption 1. *Samples from $p_\theta(\boldsymbol{x})$, namely $\hat{\boldsymbol{x}}$, can be drawn with an admissible computational cost.*

Assumption 2. *Gradients $\nabla_\theta \mathbb{E}_{p_\theta(\boldsymbol{x})}\big[f(\boldsymbol{x})\big]$ can be (approximately) computed with an admissible computational cost.*

While Assumption 1 is satisfied in most generative models, Assumption 2 is a little less obvious. We note that the efficient computation of $\nabla_\theta \mathbb{E}_{p_\theta(\boldsymbol{x})}\big[f(\boldsymbol{x})\big]$ is often inherently easy or facilitated with techniques such as the reparameterization trick [Kingma and Welling, 2014] or its variants, and thus these assumptions are satisfied by many popular methods such as factor analysis, VAEs, and GANs.

3.2 Target Type of Prior Knowledge

We suppose that we know the (partial) relationship of features that can be encoded as plausible relative dependence of the features. Such knowledge is frequently available in many practices of generative modeling, and in fact, the use of such knowledge has also been considered in other contexts of machine learning (see Section 4.2). Below we introduce several motivating examples and then give a formal definition.

Motivating Examples

The first motivating example is when we know (a part of) the data-generating process. Suppose to learn a generative model of sensor data of an industrial plant. We usually know how units and sensors in the plant are connected, which is an important source of prior knowledge. In Figure 1a, suppose that Unit i has internal state z_i for $i = 1, \ldots, 3$. Also, suppose there are relations $z_2 = h_{12}(z_1, \omega_1)$ and $z_3 = h_2(z_2, \omega_2)$, where ω's are random noises and h's are functions of physical processes. Then, $(p(z_1), p(z_2))$ would be more statistically dependent than $(p(z_1), p(z_3))$ because of the presence of ω_2. If the sensor readings, $\{x_i\}$, are determined by $x_i = g(z_i)$ with observation functions $\{g_i\}$, then, $(p(x_1), p(x_2))$ would be more statistically dependent than $(p(x_1), p(x_3))$ analogously to z. Figure 1a is revisited in Example 1. This type of prior knowledge is often available also for physical and biological phenomena.

Another type of example is when we know pairwise similarity or dissimilarity of features (Figure 1b). For example, we may estimate similarities of distributed sensors from their locations. Also, we may anticipate similarities of words using ontology or word embeddings. Moreover, we may know the similarities of molecules from their descriptions in the chemical compound analysis. We can anticipate relative dependence from such information, i.e., directly similar feature pairs are more dependent than dissimilar feature pairs. Figure 1b is revisited in Example 1. Here, we do not have to know every pairwise relation; knowledge on *some* feature pairs is sufficient to anticipate the relative dependence.

Definition of Prior Knowledge

As the exact degree of feature dependence can hardly be described precisely from prior knowledge, we use the dependence of a pair of features *relative to another pair*. This idea is formally written as follows:

Definition 1 (Knowledge of feature dependence)**.** *Suppose that the observed random variable, $\boldsymbol{x}$, is a tuple of d random variables, i.e., $\boldsymbol{x} = (x_1, \ldots, x_d)$. Knowledge of feature dependence is described as a set of triples:*

$$\mathcal{K} := \big\{ (J_s^{ref}, J_s^+, J_s^-) \mid s = 1, \ldots, |\mathcal{K}| \big\}, \qquad (5)$$

where $J^{ref}, J^+, J^- \subset \{1, \ldots, d\}$ are index sets. The semantics are as follows. Let $J = \{i_1, \ldots, i_{|J|}\}$, and let $\boldsymbol{x}_J$ be the subtuple of $\boldsymbol{x}$ by J, i.e., $\boldsymbol{x}_J = (x_{i_1}, \ldots, x_{i_{|J|}})$. Then, triple $(J_s^{ref}, J_s^+, J_s^-)$ encodes the following piece of knowledge: "$\boldsymbol{x}_{J_s^{ref}}$ is more dependent on $\boldsymbol{x}_{J_s^+}$ than on $\boldsymbol{x}_{J_s^-}$."

