Learning from permutations

Jean-Philippe Vert

Google AI

MINES ParisTech

TECH

2017 is the year of Machine Learning. Here's why

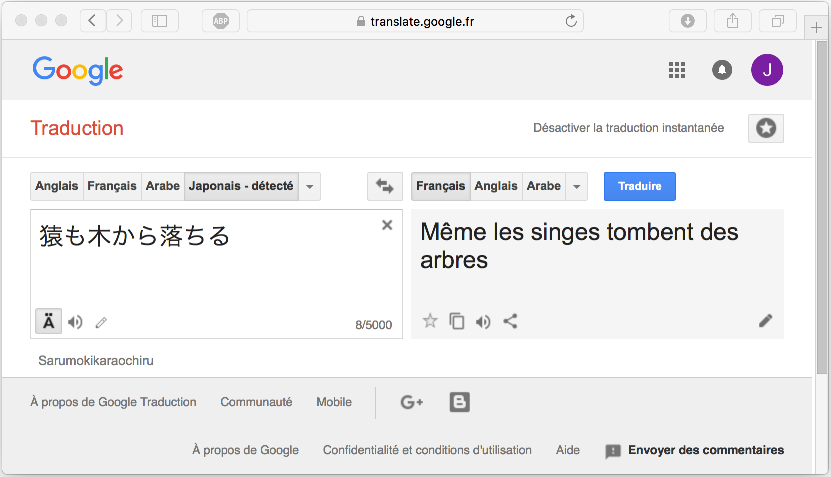
GAURAV SANGWANI | 0 | JAN 13, 2017, 12.51 PM

Machine learning is maybe the most sweltering thing in Silicon Valley at this moment. Particularly deep learning. The reason why it is so hot is on the grounds that it can assume control of numerous repetitive, thoughtless tasks. It'll improve doctors, and make lawyers better lawyers. What's more, it makes cars drive themselves.

Perception

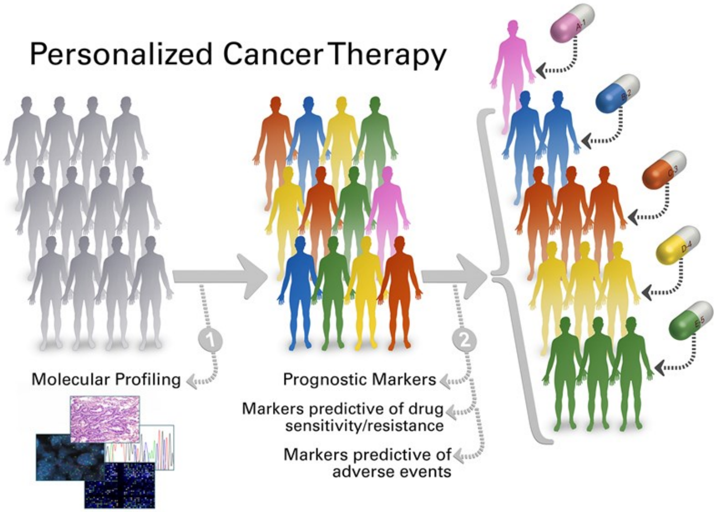

Dr. France Prešeren monument
(1905) by Ivan Zajc and Maks Fabiani... >
History
Audio
Video

Communication



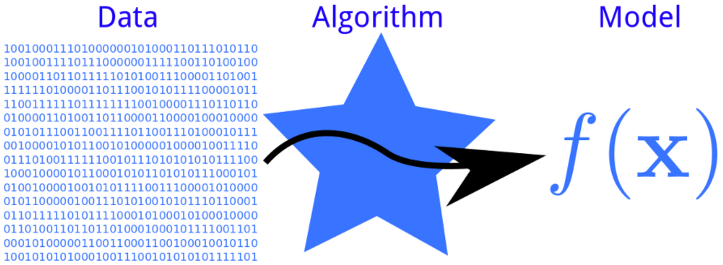
Mobility

Health



https://pct.mdanderson.org

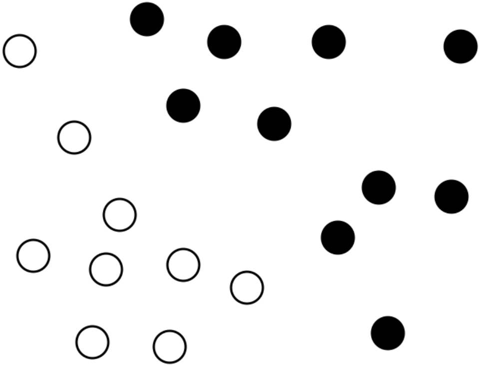
A common process: learning from data



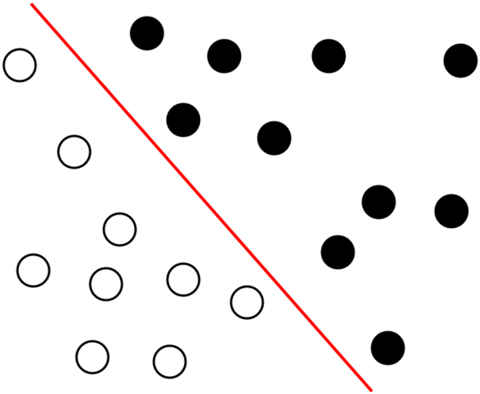
https://www.linkedin.com/pulse/supervised-machine-learning-pega-decisioning-solution-nizam-muhammad

