Quality Control

Denis Derkach, Maksim Artemev, Artem Ryzhikov

CS HSE faculty, spring 2020

Contents

Natural Metrics
- Motivation

"Nonparametric" assessment of PDF

Distances between PDFs
- Total Variation Distance
- Kullback-Leibler Divergence
- Metrics from divergences
- Specially developed metrics

Quote

All models are wrong, but some useful.

-George Box

Natural Metrics

Motivation

- Assessing the quality of machine learning models is a problem by itself — it is often necessary to develop proxy metrics because basic metrics are not available;
- Some ways of evaluating the final response may give inaccurate results (e.g. annealing);
- Generative modeling is affected by several things, like preprocessing: multiplication of input data by 0.1 can artificially increase the likelihood of the final sample by 10 times.

Example of assessment

4.3 Human Evaluation of Samples

To obtain a quantitative measure of quality of our samples, we asked 15 volunteers to participate in an experiment to see if they could distinguish our samples from real images. The subjects were presented with the user interface shown in Fig. 6(right) and shown at random four different types of image: samples drawn from three different GAN models trained on CIFAR10 ((i) LAPGAN, (ii) class conditional LAPGAN and (iii) standard GAN [10]) and also real CIFAR10 images. After being presented with the image, the subject clicked the appropriate button to indicate if they believed the image was real or generated. Since accuracy is a function of viewing time, we also randomly pick the presentation time from one of 11 durations ranging from 50ms to 2000ms, after which a gray mask image is displayed. Before the experiment commenced, they were shown examples of real images from CIFAR10. After collecting $\sim$10k samples from the volunteers, we plot in Fig. 6 the fraction of images believed to be real for the four different data sources, as a function of presentation time. The curves show our models produce samples that are far more realistic than those from standard GAN [10].

From :https://arxiv.org/abs/1506.05751

Example of assessment

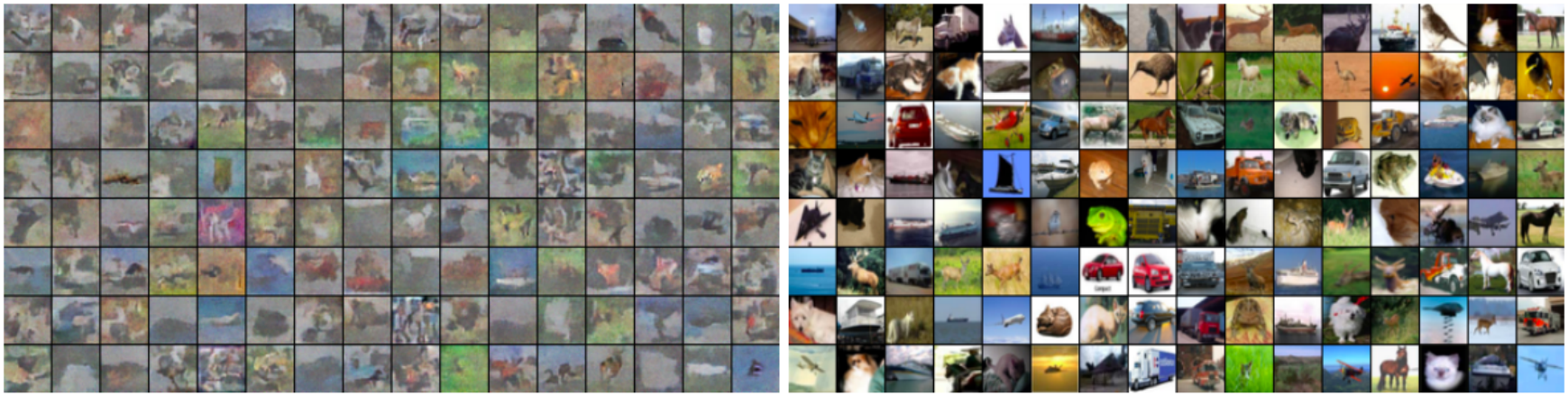


Figure 2. (Left) Samples from a convolutionally trained μ-ssRBM exhibit global coherence, sharp region boundaries, a range of colours, and natural-looking shading. (Right) The images in the CIFAR-10 training set closest (L2 distance with contrast normalized training images) to the corresponding model samples. The model does not appear to be capturing the natural image statistical structure by overfitting particular examples from the dataset.

From:https://icml.cc/Conferences/2011/papers/591_icmlpaper.pdf

Difficulties particular to GM

The model that should produce pictures of cats and dogs only generates pictures of dogs. The photos generated are very close to the training kit. Thus, we have an undertrained and overtrained network at the same time.

Choosing the best metric

We need a metric that tells us that our modeled distribution is somewhere close to the real one. Ideally, it should be differentiable and have nice properties for convergence. From now on, lets denote

- $p(x)$ as true pdf;
- q_θ as its estimate.

Distance

- It seems quite natural to measure the quality by evaluating some distance function to the result obtained;
- The problem is (as usual) we do not have access to $p(x)$;
- idea: use other methods for indirect quality assessment.

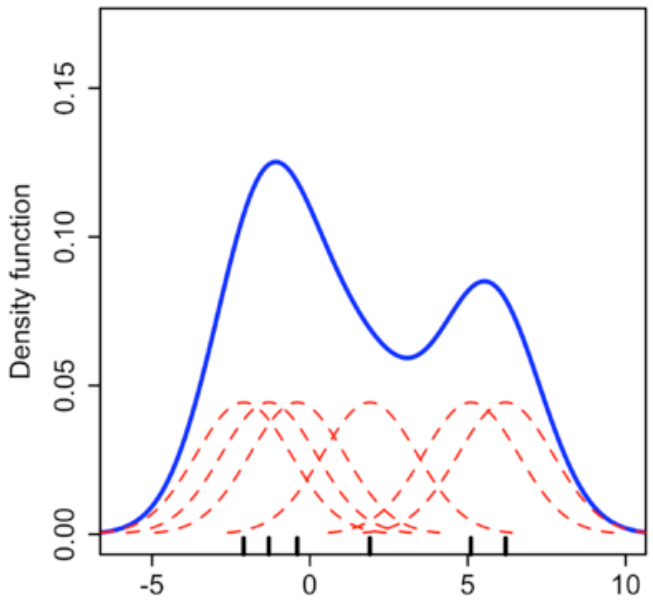
”Nonparametric” assessment of PDF

Kernel density estimation

Estimate of the pdf that looks like:

$$\hat{p}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

h — Bandwidth.