Example 1. $\mathcal{K} = \{(\{1\}, \{2\}, \{3\})\}$ *in Figure 1a and* $\mathcal{K} = \{(\{1\}, \{2\}, \{3, 4\})\}$ *in Figure 1b.*

More examples of $\mathcal{K}$ are in Section 5.

3.3 Proposed Regularization Method

We define the proposed regularization method and give its interpretation as a probabilistic penalty method.

Definition of Regularizer

We want to force a generative model $p_\theta(\boldsymbol{x})$ to follow the relations encoded in $\mathcal{K}$. To this end, the order of HSIC between the marginals of $p_\theta(\boldsymbol{x})$ should be as consistent as possible to the relations in $\mathcal{K}$. More concretely, the HSIC of $(p_\theta(\boldsymbol{x}_{J^{ref},s}), p_\theta(\boldsymbol{x}_{J^+,s}))$ should be larger than that of $(p_\theta(\boldsymbol{x}_{J^{ref},s}), p_\theta(\boldsymbol{x}_{J^-,s}))$, for $s = 1, \ldots, |\mathcal{K}|$.

Because directly imposing constraints on the true HSIC of the marginals is intractable, we resort to a penalty method using empirical HSIC. Concretely, we add a term that penalizes the violation of the knowledge in $\mathcal{K}$ as follows.

Definition 2 (Knowledge-based regularization)**.** *Let $L(\theta)$ be the original loss function in* (1)*. The learning problem regularized using $\mathcal{K}$ in* (5) *is posed as*

$$\underset{\theta}{\text{minimize}} \ L(\theta) + \lambda R_\mathcal{K}(\theta), \qquad (6)$$

where $\lambda \geq 0$ is a regularization hyperparameter, and

$$R_\mathcal{K}(\theta) := \frac{1}{|\mathcal{K}|} \sum_{s=1}^{|\mathcal{K}|} R_{\mathcal{K},s}(\theta), \qquad (7)$$

$$R_{\mathcal{K},s}(\theta) := \max(0, \ \nu_\alpha - \hat{\rho}_s(\theta)).$$

Here, $\nu_\alpha \geq 0$ is another hyperparameter. $\hat{\rho}_s(\theta)$ is the empirical estimation of the relative HSIC corresponding to the s-th triple in $\mathcal{K}$, that is,

$$\hat{\rho}_s(\theta) := \widehat{HSIC}_{x_{J_s^{ref}}, x_{J_s^+}}(\hat{\mathcal{D}}) - \widehat{HSIC}_{x_{J_s^{ref}}, x_{J_s^-}}(\hat{\mathcal{D}}), \quad (8)$$

where $\hat{\mathcal{D}}$ denotes a set of samples drawn from $p_\theta(\boldsymbol{x})$.

Interpretation

By imposing the regularizer $R_{\mathcal{K}}$, we expect that the hypothesis space to be explored is made smaller implicitly, which would result in better generalization capability. This is also supported by the following interpretation of the regularizer.

While $R_{\mathcal{K}}(\theta)$ depends on the *empirical* HSIC values, making (each summand of) $R_{\mathcal{K}}(\theta)$ small can be interpreted as imposing "probabilistic" constraints on the order of the *true* HSIC values via a penalty method. Now let ρ_s denote the true value of $\hat{\rho}_s$ (we omit argument θ for simplicity), i.e.,

$$\rho_s := \mathrm{HSIC}(\mathcal{F}_{J_s^{\mathrm{ref}}}, \mathcal{F}_{J_s^+}) - \mathrm{HSIC}(\mathcal{F}_{J_s^{\mathrm{ref}}}, \mathcal{F}_{J_s^-}), \quad (9)$$

where $\mathcal{F}_J$ denotes a separable RKHS on the space of $\boldsymbol{x}_J$. Moreover, consider a test with hypotheses:

$$H_{s,0}: \ \rho_s \leq 0 \quad \text{and} \quad H_{s,1}: \ \rho_s > 0, \quad (10)$$

for $s = 1, \ldots, |\mathcal{K}|$. On the test with $H_{s,0}$ and $H_{s,1}$, the following proposition holds:

Proposition 1. *If $R_{\mathcal{K},s} = 0$ is achieved, then the null hypothesis $H_{s,0}$ can be rejected in favor of the alternative $H_{s,1}$ with p-value p_s upper bounded by*

$$p_s \leq 1 - \Phi(\nu_\alpha/\tau_s), \quad \exists \tau_s > 0. \quad (11)$$

Proof. From (4), we have $p_s \leq 1 - \Phi(\hat{\rho}_s/\tau_s)$, where

$$\tau_s = (\sigma^2_{J_s^{\mathrm{ref}} J_s^+} + \sigma^2_{J_s^{\mathrm{ref}} J_s^-} - 2\sigma_{J_s^{\mathrm{ref}} J_s^+ J_s^{\mathrm{ref}} J_s^-})^{1/2} > 0.$$

Also, when $R_{\mathcal{K},s} = 0$, we have $\hat{\rho}_s \geq \nu_\alpha$. Consequently, as Φ is monotonically increasing, we have (11). □

Choice of Hyperparameters

The proposed regularizer has two hyperparameters, λ and ν_α. While λ is a standard regularization parameter that balances the loss and the regularizer, ν_α can be interpreted as follows. Now let $\alpha := 1 - \Phi(\nu_\alpha/\tau_s)$ (i.e., the right-hand side of (11)), which corresponds to the required significance of our probabilistic constraints. We can determine $0 < \alpha < 1$ based on the plausibility of the prior knowledge, such as $\alpha = 0.05$ or $\alpha = 0.1$ (a smaller α requires higher significance). Also, τ_s can be estimated as it is defined with the variances of the asymptotic distribution of HSIC [Bounliphone *et al.*, 2015]. Hence, as $\nu_\alpha = \tau_s \Phi^{-1}(1 - \alpha)$, we can roughly determine the value of ν_α that corresponds to a specific significance, α. The need to tune ν_α (or α) remains, but the above interpretation is useful to determine the search range for ν_α.

Optimization Method

From Assumptions 1 and 2, the gradient of $\hat{\rho}_s$ with regard to θ can be computed with the backpropagation via the samples from $p_\theta(\boldsymbol{x})$, which are used to compute empirical HSIC. Hence, if the solution of the original optimization, (1), is obtained via a gradient-based method, the regularized version, (6), can also be solved using the same gradient-based method. The situation is especially simplified if the dataset is a set of d-dimensional vectors, i.e., $\mathcal{D} = \{\boldsymbol{x}_i \in \mathbb{R}^d \mid i = 1, \ldots, n\}$. In the optimization process, sample $\hat{\mathcal{D}} = \{\hat{\boldsymbol{x}}_i \in \mathbb{R}^d \mid i = 1, \ldots, m\}$ from $p_\theta(\boldsymbol{x})$ being learned, and use them for computing the the regularization term in (7) and their gradients.

