ECE 5630: Information Theory for Data Transmission, Security and Machine Learning 02/20/20

Lecture 8: Duality for f-divergences

Lecturer: Prof. Ziv Goldfeld

Scriber: Ben You, *Net ID*: by284

Assistant Editor: Kia Khezeli

8.1 Primer: Convex Conjugates

Definition 8.1 (Convex Conjugate) *Let $f : I \to \mathbb{R}$ be a convex function, where $I \subseteq \mathbb{R}$ is an interval. The convex conjugate of f is another function $f^\star : I^\star \to \mathbb{R}$ defined as*

$$f^\star(y) = \sup_{x \in I} (yx - f(x)),$$

where $I^\star := \{y \in \mathbb{R} : \sup_{x \in I} (yx - f(x)) < \infty\}$.

The convex conjugate of f at a point y_0 is given by $x_0 \in I$ with the largest difference between a linear function with slope y_0 and the function f as depicted in Figure 1.

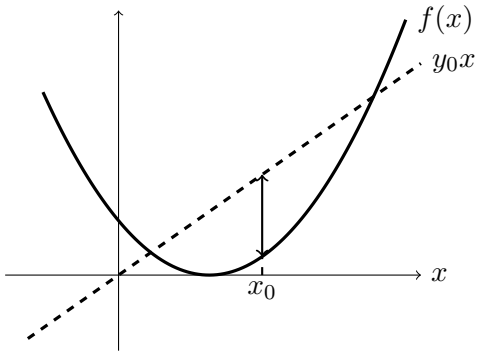


Figure 1: The illustration of $x_0 = f^\star(y_0)$.

Proposition 8.1 (Properties) *The convex conjugate $f^\star$ of f satisfies:*

(i) $f^\star$ is continuous on its domain.

(ii) $f^\star$ is convex.

(iii) Biconjugation: $(f^\star)^\star = f$.

8.2 Duality

We can utilize convex conjugation to formulate computation of f-divergences as an optimization problem. This form is referred to as dual (or variational) representation of D_f.

Theorem 1 (f-Divergence Duality) *For any f-divergence, we have:*

$$D_f(P\|Q) = \sup_{g:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[f^\star(g(X))],$$

where $f^\star$ is the convex conjugate of f and the supremum is taken over all functions g for which both expectations are finite.

Proof: Recall $D_f(P\|Q) = \int_{\mathcal{X}} f\left(\frac{\mathsf{d}P}{\mathsf{d}Q}(x)\right)\mathsf{d}Q(x)$. By Property (iii) in Proposition 8.1, we have

$$\begin{aligned} D_f(P\|Q) &= \int_{\mathcal{X}} f\left(\frac{\mathsf{d}P}{\mathsf{d}Q}(x)\right)\mathsf{d}Q(x) \\ &= \int_{\mathcal{X}} \sup_{y\in\mathsf{dom}(f^\star)} \left(y\frac{\mathsf{d}P}{\mathsf{d}Q}(x) - f^\star(y)\right)\mathsf{d}Q(x) \\ &\geq \int_{\mathcal{X}} \left(g(x)\frac{\mathsf{d}P}{\mathsf{d}Q}(x) - f^\star(g(x))\right)\mathsf{d}Q(x) \end{aligned}$$

for all measurable $g : \mathcal{X} \to \mathbb{R}$ (y inside the supremum generally depends on x). Finally, for any $g : \mathcal{X} \to \mathbb{R}$, we write

$$\begin{aligned} D_f(P\|Q) &\geq \int_{\mathcal{X}} g(x)\frac{\mathsf{d}P}{\mathsf{d}Q}(x)\mathsf{d}Q(x) - \int_{\mathcal{X}} f^\star(g(x))\mathsf{d}Q(x) \\ &= \int_{\mathcal{X}} g(x)\mathsf{d}P(x) - \int_{\mathcal{X}} f^\star(g(x))\mathsf{d}Q(x) \\ &= \mathbb{E}_P[g(X)] - \mathbb{E}_Q[f^\star(g(X))]. \end{aligned}$$

Then, by supremizing over all measurable $g : \mathcal{X} \to \mathbb{R}$, we have

$$D_f(P\|Q) \geq \sup_{g:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[f^\star(g(X))].$$

It can be shown that the above lower bound is tight and achieved by $g(x) = f'\left(\frac{\mathsf{d}P}{\mathsf{d}Q}(x)\right)$, where f' is the derivative of f. ■

Example 8.1 *We need to find convex conjugates of respective f functions. Let $h(x,y) = xy - f(x)$. Notice that $h(x,y)$ is concave in x as $f(x)$ is convex. So we can use the first-order optimality condition to find $f^\star(y) = \sup_x h(x,y)$.*

1. *<u>KL</u>: $f(x) = x\log x$ and $h(x,y) = xy - xlog(x)$. From the first order optimality condition $\frac{\mathsf{d}h}{\mathsf{d}x} = 0$, it follows that $x^\star = \text{argmax}_{x>0}\, h(x,y) = e^{y-1}$ and*

$$f^\star(y) = ye^{y-1} - (y-1)e^{y-1} = e^{y-1}.$$

Then,

$$D_f(P\|Q) = \sup_{g:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[e^{g(X)-1}] = 1 + \sup_{g:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[e^{g(X)}],$$

where the last equation follows from a change of variable of the form $\tilde{g}(x) = g(x) - 1$.