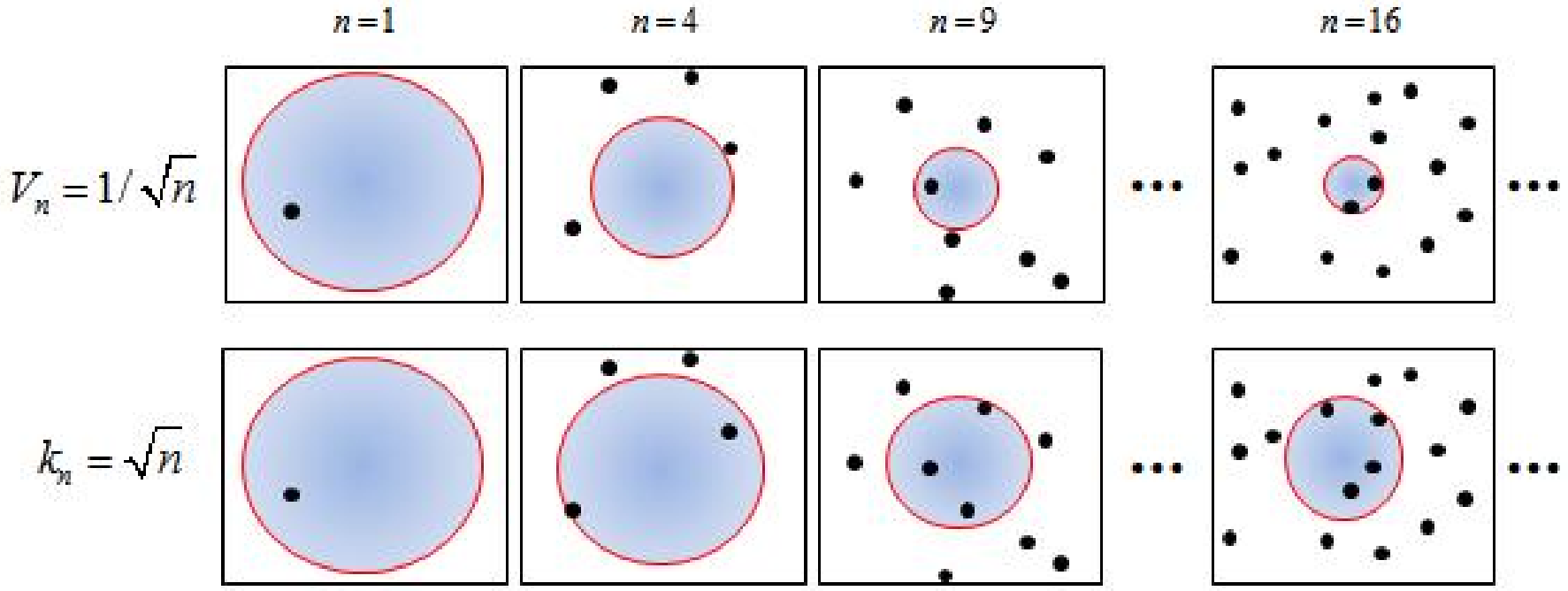
*More detail on these methods next week.

Properties: KDE

- Pros:
  - can be applied to data with any distribution;
  - simple for understanding;
  - good quality in case big samples.
- Cons:
  - choice of h may be hard;
  - CPU intensive;
  - does not take into account local fluctuations.
  - dimensionality problem.

Local Fluctuation problem

We can overcome it by considering a floating width of h, depending, for example, based on the number of neighbors (k-nearest neighbors, knn).



knn - properties

- Pros:
  - can be applied to data with any distribution;
  - simple for understanding;
  - good quality in case big samples.
- Cons:
  - choice of k is also difficult;
  - CPU intensive;
  - quality depends greatly on the number of samples.

Curse of dimensionality

The speed of convergence (while being close to optimal) of these methods depends greatly on the number of dimensions to consider $O(n^{-\frac{4}{4+d}})$. For example, the same quality on different number of dimensions d would require n samples.:

d	1	2	3	4	5	6	7	8	9
n	4	19	67	223	768	2790	10700	43700	187000

We probably need something that converges faster than this. And than check against this.

Distances between PDFs

Motivation

We need the way to estimate how close two distributions are.

f-divergence

NB: more motivation on f-divergence, Bregman divergence.

Definition

Let P and Q - be $\mathcal{X}$, with P absolutely continuous over Q. Than for convex function $f : (0, \infty) \to \mathbb{R}$, which is strictly convex at 1 and $f(1) = 0$, f-divergence between P and Q is called:

$$D_f(P||Q) \equiv \int_\Omega f\left(\frac{dP}{dQ}\right) dQ.$$

We thus only need to choose the function f to define the divergence.

First ideas

Definition

The first idea is to use something very straightforward, like for pdf's $p(x)$ and $q_\theta(x)$, $x \in \mathbb{R}^n$:

$$D(p(x), q_\theta(x)) = \sup_{\mathrm{A}} \left| \int_{\mathrm{A}} \mathrm{p(x)dx} - \int_{\mathrm{A}} \mathrm{q_\theta(x)dx} \right|,$$

where sup is calculated over all measurable A.

Total Variation Distance

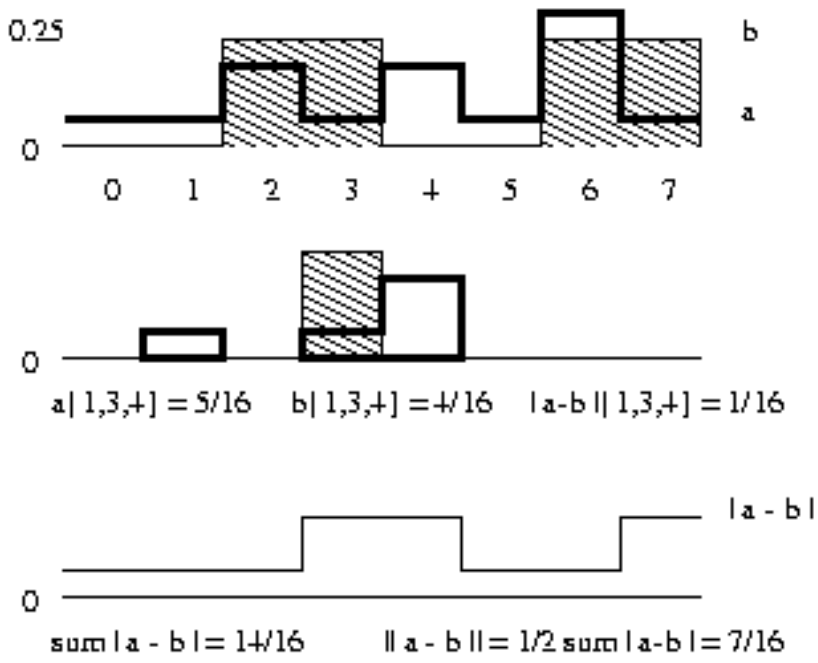
Which in fact can be rewritten like this (using Scheffé's lemma):

$$D(p(x), q_\theta(x)) = \frac{1}{2} \int_{\mathbb{R}^n} |p(x) - q_\theta(x)| dx,$$

