Mach Learn (2015) 100:399–424
DOI 10.1007/s10994-015-5497-9

Generalized Twin Gaussian processes using Sharma–Mittal divergence

Mohamed Elhoseiny[1] · **Ahmed Elgammal**[1]

Received: 2 February 2014 / Accepted: 13 April 2015 / Published online: 23 June 2015

Abstract There has been a growing interest in mutual information measures due to their wide range of applications in machine learning and computer vision. In this paper, we present a generalized structured regression framework based on Sharma–Mittal (SM) divergence, a relative entropy measure, which is introduced to in the machine learning community in this work. SM divergence is a generalized mutual information measure for the widely used Rényi, Tsallis, Bhattacharyya, and Kullback–Leibler (KL) relative entropies. Specifically, we study SM divergence as a cost function in the context of the Twin Gaussian processes (TGP) (Bo and Sminchisescu 2010), which generalizes over the KL-divergence without computational penalty. We show interesting properties of Sharma–Mittal TGP (SMTGP) through a theoretical analysis, which covers missing insights in the traditional TGP formulation. However, we generalize this theory based on SM-divergence instead of KL-divergence which is a special case. Experimentally, we evaluated the proposed SMTGP framework on several datasets. The results show that SMTGP reaches better predictions than KL-based TGP, since it offers a bigger class of models through its parameters that we learn from the data.

Keywords Sharma–Mittal entropy · Structured regression · Twin Gaussian processes · Pose estimation · Image reconstruction

Editors: Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo.

This research was partially funded by NSF award # 1409683.

✉ Mohamed Elhoseiny
m.elhoseiny@cs.rutgers.edu

Ahmed Elgammal
elgammal@cs.rutgers.edu

[1] Computer Science Department, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854-8019, USA

1 Introduction

Since 1950s, a lot of work has been done to measure information and probabilistic metrics. Claude Shannon (Shannon 2001) proposed a powerful framework to mathematically quantify information, which has been the foundation of the information theory and the development in communication, networking, and a lot of Computer Science applications. Many problems in Physics and Computer Science require a reliable measure of information divergence, which have motivated many mathematicians, physicists, and computer scientists to study different divergence measures. For instance, Rényi (Rényi 1960), Tsallis (Tsallis 1988) and Kullback–Leibler divergences (Gray 1990) have been applied in many Computer Science applications. They have been effectively used in machine learning for many tasks including subspace analysis (Learned-Miller and Fisher-III 2003; Póczos and Lõrincz 2005; Van Hulle 2008; Szab et al. 2007), facial expression recognition (Shan et al. 2005), texture classification (Hero et al. 2001), image registration (Kybic 2006), clustering (Aghagolzadeh et al. 2007), non-negative matrix factorization (Wang and Zhang 2013) and 3D pose estimation (Bo and Sminchisescu 2010).

In the Machine Learning community, a lot of attempts have been done to understand information and connect it to uncertainty. Many of proposed terminologies turns out to be different views of the same measure. For instance, Bregman Information (Banerjee et al. 2005), Statistical Information (DeGroot 1962), Csiszr–Morimoto f-divergence, and the gap between the expectations in Jensen's inequality (i.e., the Jensen gap) (Jensen 1906) turn out to be equivalent to the maximum reduction in uncertainty for convex functions, in contrast with the prior probability distribution (Reid and Williamson 2011).

A lot of work has been proposed in order to unify divergence functions (Amari and Nagaoka 2000; Reid and Williamson 2011; Zhang 2007, 2004). Cichocki and Ichi Amari (2010) considered explicitly the relationships between Alpha-divergence (Cichocki et al. 2008), Beta-divergence (Kompass 2007) and Gamma-divergence (Cichocki and Ichi Amari 2010); each of them is a single-parameter divergence measure. Then, Cichocki et al. (2011) introduced a two-parameter family. However, we study here a two-parameter divergence measure (Sharma 1975), investigated in the Physics community, which is interesting to be considered in the Machine Learning community.

Akturk et al. (2007), physicists,[1] studied an entropy measure called Sharma–Mittal on theormostatics in 2007, which was originally introduced by Sharma BD et al. (Sharma 1975). Sharma–Mittal (SM) divergence has two parameters (α and β), detailed later in Sect. 2. Akturk et al. (2007) discussed that SM entropy generalizes both Tsallis ($\beta \to \alpha$) and Rényi entropy ($\beta \to 1$) in the limiting cases of its two parameters; this was originally showed by (Masi 2005). In addition, it can be shown that SM entropy converges to Shannon entropy as $\alpha, \beta \to 1$. Aktürk et al also suggested a physical meaning of SM entropy, which is the free energy difference between the equilibrium and the off-equilibrium distribution. In 2008, SM entropy was also investigated in multidimensional harmonic oscillator systems (Aktürk et al. 2008). Similarly, SM relative entropy (mutual information) generalizes each of the Rényi, Tsallis and KL mutual information divergences. This work in physics domain motivated us to investigate SM Divergence in the Machine Learning domain.

A closed-form expression for SM divergence between two Gaussian distributions was recently proposed (Nielsen and Nock 2012), which motivated us to study this measure in structured regression setting. In this paper, we present a generalized framework for structured

[1] This work was proposed four years before Cichocki et al. (2011) and it was not considered either as a prior work in the Machine learning community as far as we know.

regression utilizing a family of divergence measures that includes SM divergence, Rényi divergence, Tsallis divergence and KL divergence. In particular, we study SM divergence within the context of Twin Gaussian processes (TGP), a state-of-the-art structured-output regression method. Bo and Sminchisescu (2010) proposed TGP as a structured prediction approach based on estimating the KL divergence from the input to output Gaussian Processes, denoted by KLTGP.[2] Since KL divergence is not symmetric, Bo and Sminchisescu (2010) also studied TGP based on KL divergence from the output to the input data, denoted by IKLTGP (Inverse KLTGP). In this work, we present a generalization for TGP using the SM divergence, denoted by SMTGP. Since SM divergence is a two-parameter family, we study the effect of these parameters and how they are related to the distribution of the data. In the context TGP, we show that these two parameters, α and β, could be interpreted as distribution bias and divergence order in the context of structured learning. We also highlight probabilistic causality direction of the SM objective function.[3] More specifically, there are six contributions to this paper

1. The first presentation of SM divergence in the Machine Learning Community
2. A generalized version of TGP based on of SM divergence to predict structured outputs; see Sect. 3.2.
3. A simplification to the SM divergence closed-form expression in (Nielsen and Nock 2012) for Multi-variate Gaussian Distribution,[4] which reduced both the cost function evaluation and the gradient computation, used in our prediction framework; see Sects. 3.3 and 3.4.
4. Theoretical analysis of TGP under SM divergence in Sect. 4.
5. A certainty measure, that could be associated with each structured output prediction, is argued in Sect. 4.2.
6. An experimental demonstration that SM divergence improves on KL divergence under TGP prediction by correctly tuning α and β through cross validation on two toy examples and three real datasets; see Sect. 5.

The rest of this paper is organized as follows: Sect. 2 presents background on SM Divergence and its available closed-form expression for multivariate Gaussians. Section 3 presents the optimization problem used in our framework and the derived analytic gradients. Section 4 presents our theoretical analysis on TGP under our framework from spectral perspective. Section 5 presents our experimental validation. Finally, Sect. 6 discusses and concludes our work.

2 Sharma–Mittal divergence

This section addresses a background on SM-divergence and its closed form for the multivariate Gaussian distribution.

[2] That is why it is called Twin Gaussian processes.

[3] This is mainly detailed in Sect. 4.

[4] This simplification could be useful out of the context TGP, while computing SM-divergence between two multi-variate distributions.

2.1 SM family divergence measures

The SM divergence, $D_{\alpha,\beta}(p : q)$, between two distributions $p(t)$ and $q(t)$ is defined as (Sharma 1975)

$$D_{\alpha,\beta}(p:q) = \frac{1}{\beta-1}\left(\int_{-\infty}^{\infty} p(t)^{\alpha} q(t)^{1-\alpha} dt\right)^{\frac{1-\beta}{1-\alpha}} - 1\Bigg), \quad \forall \alpha > 0, \alpha \neq 1, \beta \neq 1. \tag{1}$$

It was shown in (Akturk et al. 2007) that most of the widely used divergence measures are special cases of SM divergence. Each of the Rényi, Tsallis and KL divergences can be defined as limiting cases of SM divergence as follows:

$$\begin{aligned}
R_{\alpha}(p:q) &= \lim_{\beta\to 1} D_{\alpha,\beta}(p:q) = \frac{1}{\alpha-1} ln\left(\int_{-\infty}^{\infty} p(t)^{\alpha} q(t)^{1-\alpha} dt)\right), \quad \forall \alpha > 0, \alpha \neq 1.\\
T_{\alpha}(p:q) &= D_{\alpha,\alpha}(p:q) = \frac{1}{\alpha-1}\left(\int_{-\infty}^{\infty} p(t)^{\alpha} q(t)^{1-\alpha} dt - 1\right), \quad \forall \alpha > 0, \alpha \neq 1,\\
KL(p:q) &= \lim_{\beta\to 1,\alpha\to 1} D_{\alpha,\beta}(p:q) = \int_{-\infty}^{\infty} p(t).ln\left(\frac{p(t)}{q(t)} dt\right)
\end{aligned} \tag{2}$$

where $R_{\alpha}(p:q)$, $T_{\alpha}(p:q)$ and $KL(p:q)$ denotes Rényi, Tsallis, KL divergences respectively. We also found that Bhattacharyya divergence (Kailath 1967), denoted by $B(p:q)$ is a limit case of SM and Rényi divergences as follows

$$\begin{aligned}
B(p:q) &= 2 \cdot \lim_{\beta\to 1,\alpha\to 0.5} D_{\alpha,\beta}(p:q) = 2 \cdot \lim_{\alpha\to 0.5} R_{\alpha}(p:q)\\
&= -ln\left(\int_{-\infty}^{\infty} p(x)^{0.5} q(x)^{0.5} dx\right).
\end{aligned}$$

While SM is a two-parameter generalized entropy measure originally introduced by Sharma (1975), it is worth to mention that two-parameter family of divergence functions has been recently proposed in the machine learning community since 2011 (Cichocki et al. 2011; Zhang 2013). It is shown in (Cichocki and Ichi Amari 2010) that the Tsallis entropy is connected to the Alpha-divergence (Cichocki et al. 2008), and Beta-divergence (Kompass 2007),[5] while the Rényi entropy is related to the Gamma-divergences (Cichocki and Ichi Amari 2010). The Tsallis and Rényi relative entropies are two different generalization of the standard Boltzmann–Gibbs entropy (or Shannon information). However, we focus here on SM divergence for three reasons (1) It generalizes over a considerable family of functions suitable for structured regression problems (2) Possible future consideration of this measure in works that study entropy and divergence functions, (3) SM divergence has a closed-form expression, recently proposed for multivariate Gaussian distributions (Nielsen and Nock 2012), which is interesting to study.

Another motivations of this work is to study how the two parameters of the SM Divergence, as a generalized entropy measure, affect the performance of the structured regression problem. Here we show an analogy in the physics domain that motivates our study. As indicated by Masi (2005) in physics domain, it is important to understand that Tsallis and Rényi entropies are two different generalizations along two different paths. Tsallis generalizes to non-extensive

[5] Alpha and Beta divergence should not be confused with α and β parameters of Sharma Mittal divergence.

systems,[6] while Rényi to quasi-linear means.[7] SM entropy generalizes to non-extensive sets and non-linear means having Tsallis and Rényi measures as limiting cases. Hence, in TGP regression setting, this indicates resolving the trade-off of having a control of the direction of bias towards one of the distributions (i.e. input and output distributions) by changing α. It also allows higher-order divergence measure by changing β. Another motivation from Physics is that SM entropy is the only entropy that gives rise to a thermostatistics based on escort mean values[8] and admitting of a partition function (Frank and Plastino 2002).

2.2 SM-divergence closed-form expression for multivariate Gaussians

In order to solve optimization problems efficiently over relative entropy, it is critical to have a closed-form formula for the optimized function, which is SM relative entropy in our framework. Prediction over Gaussian Processes (Rasmussen and Williams 2005) is performed practically as a multivariate Gaussian distribution. Hence, we are interested in finding a closed-form formula for SM relative entropy of distribution $\mathcal{N}_q$ from $\mathcal{N}_p$, such that $\mathcal{N}_p = \mathcal{N}(\mu_p, \Sigma_p)$, and $\mathcal{N}_q = \mathcal{N}(\mu_q, \Sigma_q)$. In 2012, Frank Nielsen proposed a closed form expression for SM divergence (Nielsen and Nock 2012) as follows

$$D_{\alpha,\beta}(\mathcal{N}_p : \mathcal{N}_q) = \frac{1}{\beta - 1}\Bigg[\Bigg(\frac{|\Sigma_p|^{\alpha}|\Sigma_q|^{1-\alpha}}{|(\alpha\Sigma_p{}^{-1} + (1-\alpha)\Sigma_q{}^{-1})^{-1}|}\Bigg)^{-\frac{1-\beta}{2(1-\alpha)}} \cdot e^{-\frac{\alpha(1-\beta)}{2}\Delta\mu^T(\alpha\Sigma_p{}^{-1}+(1-\alpha)\Sigma_q{}^{-1})^{-1}\Delta\mu} - 1\Bigg] \tag{3}$$

where $0 \leq \alpha \leq 1$, $\Delta\mu = \mu_p - \mu_q$, $\alpha\Sigma_p{}^{-1} + (1-\alpha)\Sigma_q{}^{-1}$ is a positive definite matrix, and $|\cdot|$ denotes the matrix determinant. The following section builds on this SM closed-form expression to predict structured output under TGP, which leads an analytic gradient of the SMTGP cost function with cubic computational complexity. We then present a simplified expression of the closed-form expression in Eq. 3, which results in an equivalent SMTGP analytic gradient of quadratic complexity.