Algorithm 1 Knowledge-regularized gradient method

Input: Data $\mathcal{D} = \{\boldsymbol{x}_i\}$, knowledge $\mathcal{K} = \{(J_s^{\mathrm{ref}}, J_s^+, J_s^-)\}$, hyperparameters $\lambda \geq 0$, $\nu_\alpha \geq 0$, sample size m
Output: A set of parameters θ^* of $p_\theta(\boldsymbol{x})$
1: initialize θ
2: **repeat**
3: $\quad$ draw $\hat{\mathcal{D}} = \{\hat{\boldsymbol{x}}_i \in \mathbb{R}^d \mid i = 1, \ldots, m\}$ from $p_\theta(\boldsymbol{x})$
4: $\quad$ **for** $s = 1, \ldots, |\mathcal{K}|$ **do**
5: $\quad\quad$ $J_s \leftarrow J_s^{\mathrm{ref}} \cup J_s^+ \cup J_s^-$
6: $\quad\quad$ $\hat{\mathcal{D}}_s \leftarrow \{[\hat{\boldsymbol{x}}_i]_{J_s} \in \mathbb{R}^{|J_s|} \mid i = 1, \ldots, m\}$
$\quad\quad\quad$ ▷ $[\hat{\boldsymbol{x}}_i]_J$ is the subvector of $\hat{\boldsymbol{x}}_i$ indexed by J
7: $\quad\quad$ compute $\partial\hat{\rho}_s/\partial\theta$ using $\hat{\mathcal{D}}_s$
8: $\quad$ **end for**
9: $\quad$ compute $\partial R_{\mathcal{K}}/\partial\theta$ using $\{\partial\hat{\rho}_s/\partial\theta \mid s = 1, \ldots, |\mathcal{K}|\}$
10: $\quad$ update θ using $\partial L/\partial\theta + \lambda\partial R_{\mathcal{K}}/\partial\theta$
11: **until** convergence

Then, incorporate them into the original gradient-based updates. These procedures are summarized in Algorithm 1.

The computation of $R_{\mathcal{K}}$ requires $O(m^2|\mathcal{K}|)$ operations in a naive implementation, where m is the number of samples drawn from $p_\theta(\boldsymbol{x})$. This will be burdensome for a very large $|\mathcal{K}|$, which can be alleviated by carefully choosing J^{ref}'s so that the computed HSIC values can be reused for many times.

4 Related Work

4.1 Generative Modeling with Prior Knowledge

A perspective related to this work is model design based on prior knowledge. For example, the object-oriented Bayesian network language [Koller and Pfeffer, 1997] helps graphical model designs when the structures behind data can be described in an object-oriented way. Such tools are useful when we can prepare prior knowledge enough for building a full model. However, it is not apparent how to utilize them when knowledge is only partially given, which is often the case in practice. In contrast, our method can exploit prior knowledge even if only a part of the structures is known in advance. Another related perspective is the structure learning of Bayesian networks with constraints from prior knowledge [Cussens *et al.*, 2017; Li and van Beek, 2018].

Posterior regularization (PR) [Ganchev *et al.*, 2010; Zhu *et al.*, 2014; Mei *et al.*, 2014; Hu *et al.*, 2018] is known as a framework for incorporating prior knowledge into probabilistic models. In fact, our method can be included in the most general class of PR, which was briefly mentioned in [Zhu *et al.*, 2014]. However, practically, existing work [Ganchev *et al.*, 2010; Zhu *et al.*, 2014; Mei *et al.*, 2014; Hu *et al.*, 2018] only considered limited cases of PR, where constraints were written in terms of a linear monomial of expectation with regard to the target distribution. In contrast, our method tries to fulfill the constraints on statistical dependence, which needs more complex expressions.

The work by Lopez *et al.* [2018] should be understood as a kind of technical complement of ours (and *vice versa*). While Lopez *et al.* [2018] regularize the amortized inference of VAEs to make *independence of latent variables*, we regu-
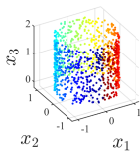
Figure 2: TOY dataset.

Figure 3: CCMNIST dataset.

dim(z)	L2 only	out-layer	proposed	dedicated decoder
25	332 (3.4)	331 (3.6)	**321** (2.4)	**325** (1.5)
50	326 (3.3)	325 (3.1)	**312** (2.6)	**301** (4.3)
100	332 (2.9)	329 (2.1)	**317** (2.0)	**305** (5.9)
200	333 (4.4)	333 (3.2)	**322** (2.4)	**305** (4.2)

Table 1: Test mean cross-entropy of VAEs on CCMNIST. Only the case of $\dim(\text{MLP}) = 2^{10}$ is reported, and the other cases are similar.

larize a general generative model itself to make *dependence of observed variables* compatible with prior knowledge. In other words, the former regularizes an encoder, whereas the latter regularizes a decoder. Moreover, our method is more flexible than [Lopez *et al.*, 2018] in terms of applicable type of knowledge; they considered only independence, but our method can incorporate both dependence and independence. Also, while Lopez *et al.* [2018] discussed the method only for VAEs, our method applies to any generative models as long as the mild assumptions in Section 3.1 are satisfied.