- Given examples (training data), make a machine learn how to predict on new samples, or discover patterns in data
- Statistics + optimization + computer science
- Gets better with more training examples and bigger computers

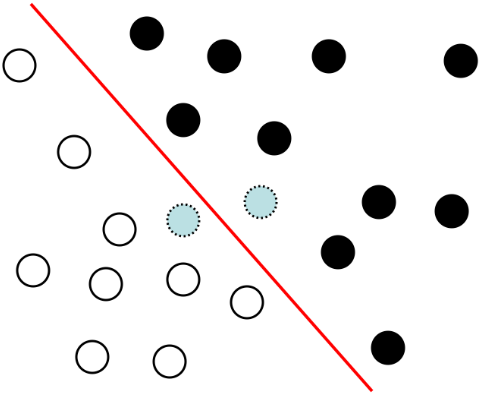
It (more or less) boils down to



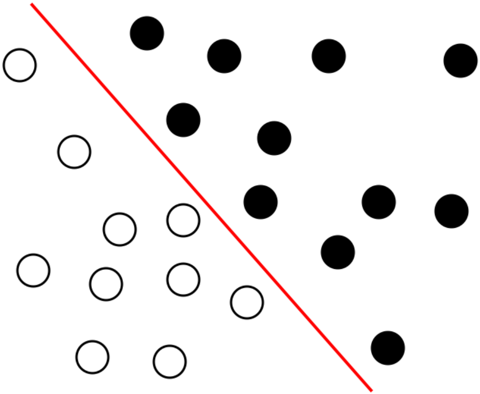
It (more or less) boils down to



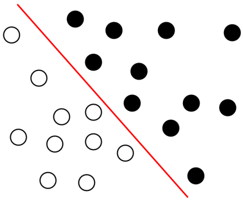
It (more or less) boils down to



It (more or less) boils down to



In practice (eg, linear ridge logistic regression)



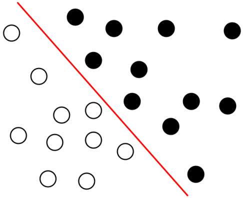
- Input $X_1, \ldots, X_n \in \mathbb{R}^p$
- Output $Y_1, \ldots, Y_n \in \{-1, 1\}$
- Classifier: $f_\beta(X) = \text{sign}(\beta^\top X)$ for $\beta \in \mathbb{R}^p$
- Training

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(\beta^\top X_i \right) + \lambda \|\beta\|^2 \right\}$$

with $\ell_i \left(\beta^\top X_i \right) = \ln \left(1 + e^{-Y_i \beta^\top X_i} \right)$

- Convex optimization problem, scalable (SGD)

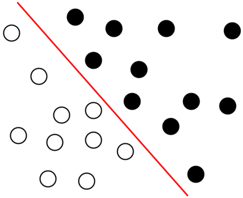
Questions



$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-Y_i \beta^\top X_i}\right) + \lambda \|\beta\|^2 \right\}$$

- How to compute $\hat{\beta}$ in practice?
- Will $f_{\hat{\beta}}$ make good predictions?
- How to train nonlinear models?
- What if inputs are not vectors?
- What if outputs are not binary?
- ...

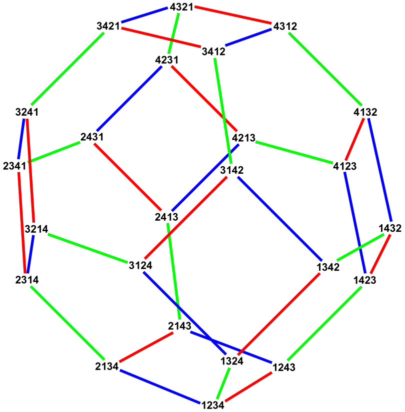
Questions



$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-Y_i \beta^\top X_i}\right) + \lambda \|\beta\|^2 \right\}$$

- How to compute $\hat{\beta}$ in practice?
- Will $f_{\hat{\beta}}$ make good predictions?
- How to train nonlinear models?
- What if inputs are not vectors?
- What if outputs are not binary?
- ...

What if inputs are permutations?



