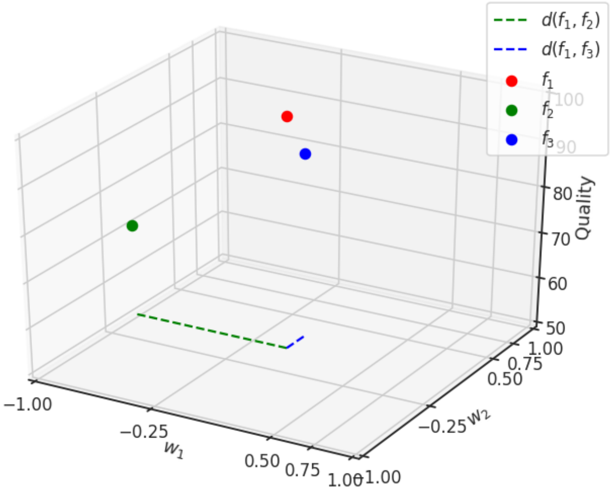
Probabilistic metric spaces, projections

MIPT

2023

Motivation

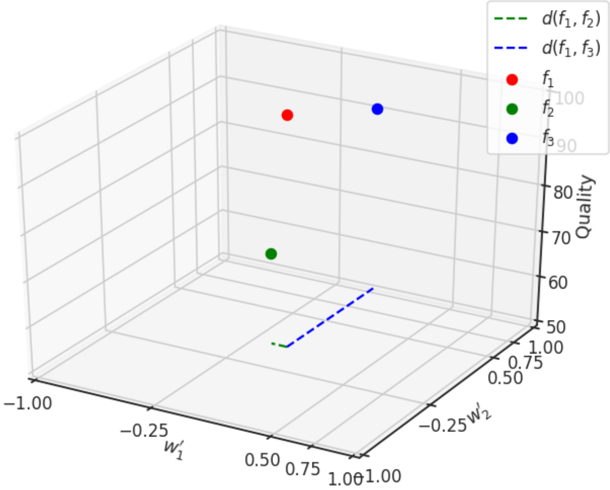
Which model is closer to f_1?



Motivation

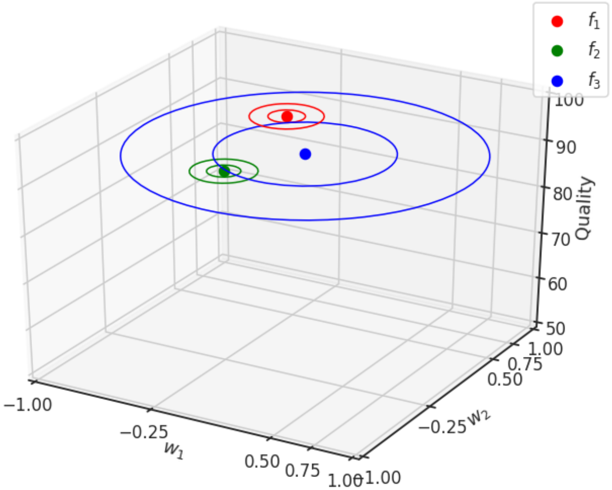
Which model is closer to f_1?

Metric change≈coordinate change. Different metrics represent different model space properties.



Motivation

Which model is closer to f_1?



Definition and properties

Given a parameter space w.
A distance function d is a function, defined on the pair of distributions $p_1, p_2 \to \mathbb{R}_+$.
Probable Properties

- Metric axioms
  - $d(p_1, p_1) = 0$
  - $d(p_1, p_2) = d(p_2, p_1)$
  - $d(p_1, p_2) \leq d(p_1, p_3) + d(p_3, p_2)$
- (Aduenko, 2017)
  - $d \in [0, 1]$
  - d is defined in case of different support for p_1, p_2
  - d is nearly zero, if p_2 is a low-informative distribution
- Performance criteria
  - Tractable
  - Easy to compute

Total variation

For two probability measures P_1, P_2 on the set $\mathfrak{A}$

$$TV = \sup_{\mathfrak{a}\in\mathfrak{A}} |P_1(\mathfrak{a}) - P_2(\mathfrak{a})|$$

Properties:

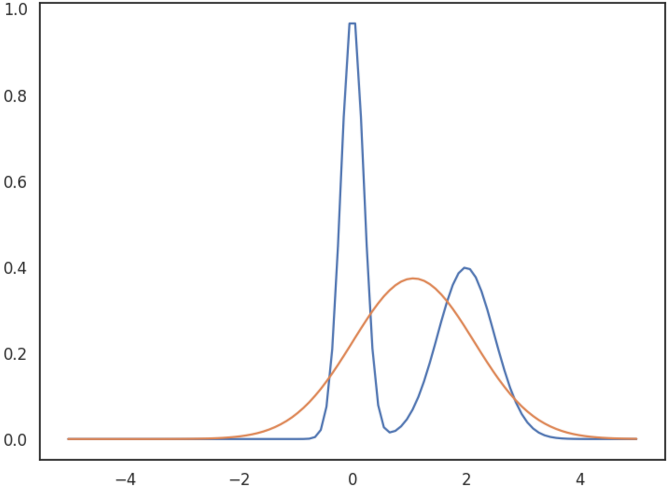
- $0 \leq TV \leq 1$
- TV is a metric
- $TV = 0 \iff P_1 = P_2$
- Scheffe lemma: for differentiable distributions with PDF f_i defined on$\mathbb{R}^d$:

$$TV = \frac{1}{2}\int |f_1(x) - f_2(x)| dx = \frac{1}{2}||f_1 - f_2||_1.$$

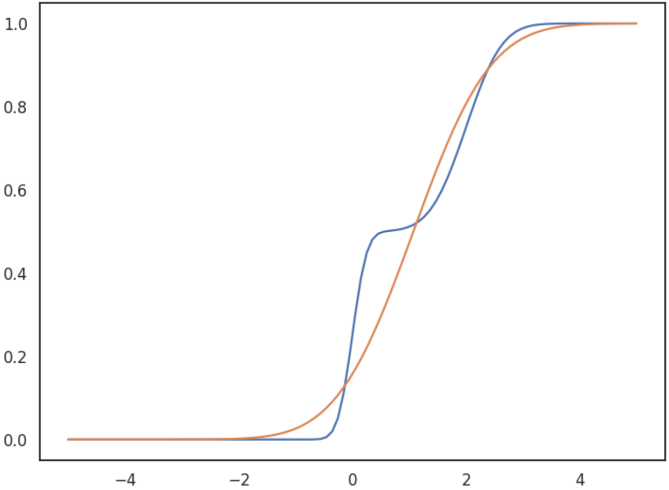
- $TV(\prod_i P_1^i, \prod_i P_2^i) \leq \sum_i TV(P_1^i, P_2^i)$
- Corresponds to statistics in KS-test

Total variation: example

Approximation of Gaussian mixture by Gaussian distribution.