4.2 Knowledge of Feature Dependence

In fact, the use of knowledge of feature relation has been discussed in different or more specific contexts. In natural language processing, feature similarity is often available as the similarity between words. Xie *et al.* [Xie *et al.*, 2015] exploited correlations of words for topic modeling, and Liu *et al.* [Liu *et al.*, 2015] incorporated semantic similarity of words in learning word embeddings. These are somewhat related to generative modeling, but the scope of data and model is limited. Moreover, there are several pieces of research on utilizing feature similarity [Krupka and Tishby, 2007; Li and Li, 2008; Sandler *et al.*, 2009; Li *et al.*, 2017; Mollaysa *et al.*, 2017] for discriminative problems. Despite these interests, there have been surprisingly few studies on the use of such knowledge for generative modeling in general.

5 Experiments

5.1 Datasets and Prior Knowledge

We used the following four datasets, for which we can prepare plausible prior knowledge of feature relations.

Toy data. We created a cylinder-like toy dataset as exemplified in Figure 2. On this dataset, we know that x_1 and x_2 are more statistically dependent than x_1 and x_3 are. This knowledge is expressed in the manner of (5) as

$$\mathcal{K}_{\text{Toy}} = \{(J_1^{\text{ref}} = \{1\},\ J_1^+ = \{2\},\ J_1^- = \{3\})\}.$$

Constrained concatenated MNIST. We created a new dataset from MNIST, namely *constrained concatenated MNIST* (CCMNIST, examples in Figure 3) as follows. Each image in this dataset was generated by vertically concatenating three MNIST images. Here, the concatenation is constrained so that the top and middle images have the same label, whereas the bottom image is independent of the top two. We created training and validation sets from MNIST's training dataset and a test set from MNIST's test dataset. On this dataset, we can anticipate that the top third and the middle third of each image are more dependent than the top and the

bottom are. If each image is vectorized in a row-wise fashion as a 2352-dim vector, this knowledge is expressed as

$$\mathcal{K}_{\text{ccMNIST}} = \{(J_1^{\text{ref}} = \{1, ..., 784\},\ J_1^+ = \{785, ..., 1568\},$$
$$J_1^- = \{1569, ..., 2352\})\}.$$

Plant sensor data. We used a simulated sensor dataset (PLANT), which corresponds to the example in Figure 1a. The simulation is based on a real industrial chemical plant called the Tennessee Eastman process [Downs and Vogel, 1993]. In the plant, there are four major unit operations: reactor, vapor-liquid separator, recycle compressor, and product stripper. On each unit operation, sensors such as level sensors and thermometers are attached. We used readings of 22 sensors. On this dataset, we can anticipate the relative dependence between sets of sensors based on the process diagram of the Tennessee Eastman process [Downs and Vogel, 1993]. The knowledge set for this dataset ($\mathcal{K}_{\text{Plant}}$) contains, e.g.,

$$(J_1^{\text{ref}} = J_{\text{compressor}},\ J_1^+ = J_{\text{reactor}},\ J_1^- = J_{\text{stripper}}), \dots$$

where $J_{\text{compressor}}$ is the set of indices of sensors attached to the compressor, and so on. This is plausible because the reactor is directly connected to the compressor while the stripper is not. We prepared $\mathcal{K}_{\text{Plant}}$ comprising $|\mathcal{K}_{\text{Plant}}| = 12$ relations based on the structure of the plant [Downs and Vogel, 1993].