- Permutation: a bijection

$$\sigma : [1, N] \to [1, N]$$

- $\sigma(i)$ = rank of item i
- Composition

$$(\sigma_1 \sigma_2)(i) = \sigma_1(\sigma_2(i))$$

- $\mathbb{S}_N$ the symmetric group
- $|\mathbb{S}_N| = N!$

Examples

- Ranking data

- Ranks extracted from data

(histogram equalization, quantile normalization...)

Learning from permutations

- Assume your data are permutations and you want to learn

$$f : \mathbb{S}_N \to \mathbb{R}$$

- A solutions: embed $\mathbb{S}_N$ to a Euclidean (or Hilbert) space

$$\Phi : \mathbb{S}_N \to \mathbb{R}^p$$

  and learn a linear function:

$$f_\beta(\sigma) = \beta^\top \Phi(\sigma)$$

- The corresponding kernel is

$$K(\sigma_1, \sigma_2) = \Phi(\sigma_1)^\top \Phi(\sigma_2)$$

How to define the embedding $\Phi : \mathbb{S}_N \to \mathbb{R}^p$?

- Should encode interesting features
- Should lead to efficient algorithms
- Should be invariant to renaming of the items, i.e., the kernel should be right-invariant

$$\forall \sigma_1, \sigma_2, \pi \in \mathbb{S}_N, \quad K(\sigma_1\pi, \sigma_2\pi) = K(\sigma_1, \sigma_2)$$

Harmonic analysis on $\mathbb{S}_N$

- A representation of $\mathbb{S}_N$ is a matrix-valued function $\rho : \mathbb{S}_N \to \mathbb{C}^{d_\rho \times d_\rho}$ such that
$$\forall \sigma_1, \sigma_2 \in \mathbb{S}_N\,, \quad \rho(\sigma_1 \sigma_2) = \rho(\sigma_1)\rho(\sigma_2)$$
- A representation is irreductible (irrep) if it is not equivalent to the direct sum of two other representations
- $\mathbb{S}_N$ has a finite number of irreps $\{\rho_\lambda \,:\, \lambda \in \Lambda\}$ where $\Lambda = \{\lambda \vdash N\}$[1] is the set of partitions of N
- For any $f : \mathbb{S}_N \to \mathbb{R}$, the Fourier transform of f is
$$\forall \lambda \in \Lambda\,, \quad \hat{f}(\rho_\lambda) = \sum_{\sigma \in \mathbb{S}_N} f(\sigma)\rho_\lambda(\sigma)$$

[1] $\lambda \vdash N$ iff $\lambda = (\lambda_1, \ldots, \lambda_r)$ with $\lambda_1 \geq \ldots \geq \lambda_r$ and $\sum_{i=1}^{r} \lambda_i = N$

Right-invariant kernels

Bochner's theorem

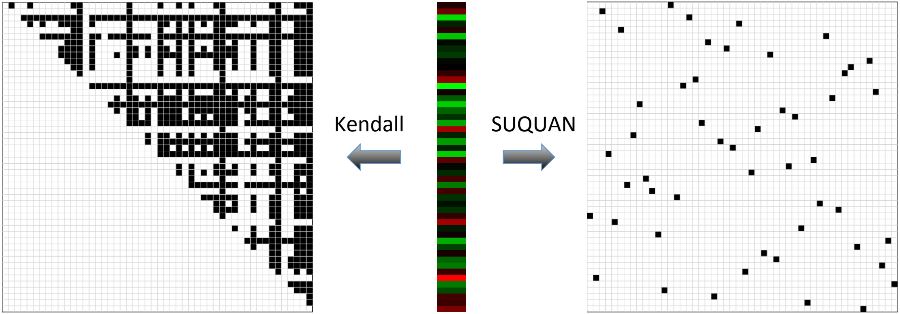
An embedding $\Phi : \mathbb{S}_N \to \mathbb{R}^p$ defines a right-invariant kernel $K(\sigma_1, \sigma_2) = \Phi(\sigma_1)^\top \Phi(\sigma_2)$ if and only there exists $\phi : \mathbb{S}_N \to \mathbb{R}$ such that

$$\forall \sigma_1, \sigma_2 \in \mathbb{S}_N \,, \quad K(\sigma_1, \sigma_2) = \phi(\sigma_2^{-1} \sigma_1)$$

and

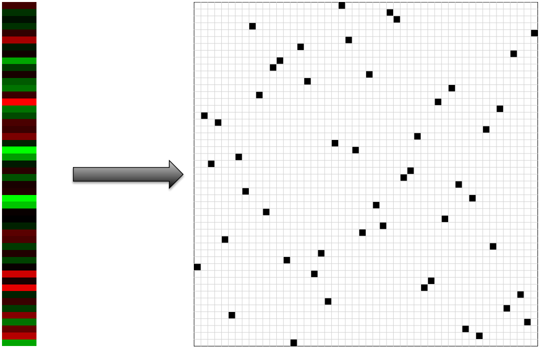
$$\forall \lambda \in \Lambda \,, \quad \hat{\phi}(\rho_\lambda) \succeq 0$$