You can see for example David Pollard A User's Guide to Measure Theoretic Probability, Chap 3.

Total Variation Distance: 1D example

- Probability distributions a (solid) and b (shaded).
- Subset $x = \{1, 3, 4\}$. $P_a(x) = 5/16$ and $P_b(x) = 4/16$.
- $||a - b||$ is the largest difference across all 256 possible subsets.
- $||a - b|| = 1/2 \sum |a - b|$



From GECCO-2002

Observations

- Symmetric: $D(P, Q) = D(Q, P)$.
- Connected to the hypotheses testing: $1 - D(P, Q)$ is equal to sum of false positives и false negatives.
- With growing number of trials, n, distance $D(f_{X^n}, g_{Y^n}) \to 1$. Moreover, if $D(f_X, g_Y) = \delta$, than for any $k \in \mathbb{N}$:
  $1 - 2e^{-k\frac{\delta^2}{2}} \leq D(f_{X^n}, g_{Y^n})$.
- Too strong. The distance might ignore the growing number of trials: $D(f_{X^2}, g_{Y^2}) = D(f_X, g_Y)$ (for example, $X_1, \ldots, X_n \sim \pm 1$, $S_n = \sum_n X_i$. Than

  $$S_n/\sqrt{n} \to \mathcal{N}(0, 1),$$

  but $D(S_n, Z) = 1$ for any n).

Kullback-Leibler Divergence: Definition

Let $p(x)$ and $q_\theta(x)$ are two probability distributions

$$KL(P||Q) = \int_{\mathbb{R}^n} p(x) \log \left(\frac{p(x)}{q_\theta(x)} \right) dx.$$

Although the KL divergence measures the "distance" between two distributions, it is not a distance measure.

Properties of KL divergence

- not symmetric $KL(P||Q) \neq KL(Q||P)$;
- invariant under change of variables;
- additive for independent rv: if $P(X,Y)$ and $Q(X,Y)$ are factorisable, than $KL(P\|Q) = KL(P_1\|Q_1) + KL(P_2\|Q_2)$.
- chain rule: $KL(p_{X^n}, p_{Y^n}) = nKL(p_X, p_Y)$.

KL and maximum likelihood estimate

Let θ^* be the true value of θ. Denote

$$M_n(\theta) = \frac{1}{n}\sum_i \log\frac{f(X_i;\theta)}{f(X_i;\theta_*)}$$

and $M(\theta) = -KL(\theta_*, \theta)$.

Let $\sup_{\theta\in\Theta}|M_n(\theta) - M(\theta)| \xrightarrow{P} 0$ and for any $\epsilon > 0$ $\sup_{\theta:|\theta-\theta_*|\geq\epsilon} M(\theta) < M(\theta_*)$.

If $\widehat{\theta}_n$ is the maximum likelihood estimate, than $\widehat{\theta}_n \xrightarrow{P} \theta_*$.

Cross Entropy and KL Divergence

Given two distributions p and q over a given variable X, the cross entropy is defined as

$$H(p, q) = \mathbb{E}_p(\log q)$$

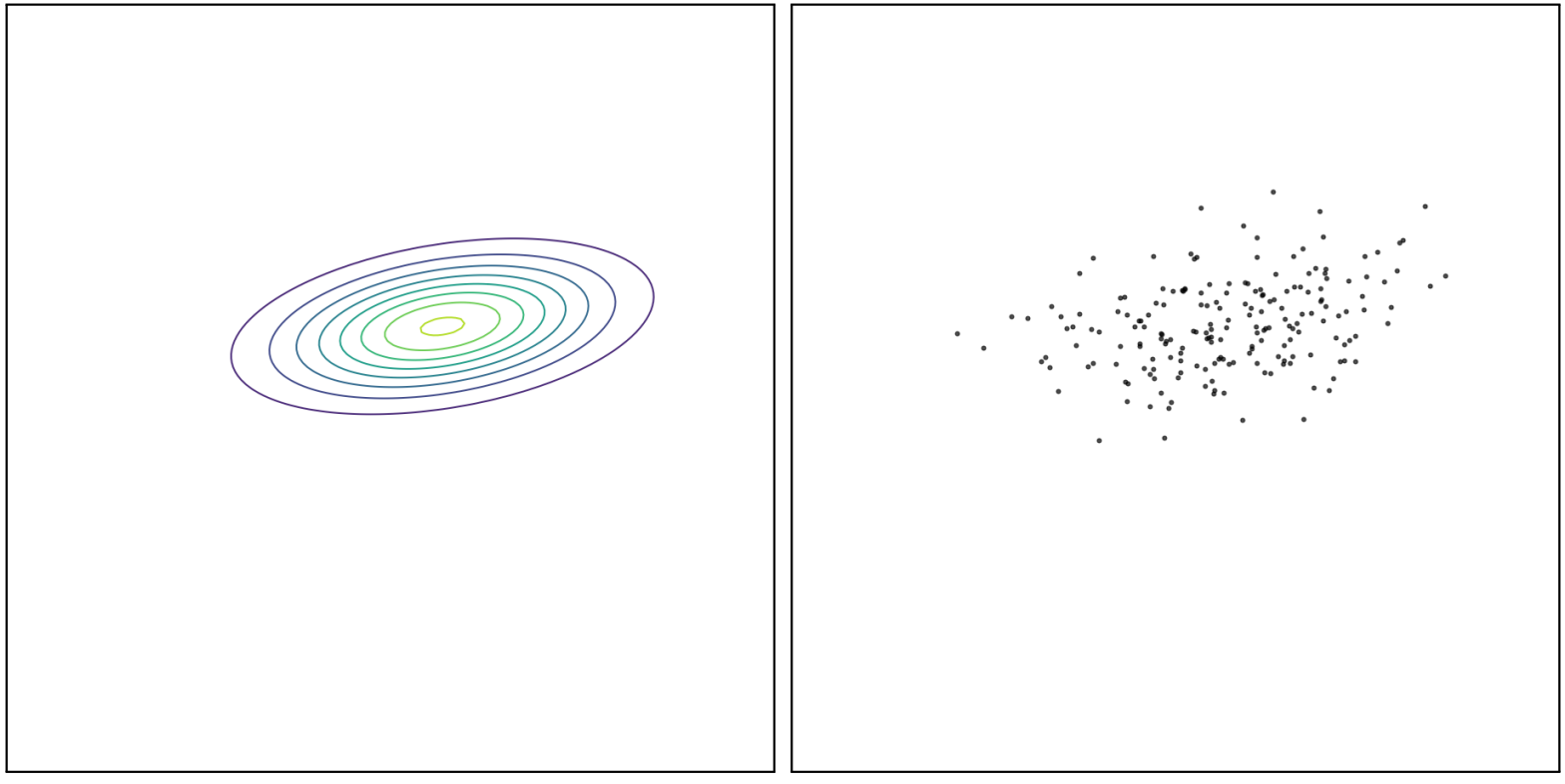
KL divergence is connected to it:

$$KL(p, q) = H(p) + H(p, q).$$

Since we normally optimise $L(\theta) = H(p_{data}, q(x))$, than optimisation of $KL \leftrightarrow$ optimisation of H.

Trying to converge

Let's check the convergence properties.Unfortunately, we do not have access to the true $p(x)$ during the study, so we must sample from it:



In the first part of our study we will using the 2D correlated Gaussian.

Here and later some examples are motivated by this blog.

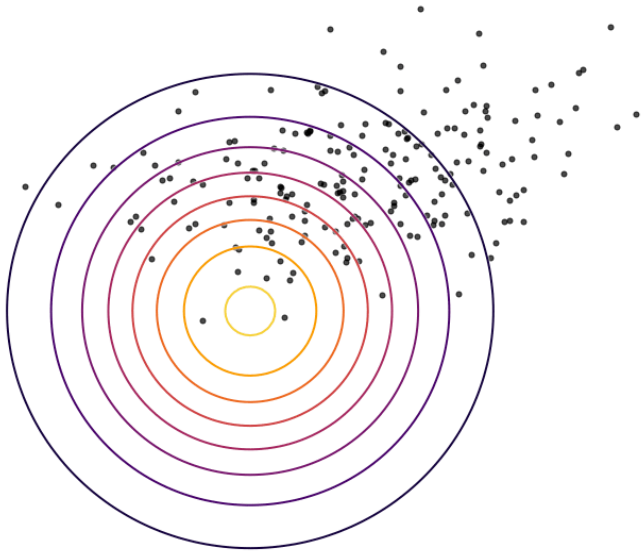
Optimal parameters

We need to get the optimal parameter, θ^* for our study. Let's do it by minimizing KL divergence.

$$
\begin{aligned}
\theta^* &= \arg\min_{\theta} KL(p(x)||q_\theta(x)) = \\
&= \arg\min_{\theta}(\mathbb{E}_{x\sim p}[\log p(x)] - \mathbb{E}_{x\sim p}[\log q_\theta(x)]) \\
&\quad \text{since } p(x) \text{ does not depend on } \theta = \\
&= \arg\min_{\theta} -\mathbb{E}_{x\sim p}[\log q_\theta(x)] = \\
&= \arg\max_{\theta} \mathbb{E}_{x\sim p}[\log q_\theta(x)].
\end{aligned}
$$

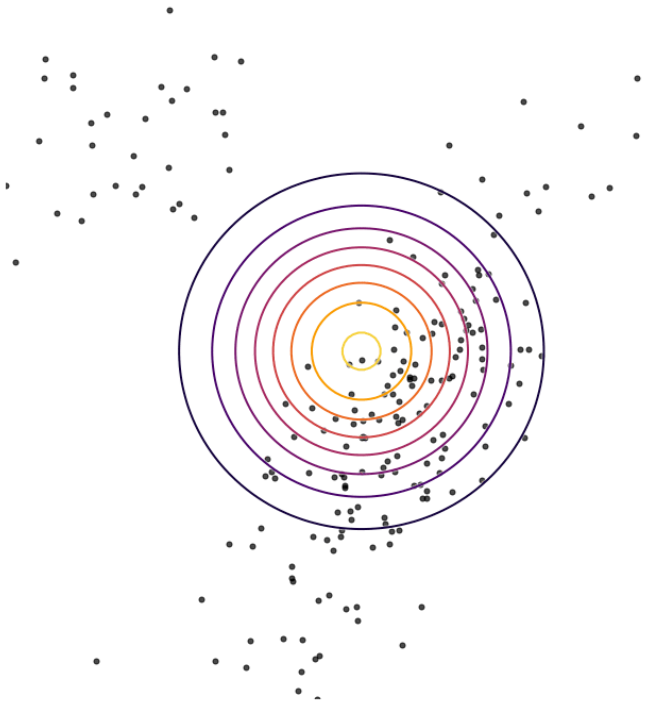
Which means that we want to find θ^* which assigns samples from $p(x)$ the highest possible $\log$ probability under $q_{\theta^*}(x)$.