Solar energy production data. We used the records of solar power production[1] (SOLAR) of 137 solar power plants in Alabama in June 2006. The data of the first 20 days were used for training, the next five days were for validation, and the last five days were for test. This dataset corresponds to the example in Figure 1b because we can anticipate the dependence of features from the pairwise distances between the solar plants. We created a knowledge set $\mathcal{K}_{\text{Solar}}$ as follows; for the i-th plant ($i = 1, \dots, 137$), if the nearest plant is within 10 [km] and the distance to the second-nearest one is more than 12 [km], then add $(\{i\}, \{j_i\}, \{k_i\})$ to $\mathcal{K}_{\text{Solar}}$, where j_i and k_i are the indices of the nearest and the second-nearest plants, respectively. This resulted in $|\mathcal{K}_{\text{Solar}}| = 31$.

5.2 Settings

Models. We used factor analysis (FA, see also [Zhang, 2009]), VAE, and GAN. We trained FA and VAEs on all datasets and GANs on the TOY dataset. For VAEs and GANs, encoders, decoders, and discriminators were modeled using multi-layer perceptrons (MLPs) with three hidden layers. Hereafter we denote the dimensionality of VAE/GAN's latent variable by $\dim(z)$ and the dimensionality of MLP's hidden layer by $\dim(\text{MLP})$. We tried $\dim(\text{MLP}) = 2^5, 2^6, \dots, 2^{10}$ and used several values of $\dim(z)$ for the different datasets.

---

[1] www.nrel.gov/grid/solar-power-data.html

dataset	$\dim(\boldsymbol{z})$	L2 only	out-layer	proposed
TOY	4	0.76 (.15)	**0.41** (.08)	**0.41** (.07)
PLANT	4	8.73 (.17)	—	8.67 (.10)
	7	8.21 (.15)	—	*8.06* (.16)
	11	8.11 (.23)	—	*7.95* (.17)
SOLAR	4	4.51 (.96)	4.32 (.46)	**2.57** (.33)
	13	3.22 (.54)	3.15 (.47)	**1.97** (.16)
	54	2.73 (.43)	2.71 (.49)	**1.90** (.16)

Significant diff. from "L2 only" with *italic*: $p < .01$ or **bold**: $p < .001$.

Table 2: Test mean per-feature reconstruction errors by VAEs on the three datasets. Here only the cases of $\dim(\mathrm{MLP}) = 2^5, 2^6, 2^7$ (resp. for the three datasets) are reported. Other cases are similar.

Baselines. As the simplest baseline, we trained every model only with L2 regularization (i.e., weight decay). We tried another baseline, in which the weight of the last layer of decoder MLPs are regularized using $\mathcal{K}$; e.g., for $\mathcal{K}_{\mathrm{Toy}} = \{\{1\}, \{2\}, \{3\}\}$, we used a regularization term

$$R_{\mathcal{K}_{\mathrm{toy}}}^{\mathrm{out\text{-}layer}} := \max(0, \ \epsilon + \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2^2 - \|\boldsymbol{w}_1 - \boldsymbol{w}_3\|_2^2),$$

where $\boldsymbol{w}_i$ denotes the i-th row of the weight matrix of the last layer of the decoder MLP, and ϵ was tuned in the same manner as ν_α. We refer to this baseline as "out-layer." It cannot be applied to the PLANT dataset because the feature sets in $\mathcal{K}_{\mathrm{Plant}}$ do not have any one-to-one correspondences.

Hyperparameters. We computed HSIC with $m = 128$ using Gaussian kernels with the bandwidth set by the median heuristics. No improvement was observed with $m > 128$. The hyperparameters were chosen based on the performance on the validation sets. The search was not intensive; λ was chosen from three candidate values that roughly adjust orders of L and $R_{\mathcal{K}}$, and ν_α was chosen from .01 or .05.

5.3 Results

Below, the quantitative results are reported mainly with cross entropy (for CCMNIST) or reconstruction errors (for the others) as they are a universal performance criterion to examine the generalization capability. For VAEs, the significance of the improvement by the proposed method did not change even when we examined the ELBO values.

Evaluation by test set performance. The test set performance of VAEs are shown in Tables 1 and 2. We can observe that, while the improvement by the out-layer baseline was quite marginal, the proposed regularization method resulted in significant improvement. The performance of FA slightly improved (details omitted due to space limitations).