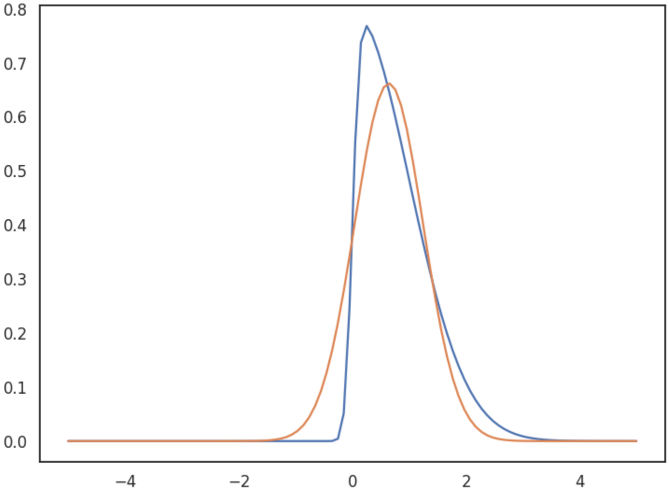


Total variation: example

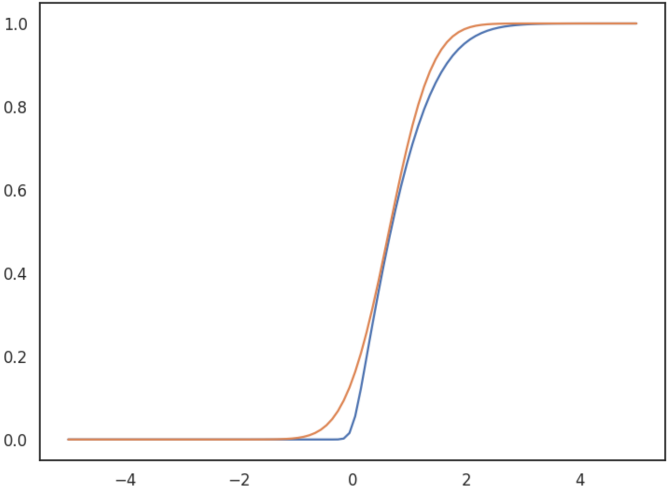


Total variation: example

Approximation of skewed distribution by Gaussian.



Total variation: example

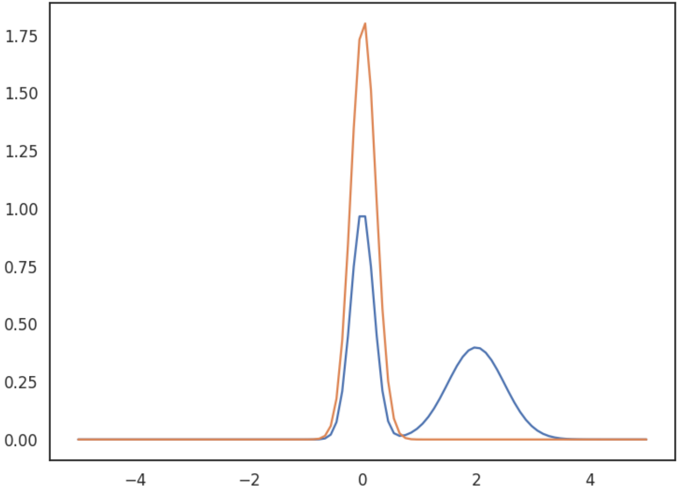


Hellinger distance

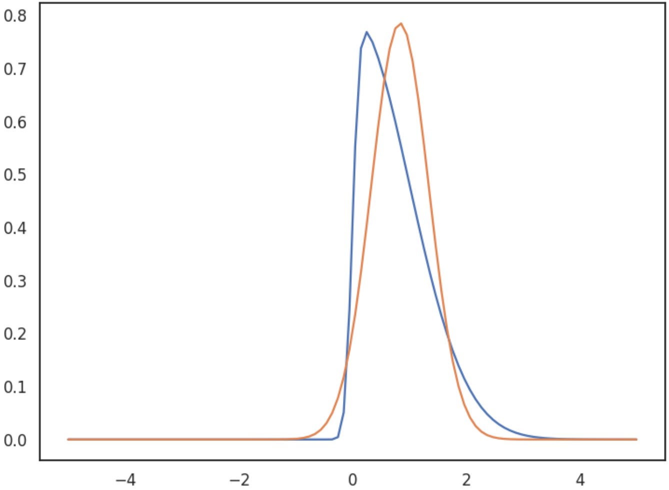
$$H = \sqrt{\int (f_1(x) - f_2(x))^2 \, dx} = ||\sqrt{f_1} - \sqrt{f_2}||_2$$

- $0 \leq H \leq 2$
- H is metric
- $H = 0 \iff P_1 = P_2$
- $H^2(\prod_i P_1^i, \prod_i P_2^i) \leq \sum_i H^2(P_1^i, P_2^i)$
- $1 - H^2 = 1 - \int \sqrt{f_1(x) f_2(x)} dx$

Hellinger distance: example



Hellinger distance: example



KL divergence

$$KL(P_1, P_2) = \int \log \frac{f_1(x)}{f_2(x)} f_1(x) dx$$

- $KL \geq 0$
- KL is not a metric: not a symmetric
- KL is not a metric: does not respect triangle inequality
- $KL = 0 \iff P_1 = P_2$
- $KL(\prod_i P_1^i, \prod_i P_2^i) = \sum_i KL(P_1^i, P_2^i)$
- If we have a dependence between 2 random values w, γ, then

$$KL\left(p_1(\mathrm{w}, \gamma), p_2(\mathrm{w}, \gamma)\right) = KL(p_1(\mathrm{w}), p_2(\mathrm{w})) + \int_{\mathrm{w}} p_1(\mathrm{w}) \int_{\gamma} \log \frac{p_1(\gamma|\mathrm{w})}{p_2(\gamma|\mathrm{w})} p_1(\gamma|\mathrm{w}) d\gamma d\mathrm{w}$$