Converging with KL



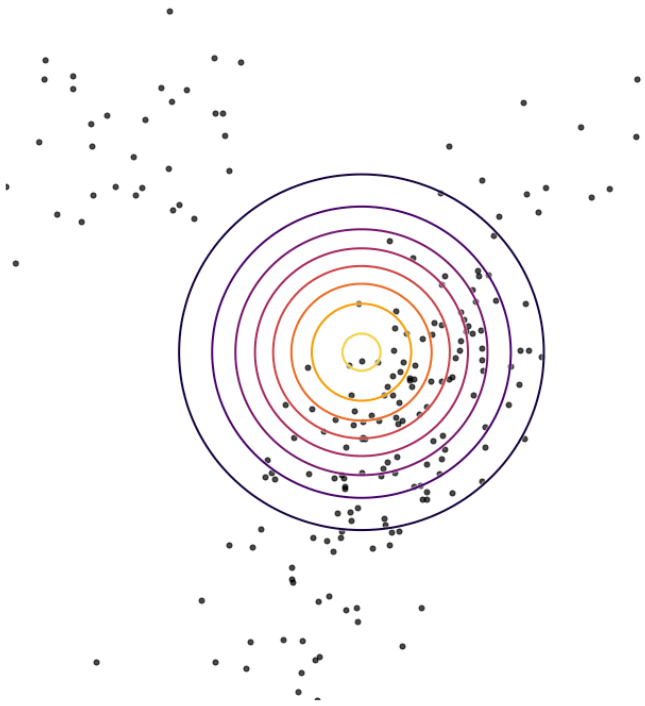
- The procedure works!
- Does it mean that the problem of building a model is solved?

Converging with KL: multimodal case



- The procedure works!
- Does it mean that the problem of building a model is solved?
- Not really, for the multimodal case, we will have problems.

Converging with KL: multimodal case intuition



- It's natural if we look at the quantity we optimize:

$$\arg\max_{\theta} \mathbb{E}_{x\sim p}[\log q_\theta(x)]$$

- If there is no $q_\theta(x)$ support in the place, where we have $x \sim p(x)$ than the optimised function goes to ∞.
- Automatically, we also have $q_\theta(x)$ support in places with no $x \sim p(x)$, which is also bad.

Reverse KL divergence

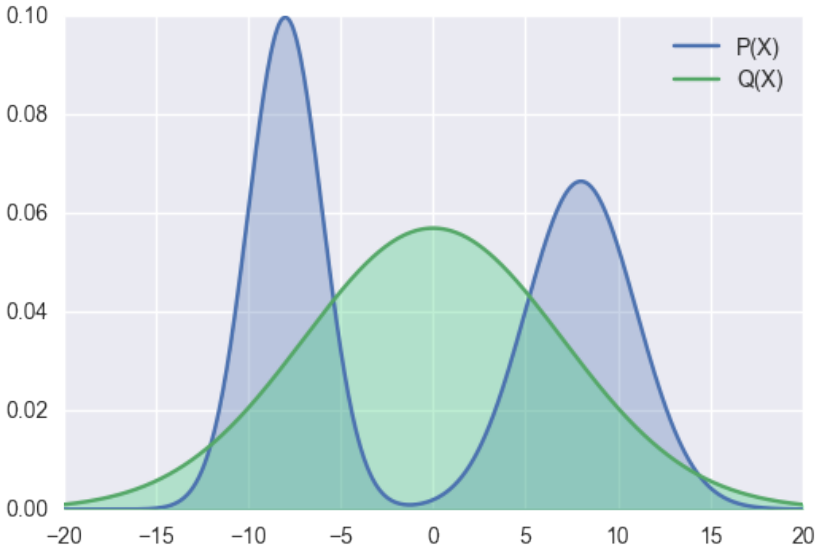
In order to overcome the problems, we can define a reverse divergence:

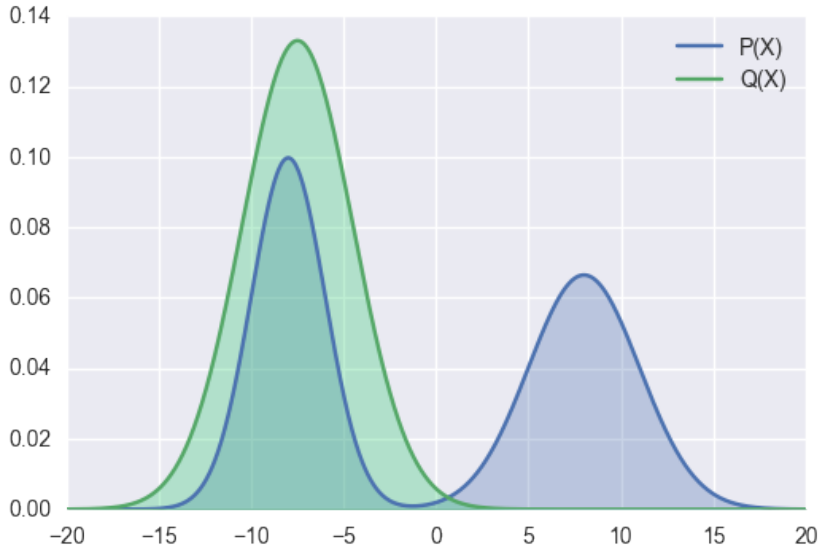
$$rKL(q_\theta||p) = \int_{\mathbb{R}^n} p(x) \log\left(\frac{p(x)}{q_\theta(x)}\right) dx.$$

Intuition

$$KL = \int p(X) \log \frac{p(x)}{q_\theta(x)} dx$$

$$rKL = \int q(x) \log \frac{q_\theta(x)}{p(x)} dx$$





Forward KL is known as zero avoiding, as it is avoiding $q(x) = 0$ whenever $P(x) > 0$.

Reverse KL Divergence is known as zero forcing, as it forces $Q(X)$ to be 0 on some areas, even if $P(X) > 0$.

Picture credit:https://wiseodd.github.io/techblog/2016/12/21/forward-reverse-kl/

rKL: optimisation

In fact, we are optimizing a very similar thing:

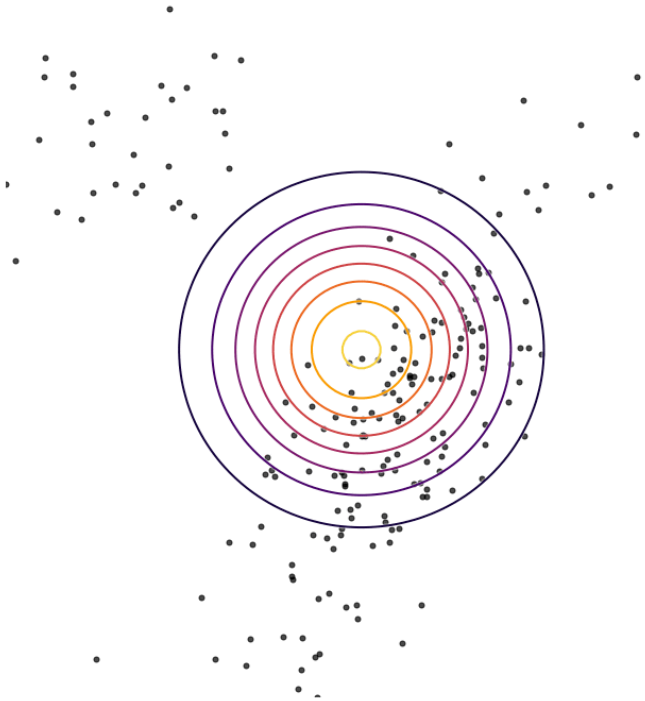
$$\begin{aligned}\theta^* &= \arg\min_\theta KL(q_\theta(x)||p(x)) = \\ &= \arg\min_\theta(\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] - \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)]) = \\ &= \arg\max_\theta(-\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] + \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)])\end{aligned}$$

But we do not have the previous problem of likelihood going to infinity in unreasonable places.