Comparison to a dedicated model. We compared the performance of the proposed method to that of a model designed specifically for CCMNIST dataset (termed "dedicated"). The dedicated model uses an MLP designed in accordance with the prior knowledge that the top and the middle parts are from the same digit. The performance of the dedicated model is shown in the right-most column of Table 1. We can observe that the proposed method performed intermediately between the most general case (L2 only) and the dedicated model. As
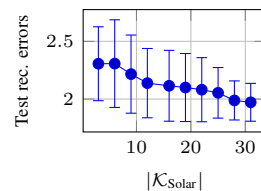

Figure 4: Performance of VAEs learned on the SOLAR dataset with $\mathcal{K}_{\mathrm{Solar}}$ of different sizes.
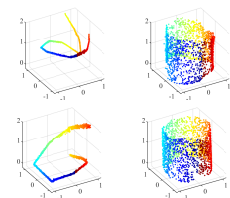

Figure 5: Samples from GAN: (*left*) w/o and (*right*) with the proposed method.

designing models meticulously is time-consuming and often even infeasible, the proposed method will be useful to give a trade-off between performance and a user's workload.

Effect of knowledge set size. Another interest lies in how the amount of provided prior knowledge affects the performance of the regularizer. We investigated this by changing the number of tuples in $\mathcal{K}_{\mathrm{Solar}}$ used in learning VAEs. We prepared knowledge subsets by extracting some tuples from the original set $\mathcal{K}_{\mathrm{Solar}}$. When creating the subsets, tuples were chosen so that a larger subset contained all elements of the smaller ones. Figure 4 shows the test set performances along with the subset size. The performance is improved with a larger knowledge set. For example, the difference between $|\mathcal{K}| = 6$ and $|\mathcal{K}| = 31$ is significant with $p \approx .003$.

Inspection of generated samples. We inspected the samples drawn from GANs trained on the TOY dataset. When the proposed regularizer was not applied, we often observed a phenomenon similar to "mode collapse" (see, e.g., [Metz *et al.*, 2017]) occurred as in the left-side plots of Figure 5. In contrast, the whole geometry of the TOY dataset was always captured successfully when the proposed regularizer was applied, as in the right-side plots of Figure 5.

Running time. For example, the time needed for training a VAE for 50 epochs on the CCMNIST dataset was 150 or 117 seconds, with or without the proposed method, respectively. As the complexity is linear in $|\mathcal{K}|$, this would not be prohibitive, while any speed-up techniques will be useful.

6 Conclusion

In this work, we developed a method for regularizing generative models using prior knowledge of feature dependence, which is frequently encountered in practice and has been studied in contexts other than generative modeling. The proposed regularizer can be incorporated in off-the-shelf learning methods of many generative models. The direct extension of the current method includes the use of higher-order dependence between multiple sets of features [Pfister *et al.*, 2017].

Acknowledgments

References

[Bounliphone *et al.*, 2015] Wacha Bounliphone, Arthur Gretton, Arthur Tenenhaus, and Matthew B. Blaschko.

A low variance consistent test of relative dependency. In *Proc. of the 32nd Int. Conf. on Machine Learning*, pages 20–29, 2015.

[Cussens *et al.*, 2017] James Cussens, Matti Järvisalo, Janne H. Korhonen, and Mark Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *Journal of Artificial Intelligence Research*, 58:185–229, 2017.

[Dinh *et al.*, 2018] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Proc. of the 6th Int. Conf. on Learning Representations*, 2018.

[Downs and Vogel, 1993] James J. Downs and Ernest F. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, 1993.

[Ganchev *et al.*, 2010] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *J. of Machine Learning Research*, 11:2001–2049, 2010.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680, 2014.

[Gretton *et al.*, 2005] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert–Schmidt norms. In *Algorithmic Learning Theory*, pages 63–77, 2005.