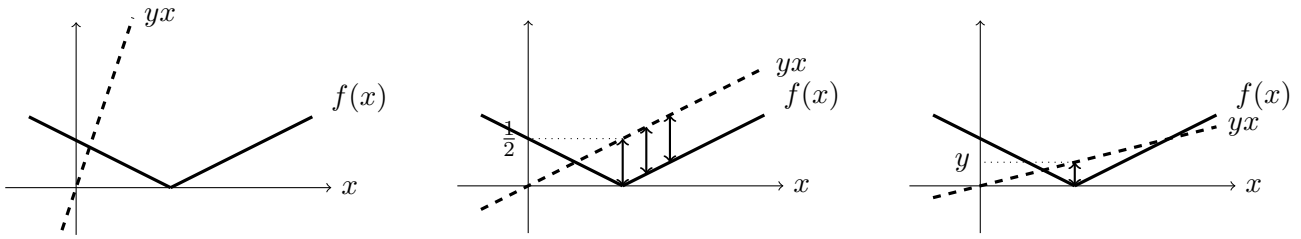
2. *<u>TV</u>: $f(x) = \frac{1}{2}|x-1|$ and $h(x,y) = xy - \frac{1}{2}|x-1|$. So,*

$$f^\star(y) = \begin{cases} y, & |y| \leq \frac{1}{2}, \\ \infty, & \text{o.w.} \end{cases}$$

Then,

$$\delta_{\mathsf{TV}}(P,Q) = \sup_{||g||_\infty \leq \frac{1}{2}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[g(X)] = \sup_{||g||_\infty \leq 1} \frac{1}{2}(\mathbb{E}_P[g(X)] - \mathbb{E}_Q[g(X)]),$$

where the uniform norm (sup norm) $||g||_\infty$ is defined as $||g||_\infty = \sup_{y\in\mathsf{dom}(g)} |g(y)|$.



(a) $y > \frac{1}{2}$. (b) $y = \frac{1}{2}$. (c) $y < \frac{1}{2}$.

Figure 2: The illustration of $f^\star(y)$ for $f(x) = \frac{1}{2}|x - 1|$.

8.3 Generative Modeling

Generative modeling is an unsupervised learning task, where we are given unlabeled data $\mathcal{X}_n := \{X_i\}_{i=1}^n$ drawn in an i.i.d. manner according to $P \in \mathcal{P}(\mathbb{R}^d)$. Our objective is to use $\mathcal{X}_n$ to learn some underlying structure, e.g., clustering, dimensionality reduction, or fit a model to P itself.

A common approach to generative modeling to consider a parametric model class $\{Q_\theta\}_{\theta\in\Theta}$, where $\Theta \subseteq \mathbb{R}^{d'}$. We aim to find a model such that $Q_\theta \approx P$. Note that by "learning" Q_θ, does not necessarily require to explicitly know it, but we do want to the very least to be able to sample from it. The state-of-the-art systems for learning such generative model (that can be readily sampled) are Generative Adversarial Networks (GANs).

8.3.1 Generative Adversarial Networks (GANs)

A GAN [1] pits two deep neural networks, a *generator* and a *discriminator*, against each other in a zero-sum game to improve their performance. The generator produces new data instances, while the discriminator evaluates them for authenticity and penalizes the generator when samples are not realistic enough (see Fig. 3). Training the two networks via an alternating optimization procedure until convergence results in a generator capable of producing strikingly realistic samples. The details are as follows.

Resources: In addition to the data set $\mathcal{X}_n$, we have access to an exogenous source of randomness, which we can freely sample. A common choice is an isotropic Gaussian $Z \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_{d_0})$, where typically $d_0 \ll d$.

Structure: A GAN consists of two DNNs:

1. **Generator**: A DNN $g_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^d$ that takes random noise Z as an input and outputs synthetic (fake) samples. We denote the probability law of $g_\theta(Z)$ by Q_θ.
2. **Discriminator**: A DNN $d_\phi : \mathbb{R}^d \to \mathbb{R}$ that takes in both "real" and "fake" samples and tries to tell them apart.