3 Sharma–Mittal TGP

In prediction problems, we expect that similar inputs produce similar predictions. This notion was adopted in (Bo and Sminchisescu 2010; Yamada et al. 2012) to predict structured output based on KL divergence between two Gaussian Processes. This section presents TGP for structured regression by minimizing SM relative entropy. We follow that by our theoretical analysis of TGPs in Sect. 4. We begin by introducing some notation. Let the joint distributions of the input and the output be defined as follows

$$\begin{aligned} &p(X, x) = \mathcal{N}_X(0, K_{X\cup x}),\ p(Y, y) = \mathcal{N}_Y(0, K_{Y\cup y}),\\ &K_{X\cup x} = \begin{bmatrix} K_X & K_X^x \\ K_X^{x\,T} & K_X(x,x) \end{bmatrix},\ K_{Y\cup y} = \begin{bmatrix} K_Y & K_Y^y \\ K_Y^{y\,T} & K_Y(y,y) \end{bmatrix} \end{aligned} \tag{4}$$

[6] i.e., In Physics, Entropy is considered to have an extensive property if its value depends on the amount of material present; Tsallis is an non-extensive entropy.

[7] i.e., Rényi entropy is could be interpreted as an averaging of quasi-arithmetic function Akturk et al. (2007).

[8] Escort mean values are useful theoretical tools, used in thermostatistics,for describing basic properties of some probability density function (Tsallis et al. 2009).

where $x_{(d_x \times 1)}$ is a new input test point, whose unknown outcome is $y_{(d_y \times 1)}$ and the training set is $X_{(N \times d_x)}$ and $Y_{(N \times d_y)}$ matrices. K_X is an $N \times N$ matrix with $(K_X)_{ij} = k_X(x_i, x_j)$, such that $k_X(x_i, x_j)$ is the similarity kernel between x_i and x_j. K_X^x is an $N \times 1$ column vector with $(K_X^x)_i = k_X(x_i, x)$. Similarly, K_Y is an $N \times N$ matrix with $(K_Y)_{ij} = k_Y(y_i, y_j)$, such that $k_Y(y_i, y_j)$ is the similarity kernel between y_i and y_j, and K_Y^y is an $N \times 1$ column vector with $(K_Y^y)_i = k_Y(y_i, y)$. By applying Gaussian-RBF kernel functions, the similarity kernels for inputs and outputs will be in the form of $k_X(x_i, x_j) = exp(\frac{-\|x_i - x_j\|^2}{2\rho_x^2}) + \lambda_X \delta_{ij}$ and $k_Y(y_i, y_j) = exp(\frac{-\|y_i - y_j\|^2}{2\rho_y^2}) + \lambda_Y \delta_{ij}$, respectively, where ρ_x and ρ_y are the corresponding kernel bandwidths, λ_X and λ_Y are regularization parameters to avoid overfitting and to handle noise in the data, and $\delta_{ij} = 1$ if $i = j$, 0 otherwise.

3.1 KLTGP and IKLTGP prediction

Bo and Sminchisescu (2010) firstly proposed TGP which minimizes the Kullback–Leibler divergence between the marginal GP of inputs and outputs. However, they were focusing on the Human Pose Estimation problem. As a result, the estimated pose using TGP is given as the solution of the following optimization problem (Bo and Sminchisescu 2010)

$$\hat{y} = \underset{y}{\text{argmin}}[L_{KL}(x, y) = k_Y(y, y) - 2K_Y^{yT} u_x - \eta_x log(k_Y(y, y) - K_Y^{yT}(K_Y)^{-1} K_Y^y)] \tag{5}$$

where $u_x = (K_X)^{-1} K_X^x$, $\eta_x = k_X(x, x) - K_X^{xT} u_X$. The analytical gradient of this cost function is defined as follows (Bo and Sminchisescu 2010)

$$\frac{\partial L_{KL}(x, y)}{\partial y^{(d)}} = \frac{\partial k_Y(y, y)}{\partial y^{(d)}} - 2u_x^T \frac{\partial K_Y^y}{\partial y^{(d)}} - \eta_x \frac{log\left(\frac{\partial k_Y(y,y)}{\partial y^{(d)}} - 2K_Y^{yT}(K_Y)^{-1}\frac{\partial K_Y^y}{\partial y^{(d)}}\right)}{k_Y(y, y) - K_Y^{yT}(K_Y)^{-1} K_Y^y} \tag{6}$$

where d is the dimension index of the output y. For Gaussian kernels, we have

$$\frac{\partial k_Y(y, y)}{\partial y^{(d)}} = 0, \frac{\partial K_Y^y}{\partial y^{(d)}} = \begin{bmatrix} -\frac{1}{\rho_y^2}(y^{(d)} - y_1^{(d)}) k_Y(y, y_1) \\ -\frac{1}{\rho_y^2}(y^{(d)} - y_2^{(d)}) k_Y(y, y_2) \\ \cdots \\ -\frac{1}{\rho_y^2}(y^{(d)} - y_N^{(d)}) k_Y(y, y_N) \end{bmatrix}.$$

This optimization problem can be solved using a second order BFGS quasi-Newton optimizer with cubic polynomial line search for optimal step size selection. Since KL divergence is not symmetric, Bo and Sminchisescu (2010) also studied inverse KL-divergence between the output and the input distribution under TGP; we denote this model as IKLTGP. Equations 7 and 8 show the IKLTGP cost function and its corresponding gradient.[9]

[9] We derived this equation since it was not provided in (Bo and Sminchisescu 2010).

$$\hat{y} = \underset{y}{\text{argmin}}[L_{IKL}(x, y) = -2K_X^{x\,T} u_y + u_y^T K_X u_y + \eta_y(log(\eta_y) - log(\eta_x))], \quad u_y = K_Y^{-1} K_Y^y, \eta_y = k_Y(y, y) - K_Y^{y\,T} u_y \tag{7}$$

$$\frac{\partial L_{IKL}(x, y)}{\partial y^{(d)}} = -2K_X^{x\,T} K_Y^{-1} \frac{\partial K_Y^y}{\partial y^{(d)}} + 2u_y^T K_X K_Y^{-1} \frac{\partial K_Y^y}{\partial y^{(d)}} - 2(log(\eta_y) - log(\eta_x) + 1) K_Y^{y\,T} K_Y^{-1} \frac{\partial K_Y^y}{\partial y^{(d)}} \tag{8}$$

From Eqs. 6 and 8, it is not hard to see that the gradients of KLTGP and IKLTGP can be computed in quadratic complexity, given that K_X^{-1} and K_Y^{-1} are precomputed once during training and stored, as it depends only on the training data. This quadratic complexity of KLTGP gradient presents a benchmark for us to compute the gradient for SMTGP in $O(N^2)$. Hence, we address this benchmark in our framework, as detailed in the following subsections.

3.2 SMTGP prediction

By applying the closed-form in Eq. 3, SM divergence between $p(X, x)$ and $p(Y, y)$ becomes in the following form

$$D_{\alpha,\beta}(p(X, x) : p(Y, y)) = \frac{1}{\beta - 1}\left[\left(\frac{|K_{X\cup x}|^{\alpha} |K_{Y\cup y}|^{1-\alpha}}{|(\alpha K_{X\cup x}^{-1} + (1-\alpha) K_{Y\cup y}{}^{-1})^{-1}|}\right)^{-\frac{1-\beta}{2(1-\alpha)}} - 1\right] \tag{9}$$

From matrix algebra, $|K_{X\cup x}| = |K_X|(k_X(x, x) - K_X^{x\,T} K_X{}^{-1} K_X^x)$. Similarly, $|K_{Y\cup y}| = |K_Y|(k_Y(y, y) - K_Y^{y\,T} K_Y{}^{-1} K_Y^y)$. Hence, Eq. 9 could be rewritten as follows

$$D_{\alpha,\beta}(p(X, x) : p(Y, y)) = \frac{|K_X|^{\frac{-\alpha(1-\beta)}{2(1-\alpha)}} |K_Y|^{\frac{-(1-\beta)}{2}}}{\beta - 1} \cdot (k_X(x, x) - K_X^{x\,T} K_X{}^{-1} K_X^x)^{\frac{-\alpha(1-\beta)}{2(1-\alpha)}} \cdot (k_Y(y, y) - K_Y^{y\,T} K_Y{}^{-1} K_Y^y)^{\frac{-(1-\beta)}{2}} \cdot |\alpha K_{X\cup x}^{-1} + (1-\alpha) K_{Y\cup y}{}^{-1}|^{\frac{-(1-\beta)}{2(1-\alpha)}} - \frac{1}{\beta - 1} \tag{10}$$

$|K_X|^{\frac{-\alpha(1-\beta)}{2(1-\alpha)}} |K_Y|^{\frac{-(1-\beta)}{2}}$ is a positive constant, since K_X and K_Y are positive definite matrices. Hence, it could be removed from the optimization problem. Same argument holds for $|K_{X\cup x}| = |K_X|(k_X(x, x) - K_X^{x\,T} K_X{}^{-1} K_X^x) > 0$, so $(k_X(x, x) - K_X^{x\,T} K_X{}^{-1} K_X^x) > 0$ could be also removed from the cost function. Having removed these constants, the prediction function reduces to minimizing the following expression

$$L_{\alpha,\beta}(p(X, x) : p(Y, y)) = \frac{1}{\beta - 1}(k_Y(y, y) - K_Y^{y\,T} K_Y{}^{-1} K_Y^y)^{\frac{-(1-\beta)}{2}} \cdot |\alpha K_{X\cup x}^{-1} + (1-\alpha) K_{Y\cup y}{}^{-1}|^{\frac{-(1-\beta)}{2(1-\alpha)}} \tag{11}$$

It is worth mentioning that $K_{X\cup x}^{-1}$ is quadratic to compute, given that K_X^{-1} is precomputed during the training; see Appendix 1.

To avoid numerical instability problems in Eq. 11 (introduced by determinant of the large matrix $(\alpha K_{X\cup x}^{-1} + (1-\alpha) K_{Y\cup y}{}^{-1})$, we optimized $\log(L_{\alpha,\beta}(N_X : N_Y))$ instead of $L_{\alpha,\beta}(N_X : N_Y)$. We derived the gradient of $\log(L_{\alpha,\beta}(N_X : N_Y))$ by applying the matrix calculus directly on the logarithm of Eq. 11, presented below; the derivation steps are detailed in Appendix 2.

$$\frac{\partial L_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y^{(d)}} = (1-\beta)\left[\frac{{K_Y^y}^T {K_Y}^{-1} \frac{\partial K_Y^y}{\partial y^{(d)}}}{(k_Y(y,y) - {K_Y^y}^T {K_Y}^{-1} K_Y^y)} + \mu_y^T \cdot \frac{\partial K_Y^y}{\partial y^{(d)}}\right] \quad (12)$$

μ_y is computed by solving the following linear system of equations $(\alpha K_{Y\cup y} K_{X\cup x}^{-1} K_{Y\cup y} + (1-\alpha)K_{Y\cup y})\mu_y^{'} = [0,0,\ldots 0,1]^T$, μ_y is the first N elements in $\mu_y^{'}$, which is a vector of $N+1$ elements. The computational complexity of the gradient in Eq. 12 is cubic at test time, due to solving this system. On the other hand, the gradient for KLTGP is quadratic. This problem motivated us to investigate the cost function to achieve a quadratic complexity of the gradient computation for SMTGP.

3.3 Quadratic SMTGP prediction

We start by simplifying the closed-form expression introduced in (Nielsen and Nock 2012), which led to the $O(N^3)$ gradient computation.