Entropy

Differential entropy is a generalization of Shannon entropy:

$$h(\mathrm{w}) = -\int_{\mathrm{w}} \log f(\mathrm{w}) f(\mathrm{w}) d\mathrm{w}$$

- Non-invariant under change of variables
  - $h(\mathrm{F}(\mathrm{w})) \leq h(\mathrm{w}) + \int f(\mathrm{w}) \log \left| \frac{\partial \mathrm{F}}{\partial \mathrm{w}} \right| d\mathrm{w}$
  - If F is a bijection, inequality turns into equality
- Can be negative

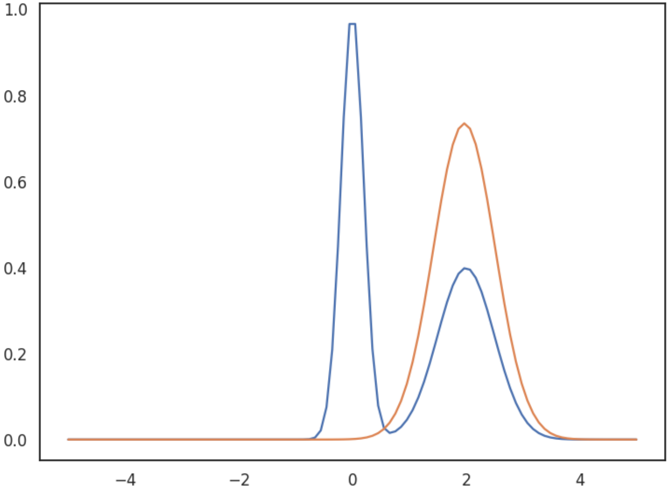
KL is a special case of entropy that

- Invariant under change of variables
- Always positive

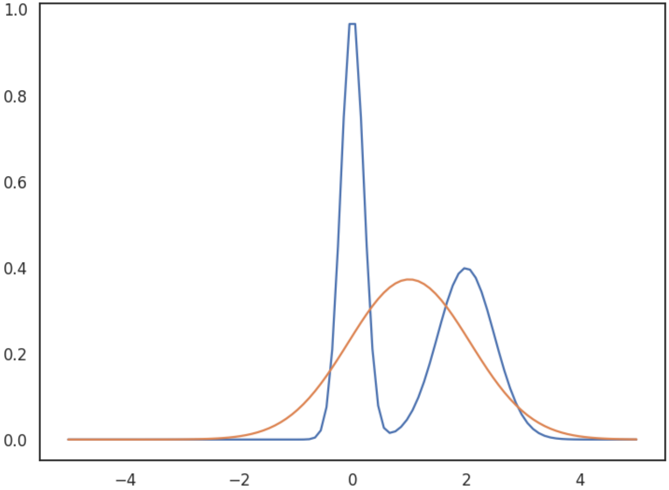
Interpretation of $KL(P_1, P_2)$:

- Amount of information that we can get if use P_1 instead of P_2
- Amount of information that we need to use for coding of data distributed by P_1, if the decoder uses P_2.