The first term is related to entropy of the generating model, the second penalises generated samples that are not similar to real distribution.

Let's check whether it works.

Converging with rKL



- We no longer have $q_\theta(x)$ support in the regions with no $x \sim p(x)$ population.
- The converged distribution looks reasonable but only for one solution.

The main problem: optimised expression depends on the $p(x)$

$$\arg\max_\theta(-\mathbb{E}_{\tilde{x}\sim q_\theta}[\log q_\theta(x)] + \mathbb{E}_{\tilde{x}\sim q_\theta}[\log p(x)]),$$

which we normally do not have.

Jensen-Shannon Divergence

We can try to optimize different divergences however, the problems normally stay. A distinguishable attempt is to construct the mixture of KL and rKL:

$$\begin{aligned} JS(p(x)||q_\theta(x)) = & \quad \frac{1}{2} \quad KL(p(x)||\frac{p(x)+q_\theta(x)}{2}) + \\ & + \quad \frac{1}{2} \quad KL(q_\theta(x)||\frac{p(x)+q_\theta(x)}{2}), \end{aligned}$$

It is symmetric and does not ignore zeroes like KL and does not ignore x like rKL.

Metrics from divergences

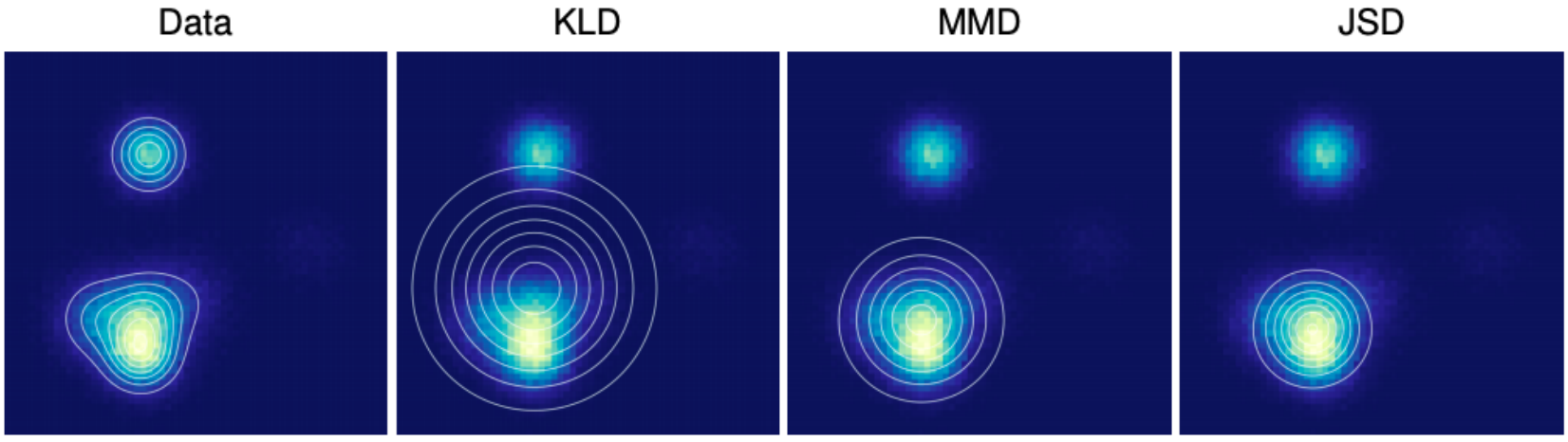


Figure 1: An isotropic Gaussian distribution was fit to data drawn from a mixture of Gaussians by either minimizing Kullback-Leibler divergence (KLD), maximum mean discrepancy (MMD), or Jensen-Shannon divergence (JSD). The different fits demonstrate different tradeoffs made by the three measures of distance between distributions.

Problems:

- need to use metrics different from optimised;
- difficulties in case of high dimension problem;
- not evident choice of a good metric.