Lemma 3.1 *SM-divergence between two N-dimensional multivariate Gaussians* $\mathcal{N}_p = \mathcal{N}(0, \Sigma_p)$ *and* $\mathcal{N}_q = \mathcal{N}(0, \Sigma_q)$ *can be written as*

$$D'_{\alpha,\beta}(\mathcal{N}_p : \mathcal{N}_q) = \frac{1}{\beta - 1}\left[\left(\frac{|\Sigma_p|^{1-\alpha}|\Sigma_q|^{\alpha}}{|\alpha\Sigma_q + (1-\alpha)\Sigma_p|}\right)^{\frac{(1-\beta)}{2(1-\alpha)}} - 1\right] \quad (13)$$

Proof Under TGP setting, the exponential term in Eq. 3 vanishes to 1, since $\Delta\mu = 0$ (i.e. $\mu_p = \mu_q = 0$). Then, $\frac{|\Sigma_p|^{\alpha}|\Sigma_q|^{1-\alpha}}{|(\alpha\Sigma_p^{-1} + (1-\alpha){\Sigma_q}^{-1})^{-1}|}$ could be simplified as follows:

$$\begin{aligned}
&= \frac{|\Sigma_p|^{\alpha}|\Sigma_q|^{1-\alpha}}{|(\alpha\Sigma_p^{-1} + (1-\alpha){\Sigma_q}^{-1})|^{-1}}, \text{ since } |A^{-1}| = \frac{1}{|A|} \\
&= \frac{|\Sigma_p|^{\alpha}|\Sigma_q|^{1-\alpha}}{|\Sigma_p^{-1}(\alpha\Sigma_q + (1-\alpha)\Sigma_p){\Sigma_q}^{-1}|^{-1}}, \text{ by factorization} \\
&= \frac{|\Sigma_p|^{\alpha}|\Sigma_q|^{1-\alpha}}{|\Sigma_p||\alpha\Sigma_q + (1-\alpha)\Sigma_p|^{-1}|\Sigma_q|}, \text{ since } |AB| = |A||B| \\
&= \frac{|\alpha\Sigma_q + (1-\alpha)\Sigma_p|}{|\Sigma_p|^{1-\alpha}|\Sigma_q|^{\alpha}}, \text{ by rearrangement}
\end{aligned} \quad (14)$$

□

We denote the original closed-form expression as $D_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$, while the simplified form as $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$. After applying the simplified SM expression in Lemma 3.1 to measure the divergence between $p(X,x)$ and $p(Y,y)$, the new cost function becomes in the following form

$$\begin{aligned}
&D'_{\alpha,\beta}(p(X,x):p(Y,y)) = \frac{1}{\beta-1}\left[\left(\frac{|K_{X\cup x}|^{1-\alpha}|K_{Y\cup y}|^{\alpha}}{|(1-\alpha)K_{X\cup x} + \alpha K_{Y\cup y}|}\right)^{\frac{1-\beta}{2(1-\alpha)}} - 1\right] \\
&= \frac{1}{\beta-1}\left(|K_{X\cup x}|^{\frac{1-\beta}{2}}|K_{Y\cup y}|^{\frac{\alpha(1-\beta)}{2(1-\alpha)}}|(1-\alpha)K_{X\cup x} + \alpha K_{Y\cup y}|^{\frac{-(1-\beta)}{2(1-\alpha)}}\right) - \frac{1}{\beta-1}, \\
&|K_{Y\cup y}|^{\frac{\alpha(1-\beta)}{1-\alpha}} = |K_Y|^{\frac{\alpha(1-\beta)}{(1-\alpha)}} \cdot (k_Y(y,y) - {K_Y^y}^T {K_Y}^{-1} K_Y^y)^{\frac{\alpha(1-\beta)}{(1-\alpha)}}, \\
&|(1-\alpha)K_{X\cup x} + \alpha K_{Y\cup y}|^{\frac{-(1-\beta)}{2(1-\alpha)}} = |(1-\alpha)K_X + \alpha K_Y|^{\frac{-(1-\beta)}{2(1-\alpha)}} \cdot \\
&({K_{xy}}^{\alpha} - {K_{XY}^{xy}}^T((1-\alpha)K_X + \alpha K_Y)^{-1} K_{XY}^{xy})^{\frac{-(1-\beta)}{2(1-\alpha)}}
\end{aligned} \quad (15)$$

where ${K_{xy}}^{\alpha} = (1-\alpha)k_X(x,x) + \alpha k_Y(y,y)$, $K_{XY}^{xy} = (1-\alpha)K_X^x + \alpha K_Y^y$. Since $|K_{X\cup x}|^{\frac{1-\beta}{2}}$, $|K_Y|^{\frac{\alpha(1-\beta)}{(1-\alpha)}}$, and $|(1-\alpha)K_X + \alpha K_Y|^{\frac{-(1-\beta)}{2(1-\alpha)}}$ are multiplicative positive constants that do not depend on y, they can be dropped from the cost function. Also, $-\frac{1}{\beta-1}$ is an additive constant that can be ignored under optimization. After ignoring these multiplicative positive constants and the added constant, the improved SMTGP cost function reduces to

$$L'_{\alpha,\beta}(p(X,x):p(Y,y)) = \frac{1}{\beta-1}\Big[(k_Y(y,y) - {K_Y^y}^T {K_Y}^{-1} K_Y^y)^{\frac{\alpha(1-\beta)}{2(1-\alpha)}} \cdot ({K_{xy}}^{\alpha} - {K_{XY}^{xy}}^T((1-\alpha)K_X + \alpha K_Y)^{-1} K_{XY}^{xy})^{\frac{-(1-\beta)}{2(1-\alpha)}}\Big] \tag{16}$$

In contrast to $L_{\alpha,\beta}$ in Eq. 11, $L'_{\alpha,\beta}$ does not involve a determinant of a large matrix. Hence, we predict the output y by directly[10] minimizing $L'_{\alpha,\beta}$ in Eq. 16. Since the cost function has two factors that does depend on y, we follow the rule that if $g(y) = c \cdot f(y) \cdot r(y)$ where c is a constant, $f(y)$ and $r(y)$ are functions, then $\frac{\partial g(y)}{\partial y} = c \cdot (\frac{\partial f(y)}{\partial y} r(y) + f(y)\frac{\partial r(y)}{\partial y})$, which interprets the two terms of the derived gradient below, where $f(y) = (k_Y(y,y) - {K_Y^y}^T {K_Y}^{-1} K_Y^y)^{\frac{\alpha(1-\beta)}{2(1-\alpha)}}$, $r(y) = ({K_{xy}}^{\alpha} - {K_{XY}^{xy}}^T((1-\alpha)K_X + \alpha K_Y)^{-1} K_{XY}^{xy})^{\frac{-(1-\beta)}{2(1-\alpha)}}$, $c = \frac{1}{\beta-1}$

$$\begin{aligned}\frac{\partial L'(\alpha,\beta)}{\partial y^{(d)}}(p(X,x):p(Y,y)) = \frac{1}{\beta-1}\Big[&\frac{\alpha(1-\beta)}{2(1-\alpha)}(k_Y(y,y) - {K_Y^y}^T {K_Y}^{-1} K_Y^y)^{\frac{\alpha(1-\beta)}{2(1-\alpha)}-1} \cdot \\ &(\frac{\partial k_Y(y,y)}{\partial y^{(d)}} - 2 \cdot {K_Y^y}^T {K_Y}^{-1} \frac{\partial K_Y^y}{\partial y^{(d)}}) \cdot \\ &({K_{xy}}^{\alpha} - {K_{XY}^{xy}}^T((1-\alpha)K_X + \alpha K_Y)^{-1} K_{XY}^{xy})^{\frac{-(1-\beta)}{2(1-\alpha)}} \\ &+ (k_Y(y,y) - {K_Y^y}^T {K_Y}^{-1} K_Y^y)^{\frac{\alpha(1-\beta)}{2(1-\alpha)}} \cdot \frac{-(1-\beta)}{2(1-\alpha)} \\ &({K_{xy}}^{\alpha} - {K_{XY}^{xy}}^T((1-\alpha)K_X + \alpha K_Y)^{-1} K_{XY}^{xy})^{\frac{-(1-\beta)}{2(1-\alpha)}-1} \cdot \\ &(\alpha\frac{\partial k_Y(y,y)}{\partial y^{(d)}} - 2 \cdot {K_{XY}^{xy}}^T((1-\alpha)K_X + \alpha K_Y)^{-1} \cdot \alpha \frac{\partial K_Y^y}{\partial y^{(d)}})\Big]\end{aligned} \tag{17}$$

The computational complexity of the cost function in Eq. 16 and the gradient in Eq. 17 is quadratic at test time (i.e. $O(N^2)$) on number of the training data. Since ${K_Y}^{-1}$ and $(\alpha K_X + (1-\alpha)K_Y)^{-1}$ depend only on the training points, they are precomputed in the training time. Hence, our hypothesis, about the quadratic computational complexity of improved SMTGP prediction function and gradient, is true since the remaining computations are $O(N^2)$. This indicates the advantage of using our closed-form expression for SM divergence in Lemma 3.1 against the closed-form proposed in (Nielsen and Nock 2012) with cubic complexity. However, both expression are equivalent, it is straight forward to compute the gradient in quadratic complexity from $D'(\alpha,\beta)$ expression.

3.4 Advantage of $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ against $D_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ out of SMTGP context

The previous subsection shows that the computational complexity of SMTGP prediction was decreased significantly using our $D'_{\alpha,\beta}$ at test time to be quadratic, compared to cubic

[10] There is no need to optimize over the logarithm of $L'_{\alpha,\beta}$ because there is no numerical stability problem.

complexity for $D_{\alpha,\beta}$. Out of the TGP context, we show here another general advantage of using our proposed closed-form expression to generally compute SM-divergence between two Gaussian distributions $\mathcal{N}_p$ and $\mathcal{N}_q$. $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ is 1.67 times faster to compute than $D_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ under $\Delta\mu = 0$ condition. This is since $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ needs N^3 operations which is much less than $5N^3/3$ operations needed to compute $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$(i.e., requires less matrix operations); see Appendix 3 for the proof. We conclude this section by a general form of Lemma 3.1 in Eq. 18, where $\Delta\mu \neq 0$. This equation was achieved by refactorizing the exponential term and using matrix identities.

$$D'_{\alpha,\beta}(\mathcal{N}_p : \mathcal{N}_q) = \frac{1}{\beta - 1}\left[\left(\frac{|\Sigma_p|^{1-\alpha}|\Sigma_q|^{\alpha}}{|\alpha\Sigma_q + (1-\alpha)\Sigma_p|}\right)^{\frac{(1-\beta)}{2(1-\alpha)}} \cdot e^{-\frac{\alpha(1-\beta)}{2}\Delta\mu^T \Sigma_q(\alpha\Sigma_q + (1-\alpha)\Sigma_q)^{-1}\Sigma_p\Delta\mu} - 1\right] \tag{18}$$

In case $\Delta\mu \neq 0$, $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ is 1.5 times faster than computing $D_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$. This is since $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ needs $4N^3/3$ operations in this case which is less than $2N^3$ operations needed to compute $D'_{\alpha,\beta}(\mathcal{N}_p, \mathcal{N}_q)$ under $\Delta\mu \neq 0$; see Appendix 3. This indicates that the simplifications, we provided in this work, could be used to generally speedup the computation of SM divergence between two Gaussian Distributions, beyond the context of TGPs.

4 Theoretical analysis

In order to understand the role of α and β parameters of SMTGP, we performed an eigen analysis of the cost function in Eq. 15. Generally speaking, the basic notion of TGP prediction, is to extend the dimensionality of the divergence measure from N training examples to $N+1$ examples, which involves the test point x and the unknown output y. Hence, we start by discussing the extension of a general Gaussian Process from K_Z (e.g. K_X and K_Y) to $K_{Z\cup z}$ (e.g. $K_{X\cup x}$ and $K_{Y\cup y}$), where Z is any domain and z is the point that extends K_Z to $K_{Z\cup z}$, detailed in Sect. 4.1. Based on this discussion, we will derive two lemmas to address some properties of the SMTGP prediction in Sect. 4.2, which will lead to a probabilistic interpretation that we provide in Sect. 4.3.

4.1 A Gaussian process from N to $N+1$ points

In this section, we will use a superscript to disambiguate between the kernel matrix of size N and $N+1$, i.e. K^N and K^{N+1}. Let $f(\mathbf{z}) = \mathcal{GP}(m(z) = 0,\ k(\mathbf{z}, \mathbf{z}'))$ be a Gaussian process on an arbitrary domain Z. Let $GP^N = \mathcal{N}(0, K^N)$ be the marginalization of the given Gaussian process over the N training points (i.e. $\{z_i\}, i = 1 : N$). Let $GP^{N+1} = \mathcal{N}(0, K^{N+1})$ be the extension of the GP^N be the marginalization of $f(z)$ over $N+1$ points after adding the $N+1^{th}$ point (i.e. z).[11] The kernel matrix K^{N+1} is written in terms of K^N as follows

$$K^{N+1} = \begin{bmatrix} K^N & v \\ v^T & k(z,z) \end{bmatrix} \tag{19}$$

where $v = [k(z, z_1) \ldots k(z, z_N)]^T$. The matrix determinant of K^{N+1} is related to K^N by

$$|K^{N+1}| = \eta \cdot |K^N|, \quad \eta = k(z,z) - v^T(K^N)^{-1}v. \tag{20}$$

[11] This is linked to the extending $p(X)$ to $p(X, x)$ and $p(Y)$ to $p(Y, y)$ by x and y respectively.