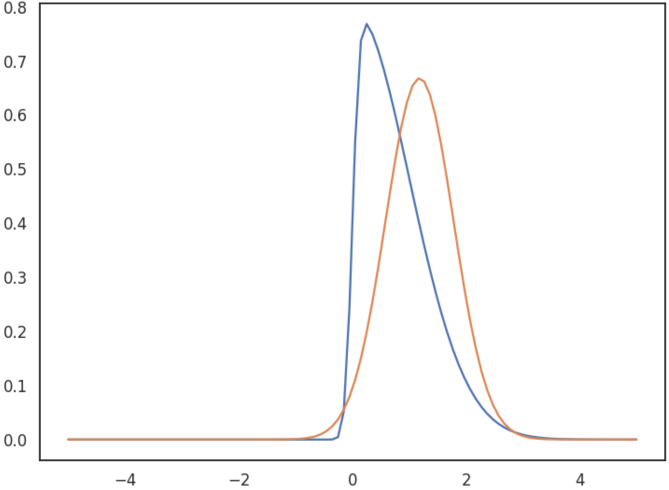
KL: example



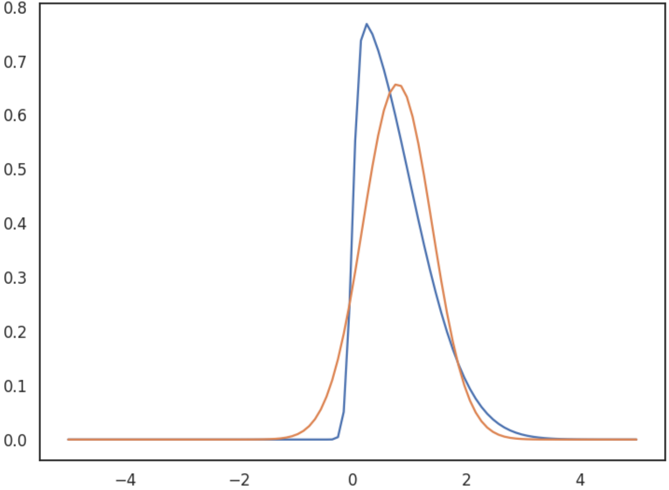
KL: example



KL: example



KL: example

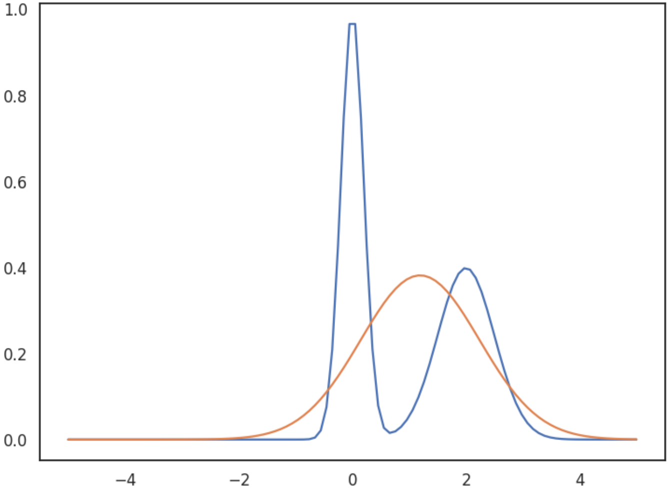


JS

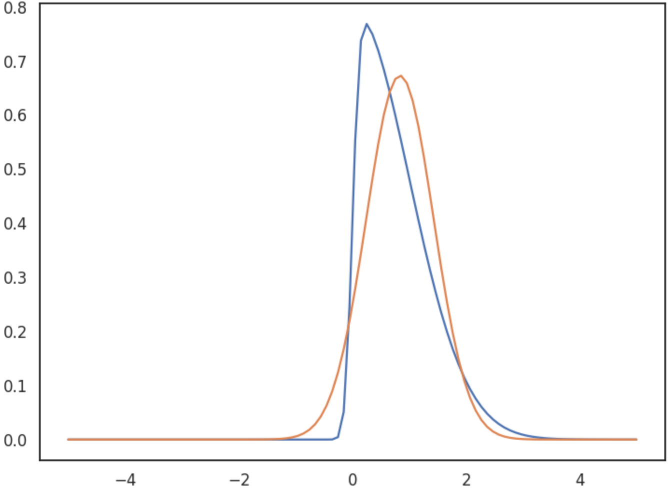
$$JS(P_1, P_2) = \frac{1}{2}KL\left(P_1 \middle| \frac{1}{2}P_1 + \frac{1}{2}P_2\right) + \frac{1}{2}KL\left(P_2 \middle| \frac{1}{2}P_1 + \frac{1}{2}P_2\right)$$

- $0 \leq JS \leq 1$
- $\sqrt{JS}$ is a metric
- $JS = 0 \iff P_1 = P_2$

JS: example

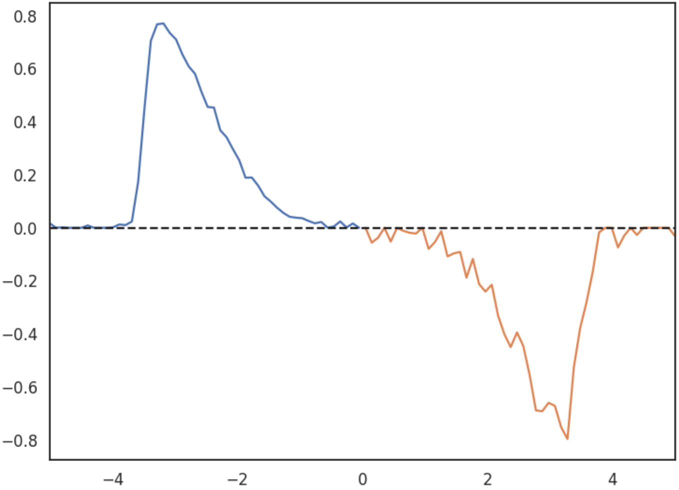


JS: example



Wasserstein distance: motivation

Gaspard Monge: how to move sand into hole in a cheapest way?



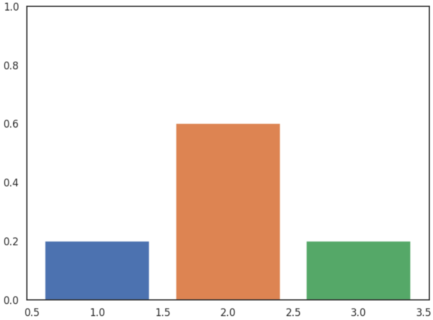
Wasserstein distance: discrete problem

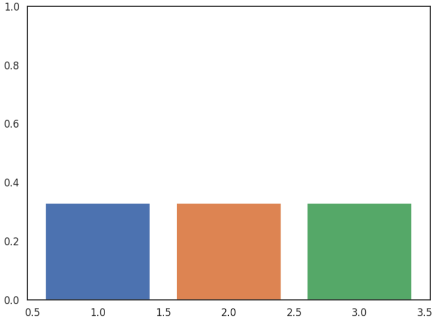
Given two discrete probability measures $p_1(\mathrm{w}_i^1), i \in \{1, \dots n_1\}$, $p_2(\mathrm{w}_j^2), j \in \{1, \dots n_2\}$.
Given a cost matrix C: $c_{ij} \in \mathbb{R}_+$.

We need to find a mapping induced my matrix t_{ij} that:

- $\sum_i t_{ij} = p_2(\mathrm{w}_j^2), \sum_j t_{ij} p_2(\mathrm{w}_i^1)$
- $\sum_i \sum_j c_{ij} t_{ij} \to \min.$

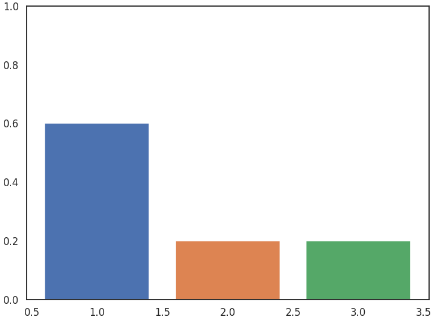
Discrete problem: example

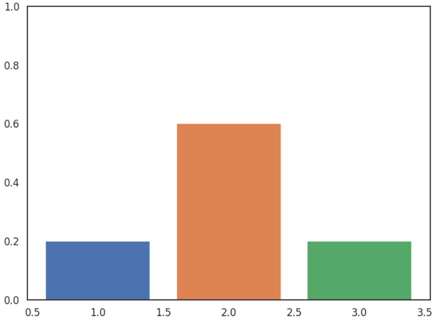




Cost: 0.4

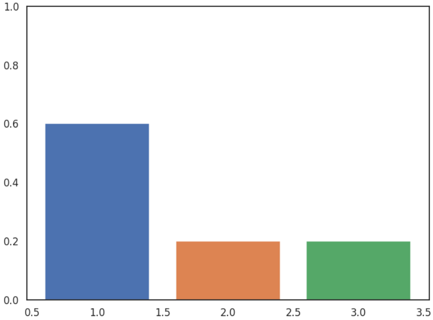
Discrete problem: example

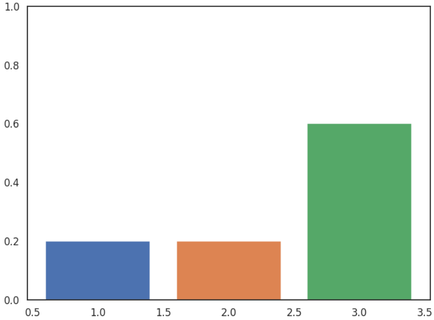




Cost: 0.4

Discrete problem: example





Cost: 0.8

Continuos problem

Given 2 continuos measures $P_1(\mathrm{w}^1), \mathrm{w}^1 \in \mathbb{W}_1,\ P_2(\mathrm{w}^2), \mathrm{w}^2 \in \mathbb{W}_2$.
Given a cost function $C : \mathbb{W}_1 \times \mathbb{W}_2 \to \mathbb{R}_+$.