Some attempts



(Jiao and Vert, 2015, 2017, 2018; Le Morvan and Vert, 2017)

SUQUAN embedding (Le Morvan and Vert, 2017)



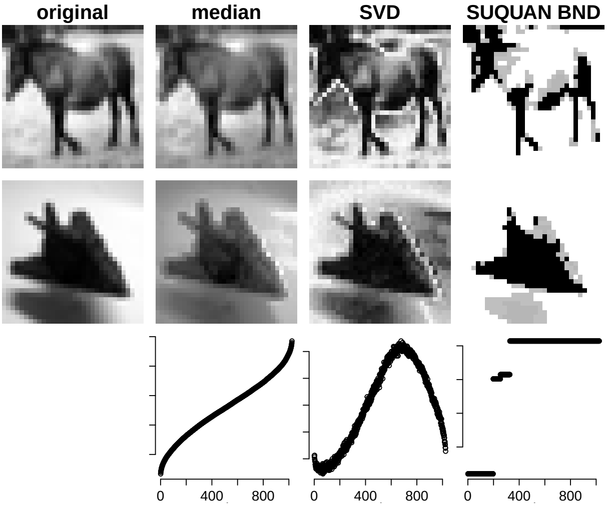
- Let $\Phi(\sigma) = \Pi_\sigma$ the permutation representation (Serres, 1977):

$$[\Pi_\sigma]_{ij} = \begin{cases} 1 & \text{if } \sigma(j) = i\,, \\ 0 & \text{otherwise.} \end{cases}$$

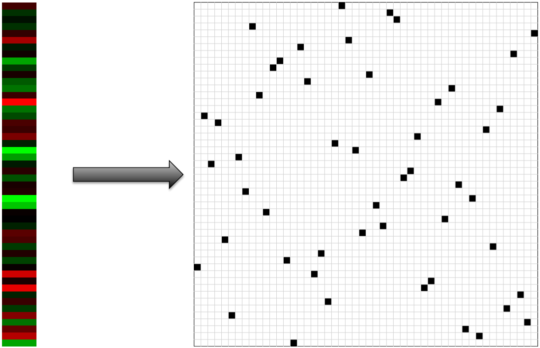
- Leads to new approaches for supervised quantile normalization (SUQUAN) and vector quantization

Example: CIFAR-10

- Discriminate images of horse vs. plane
- Different methods learn different quantile functions

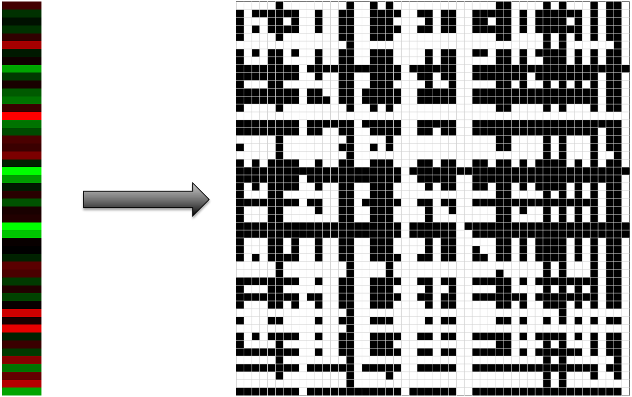


Limits of the SUQUAN embedding



- Linear model on $\Phi(\sigma) = \Pi_\sigma \in \mathbb{R}^{N \times N}$
- Captures first-order information of the form "*i-th feature ranked at the j-th position*"
- What about higher-order information such as "*feature i larger than feature j*"?

The Kendall embedding (Jiao and Vert, 2015, 2017)



$$\Phi_{i,j}(\sigma) = \begin{cases} 1 & \text{if } \sigma(i) < \sigma(j)\,, \\ 0 & \text{otherwise.} \end{cases}$$

Kendall and Mallows kernels

- The Kendall kernel is

$$K_\tau(\sigma, \sigma') = \Phi(\sigma)^\top \Phi(\sigma')$$

- The Mallows kernel is

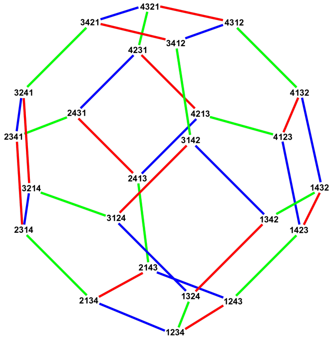
$$\forall \lambda \geq 0 \quad K_M^\lambda(\sigma, \sigma') = e^{-\lambda \| \Phi(\sigma) - \Phi(\sigma') \|^2}$$

Theorem (Jiao and Vert, 2015, 2017)

The Kendall and Mallows kernels are positive definite and can be evaluated in $O(N \log N)$ time