From :https://arxiv.org/abs/1506.05751

sliced-Wasserstein distance

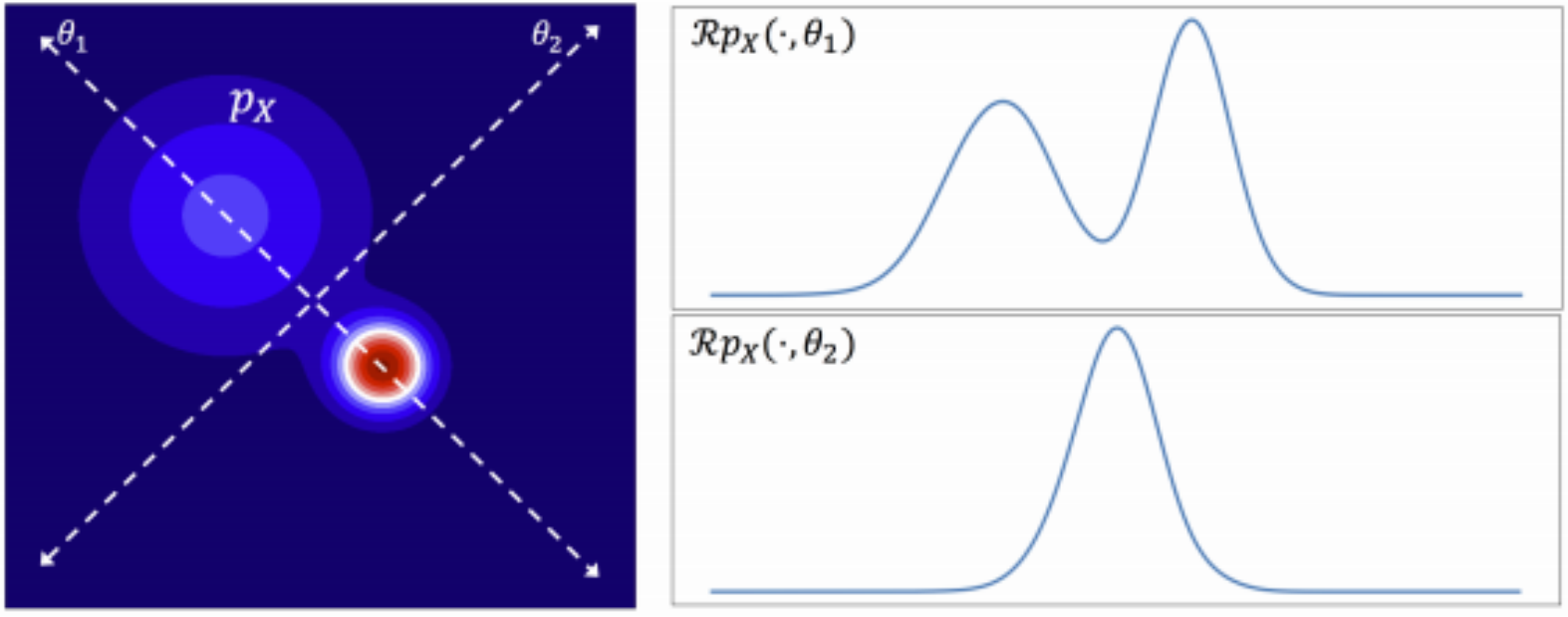
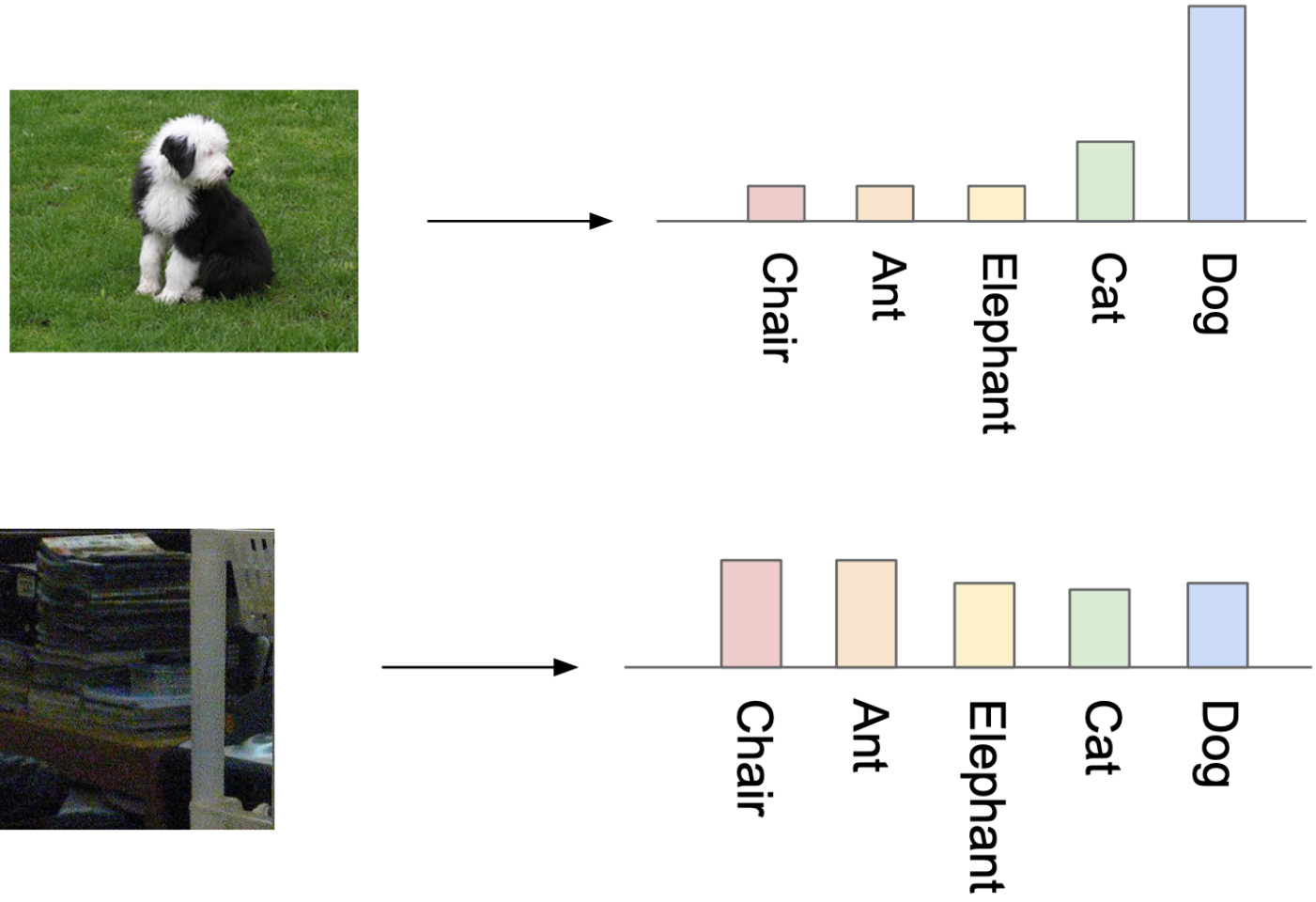


Fig. 1: Visualization of the slicing process defined in Eq. (10)

We can cut multidimensional space into one-dimensional projections and calculate W by 1D. It will strongly accelerate calculations.