We need to find a join distribution T on $\mathbb{W}_1 \times \mathbb{W}_2$ that:

- $\int_{\mathbb{W}_1} dT(\mathrm{w}_1, \mathrm{w}_2) = P_1, \quad \int_{\mathbb{W}_2} dT(\mathrm{w}_1, \mathrm{w}_2) = P_2$
- $\int_{\mathbb{W}_1 \times \mathbb{W}_2} C(\mathrm{w}_1, \mathrm{w}_2) dT(\mathrm{w}_1, \mathrm{w}_2) \to \min.$

Dual problem

$$\max_{\hat{T}_1, \hat{T}_2} \int_{\mathbb{W}_1} \hat{T}_1(\mathrm{w}_1) f_1(\mathrm{w}_1) d\mathrm{w}_1 + \int_{\mathbb{W}_2} \hat{T}_2(\mathrm{w}_2) f_2(\mathrm{w}_2) d\mathrm{w}_2$$

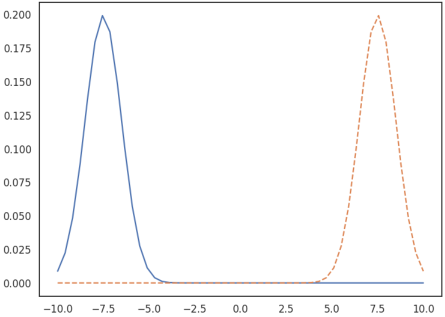
when $\hat{T}_1(\mathrm{w}_1) + \hat{T}_2(\mathrm{w}_2) \leq C(\mathrm{w}_1, \mathrm{w}_2)$

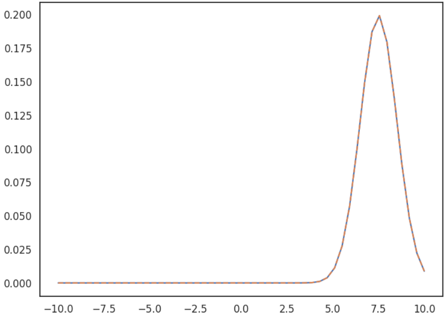
Kantorovich–Rubinstein theorem

Let $\mathbb{W}_1 = \mathbb{W}_2$ and $C = ||\cdot||_1$. Then:

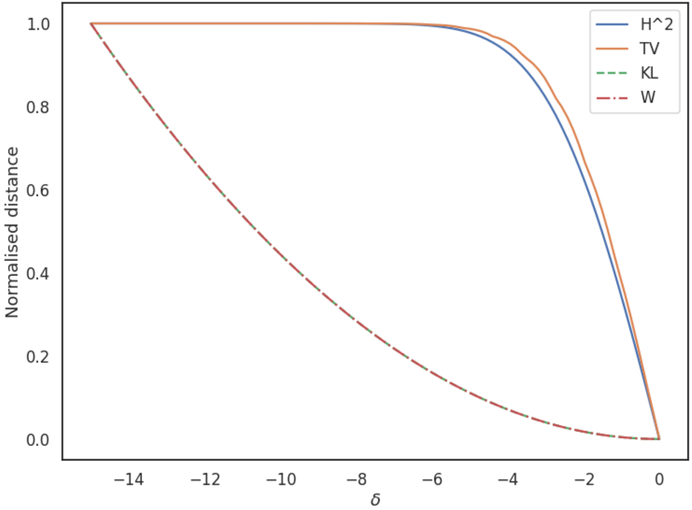
$$\max \hat{T} \in \mathrm{Lip}_1 \int_{\mathbb{W}} \hat{T}(\mathrm{w}) f_1(\mathrm{w}) d\mathrm{w} - \int_{\mathbb{W}} \hat{T}(\mathrm{w}) f_2(\mathrm{w}) d\mathrm{w}$$

Distance between peaks: example

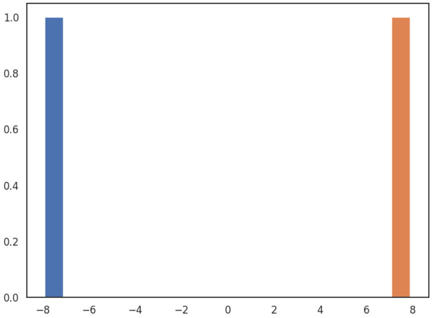


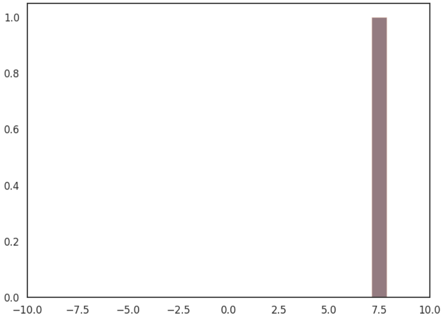


Distance between peaks: example



Distance between peaks: example





Distance between peaks: example

$$TV = 0$$

$$H = 0$$

$$KL = \begin{cases} 0, & \delta = 0 \\ \infty, & \text{otherwise} \end{cases}$$

$$JS = \begin{cases} 0, & \delta = 0 \\ \log 2, & \text{otherwise} \end{cases}$$

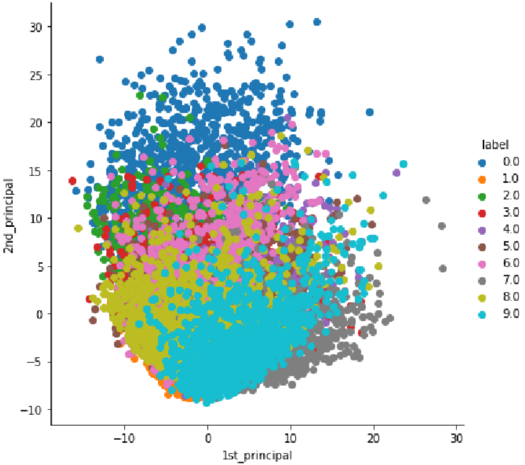
$$W = |\delta|.$$

Conclusion: W-distance has good properties to work with different support sets.

How we can embed models into (probabilistic) vector space?