Since multivariate Gaussian distribution is a special case of the elliptical distributions, the eigen values of any covariance matrix (e.g. K^N, K^{N+1}) are interpreted as variance of the distribution in the direction of the corresponding eigen vectors. Hence, the determinant of the matrix (e.g. $|K^N|$, $|K^{N+1}|$) generalizes the notion of the variance in multiple dimensions as the volume of this elliptical distribution, which is oriented by the eigen vectors. From this notion, one could interpret η as the ratio by which the variance (uncertainty) of the marginalized Gaussian process is scaled, introduced by the new data point z. Looking closely at η, we can notice

(1) $0 \leq \eta \leq k(z, z)$, since $|K^N| > 0$, $|K^{N+1}| > 0$, and $v^T (K^N)^{-1} v \geq 0$.
(2) In the case of the regularized Gaussian kernel, we used in our work, $k(z, z) = 1 + \lambda$, and hence $0 \leq \eta \leq 1 + \lambda$
(3) η decreases as the new data point get closer to the N points. This situation makes v highly correlated with the eigen vectors of small eigen values of K^N, since the term $v^T (K^N)^{-1} v$ is maximized as v points to the smallest principal component of K^N (i.e. the direction of the maximum certainty). Hence, η is an uncertainty measure, which is minimized as the new data point z produces a vector v, that maximizes the certainty of the data under $\mathcal{N}(0, K^N)$, which could be thought as a measurement proportional to $1/p(z|z_1 : z_N)$. Computing η on the input space X makes it equivalent to the predictive variance of Gaussian Process Regression (GPR) prediction (Rasmussen and Williams 2005) (Chapter 2), which depends only on the input space. However, we are discussing η as an uncertainty extension from N to $N + 1$ on an arbitrary domain, which is beneficial for SMTGP analysis that follows.

4.2 TGP cost function analysis

We start by the optimization function of the SMTGP prediction, defined as

$$\begin{aligned} \hat{y}(\alpha, \beta) &= \underset{y}{\operatorname{argmin}} \left[D'_{\alpha,\beta} \left(GP_{X \cup x} : GP_{Y \cup y} \right) \right. \\ &= \left. \frac{1}{\beta - 1} \left(\left(\frac{|K_{X \cup x}|^{1-\alpha} |K_{Y \cup y}|^{\alpha}}{|(1-\alpha) K_{X \cup x} + \alpha K_{Y \cup y}|} \right)^{\frac{1-\beta}{2(1-\alpha)}} - 1 \right) \right) \right] \end{aligned} \tag{21}$$

where $D'_{\alpha,\beta}(\cdot, \cdot)$ is as defined in Eq. 15. As detailed in Sect. 3, SM divergence, involves the determinant of three matrices of size $N + 1 \times N + 1$, namely $K_{X \cup x}$, $K_{Y \cup y}$, and $\alpha K_{Y \cup y} + (1 - \alpha) K_{X \cup x}$. Hence, We have three uncertainty extensions from N to $N + 1$, as follows

$$\begin{gathered} |K_{X \cup x}| = \eta_x \cdot |K_X|, \qquad |K_{Y \cup y}| = \eta_y \cdot |K_Y| \\ |\alpha K_{Y \cup y} + (1 - \alpha) K_{X \cup x}| = \eta_{x,y}(\alpha) \cdot |\alpha K_Y + (1 - \alpha) K_X|, \\ \text{where } \eta_{x,y}(\alpha) = \alpha K_Y(y, y) + (1 - \alpha) K_X(x, x) - v_{xy}(\alpha)^T (\alpha K_Y + (1 - \alpha) K_X)^{-1} v_{xy}(\alpha), \\ v_{xy}(\alpha) = \alpha K_Y^y + (1 - \alpha) K_X^x \end{gathered} \tag{22}$$

It might not be straightforward to think about $\alpha K_{Y \cup y} + (1 - \alpha) K_{X \cup x}$ within TGP formulation as a kernel matrix defined on $X \times Y$ space in Eq. 22. This gives an interpretation of the constraint that $0 \leq \alpha \leq 1$ in Eqs. 3 and 13. Since $\alpha k_Y(y_i, y_j) + (1 - \alpha) k_X(x_i, x_j)$ is a weighted sum of valid kernels with positive weights, then $\alpha K_{Y \cup y} + (1 - \alpha) K_{X \cup x}$ is a valid kernel matrix on $X \times Y$ space. From Eqs. 21 and 22, we derived with the following two Lemmas.

Lemma 4.1 *Under SMTGP,* $\varphi_\alpha(x,y) = \frac{\eta_x^{1-\alpha}\eta_y^{\alpha}}{\eta_{x,y}(\alpha)} \leq (\int_{-\infty}^{\infty} p_Y(t)^{\alpha} p_X(t)^{1-\alpha} dt)^{-2}$, $(\int_{-\infty}^{\infty} p_Y(t)^{\alpha} p_X(t)^{1-\alpha} dt)^{-2} = \frac{|\alpha K_Y + (1-\alpha)K_X|}{|K_X|^{1-\alpha}|K_Y|^{\alpha}} \geq 1$

Proof Directly from the definition of SM TGP in Eqs. 21 and 22, SM TGP cost function could be written as,

$$\begin{aligned} \hat{y}(\alpha,\beta) &= \underset{y}{\operatorname{argmin}} \Bigg[D'_{\alpha,\beta}(p(X,x) : p(Y,y)) \\ &= \frac{1}{\beta-1}\left(\left(\frac{|K_X|^{1-\alpha}.\eta_x^{1-\alpha}.|K_Y|^{\alpha}.\eta_y^{\alpha}}{|\alpha K_Y + (1-\alpha)K_X|.\eta_{x,y}(\alpha)}\right)^{\frac{(1-\beta)}{2(1-\alpha)}} - 1\right)\Bigg] \end{aligned} \tag{23}$$

Comparing Eq. 1 to Eq. 23, then

$$\left(\frac{|K_X|^{1-\alpha}.\eta_x^{1-\alpha}.K_Y|^{\alpha}.\eta_y^{\alpha}}{|\alpha K_Y + (1-\alpha)K_X|.\eta_{x,y}(\alpha)}\right)^{\frac{1}{2}} = \int_{-\infty}^{\infty} p_{X,x}(t)^{\alpha} p_{Y,y}(t)^{1-\alpha} dt \leq 1,$$

and since $\eta_{x,y}(\alpha) > 0$ and $\eta_y > 0$, then

$$\varphi_\alpha(x,y) = \frac{\eta_x^{1-\alpha}\eta_y^{\alpha}}{\eta_{x,y}(\alpha)} \leq \frac{|\alpha K_Y + (1-\alpha)K_X|}{|K_X|^{1-\alpha}|K_Y|^{\alpha}},$$

and since $\int_{-\infty}^{\infty} p_X(t)^{\alpha} p_Y(t)^{1-\alpha} dt = \left(\frac{|K_X|^{1-\alpha}|K_Y|^{\alpha}}{|\alpha K_Y + (1-\alpha)K_X|}\right)^{\frac{1}{2}}$ and

$\int_{-\infty}^{\infty} p_X(t)^{\alpha} p_Y(t)^{1-\alpha} dt \leq 1$, then $\frac{|K_X|^{1-\alpha}|K_Y|^{\alpha}}{|\alpha K_Y + (1-\alpha)K_X|} \leq 1$. □

Lemma 4.2 *Under SMTGP and* $0 < \alpha < 1$, $\hat{y}(\alpha,\beta)$ *maximizes* $\varphi_\alpha(x,y) = \frac{\eta_x^{1-\alpha}.\eta_y^{\alpha}}{\eta_{x,y}(\alpha)} \leq \frac{|\alpha K_Y + (1-\alpha)K_X|}{|K_X|^{1-\alpha}|K_Y|^{\alpha}}$ *and it does not depend on* β *theoretically.*

Proof We start by the claim that $\hat{y}(\alpha, 1-\tau) = \hat{y}(\alpha, 1+\zeta), 0 < \alpha < 1, \tau > 0, \zeta > 0$ and both predictions are achieved by maximizing $(\int_{-\infty}^{\infty} p_{(Y,y)}(t)^{\alpha} p_{(X,x)}(t)^{1-\alpha} dt)^2 = \frac{|K_X|^{1-\alpha}|K_Y|^{\alpha}\eta_x^{1-\alpha}\eta_y^{\alpha}}{|\alpha K_Y + (1-\alpha)K_X|\eta_{x,y}\alpha} \leq 1$, which is $\propto \varphi_\alpha(x,y)$, where $p_{(X,x)} = \mathcal{N}(0, K_{X\cup x})$ and $p(Y,y) = \mathcal{N}(0, K_{Y\cup y})$, $p(X) = \mathcal{N}(0, K_X)$ and $p(Y) = \mathcal{N}(0, K_Y)$. This claim indicates that the cost functions $D'_{\alpha,1-\tau}(p(X,x) : p(Y,y))$ and $D'_{\alpha,1+\zeta}(p(X,x) : p(Y,y))$ are equivalent.

Let us introduce $Z_\alpha(y) = \frac{|K_{X\cup x}|^{1-\alpha}|K_{Y\cup y}|^{\alpha}}{|\alpha K_{Y\cup y} + (1-\alpha)K_{X\cup x}|} = (\int_{-\infty}^{\infty} p_{X,x}(t)^{\alpha} p_{Y,y}(t)^{1-\alpha} dt)^2 \leq 1$. From this notation, $D'_{\alpha,1-\tau}(p(X,x) : p(Y,y))$ and $D'_{\alpha,1+\zeta}(p(X,x) : p(Y,y))$ could be re-written as

$$\begin{aligned} D_{\alpha,1-\tau}(p(X,x) : p(Y,y)) &= \frac{1}{-\tau}\left[\left(Z_\alpha(y)\right)^{\frac{\tau}{2(1-\alpha)}} - 1\right], \\ D_{\alpha,1+\zeta}(p(X,x) : p(Y,y)) &= \frac{1}{\zeta}\left[\left(Z_\alpha(y)\right)^{\frac{-\zeta}{2(1-\alpha)}} - 1\right] \end{aligned} \tag{24}$$

From Eq. 24 and under the assumption that $0 < \alpha < 1$ and $Z_\alpha(y) \leq 1$, then $D_{\alpha,1-\tau}(p(X,x) : p(Y,y)) \geq 0$, $D_{\alpha,1+\zeta}(p(X,x) : p(Y,y)) \geq 0$. Both are clearly minimized as $Z_\alpha(y)$ approaches 1 (i.e. maximized, since $Z_\alpha(y) \leq 1$). Comparing Eqs. 23 and 24, $Z_\alpha(y) = (\int_{-\infty}^{\infty} p_{X,x}(t)^{\alpha} p_{Y,y}(t)^{1-\alpha} dt)^2 = \frac{|K_{X\cup x}|^{1-\alpha}|K_{Y\cup y}|^{\alpha}}{|\alpha K_{Y\cup y} + (1-\alpha)K_{X\cup x}|} = \frac{|K_X|^{1-\alpha}.\eta_x^{1-\alpha}.K_Y|^{\alpha}.\eta_y^{\alpha}}{|\alpha K_Y + (1-\alpha)K_X|.\eta_{x,y}(\alpha)} \propto$

$\varphi_\alpha(x, y) = \frac{\eta_x^{1-\alpha}\eta_y^\alpha}{\eta_{x,y}(\alpha)}$, since $|\alpha K_Y + (1-\alpha)K_X|$, $|K_X|$, and $|K_Y|$ do not depend on the predicted output $\hat{y}$. This indicates that SMTGP optimization function is inversely proportional to $\varphi_\alpha(x, y)$, we upper-bounded in Lemma 4.1. Hence, it is not hard to see that ζ and τ controls whether to maximize $\varphi_\alpha(x, y)^{\frac{\tau}{1-\alpha}}$ or maximize $-\varphi_\alpha(x, y)^{\frac{-\zeta}{1-\alpha}}$, which are equivalent. This directly leads to that $\hat{y}(\alpha, \beta)$ maximizes $\frac{\eta_x^{1-\alpha}.\eta_y^\alpha}{\eta_{x,y}(\alpha)} \leq \frac{|\alpha K_Y+(1-\alpha)K_X|}{|K_X|^{1-\alpha}|K_Y|^\alpha}$ and it does not depend on β theoretically. □