Kernel trick useful with few samples in large dimensions

Remark



Cayley graph of $\mathbb{S}_4$

- Kondor and Barbarosa (2010) proposed the diffusion kernel on the Cayley graph of the symmetric group generated by adjacent transpositions.
- Computationally intensive ($O(N^{2N})$)
- Mallows kernel is written as

$$K_M^{\lambda}(\sigma, \sigma') = e^{-\lambda n_d(\sigma, \sigma')},$$

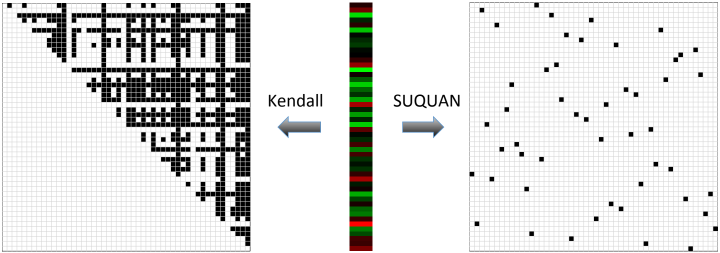
  where $n_d(\sigma, \sigma')$ is the shortest path distance on the Cayley graph.
- It can be computed in $O(N \log N)$

Higher-order kernels (Jiao and Vert, 2018)

$$\Phi(\sigma) = \Pi_\sigma^{\otimes d}$$

- For $d = 1$, this is the SUQUAN embedding
- For $d = 2$, this leads to a new weighted Kendall kernel, where weights can optimized during training

Conclusion



- Machine learning beyond vectors, strings and graphs
- Different embeddings of the symmetric group
- Scalability? Robustness to adversarial attacks?

MERCI!

References

R. E. Barlow, D. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical inference under order restrictions; the theory and application of isotonic regression*. Wiley, New-York, 1972.

Y. Jiao and J.-P. Vert. The Kendall and Mallows kernels for permutations. In *Proceedings of The 32nd International Conference on Machine Learning*, volume 37 of *JMLR:W&CP*, pages 1935–1944, 2015. URL `http://jmlr.org/proceedings/papers/v37/jiao15.html`.

Y. Jiao and J.-P. Vert. The Kendall and Mallows kernels for permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. doi: 10.1109/TPAMI.2017.2719680. URL `http://dx.doi.org/10.1109/TPAMI.2017.2719680`.

Y. Jiao and J.-P. Vert. The weighted kendall and high-order kernels for permutations. Technical Report 1802.08526, arXiv, 2018.

W. R. Knight. A computer method for calculating Kendall's tau with ungrouped data. *J. Am. Stat. Assoc.*, 61(314):436–439, 1966. URL `http://www.jstor.org/stable/2282833`.

M. Le Morvan and J.-P. Vert. Supervised quantile normalisation. Technical Report 1706.00244, arXiv, 2017.

J.-P. Serres. *Linear Representations of Finite Groups*. Graduate Texts in Mathematics. Springer-Verlag New York, 1977. doi: 10.1007/978-1-4684-9458-7. URL `http://dx.doi.org/10.1007/978-1-4684-9458-7`.

O. Sysoev and O. Burdakov. A smoothed monotonic regression via l2 regularization. Technical Report LiTH-MAT-R–2016/01–SE, Department of mathematics, Linköping University, 2016. URL `http://liu.diva-portal.org/smash/get/diva2:905380/FULLTEXT01.pdf`.

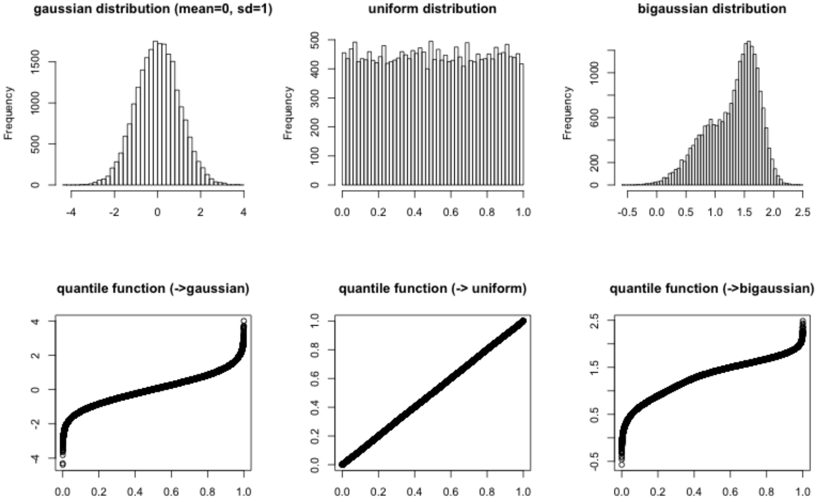
The quantile normalization (QN) embedding

- Data: permutation $\sigma \in \mathbb{S}_N$ where $\sigma(i)$= rank of item/feature i
- Fix a target quantile $q \in \mathbb{R}^N$
- Define $\Phi_q : \mathbb{S}_N \to \mathbb{R}^N$ by

$$\forall \sigma \in \mathbb{S}_N\,, \quad [\Phi_q(\sigma)]_i = q_{\sigma(i)}$$

- "Keep the order, change the values"

How to choose a "good" target distribution?