Principal compnent analysis

$$\boldsymbol{W} = \arg\max Var(\boldsymbol{X}\boldsymbol{W})$$

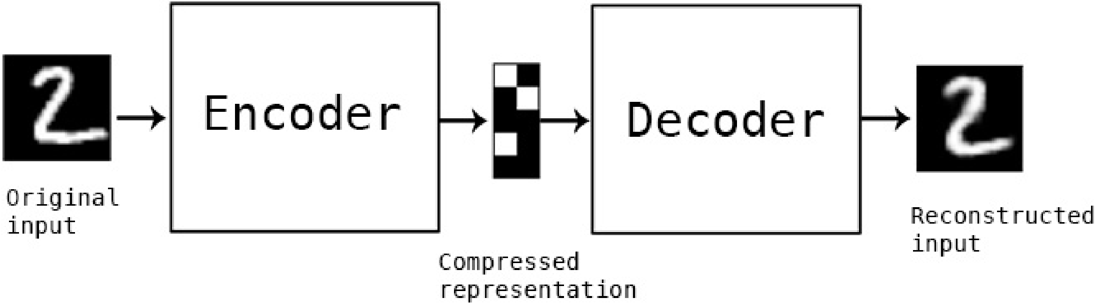


Autoencoder

Autoencoder is a model of dimension reduction:

$$H = \boldsymbol{\sigma}(W_e X),$$

$$||\boldsymbol{\sigma}(W_d H) - X||_2^2 \to \min .$$



Manifold

Manifold is space that can be locally approximated by Euclidian space.

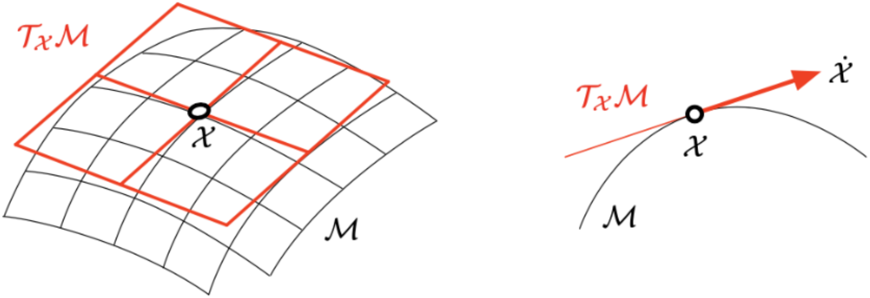
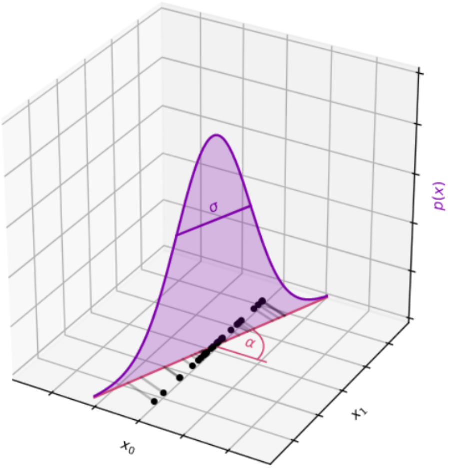
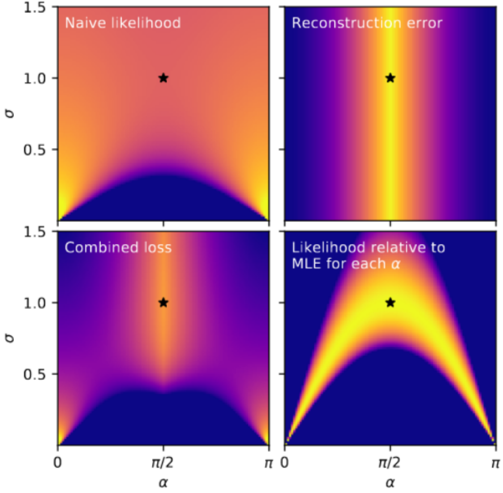


Figure 2. A manifold $\mathcal{M}$ and the vector space $T_\mathcal{X}\mathcal{M}$ (in this case $\cong \mathbb{R}^2$) tangent at the point $\mathcal{X}$, and a convenient side-cut. The velocity element, $\dot{\mathcal{X}} = \partial\mathcal{X}/\partial t$, does not belong to the manifold $\mathcal{M}$ but to the tangent space $T_\mathcal{X}\mathcal{M}$.

Manifold: do we need it?





Autoencoder: generative model?

(Alain, Bengio 2012): consider regularized autoencoder:

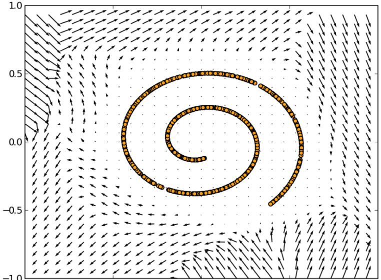
$$||f(x, \sigma) - x||^2,$$

where σ is a noise level.

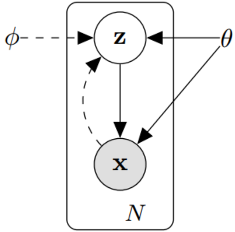
Then

$$\frac{\partial \log p(x)}{\partial x} = \frac{||f(x, \sigma) - x||^2}{\sigma^2} + o(1) \text{ with } \sigma \to 0.$$

Vector field induced by reconstruction error



Variational autoencoder



Let the objects X be generated by latent variable $\mathrm{h} \sim \mathcal{N}(0, \mathrm{I})$:

$$\mathrm{x} \sim p(\mathrm{x}|\mathrm{h}, \mathrm{w}).$$

$p(\mathrm{h}|\mathrm{x}, \mathrm{w})$ is unknown.
Maximize ELBO:

$$\log p(\mathrm{x}|\mathrm{w}) \geq \mathrm{E}_{q_\phi(\mathrm{h}|\mathrm{x})} \log p(\mathrm{x}|\mathrm{h}, \mathrm{w}) - D_{\mathrm{KL}}(q_\phi(\mathrm{h}|\mathrm{x}) || p(\mathrm{h})) \to \max.$$

Distributions $q_\phi(\mathrm{h}|\mathrm{x})$ and $p(\mathrm{x}|\mathrm{h}, \mathrm{w})$ are modeled by neural networks:

$$q_\phi(\mathrm{h}|\mathrm{x}) \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\mathrm{x}), \boldsymbol{\sigma}^2_\phi(\mathrm{x})),$$

$$p(\mathrm{x}|\mathrm{h}, \mathrm{w}) \sim \mathcal{N}(\boldsymbol{\mu}_w(\mathrm{h}), \boldsymbol{\sigma}^2_w(\mathrm{h})),$$

where $\boldsymbol{\mu}$,$\boldsymbol{\sigma}$ are neural network's outputs.