[Hu *et al.*, 2018] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Xiaodan Liang, Lianhui Qin, Haoye Dong, and Eric P. Xing. Deep generative models with learnable knowledge constraints. In *Advances in Neural Information Processing Systems 31*, 2018.

[Huang *et al.*, 2011] Junzhou Huang, Tong Zhang, and Dimitris Mataxas. Learning with structured sparsity. *J. of Machine Learning Research*, 12:3371–3412, 2011.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proc. of the 2nd Int. Conf. on Learning Representations*, 2014.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[Koller and Pfeffer, 1997] Daphne Koller and Avi Pfeffer. Object-oriented Bayesian networks. In *Proc. of the 13th Conf. on Uncertainty in Artificial Intelligence*, pages 302–313, 1997.

[Krupka and Tishby, 2007] Eyal Krupka and Naftali Tishby. Incorporating prior knowledge on features into learning. In *Proc. of the 11th Int. Conf. on Artificial Intelligence and Statistics*, pages 227–234, 2007.

[Li and Li, 2008] Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182, 2008.

[Li and van Beek, 2018] Andrew C Li and Peter van Beek. Bayesian network structure learning with side constraints. In *Proc. of the 9th Int. Conf. on Probabilistic Graphical Models*, pages 225–236, 2018.

[Li *et al.*, 2017] Yingming Li, Ming Yang, Zenglin Xu, and Zhongfei (Mark) Zhang. Learning with feature network and label network simultaneously. In *Proc. of the 31st AAAI Conf. on Artificial Intelligence*, pages 1410–1416, 2017.

[Liu *et al.*, 2015] Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int. Joint Conf. on Natural Language Processing*, pages 1501–1511, 2015.

[Lopez *et al.*, 2018] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Information constraints on auto-encoding variational Bayes. In *Advances in Neural Information Processing Systems 31*, 2018.

[Mei *et al.*, 2014] Shike Mei, Jun Zhu, and Xiaojin Zhu. Robust RegBayes: Selectively incorporating first-order logic domain knowledge into Bayesian models. In *Proc. of the 31st Int. Conf. on Machine Learning*, pages 253–261, 2014.

[Metz *et al.*, 2017] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *Proc. of the 5th Int. Conf. on Learning Representations*, 2017.

[Mollaysa *et al.*, 2017] Amina Mollaysa, Pablo Strasser, and Alexandros Kalousis. Regularising non-linear models using feature side-information. In *Proc. of the 34th Int. Conf. on Machine Learning*, pages 2508–2517, 2017.

[Pfister *et al.*, 2017] Niklas Pfister, Peter Bühlmann, Bernhard Schölkopf, and Jonas Peters. Kernel-based tests for joint independence. *J. of the Royal Statistical Society: Series B*, 80(1):5–31, 2017.

[Sandler *et al.*, 2009] Ted Sandler, John Blitzer, Partha P. Talukdar, and Lyle H. Ungar. Regularized learning with networks of features. In *Advances in Neural Information Processing Systems 21*, pages 1401–1408, 2009.

[Song *et al.*, 2012] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *J. of Machine Learning Research*, 13:1393–1434, 2012.

[van den Oord *et al.*, 2016] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proc. of the 33rd Int. Conf. on Machine Learning*, pages 1747–1756, 2016.

[Xie *et al.*, 2015] Pengtao Xie, Diyi Yang, and Eric Xing. Incorporating word correlation knowledge into topic modeling. In *Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 725–734, 2015.

[Zhang *et al.*, 2018] Qinyi Zhang, Sarah Filippi, Arthur Gretton, and Dino Sejdinovic. Large-scale kernel methods for independence testing. *Statistics and Computing*, 28(1):113–130, 2018.

[Zhang, 2009] Yi Zhang. Smart PCA. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence*, pages 1351–1356, 2009.

[Zhu *et al.*, 2014] Jun Zhu, Ning Chen, and Eric P. Xing. Bayesian inference with posterior regularization and applications to infinite latent SVMs. *J. of Machine Learning Research*, 15:1799–1847, 2014.