SUQUAN (Le Morvan and Vert, 2017)

- Learn after standard QN:
  1. Fix q arbitrarily
  2. QN all samples to get $\Phi_q(\sigma_1), \ldots, \Phi_q(\sigma_n)$
  3. Learn a model on normalized data, e.g.:

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^N} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(\beta^\top \Phi_q(\sigma_i) \right) + \lambda \|\beta\|^2 \right\}$$

- Supervised QN (SUQUAN): jointly learn q and the model:

$$\left(\hat{\beta}, \hat{q}\right) = \operatorname*{argmin}_{\beta, q \in \mathbb{R}^N} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(\beta^\top \Phi_q(\sigma_i) \right) + \lambda \|\beta\|^2 + \gamma \Omega(q) \right\}$$

SUQUAN (Le Morvan and Vert, 2017)

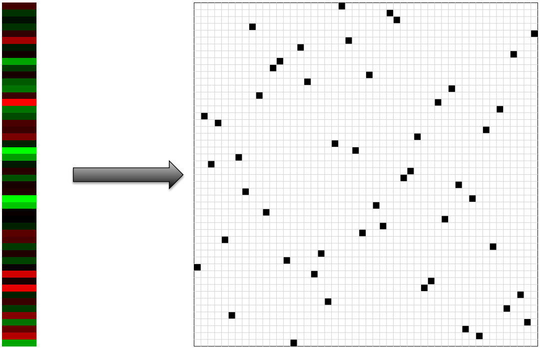
- Learn after standard QN:
  1. Fix q arbitrarily
  2. QN all samples to get $\Phi_q(\sigma_1), \ldots, \Phi_q(\sigma_n)$
  3. Learn a model on normalized data, e.g.:

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^N} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(\beta^\top \Phi_q(\sigma_i) \right) + \lambda \|\beta\|^2 \right\}$$

- Supervised QN (SUQUAN): jointly learn q and the model:

$$\left(\hat{\beta}, \hat{q}\right) = \operatorname*{argmin}_{\beta, q \in \mathbb{R}^N} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(\beta^\top \Phi_q(\sigma_i) \right) + \lambda \|\beta\|^2 + \gamma \Omega(q) \right\}$$

Computing $\Phi_q(\sigma)$



For $\sigma \in \mathbb{S}_N$ let the permutation representation (Serres, 1977):

$$[\Pi_\sigma]_{ij} = \begin{cases} 1 & \text{if } \sigma(j) = i\,, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\Phi_q(\sigma) = \Pi_\sigma^\top q$$

Linear SUQAN as rank-1 matrix regression

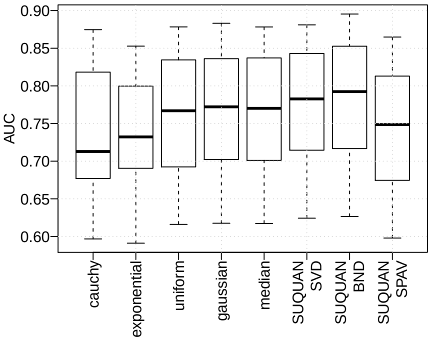
- Linear SUQAN therefore solves

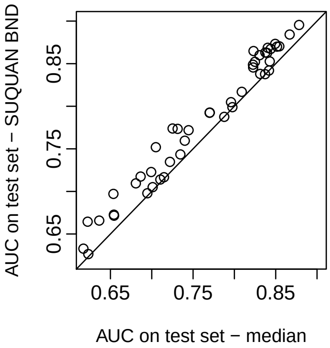
$$\min_{\beta, q \in \mathbb{R}^N} \left\{ \frac{1}{n} \ell_i \left(\beta^\top \Phi_q(\sigma_i) \right) + \lambda \|\beta\|^2 + \gamma \Omega(q) \right\}$$
$$= \min_{\beta, q \in \mathbb{R}^N} \left\{ \frac{1}{n} \ell_i \left(\beta^\top \Pi_{\sigma_i}^\top q \right) + \lambda \|\beta\|^2 + \gamma \Omega(q) \right\}$$
$$= \min_{\beta, q \in \mathbb{R}^N} \left\{ \frac{1}{n} \ell_i \left(< q\beta^\top, \Pi_{\sigma_i} >_{\text{Frobenius}} \right) + \lambda \|\beta\|^2 + \gamma \Omega(q) \right\}$$