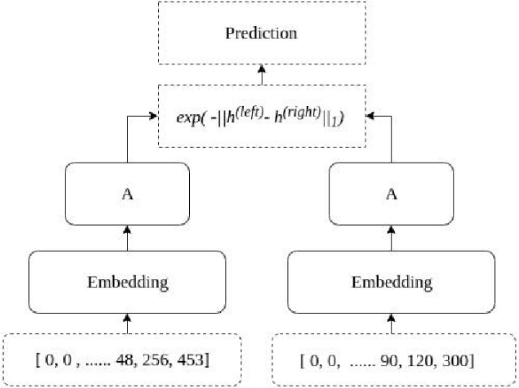
Multiple spaces

Given two spaces: X, Y.
Ho we can build a shared latent space between them?

Multiple spaces

Given two spaces: X, Y.
Ho we can build a shared latent space between them?
Naive method: $||f(x) - g(y)||_2^2 \to \min$ does not work.

Siamese networks



Metric learning

$$D(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\top \boldsymbol{M} (\boldsymbol{x}_1 - \boldsymbol{x}_2)}$$

Triplet loss

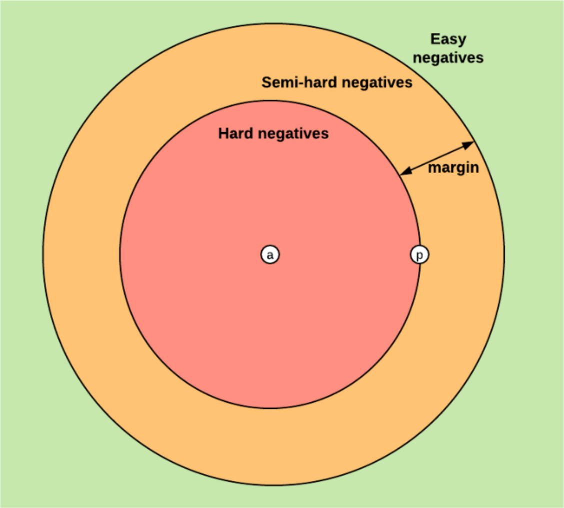
The loss function for each sample in the mini-batch is:

$$L(a, p, n) = \max\{d(a_i, p_i) - d(a_i, n_i) + \mathrm{margin}, 0\}$$

where

$$d(x_i, y_i) = \left\| \mathbf{x}_i - \mathbf{y}_i \right\|_p$$

Triplet loss



Bayesian representation learning with oracle constraints

$$p(t_{i,j,l}) = \int_z p(t_{i,j,l}|z_i, z_j, z_l)p(\boldsymbol{z}_i)p(\boldsymbol{z}_j)p(\boldsymbol{z}_k)dz_i dz_j dz_k,$$

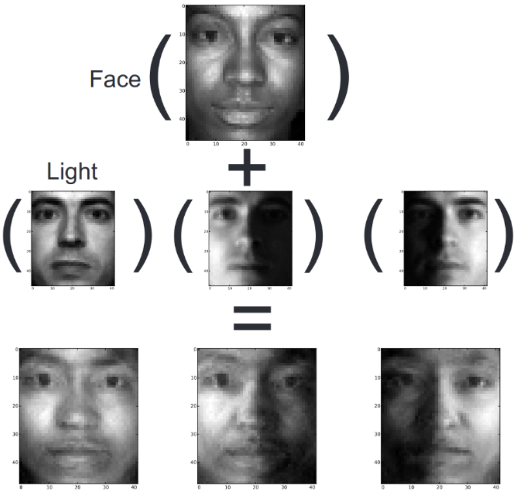
this gives the following likelihood:

$$p(t_{i,j,l}) = Ber(t_{i,j,l}) = \frac{\mathrm{e}^{-D_{i,j}}}{\mathrm{e}^{-D_{i,j}} + \mathrm{e}^{-D_{i,l}}}$$

with

$$D_{a,b} = \sum_{h=1}^{H} D_{a,b}^{h} = -\sum_{h=1}^{H} \left[\mathrm{JS}\left(p(\boldsymbol{z}_a^h)||p(\boldsymbol{z}_b^h)\right)\right].$$

Bayesian representation learning with oracle constraints



Variational learning across domains with triplet information

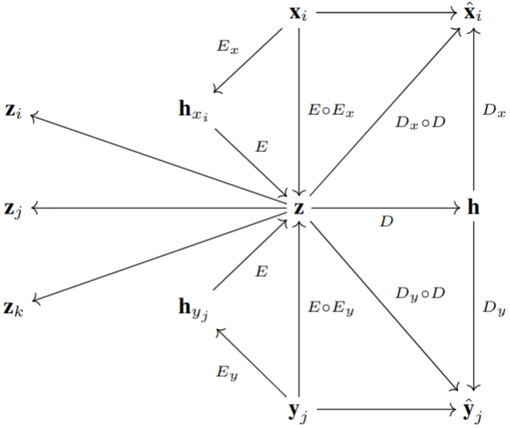


Figure 1: VBTA generative process

Variational learning across domains with triplet information

$$\mathcal{L}_{VBTA} = \mathbb{E}_{q_{\phi_x}(\mathbf{z}_x|\mathbf{x})} \log \frac{p_{\theta_x}(\mathbf{x},\mathbf{y},\mathbf{t},\mathbf{z}_x)}{q_{\phi_x}(\mathbf{z}_x|\mathbf{x})} + \mathbb{E}_{q_{\phi_y}(\mathbf{z}_y|\mathbf{y})} \log \frac{p_{\theta_y}(\mathbf{x},\mathbf{y},\mathbf{t},\mathbf{z}_y)}{q_{\phi_y}(\mathbf{z}_y|\mathbf{y})} =$$

$$= \underbrace{-\Big[KL\big(q_{\phi_{\mathbf{x}}(\mathbf{z}_x|\mathbf{x})}(\mathbf{z}_x|\mathbf{x}) \parallel p_{\theta_{\mathbf{x}}}(\mathbf{z}_x)\big) + KL\big(q_{\phi_{\mathbf{y}}(\mathbf{z}_y|\mathbf{y})}(\mathbf{z}_y|\mathbf{y}) \parallel p_{\theta_{\mathbf{y}}}(\mathbf{z}_y)\big)\Big]}_{\text{Penalty}} +$$