The proof of Lemma 4.2 shows the relationship between $\varphi_\alpha(x, y)$ and SM divergence through a derivation that starts from SMTGP cost function. From Lemmas 4.1 and 4.2, the term $\frac{|K_X|^{1-\alpha}|K_Y|^\alpha}{|\alpha K_Y+(1-\alpha)K_X|} \leq 1$ represents an agreement function between $p(X)$ and $p(Y)$. Similarly, $\frac{|K_{X\cup x}|^{1-\alpha}|K_{Y\cup y}|^\alpha}{|\alpha K_{Y\cup y}+(1-\alpha)K_{X\cup y}|} = \frac{|K_X|^{1-\alpha}|K_Y|^\alpha\eta_x^{1-\alpha}\eta_y^\alpha}{|\alpha K_Y+(1-\alpha)K_X|\eta_{x,y}\alpha} \leq 1$ is an agreement function between the extended distributions $p(X, x)$ and $p(Y, y)$. This agreement function increases as the weighted volume of the input and the output distributions (i.e..$|K_{X\cup x}|^{1-\alpha}|K_{Y\cup y}|^\alpha$, weighted by α) is as close as possible to the volume of the joint distribution (i.e. $|\alpha K_{Y\cup y} + (1-\alpha)K_{X\cup x}|$). This function reaches 1 (i.e. maximized) when the two distributions are identical, which justifies maximizing $\varphi_\alpha(x, y)$ as indicated in Lemma 4.2. From another view, maximizing $\varphi_\alpha(x, y)$ prefers minimizing $\eta_{x,y}(\alpha)$, which maximizes the $p((x, y)|(x_1, y_1), .., (x_N , y_N))$, that we abbreviate as $p(x, y)$; this is motivated by our intuition in Sect. 4.1. However, SMTGP maximizes $\varphi_\alpha(x, y) = \frac{\eta_x^{1-\alpha}\eta_y^\alpha}{\eta_{x,y}(\alpha)}$, this gives a probabilistic sense for the cost function when we follow our intuition that $\eta_x \propto 1/p(x)$, $\eta_y \propto 1/p(y)$ and $\eta_{x,y}(\alpha) \propto 1/p(x, y)$. Hence $\varphi_\alpha(x, y)$ could be seen as $\frac{p(x,y)}{p(x)^{1-\alpha}p(y)^\alpha}$, discussed in the following subsection. This understanding motivated us to plot the relation between $\varphi_\alpha(x, y)$ and the test error on SMTGP prediction. Figure 1 shows a clear correlation between $\varphi_\alpha(x, y)$ and the prediction error. Hence, it introduces a clear motivation to study it as a certainty measure, which could be associated with each structured output prediction.

4.3 Probabilistic interpretation of maximizing $\varphi_\alpha(x, y) = \frac{\eta_x^{1-\alpha}.\eta_y^\alpha}{\eta_{x,y}(\alpha)}$

As detailed in the previous subsection, one can interpret $\eta_{x,y}(\alpha) \propto 1/p(x, y)$, $\eta_x \propto 1/p(x)$, $\eta_y \propto 1/p(y)$. Hence, $\frac{\eta_{x,y}(\alpha)}{\eta_y} \propto p(y|x)$, $\frac{\eta_x}{\eta_{x,y}(\alpha)} \propto p(x|y)$. Hence, what does $\frac{\eta_x^{1-\alpha}.\eta_y^\alpha}{\eta_{x,y}(\alpha)}$ mean? Since $0 < \alpha < 1$, it is obvious that $min(\eta_x, \eta_y) < f_1(\alpha) = \eta_x^{1-\alpha}.\eta_y^\alpha < max(\eta_x, \eta_y)$. Figure 2 shows the behavior of $f_1(\alpha)$ against $f_2(\alpha) = (1-\alpha)\cdot\eta_x + \alpha\cdot\eta_y$, which is also bounded between $min(\eta_x, \eta_y)$ and $max(\eta_x, \eta_y)$. According to this figure, $f_1(\alpha)$ behaves very similar to $f_2(\alpha)$ as $|\eta_x - \eta_y|$ approaches zero, where linear approximation is accurate. However, as $|\eta_x - \eta_y|$ gets bigger, $f_1(\alpha)$ gets biased towards $min(\eta_x, \eta_y)$ as indicated in the left column of Fig. 2. Hence, $\frac{\eta_{x,y}(\alpha)}{\eta_x^{1-\alpha}.\eta_y^\alpha}$ is interpreted depending on the values of $\eta_x \propto \frac{1}{p(x)}$, $\eta_y \propto \frac{1}{p(y)}$, and $\eta_{x,y}(\alpha) \propto \frac{1}{p(x,y)}$ as follows:

1. If $\eta_x << \eta_y$, $\frac{\eta_x^{1-\alpha}.\eta_y^\alpha}{\eta_{x,y}(\alpha)} \approx^{12} \frac{\eta_x}{\eta_{x,y}(\alpha)} \propto p(y|x)$
2. If $\eta_y << \eta_x$, $\frac{\eta_x^{1-\alpha}.\eta_y^\alpha}{\eta_{x,y}(\alpha)} \approx \frac{\eta_y}{\eta_{x,y}(\alpha)} \propto p(x|y)$

[12] $\approx$ indicates equivalence for optimization/prediction.

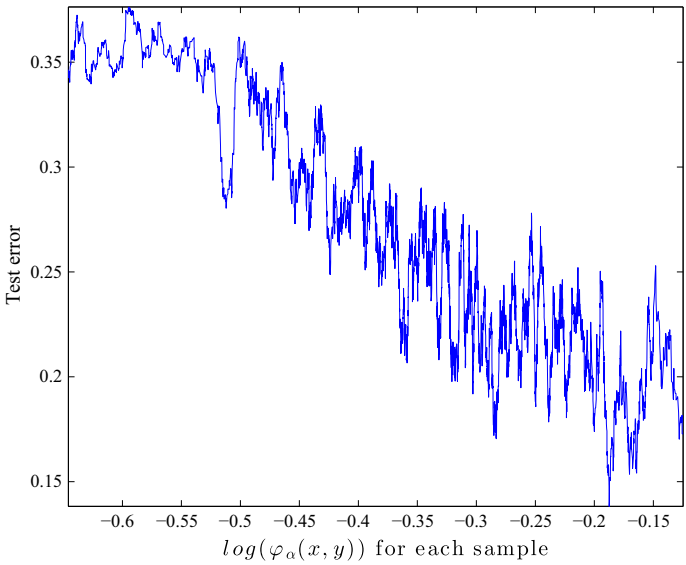

Fig. 1 $log(\varphi_\alpha(x, y))$ against test error on USPS dataset using SMTGP, $\alpha = 0.8$, $\beta = 0.5$

3. If $\eta_y \approx \eta_x$, $\frac{\eta_x^{1-\alpha}.\eta_y^{\alpha}}{\eta_{x,y}(\alpha)} \approxeq \frac{\eta_y}{\eta_{x,y}(\alpha)} \propto p(x|y) \approx \frac{\eta_x}{\eta_{x,y}(\alpha)} \propto p(y|x)$; This is less likely to happen since $p(x) = p(y)$ in this case.
4. If $|\eta_y - \eta_x| < \epsilon$, in this case α linearly control $\frac{\eta_x^{1-\alpha}.\eta_y^{\alpha}}{\eta_{x,y}(\alpha)} \approxeq \frac{(1-\alpha)\eta_x + \alpha\eta_y}{\eta_{x,y}(\alpha)} \propto (\frac{1-\alpha}{p(y|x)} + \frac{\alpha}{p(x|y)})^{-1}$

Hence, SMTGP regression predicts the output of maximum certainty on $p(x, y) = \mathcal{N}(0, (1 - \alpha)K_{X\cup x} + \alpha K_{Y\cup y})$, conditioned on the uncertainty extension on $p(x) = \mathcal{N}(0, K_{X\cup x})$ and $p(y) = GP(0, K_{Y\cup y})$. The conditioning is biased towards $max(p(x), p(y))$, which gives best discrimination relative to $p(x, y)$ and hence, maximize the certainty of the prediction. In case the difference between $p(x)$ and $p(y)$ is not high, the prediction is based on a weighted sum of $p(y|x)$ and $p(x|y)$, as shown in point 4 above.

5 Experimental results

In this section, we evaluate SMTGP on two Toy examples, USPS dataset in an image reconstruction task, and both Poser dataset (Agarwal and Triggs 2006) and HumanEva dataset (Sigal et al. 2010) for a 3D pose estimation task. It is shown in (Bo and Sminchisescu 2010; Yamada et al. 2012), that TGP outperforms Kernel Regression (KR), Gaussian Process Regression (GPR), Weighted K-Nearest Neighbor regression (Rasmussen and Williams 2005), Hilbert Schmidt independence criterion (HSIC) (Gretton et al. 2005), and Kernel Target Alignment method(KTA) (Cristianini and Kandola 2001) on a Toy example, HumanEva dataset, and Poser Dataset (i.e. Pose Estimation datasets). Hence, we extended our evaluation beyond pose estimation datasets. We compared our SMTGP with KLTGP and IKLTGP. IKLTGP stands for inverse KLTGP, which predicts the output by minimizing the KL divergence of the output probability distribution from the input probability distribution (Bo and Sminchisescu 2010). The main motivation behind this comparison

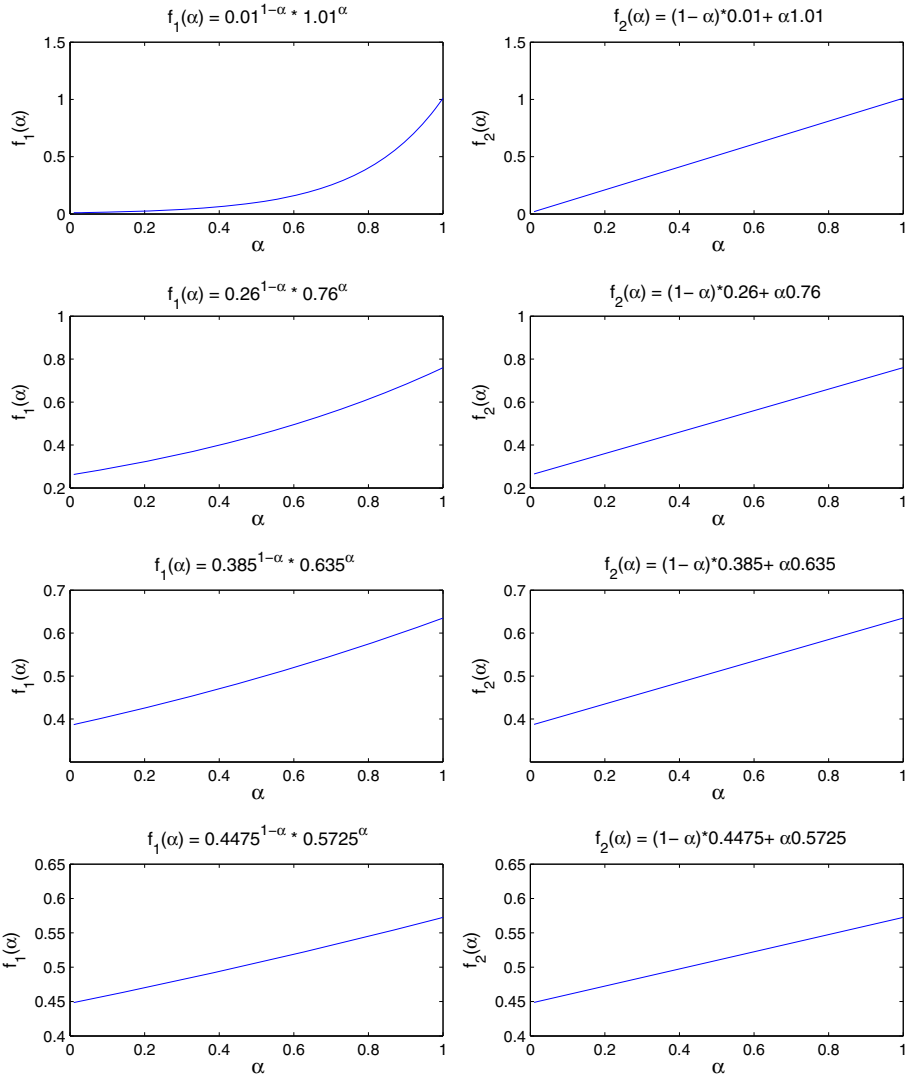

Fig. 2 *Left plot* functions are in the form $f_1(\alpha) = \eta_{D_1}^{1-\alpha}\eta_{D_2}^{\alpha}$ and their corresponding $f_2(\alpha) = (1-\alpha)\cdot\eta_{D1} + \alpha\cdot\eta_{D_2}$ are on the *right*; *rows indicate* different values of η_{D_1}, η_{D_2}, where D_1 and D_2 are arbitrary two domains

is that KLTGP and IKLTGP are biased to one of the distributions, and therefore the user has to choose either to use KLTGP or IKLTGP based on the problem. In contrast, SMTGP could be adapted by α and β on the validation set, such that the prediction error is minimized. From this point, we denote the set of KLTGP, IKLTGP and SMTGP as *TGPs*. Our presentation of the results starts by the specification of the toy examples and the datasets in Sect. 5.1. Then, we present our parameter settings and how α and β are selected in Sect. 5.2. Finally, we show our argument on the performance on these tasks in Sect. 5.3.