Optimization: By iteratively optimizing g_θ and d_ϕ via an alternating optimization procedure, once converged, we obtain a generator that "is able to fool even the best discriminator possible". Formally, the desired (θ, ϕ) pair attains Nash equilibrium of the zero-sum game

$$\inf_{\theta\in\Theta} \sup_{\phi\in\Phi} \mathbb{E}[d_\phi(x)] - \mathbb{E}[d_\phi(g_\theta(Z))]$$

Principled Form: A principled approach to design the discriminator is to model it as an f-divergence D_f, i.e., solve

$$\inf_{\theta\in\Theta} D_f\left(P \| Q_\theta\right).$$

Plugging in the dual form of D_f into the above and recover the minimax game formulation of GANs.

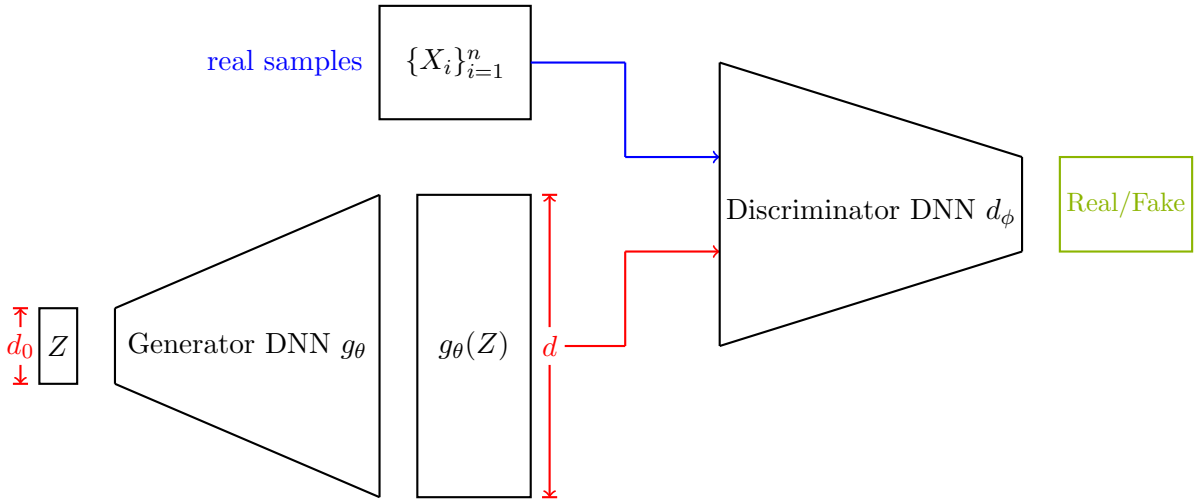


Figure 3: Generative Adversarial Networks (GANs).

Example 8.2 (Total Variation Discriminator) *Taking $D_f = \delta_{\mathsf{TV}}$, the induced GAN is*

$$\begin{aligned}\inf_{\theta\in\Theta} \delta_{\mathsf{TV}}\left(P, Q_\theta\right) &= \inf_{\theta\in\Theta} \sup_{\|d\|_\infty<\frac{1}{2}} \mathbb{E}[d(x)] - \mathbb{E}[d(g_\theta(Z))] \\ &\approx \inf_{\theta\in\Theta} \sup_{\substack{\phi\in\Phi:\\ \|d_\phi\|_\infty<\frac{1}{2}}} \mathbb{E}[d_\phi(x)] - \mathbb{E}[d_\phi(g_\theta(Z))],\end{aligned}$$

where the approximation relies on the parametric class Φ being reach enough to enjoy the universal approximation property.

Remark 8.1 (Concluding Remarks)

(i) GANs are very useful in practice but hard to study theoretically.

(ii) $\inf_\theta D_f\left(P\|Q_\theta\right)$ is a convenient mathematical formulation that lands itself well for a theoretic analysis (e.g., sample complexity, generalization, etc.).

(iii) Wasserstein GAN [2]: The GAN construction that has shown the "best" performance in practice is based on the 1-Wasserstein distance. That is, the min-max game is modeled as $\inf_{\theta\in\Theta} \mathsf{W}_1(P, Q_\theta)$ where the 1-Wasserstein distance between two distributions $P, Q \in \mathcal{P}(\mathbb{R}^d)$ is defined as

$$\mathsf{W}_1(P, Q) := \sup_{\|f\|_{\mathsf{Lip}}\leq 1} \mathbb{E}_P\left[f(X)\right] - \mathbb{E}_Q\left[f(X)\right].$$

where $\|f\|_{\mathsf{Lip}} := \sup_{x,y\in\mathbb{R}^d} \frac{|f(x)-f(y)|}{\|x-y\|}$ is the Lipschitz norm of f.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza B., Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS-2014)*, pages 2672–2680, 2014.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML-2017)*, pages 214–223, Sydney, Australia, Jul. 2017.