$$+ \underbrace{\Big[\mathbb{E}_{q_{\phi_{\mathbf{x}}}(\mathbf{z}_x|\mathbf{x})}\big[\log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}_x)\big] + \mathbb{E}_{q_{\phi_{\mathbf{y}}}(\mathbf{z}_y|\mathbf{y})}\big[\log p_{\theta_{\mathbf{y}}}(\mathbf{y}|\mathbf{z}_y)\big]\Big]}_{\text{Reconstruction}} +$$

$$+ \underbrace{\Big[\mathbb{E}_{q_{\phi_{\mathbf{x}}}(\mathbf{z}_x|\mathbf{x})}\big[\log p_{\theta_{\mathbf{x}}}(\mathbf{y}|\mathbf{z}_x)\big] + \mathbb{E}_{q_{\phi_{\mathbf{y}}}(\mathbf{z}_y|\mathbf{y})}\big[\log p_{\theta_{\mathbf{y}}}(\mathbf{x}|\mathbf{z}_y)\big]\Big]}_{\text{Cycle-consistency}} +$$

$$+ \underbrace{\mathbb{E}_{q_{\phi_{\mathbf{x}}}(\mathbf{z}_x|\mathbf{x})}\big[\log p(\mathbf{t}|\mathbf{z}_x)\big] + \mathbb{E}_{q_{\phi_{\mathbf{y}}}(\mathbf{z}_y|\mathbf{x})}\big[\log p(\mathbf{t}|\mathbf{z}_y)\big]}_{\text{Triplet likelihood}}$$

Differentiable Neural Architecture Search in Equivalent Space with Exploration Enhancement

- Structure representation: graph supervised encoder
- Structure optimization: DARTS + exploration

Table 1: Comparison results with state-of-the-art NAS approaches on NAS-Bench-201.

Method	CIFAR-10		CIFAR-100		ImageNet-16-120	
	Valid(%)	Test(%)	Valid(%)	Test(%)	Valid(%)	Test(%)
ENAS	37.51±3.19	53.89±0.58	13.37±2.35	13.96±2.33	15.06±1.95	14.84±2.10
RandomNAS*	85.63±0.44	88.58±0.21	60.99±2.79	61.45±2.24	31.63±2.15	31.37±2.51
DARTS (1st)	39.77±0.00	54.30±0.00	15.03±0.00	15.61±0.00	16.43±0.00	16.32±0.00
DARTS (2nd)	39.77±0.00	54.30±0.00	15.03±0.00	15.61±0.00	16.43±0.00	16.32±0.00
SETN	84.04±0.28	87.64±0.00	58.86±0.06	59.05±0.24	33.06±0.02	32.52±0.21
NAO*	82.04±0.21	85.74±0.31	56.36±3.14	59.64±2.24	30.14±2.02	31.35±2.21
GDAS*	90.03±0.13	93.37±0.42	70.79±0.83	70.35±0.80	40.90±0.33	41.11±0.13
E^2NAS	**90.94±0.83**	**93.89±0.47**	**71.83±1.84**	**72.05±1.58**	**45.44±1.24**	**45.77±1.00**

Does Unsupervised Architecture Representation Learning Help Neural Architecture Search?

- Structure representation: graph VAE
- Optimization: unsupervised for encoding models, then RL+BO

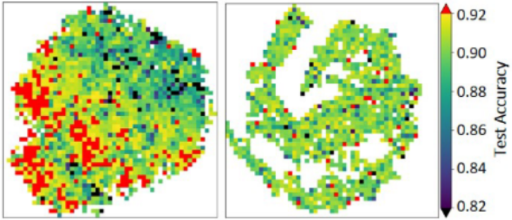


Figure 4: Latent space 2D visualization [65] comparison between *arch2vec* (left) and supervised architecture representation learning (right) on NAS-Bench-101. Color encodes test accuracy. We randomly sample 10, 000 points and average the accuracy in each small area.

References

- Адуенко А. А. Выбор мультимоделей в задачах классификации : дис. – Федер. исслед. центр"Информатика и управление"РАН, 2017.
- Andrew Nobel: Distances and Divergences for Probability Distributions, https://nobel.web.unc.edu/wp-content/uploads/sites/13591/2020/11/Distance-Divergence.pdf
- Про KL с условной вероятонстью: http://akosiorek.github.io/ml/2017/09/10/kl-hierarchical-vae.html
- Kolouri, Cattell, Rohde: Optimal Transport: A Crash Course, http://imagedatascience.com/transport/OTCrashCourse.pdf
- Computational Optimal Transport - https://arxiv.org/pdf/1803.00567.pdf
- Про GAN и WGAN: https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#kullbackleibler-and-jensenshannon-divergence
- Wasserstein GAN - https://arxiv.org/abs/1701.07875
- Bishop C. M. Pattern recognition //Machine learning. – 2006. – Т. 128. – No. 9.
- Bishop C. Bayesian pca //Advances in neural information processing systems. – 1998. – Т. 11.
- Sola J., Deray J., Atchuthan D. A micro Lie theory for state estimation in robotics //arXiv preprint arXiv:1812.01537. – 2018.
- Brehmer J., Cranmer K. Flows for simultaneous manifold learning and density estimation //Advances in Neural Information Processing Systems. – 2020. – Т. 33. – С. 442-453.
- Alain G., Bengio Y. What regularized auto-encoders learn from the data-generating distribution //The Journal of Machine Learning Research. – 2014. – Т. 15. – No. 1. – С. 3563-3593.
- Kingma D. P., Welling M. Auto-encoding variational bayes //arXiv preprint arXiv:1312.6114. – 2013.
- Ranasinghe T., Orˇasan C., Mitkov R. Semantic textual similarity with siamese neural networks //Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). – 2019. – С. 1004-1011.
- https://russianblogs.com/article/5172713037/
- Karaletsos T., Belongie S., Ratsch G. Bayesian representation learning with oracle constraints //arXiv preprint arXiv:1506.05011. – 2015.
- Kuznetsova R., Bakhteev O., Ogaltsov A. Variational learning across domains with triplet information //arXiv preprint arXiv:1806.08672. – 2018.
- Zhang M. et al. Differentiable neural architecture search in equivalent space with exploration enhancement //Advances in Neural Information Processing Systems. – 2020. – Т. 33. – С. 13341-13351.
- Luo R. et al. Neural architecture optimization //Advances in neural information processing systems. – 2018. – Т. 31.
- Yan S. et al. Does unsupervised architecture representation learning help neural architecture search? //Advances in Neural InformationProcessing Systems. – 2020. – Т. 33. – С. 12486-12498