5.1 Specification of the toy examples and the datasets

5.1.1 Toy example 1 (Bo and Sminchisescu 2010)

The training set for the first toy problem predict a 1D output variable y given a 1D control x (the input). It consists of 250 values of y generated uniformly in (0,1), for which $x = y+0.3sin(2y\pi)+\epsilon$ is evaluated with ϵ such that $\epsilon = N(\mu = 0, \sigma = 0.005)$; see Fig. 3. Stars correspond to examples where *KNN* regression and *GPR* suffer from boundary/discontinuous effects as indicated in (Bo and Sminchisescu 2010). The *TGPs* were tested with 250 equally spaced inputs x in (0, 1). We used the mean prediction error to measure the performance on this example.

5.1.2 Toy example 2

In order to introduce a more challenging situation, we generate a double S shape; see Fig. 4. Toy example 2 is constructed by concatenated two S shapes, which makes the overall predic-

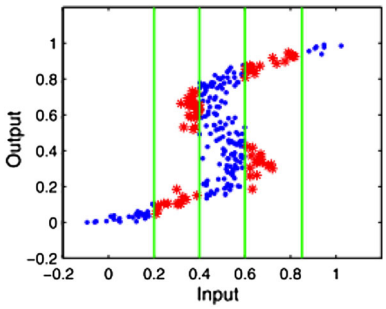

Fig. 3 Toy example 1

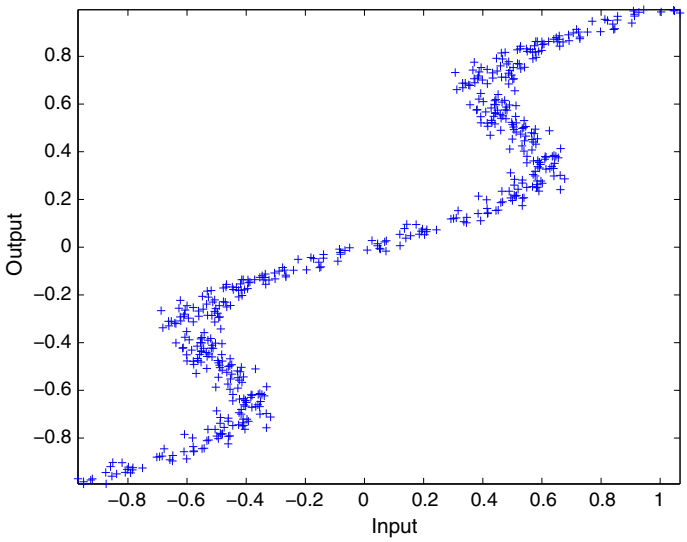

Fig. 4 Toy example 2

tion error more challenging to reduce. In addition, we down-sampled the points by 2, such that the total number of points is the same as Toy example 1. Hence, there is less information in the training data compared to Toy example 1. Similarly, the *TGPs* were tested with 500 equally spaced inputs x in $(-1, 1)$. We used the same error-measure in Toy Example 1.

5.1.3 Image reconstruction task on USPS dataset (Hull 1994)

The image reconstruction problem (Bo and Sminchisescu 2009) is that given outer 240 pixel values of a handwritten digit (16x16) from USPS data set, the goal is to predict the 16 pixel values lying in the center. We split the dataset into in 4649 test examples and 4649 training samples (No knowledge is assumed for the label of the digit). The range of the pixel values in this dataset is in $(-1, 1)$. The error measure amounts to the root mean-square error averaged over the 16 gray-scales in the center. $Error_{pose}(\hat{y}, y^*) = \|\hat{y} - y^*\|$, where $\hat{y} \in R^{16}$ is the predicted 16-values' vector lying in the center, y^* is the true 16-colors of the given outer 240 pixels values x.

5.1.4 3D pose estimation task on Poser dataset (Agarwal and Triggs 2006)

Poser dataset consists of 1927 training and 418 test images, which are synthetically generated and tuned to unimodal predictions. The image features, corresponding to bag-of-words representation with silhouette-based shape context features. The *TGPs* requires inversion of $N \times N$ matrices during the training, so the complexity of the solution is $O(N^3)$, which is impractical when N is larger. Hence, in both Poser and Human Eva datasets, we applied the *TGPs* by finding the K_{tr} nearest neighbors to each test point ($K_{tr} \approx 800$ in our experiments). This strategy was also adopted in (Bo and Sminchisescu 2010; Yamada et al. 2012). Poser dataset was generated using Poser software package, from motion capture (Mocap)data (54 joint angles per frame). The error is measured by the root mean square error (in degrees), averaged over all joints angles, and is given by: $Error_{pose}(\hat{y}, y^*) = \frac{1}{54} \sum_{m=1}^{54} \|\hat{y}^m - y^{*m} mod\ 360^\circ\|$, where $\hat{y} \in R^{54}$ is an estimated pose vector, and $y^* \in R^{54}$ is a true pose vector.

5.1.5 3D pose estimation task on HumanEva dataset (Sigal et al. 2010)

HumanEva datset contains synchronized multi-view video and Mocap data. It consists of 3 subjects performing multiple activities. We use the histogram of oriented gradient (HoG) features ($\in R^{270}$) proposed in (Bo and Sminchisescu 2010). We use training and validations sub-sets of HumanEva-I and only utilize data from 3 color cameras with a total of 9630 image-pose frames for each camera. This is consistent with experiments in (Bo and Sminchisescu 2010; Yamada et al. 2012). We use half of the data (4815 frames) for training and half (4815 frames) for testing. In HumanEva, pose is encoded by (20) 3D joint markers defined relative to the torso Distal joint in camera-centric coordinate frame, so $y = [y^{(1)}, y^{(2)}, \ldots, y^{(20)}] \in R^{60}$ and $y^{(i)} \in R^3$. Error (in mm) for each pose is measured as average Euclidean distance: $Error_{pose}(\hat{y}, y^*) = \frac{1}{20} \sum_{m=1}^{20} \|\hat{y}^m - y^{*m}\|$, where $\hat{y}$ is an estimated pose vector, and y^* is a true pose vector.

5.2 Parameter settings and learning α and β

Each SMTGP prediction is done by optimizing equation 15 by gradient descend with max steps of 50 (like Bo and Sminchisescu 2010). Since, we proved that β is mainly changing the

Table 1 Parameter settings for TGPs

	$2\rho_x^2$	$2\rho_y^2$	λ_x	λ_y	α	β
Toy 1	5	0.05	10^{-4}	10^{-4}	0.9	1.5
Toy 2	5	0.05	10^{-4}	10^{-4}	0.6	0.99
USPS	2	2	0.5×10^{-3}	0.5×10^{-3}	0.9	0.99
Poser	5	5000	10^{-4}	10^{-4}	0.7	0.5
Heva	5	500,000	10^{-3}	10^{-3}	0.99	0.99

power of the cost function, which theoretically does not affect the prediction, as detailed in Sect. 4. Hence, this motivated us to only consider only three values, which are actually edge cases ($\beta = 0.99$), ($\beta = 0.5$ for $\beta < 1$), ($\beta = 1.5$ for $\beta > 1$). We found that that the role of β in practice is mainly affecting the convergence rate and the purpose of cross validation on β is to find β that converges faster. We found that there is no specific value of β that gives the best performance for all the datasets. Hence, we suggest selecting β from only the suggested three values by cross validation like α but for a different purpose.

We performed five fold cross validation on α parameters ranging from 0 to 1 step 0.05. While, we selected three values for β. $\beta \rightarrow 1 = 0.99$ in practice, $\beta = 1.5$ (i.e. $\beta > 1$), $\beta = 0.5$ (i.e. $\beta < 1$). Our learning of the parameters covers different divergence measures and select the setting that minimize the error on the validation set. Finally, we initialize y in *SMTGP* by *KLTGP* prediction in (Bo and Sminchisescu 2010). Regarding λ_X, λ_Y, ρ_X and ρ_y, we use the values selected during the training of KLTGP (Bo and Sminchisescu 2010). Table 1 shows the parameter setting, we used for KLTGP, IKTGP, and SMTGP models. All these models share ρ_x, ρ_y, λ_x, and λ_y parameters. However, SMTGP has α and β as additional parameters.

5.3 Results

As can be noticed from Figs. 5 and 6, SMTGP improved on KLTGP on Toy 1 dataset. Further improvement has been achieved on Toy 2 dataset, which is more challenging; see Figs. 7 and 8. These results indicate the advantages of the parameter selection of α and β. From Table 2, we can notice that SMTGP improved on KLTGP by 12.70 % and also on IKLTGP by 3.51 % in Toy 2, which shows the adaptation behavior of SMTGP by tuning α and β. It was argued in (Bo and Sminchisescu 2010) that KLTGP performs better than IKLTGP in pose estimation. While, they reported that they gave almost the same performance on a toy example which we denote here by Toy 1. We presented Toy 2 to draw two conclusions. First, KLTGP does not always outperform IKLTGP as argued in (Bo and Sminchisescu 2010) in HumanEva dataset. Second, SMTGP could be tuned by cross-validation to outperform both KLTGP and IKLTGP.

Another important observation in Table 2 is that KLTGP outperforms IKLTGP on Poser and HumanEva datasets, while IKLTGP outperform KLTGP in the toy examples (slightly in the first and significantly in the second). The interesting behavior is that SMTGP performs at least as good as the best of KLTGP and IKLTGP in all of the datasets. KLTGP and IKLTGP are biased towards one of the input and the output distributions. However, SMTGP learns from the training data the bias factor (using α) towards the input or the output distributions. These results could also be justified by the fact that SM divergence is a generalization of a family of divergence measure. A powerful property in SMTGP is

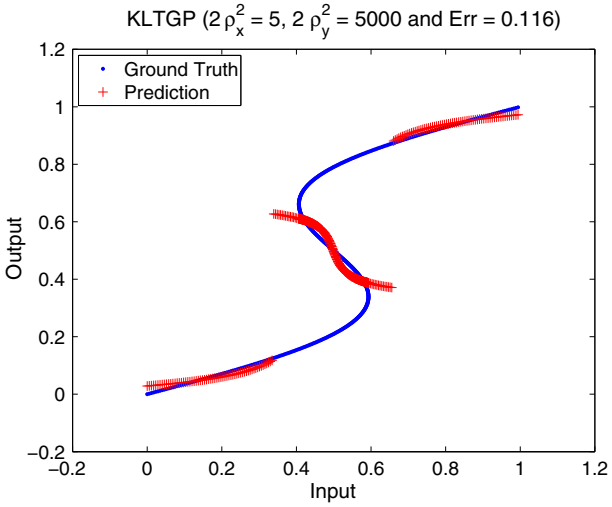

Fig. 5 Toy1: KLTGP error = 0.116(±0.152)

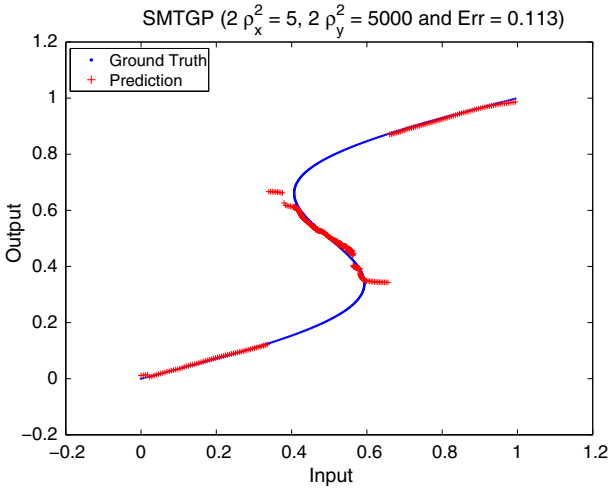

Fig. 6 Toy1:SMTGP error = 0.113(±0.158)

that by controlling α and β, SMTGP provides a set of divergence functions to optimize for prediction. However, a member of this set is selected during training by tuning α and β on a validation set. Hence, SMTGP learns α and β to make better predictions. Finally, SMTGP has a desirable generalization on the test set; see Table 2. Table 2 also shows that SMTGP does not only have same complexity as KLTGP but also it has a similar constant factor. In four of the datasets, SMTGP is faster than IKLTGP and KLTGP. [13] We optimized the matrix operations in the three methods as possible. SMTGP and KLTGP have similar number of matrix operations; this justifies why they have similar computational times.

[13] For KLTGP, we used the implementation provided by Bo and Sminchisescu (2010).