From :https://arxiv.org/abs/1804.01947

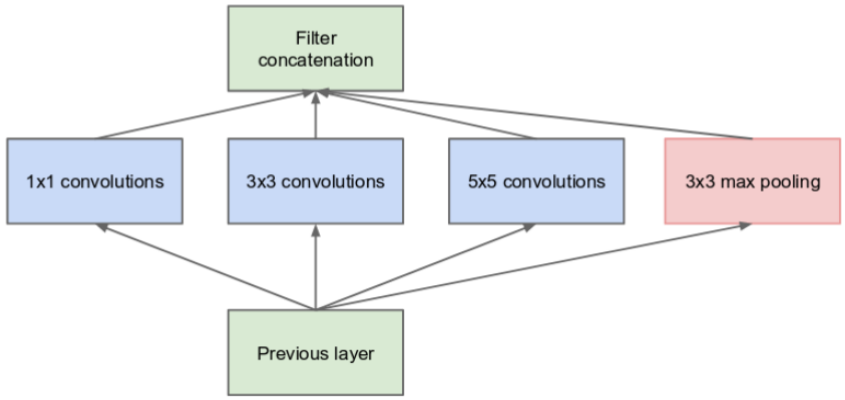
Classification of images



From :https://medium.com/octavian-ai/a-simple-explanation-of-the-inception-score-372dff6a8c7a

Inception Net

In 2014, the InceptionNet network was developed, which allowed classifying images with high quality.



(a) Inception module, naïve version

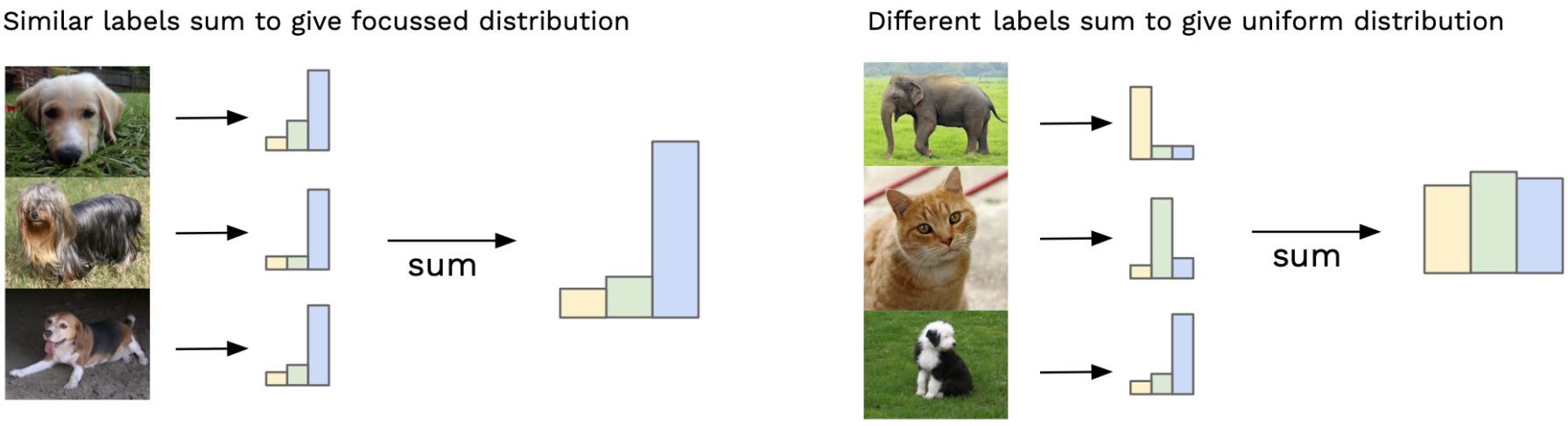
From :https://arxiv.org/abs/1409.4842

Idea

As an alternative to human annotators, we propose an automatic method to evaluate samples, which we find to correlate well with human evaluation: We apply the Inception model[1] [19] to every generated image to get the conditional label distribution $p(y|\boldsymbol{x})$. Images that contain meaningful objects should have a conditional label distribution $p(y|\boldsymbol{x})$ with low entropy. Moreover, we expect the model to generate varied images, so the marginal $\int p(y|\boldsymbol{x} = G(z))dz$ should have high entropy. Combining these two requirements, the metric that we propose is: $\exp(\mathbb{E}_{\boldsymbol{x}}\mathrm{KL}(p(y|\boldsymbol{x})||p(y)))$, where we exponentiate results so the values are easier to compare. Our *Inception score* is closely related to the objective used for training generative models in CatGAN [14]: Although we had less success using such an objective for training, we find it is a good metric for evaluation that correlates very

From :https://arxiv.org/abs/1606.03498

Merging the images



Images of one class give an obvious peak in the distribution, for several classes the distribution is more even.

Inception Score

In fact, we have some expected distribution, we just need to calculate the distance between the generated and classified images:

$$\text{IS} = \exp(\mathbb{E}_x \text{KL}(p(y|x)||p(y)))$$

$p(y|x)$ - conditional distribution for class $p(y)$ - probability of several pictures. Note:

$$1 < \text{IS}(G) < 1000$$

IS: properties

- Images should belong to one class.
- Generative algorithm should produce images for all classes.

IS: problems

- IS can only be calculated on image that InceptionNet (or other classifiers) has seen;
- InceptionNet can skip the properties of the images you are interested in, which may be well generated;
- IS does not control diversity of images;
- IS does not prevent memorisation of images.

IS fail

Figure 1. Sample of generated images achieving an Inception Score of 900.15. The maximum achievable Inception Score is 1000, and the highest achieved in the literature is on the order of 10.

From :https://arxiv.org/abs/1801.01973

Some considerations

In general, we are not prevented from using the IS idea with other quality metrics. In particular, for the Large Hadron Collider, we use regression to restore the characteristics of the particle from the picture of the experiment.

Frech'et Inception Distance

In order to solve some of the IS problems, the FID was proposed:

$$\text{FID} = ||\mu_r - \mu_g||^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2})$$

where $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ и $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ 2048-d activation of the Inception-v3 pool layer for real and generated images, respectively. Properties: solves the problem of IS deception with generating one picture per class, but do not solve the problem of storing images.

MiFID (Memorization-informed FID)

To bypass the problem with memorizing pictures it was suggested to give additional penalty for the generator producing too similar pictures. The size of the penalty is proportional to the proximity of the pictures:

$$d_{ij} = 1 - \cos(f_{g,i}, f_{r,j}) = 1 - \frac{f_{g,i} f_{r,j}}{|f_{g,i}||f_{r,j}|}$$

$$d = \frac{1}{N} \sum_i \min_j (d_{ij})$$

Than

$$\mathrm{M}IFID = \frac{1}{d_{thr}} \mathrm{F}ID,$$

where $d_{thr} = d$, if $d < \varepsilon$, otherwise $d_{thr} = 1$.

Conclusion

There is no single good way to evaluate a generative model. Most likely, the quality metrics should depend on the further use.