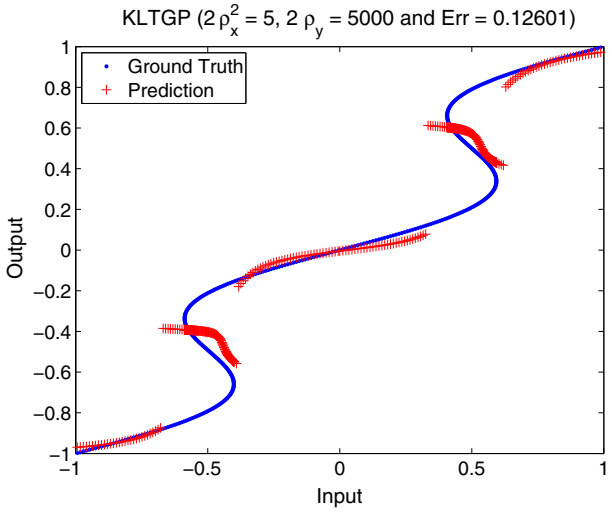

Fig. 7 Toy2:KLTGP error = 0.126(±0.14)

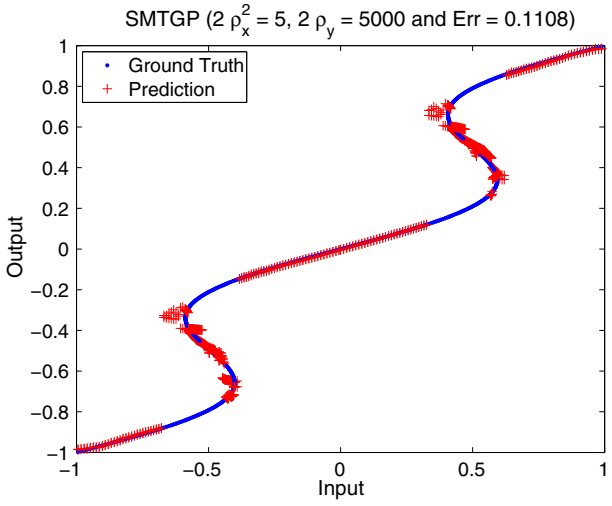

Fig. 8 Toy2: SMTGP error = 0.110(±0.15)

Table 2 Regression error and time of SMTGP, KLTGP, and IKLTGP, and error reduction of SMTGP against KLTGP and IKLTGP on the five datasets, imp. denotes the reduction

	SMTGP	KLTGP (Bo and Sminchisescu 2010)	Imp.%	IKLTGP (Bo and Sminchisescu 2010)	Imp%
Toy1	0.1126 (18.6 s)	0.116 (19.9 s)	2.93	0.115 (25.8 s)	2.09
Toy2	0.11 (20.1 s)	0.126 (19.2 s)	12.70	0.114 (25.1 s)	3.51
USPS	0.2587 (1001.7 s)	0.2665 (945 s)	2.93	0.2683 (1154 s)	3.58
Poser (deg)	5.4296 (104.3 s)	5.4296 (121.6 s)	0.00	6.484 (146.3 s)	16.26
HEva (mm)	37.59 (1631.6 s)	37.64 (2028.4 s)	0.13	55.622 (2344 s)	32.42

Table 3 Regression error of GPR, HSIC-KNN, KTA-KNN, and W-KNN regression models

	GPR (Rasmussen and Williams 2005)	WKNN (Rasmussen and Williams 2005)	HSICKNN (Gretton et al. 2005)	KTAKNN (Cristianini and Kandola 2001)
Toy1	0.17603	0.15152	0.18396	0.19333
Toy2	0.19011	0.16986	0.21294	0.19134
USPS	0.31504	0.2731	0.26832	0.26679
Poser (deg)	6.0763	5.62	7.1667	8.4739
HEva (mm)	46.6987	53.0834	57.8221	57.8733

We conclude our results by reporting the performance of GPR, HSIC-KNN, KTA-KNN, and W-KNN on the five datasets;[14] see Table 3. Comparing Tables 2 and 3, it is obvious that TGPs outperforms GPR, HSIC-KNN, KTA-KNN, and W-KNN.

6 Discussion

We proposed a framework for structured output regression based on SM-divergence. We performed a theoretical analysis to understand the properties of SMTGP prediction, which helped us learn α and β parameters of SM-divergence. As a part of our analysis, we argued on a certainty measure that could be associated with each prediction. We here discuss these main findings of our work.

A critical theoretical aspect that is missing in the KL-based TGP formulation is understanding the cost function from regression-perspective. We cover this missing theory not only by analyzing the cost function based on KL, but instead, by providing an understanding of SMTGP cost function, which covers (KL, Renye, Tsallis, Bhattacharyya as special cases of its parameters). Our claims are supported by a theoretical analysis, presented in Sect. 4. The main theoretical result is that SM-based TGP (SMTGP) prediction maximizes a certainty measure, we call $\varphi_\alpha(x, y)$, and the prediction does not depend on β theoretically. A probabilistic interpretation of $\varphi_\alpha(x, y)$ was discussed as part of our analysis and it was shown to have a negative correlation with the test error, which is an interesting result; see Fig. 1. The figure highlights the similarity between this SMTGP certainty measure and predictive variance provided by Gaussian Process Regression (GPR) (Rasmussen and Williams 2005) for single output prediction. A computationally efficient closed-form expression for SM-divergence was presented, which leads to reducing SMTGP prediction complexity from $O(N^3)$ to $O(N^2)$;[15] this makes SMTGP and KLTGP computationally equivalent. Moreover, it reduces the number of operations to compute SM-divergence between two general Gaussian distributions, out of TGP context; see Sect. 3. Practically, we achieve structured output regression by tuning α and β parameters of SM-divergence through cross validation under SMTGP cost function. We performed an intensive evaluation of different tasks on five

[14] These baseline approaches was also compared in (Bo and Sminchisescu 2010) against KLTGP, and our results is consistent with the conclusion that we reached from the comparison but only on Toy Example 1 and HumanEva dataset; see (Bo and Sminchisescu 2010) for more about the parameters of these baselines and its selection. KNN indicates that these methods were applied to training data in K-neighborhood of the testing point.

[15] N is the number of the training points.

datasets and we experimentally observed a desirable generalization property of SMTGP. Our experiments report that our resultant approach, SMTGP, outperformed KLTGP, IKLTGP, GPR, HSIC, KTA, and W-KNN methods on two toy examples and three datasets.

We conclude by highlighting a practical limitation of SMTGP, which is that it requires an additional time for tuning α and β by cross validation. However, we would like to indicate that this cross validation time is very short for the datasets (0.9 h for poser dataset and 14 h for Human Eva dataset). Using a smaller grid could significantly decrease this validation time. We used a grid of 20 steps for α. However, we found that in our experiments it is enough to use grid of size 10 (step 0.1 instead of 0.05). In addition, selecting a single randomly selected validation set like Neural networks models could save a lot of time instead of selecting α and β on the entire training set by cross validation, which we performed in our experiment.

7 Conclusion

We presented a theoretical analysis of a two-parameter generalized divergence measure, named Sharma–Mittal(SM), for structured output prediction. We proposed an alternative, yet equivalent, formulation for SM divergence whose computation is quadratic compared to cubic for the structured output prediction task (Lemma 3.1). We further investigated theoretical properties which is concluded by a probabilistic causality direction of our SM objective function; see Sect. 4. We performed extensive experiments to validate our findings on different tasks and datasets (two datasets for pose estimation, one dataset for image reconstruction and two toy examples).

Appendix 1: Relationship between $K_{X\cup x}^{-1}$ and K_X^{-1}

$K_{X\cup x}^{-1}$ is $O(N^2)$ to compute, given that the singular value decomposition of K_X is precomputed during the training, from which K_X^{-1} and K_X^{-2} are computed as well. Then, applying the matrix inversion Lemma (Alvarado 1999), $K_{X\cup x}^{-1}$ could be related to K_X^{-1} as follows

$$K_{X\cup x}{}^{-1} = \begin{bmatrix} K_X^{-1} + \frac{1}{c_x} K_X^{-1} K_X^x K_X^{x\,T} K_X^{-1} & \frac{-1}{c_x} K_X^{-1} K_X^x \\ \frac{-1}{c_x} K_X^{x\,T} K_X^{-1} & \frac{1}{c_x} \end{bmatrix} \tag{25}$$

where $c_x = K_X(x,x) - K_X^{x\,T} K_X^{-1} K_X^x$. Given that K_X^{-1} and K_X^{-2} are already computed, then computing $K_{X\cup x}^{-1}$ becomes $O(N^2)$ using Eq. 25. This equation applies to any kernel matrix (i.e., relating $K_{Y\cup y}^{-1}$ to K_Y^{-1}) .

Appendix 2: SM *D* TGP

In this "Appendix", we show mainly the gradient derivation of SMTGP $L_{\alpha,\beta}(p(X,x) : p(Y,y))$ (presented in Sect. 3.2). The derivations in Appendix 2 and 3 were mainly based on matrix calculus rules in (Petersen and Pedersen 2008); please refer to this reference for the rules.

Cost function

$$D_{\alpha,\beta}(p(X,x):p(Y,y)) = \frac{1}{\beta-1}\left[\left(\frac{|K_{X\cup x}|^{\alpha}|K_{Y\cup y}|^{1-\alpha}}{|(\alpha K_{X\cup x}^{-1}+(1-\alpha)K_{Y\cup y}{}^{-1})^{-1}|}\right)^{-\frac{1-\beta}{2(1-\alpha)}}-1\right] \quad (26)$$

$$\begin{aligned} L_{\alpha,\beta}(p(X,x):p(Y,y)) =& \frac{1}{\beta-1}(k_Y(y,y)-K_Y^{yT}K_Y{}^{-1}K_Y^y)^{\frac{-(1-\beta)}{2}}\cdot \\ & |(\alpha K_{X\cup x}^{-1}+(1-\alpha)K_{Y\cup y}{}^{-1})|^{\frac{-(1-\beta)}{2(1-\alpha)}} \end{aligned} \quad (27)$$

From the matrix inversion Lemma (Alvarado 1999),

$$K_{X\cup x}{}^{-1} = \begin{bmatrix} K_X^{-1}+\frac{1}{c_x}K_X^{-1}K_X^xK_X^{xT}K_X^{-1} & \frac{-1}{c_x}K_X^{-1}K_X^x \\ \frac{-1}{c_x}K_X^{xT}K_X^{-1} & \frac{1}{c_x} \end{bmatrix}$$

$$K_{Y\cup y}{}^{-1} = \begin{bmatrix} K_Y^{-1}+\frac{1}{c_y}K_Y^{-1}K_Y^yK_Y^{yT}K_Y^{-1} & \frac{-1}{c_y}K_Y^{-1}K_Y^y \\ \frac{-1}{c_y}K_Y^{yT}K_Y^{-1} & \frac{1}{c_y} \end{bmatrix}$$

where $c_x = K_X(x,x)-K_X^{xT}K_X^{-1}K_X^x$, $c_y = K_Y(y,y)-K_Y^{yT}K_Y^{-1}K_Y^y$. $K_{X\cup x}{}^{-1}$ and $K_{X\cup x}{}^{-1}$ could be computed in $O(N^2)$ where N is the number of points in the training set.

$$\begin{aligned} logL_{\alpha,\beta}(p(X,x):p(Y,y)) =& \frac{-(1-\beta)}{2}\cdot log(k_Y(y,y)-K_Y^{yT}K_Y{}^{-1}K_Y^y) \\ & +\frac{-(1-\beta)}{2(1-\alpha)}\cdot log|(\alpha K_{X\cup x}^{-1}+(1-\alpha)K_{Y\cup y}{}^{-1})| \end{aligned} \quad (28)$$

Gradient calculation

Following matrix calculus, $\frac{\partial logL(\alpha,\beta)}{\partial y(d)}$ could be expressed as follows

$$\begin{aligned} \frac{\partial logL_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y(d)} =& \frac{-(1-\beta)}{2}\frac{(\frac{\partial k_Y(y,y)}{\partial y(d)}-2K_Y^{yT}K_Y{}^{-1}\frac{\partial K_Y^y}{\partial y(d)})}{(k_Y(y,y)-K_Y^{yT}K_Y{}^{-1}K_Y^y)} \\ +& \frac{-(1-\beta)}{2(1-\alpha)}(1-\alpha)Tr\left(\frac{\partial K_{Y\cup y}{}^{-1}}{\partial y(d)}\cdot\left(\alpha K_{X\cup x}^{-1}+(1-\alpha)K_{Y\cup y}{}^{-1}\right)^{-1}\right) \end{aligned} \quad (29)$$

Since $\frac{\partial k_Y(y,y)}{\partial y(d)}=0$ for rbf-kernels and $\frac{-(1-\beta)}{2(1-\alpha)}(1-\alpha)=\frac{-(1-\beta)}{2}$, then

$$\begin{aligned} \frac{\partial logL_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y(d)} =& \frac{-(1-\beta)}{2}\frac{(-2K_Y^{yT}K_Y{}^{-1}\frac{\partial K_Y^y}{\partial y(d)})}{(k_Y(y,y)-K_Y^{yT}K_Y{}^{-1}K_Y^y)} \\ -& \frac{-(1-\beta)}{2}Tr\left(\left(\alpha K_{X\cup x}^{-1}+(1-\alpha)K_{Y\cup y}{}^{-1}\right)^{-1}\cdot K_{Y\cup y}{}^{-1}\frac{\partial K_{Y\cup y}}{\partial y(d)}K_{Y\cup y}{}^{-1}\right) \end{aligned} \quad (30)$$

By factorization, the gradient could be further simplified into the following form.

$$\begin{aligned} \frac{\partial logL_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y(d)} =& \frac{-(1-\beta)}{2}\frac{(-2K_Y^{yT}K_Y{}^{-1}\frac{\partial K_Y^y}{\partial y(d)})}{(k_Y(y,y)-K_Y^{yT}K_Y{}^{-1}K_Y^y)} \\ -& \frac{-(1-\beta)}{2}Tr\left(K_{Y\cup y}{}^{-1}\left(\alpha K_{X\cup x}^{-1}+(1-\alpha)K_{Y\cup y}{}^{-1}\right)^{-1}\cdot K_{Y\cup y}{}^{-1}\frac{\partial K_{Y\cup y}}{\partial y(d)}\right) \end{aligned} \quad (31)$$

Since $(AB)^{-1} = B^{-1}A^{-1}$, where A and B are invertible matrices, then

$$\frac{\partial log L_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y(d)} = \frac{-(1-\beta)}{2}\frac{(-2{K_Y^y}^T {K_Y}^{-1}\frac{\partial K_Y^y}{\partial y(d)})}{(k_Y(y,y)-{K_Y^y}^T {K_Y}^{-1}K_Y^y)} - \frac{-(1-\beta)}{2}Tr\left(\left(K_{Y\cup y}\left(\alpha K_{X\cup x}^{-1}+(1-\alpha){K_{Y\cup y}}^{-1}\right)K_{Y\cup y}\right)^{-1}\cdot\frac{\partial K_{Y\cup y}}{\partial y(d)}\right) \tag{32}$$

After applying matrix multiplications, then

$$\frac{\partial log L_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y(d)} = \frac{-(1-\beta)}{2}\frac{(-2{K_Y^y}^T {K_Y}^{-1}\frac{\partial K_Y^y}{\partial y(d)})}{(k_Y(y,y)-{K_Y^y}^T {K_Y}^{-1}K_Y^y)} - \frac{-(1-\beta)}{2}Tr\left(\left(\alpha K_{Y\cup y}K_{X\cup x}^{-1}K_{Y\cup y}+(1-\alpha)K_{Y\cup y}\right)^{-1}\cdot\frac{\partial K_{Y\cup y}}{\partial y(d)}\right) \tag{33}$$

where $\frac{\partial K_{Y\cup y}}{\partial y(d)} = \begin{bmatrix} 0 & \frac{\partial K_Y^y}{\partial y(d)} \\ \frac{\partial {K_Y^y}^T}{\partial y(d)} & 0 \end{bmatrix}$

After analyzing Eq. 33, it is not hard to see that

$$Tr\left(\left(\alpha K_{Y\cup y}K_{X\cup x}^{-1}K_{Y\cup y}+(1-\alpha)K_{Y\cup y}\right)^{-1}\cdot\frac{\partial K_{Y\cup y}}{\partial y(d)}\right) = 2\cdot\mu_y^T\cdot\frac{\partial K_Y^y}{\partial y(d)} \tag{34}$$

where $(\alpha K_{Y\cup y}K_{X\cup x}^{-1}K_{Y\cup y}+(1-\alpha)K_{Y\cup y})\mu_y^{'} = [0,0,\ldots 0,1]^T$, μ_y is a vector of all elements in $\mu_y^{'}$ except the last element. Hence,

$$\frac{\partial log L_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y(d)} = \frac{-(1-\beta)}{2}\frac{(-2\cdot{K_Y^y}^T {K_Y}^{-1}\frac{\partial K_Y^y}{\partial y(d)})}{(k_Y(y,y)-{K_Y^y}^T {K_Y}^{-1}K_Y^y)} - \frac{-(1-\beta)}{2}\cdot 2\cdot\mu_y^T\cdot\frac{\partial K_Y^y}{\partial y(d)} \tag{35}$$

which directly leads to the final form of $\frac{\partial \log L_{\alpha,\beta}(p(X,x):p(Y,y))}{\partial y^{(d)}}$

$$\frac{\partial \log L(\alpha,\beta)}{\partial y^{(d)}} = (1-\beta)\left[\frac{{K_Y^y}^T {K_Y}^{-1}\frac{\partial K_Y^y}{\partial y^{(d)}}}{(k_Y(y,y)-{K_Y^y}^T {K_Y}^{-1}K_Y^y)} + \mu_y^T\cdot\frac{\partial K_Y^y}{\partial y^{(d)}}\right] \tag{36}$$

Appendix 3: Advantage of computing SM divergence between two multivariate Gaussians using Lemma 3.1

As far as we know, an efficient way to compute $D_{\alpha,\beta}(\mathcal{N}_p,\mathcal{N}_q)$ in Eq. 3 where $\Delta\mu = 0$,[16] requires $\approx \frac{5N^3}{3}$ operations; we illustrate as follows. Cholesky decompistion of Σ_p and Σ_q requires $\frac{2N^3}{3}$ operations, with additional $\frac{2N^3}{3}$ operations for computing Σ_p^{-1} and Σ_q^{-1} from the computed decompositions (Trefethen and Bau 1997). Then, choseskly decompition of

[16] derived directly from the closed form in (Nielsen and Nock 2012).

$\alpha\Sigma_p^{-1}+(1-\alpha)\Sigma_q^{-1}$ is computed in additional $\frac{N^3}{3}$ operations. From the computed decompositions, $|\Sigma_p|$, $|\Sigma_q|$ and $|\alpha\Sigma_p^{-1}+(1-\alpha)\Sigma_q^{-1}|^{-1}=|(\alpha\Sigma_p^{-1}+(1-\alpha)\Sigma_q^{-1})^{-1}|$ are computed in $3N$ operations, which we ignore. Hence, the required computations for $D_{\alpha,\beta}(\mathcal{N}_p,\mathcal{N}_q)$ are $\frac{2N^3}{3}+\frac{2N^3}{3}+\frac{N^3}{3}=\frac{5N^3}{3}$ operations if $\Delta\mu=0$. In case $\Delta\mu\neq 0$, an additional $\frac{N^3}{3}$ operations are required to compute $(\alpha\Sigma_p^{-1}+(1-\alpha)\Sigma_q^{-1})^{-1}$, which leads to total of $\frac{6N^3}{3}=2N^3$ operations.[17]

In contrast to $D_{\alpha,\beta}(\mathcal{N}_p,\mathcal{N}_q)$, $D'_{\alpha,\beta}(\mathcal{N}_p,\mathcal{N}_q)$ in Lemma 3.1 could be computed similarly in only N^3 operations, if $\Delta\mu=0$, required to compute the determinants of Σ_p, Σ_q, and $\alpha\Sigma_q+(1-\alpha)\Sigma_p$ by Cholesky decomposition. In case $\Delta\mu\neq 0$, an additional $\frac{N^3}{3}$ operations are needed to compute $(\alpha\Sigma_q+(1-\alpha)\Sigma_q)^{-1}$.[18] So, total of $\frac{4N^3}{3}$ operations are needed if $\Delta\mu\neq 0$. Accordingly, $D'_{\alpha,\beta}(\mathcal{N}_p,\mathcal{N}_q)$ is $1.67=(\frac{5N^2}{3}/N^3)$ times faster to compute than $D_{\alpha,\beta}(\mathcal{N}_p,\mathcal{N}_q)$ if $\Delta\mu=0$, and $1.5=(2N^3/\frac{4N^3}{3})$ times faster, otherwise.

References

Agarwal, A., & Triggs, B. (2006). Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence*, *28*, 44–58.

Aghagolzadeh, M., Soltanian-Zadeh, H., Araabi, B., & Aghagolzadeh, A. (2007). A hierarchical clustering based on mutual information maximization. In *ICIP*.

Akturk, E., Bagci, G., & Sever, R. (2007). Is sharma-mittal entropy really a step beyond tsallis and rényi entropies? http://arxiv.org/abs/cond-mat/0703277

Aktürk, O. Ü., Aktürk, E., & Tomak, M. (2008). Can Sobolev inequality be written for Sharma-Mittal entropy? *International Journal of Theoretical Physics*, *47*, 3310–3320.

Alvarado, F. L. (1999). The matrix inversion lemma. Technical report, The University of Wisconsin, Madison, Wisconsin, 53706, USA.

Amari, S. I., & Nagaoka, H. (2000). *Methods of information geometry, translations of mathematical monographs* (Vol. 191). Oxford: Oxford University Press.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with Bregman divergences. *The Journal of Machine Learning Research*, *6*, 1705–1749.

Bo, L., & Sminchisescu, C. (2009). Structured output-associative regression. In *CVPR*.

Bo, L., & Sminchisescu, C. (2010). Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, *87*, 28–52.

Cichocki, A., & Ichi Amari, S. (2010). Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, *12*, 1532–1568.

Cichocki, A., Lee, H., Kim, Y. D., & Choi, S. (2008). Non-negative matrix factorization with α-divergence. *Pattern Recognition Letters*, *29*(9), 1433–1440.

Cichocki, A., Cruces, S., & Si, Amari. (2011). Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, *13*, 134–170.

Cristianini, J. N.Shawe-Taylor., & Kandola, J. S. (2001). Spectral kernel methods for clustering. In *NIPS*.

DeGroot, M. H. (1962). Uncertainty, information, and sequential experiments. *Annals of Mathematical Statistics*, *33*, 404–419.

Frank, T., & Plastino, A. (2002). Generalized thermostatistics based on the sharma-mittal entropy and escort mean values. *European Physical Journal B*, *30*, 543–549.

Gray, R. M. (1990). *Entropy and information theory*. New York: Springer.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*.

Hero, A. O., Ma, B., Michel, O., & Gorman, J. (2001). Alpha-divergence for classification, indexing and retrieval. Technical report, University of Michigan.

[17] There are additional N^2 (matrix vector multiplication), and N (dot product operations), which we ignore since they are not cubic.

[18] Additional $2\cdot O(N^{2.33})$ for 2 matrix multiplications are ignored.

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*, 550–554.
Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les in eǵalitesś entre les valeurs moyennes. *Acta Mathematica*, *30*, 175–193.
Kailath, T. (1967). The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, *15*, 52–60.
Kompass, R. (2007). A generalized divergence measure for nonnegative matrix factorization. *Neural Computation*, *19*, 780–791.
Kybic, J. (2006). Incremental updating of nearest neighbor-based high-dimensional entropy estimation. In *ICASSP*.
Learned-Miller, E. G., & Fisher-III, J. W. (2003). Ica using spacings estimates of entropy. *The Journal of Machine Learning Research*, *4*, 1271–1295.
Masi, M. (2005). A step beyond tsallis and rényi entropies. *Physics Letters A*, *338*(3), 217–224.
Nielsen, F., & Nock, R. (2012). A closed-form expression for the sharmamittal entropy of exponential families. *Journal of Physics A: Mathematical and Theoretical*, *45*(3).
Petersen, K. B., & Pedersen, M. S. (2008). The matrix cookbook. Technical University of Denmark, pp. 7–15.
Póczos, B., & Lõrincz, A. (2005). Independent subspace analysis using geodesic spanning trees. In *ICML*.
Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. Cambridge: The MIT Press.
Reid, M. D., & Williamson, R. C. (2011). Information, divergence and risk for binary experiments. *The Journal of Machine Learning Research*, *12*, 731–817.
Rényi, A. (1960). On measures of entropy and information. In *Berkeley symposium on mathematics, statistics and probability*.
Shan, C., Gong, S., & Mcowan, P. W. (2005). Conditional mutual information based boosting for facial expression recognition. In *BMVC*.
Shannon, C. E. (2001). A mathematical theory of communication. *SIGMOBILE*, *5*, 3–55.
Sharma, B. D., & Mittal, D. (1975). New non-additive measures of entropy for discrete probability distributions. *Journal of Mathematical Sciences*, *10*, 122–133.
Sigal, L., Balan, A. O., & Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, *87*, 4–27.
Szab, Z., Pczos, B., & Lrincz, A. (2007). Undercomplete blind subspace deconvolution via linear prediction. In *ECML*.
Trefethen, L. N., & Bau, D. (1997). *Numerical linear algebra. Society for Industrial and Applied Mathematics*. Philadelphia: SIAM.
Tsallis, C. (1988). Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, *52*, 479–487.
Tsallis, C., Plastino, A. R., & Alvarez-Estrada, R. F. (2009). Escort mean values and the characterization of power-law-decaying probability densities. *Journal of Mathematical Physics*. doi:10.1063/1.3104063
Van Hulle, M. M. (2008). Constrained subspace ica based on mutual information optimization directly. *Neural Computing*, *20*, 964–973.
Wang, Y. X., & Zhang, Y. J. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, *25*, 1336–1353.
Yamada, M., Sigal, L., & Raptis, M. (2012). No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In *ECCV*.
Zhang, J. (2004). Divergence function, duality, and convex analysis. *Neural Computation*, *16*, 159–195.
Zhang, J. (2007). A note on curvature of α-connections of a statistical manifold. *Annals of the Institute of Statistical Mathematics*, *59*(1), 161–170.
Zhang, J. (2013). Nonparametric information geometry: From divergence function to referential-representational biduality on statistical manifolds. *Entropy*, *15*, 5384–5418.