- A particular linear model to estimate a rank-1 matrix $M = q\beta^\top$
- Each sample $\sigma \in \mathbb{S}_N$ is represented by the matrix $\Pi_\sigma \in \mathbb{R}^{n \times n}$
- Non-convex
- Alternative optimization of f and w is easy

Experiments: CIFAR-10

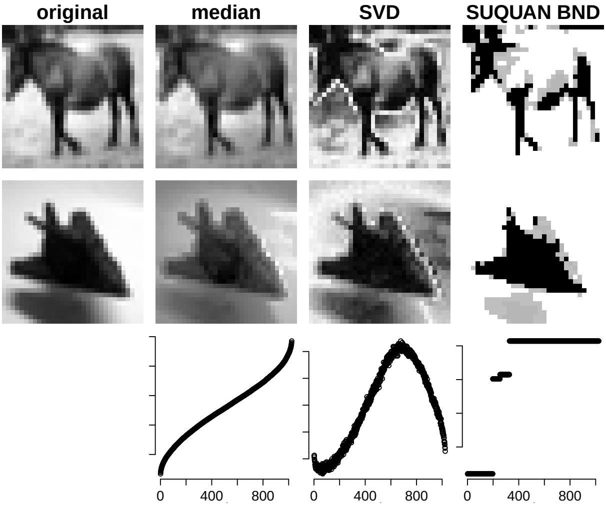
- Image classification into 10 classes (45 binary problems)
- $N = 5,000$ per class, $p = 1,024$ pixels





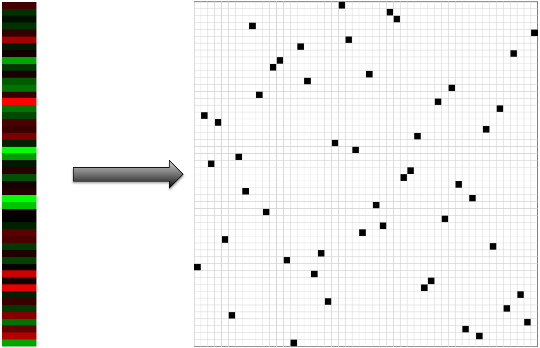
Experiments: CIFAR-10

- Example: horse vs. plane
- Different methods learn different quantile functions



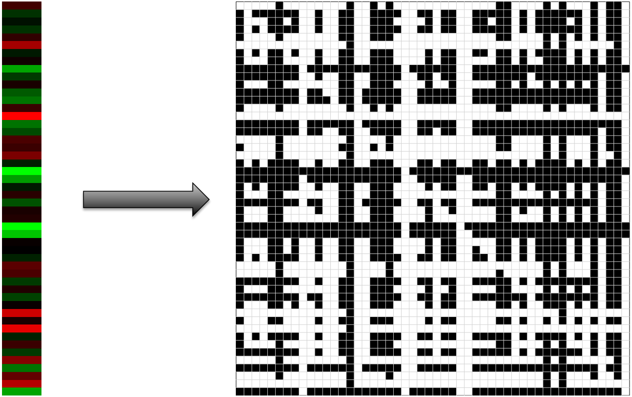
Outline

Limits of the QN embedding



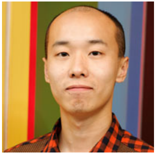
- Linear model on $\Phi(\sigma) = \Pi_\sigma \in \mathbb{R}^{N \times N}$
- Captures first-order information of the form "*i-th feature ranked at the j-th position*"
- What about higher-order information such as "*feature i larger than feature j*"?

Another representation



$$\Phi_{i,j}(\sigma) = \begin{cases} 1 & \text{if } \sigma(i) < \sigma(j), \\ 0 & \text{otherwise.} \end{cases}$$

Kendall and Mallows kernels



- The Kendall kernel is

$$K_\tau(\sigma, \sigma') = \Phi(\sigma)^\top \Phi(\sigma')$$

- The Mallows kernel is

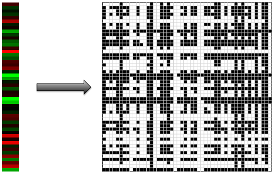
$$\forall \lambda \geq 0 \quad K_M^\lambda(\sigma, \sigma') = e^{-\lambda \| \Phi(\sigma) - \Phi(\sigma') \|^2}$$

Theorem (Jiao and Vert, 2015, 2017)

The Kendall and Mallows kernels are positive definite and can be evaluated in $O(N \log N)$ time

Kernel trick useful with few samples in large dimensions

Proof



For any two permutations $\sigma, \sigma' \in \mathbb{S}_N$:

- Inner product

$$\Phi(\sigma)^\top \Phi(\sigma') = \sum_{1 \leq i \neq j \leq N} \mathbb{1}_{\sigma(i)<\sigma(j)} \mathbb{1}_{\sigma'(i)<\sigma'(j)} = n_c(\sigma, \sigma')$$

  n_c = number of concordant pairs

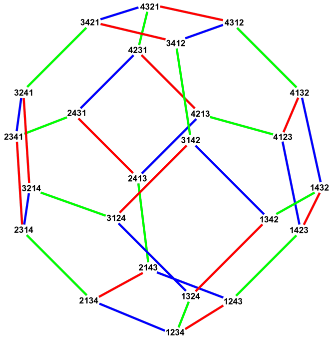
- Distance

$$\| \Phi(\sigma) - \Phi(\sigma') \|^2 = \sum_{1 \leq i,j \leq N} (\mathbb{1}_{\sigma(i)<\sigma(j)} - \mathbb{1}_{\sigma'(i)<\sigma'(j)})^2 = 2n_d(\sigma, \sigma')$$

  n_d = number of discordant pairs

n_c and n_c can be computed in $O(N \log N)$ (Knight, 1966)

Related work



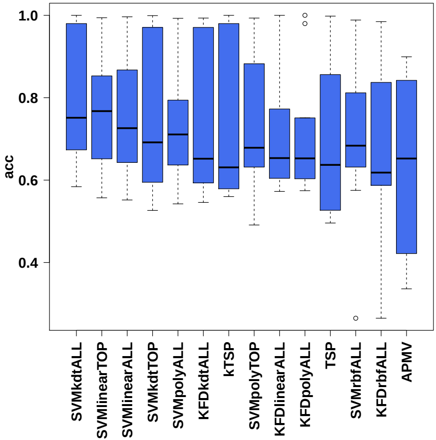
Cayley graph of $\mathbb{S}_4$

- Kondor and Barbarosa (2010) proposed the diffusion kernel on the Cayley graph of the symmetric group generated by adjacent transpositions.
- Computationally intensive ($O(N^{2N})$)
- Mallows kernel is written as

$$K_M^{\lambda}(\sigma, \sigma') = e^{-\lambda n_d(\sigma, \sigma')},$$

where $n_d(\sigma, \sigma')$ is the shortest path distance on the Cayley graph.
- It can be computed in $O(N \log N)$

Applications



Average performance on 10 microarray classification problems (Jiao and Vert, 2017).

Constraints on f

- Ridge

$$\mathcal{F}_0 = \left\{ f \in \mathbb{R}^p \, : \, \frac{1}{p} \sum_{i=1}^{p} f_i^2 \leq 1 \right\} .$$

- Non-decreasing

$$\mathcal{F}_{\mathrm{BND}} = \mathcal{F}_0 \cap \mathcal{I}_0 \, , \quad \text{where} \quad \mathcal{I}_0 = \{ f \in \mathbb{R}^p \, : \, f_1 \leq f_2 \leq \ldots \leq f_p \}$$

- Non-decreasing and smooth

$$\mathcal{F}_{\mathrm{SPAV}} = \left\{ f \in \mathcal{I}_0 \, : \, \sum_{j=1}^{p-1} (f_{j+1} - f_j)^2 \leq 1 \right\} .$$

SUQUAN-BND and SUQUAN-PAVA

Algorithm 2: SUQUAN-BND and SUQUAN-SPAV

Input: $(x_1, y_1), \ldots, (x_n, y_n)$, $f_{init} \in \mathcal{I}_0$, $\lambda \in \mathbb{R}$

Output: $f \in \mathcal{I}_0$ target quantile

1: **for** $i = 1$ to n **do**

2: $\quad rank_i, order_i \leftarrow \text{sort}(x_i)$

3: **end for**

4: $w, b \leftarrow \underset{w,b}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(w^\top f_{init}[rank_i] + b \right) + \lambda ||w||^2$

(standard linear model optimisation)

5: $f \leftarrow \underset{f \in \mathcal{F}_{BND}}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(f^\top w[order_i] + b \right)$

(isotonic optimisation problem using PAVA as prox)

OR

$f \leftarrow \underset{f \in \mathcal{F}_{SPAV}}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(f^\top w[order_i] + b \right)$

(smoothed isotonic optimisation problem using SPAV as prox)

- Alternate optimization in *w* and *f*, monotonicity constraint on *f*
- Accelerated proximal gradient optimization for *f*, using the Pool Adjacent Violators Algorithm (PAVA, Barlow et al. (1972)) or the Smoothed Pool Adjacent Violators algorithm (SPAV, Sysoev and Burdakov (2016)) as proximal operator.

A variant: SUQUAN-SVD

Algorithm 1: SUQUAN-SVD

Input:
$(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \{-1, 1\}$

Output: $f \in \mathcal{F}_0$ target quantile

```
M_LDA ← 0 ∈ R^{p×p}
n_{+1} ← |{i : y_i = +1}|
n_{-1} ← |{i : y_i = −1}|
for i = 1 to n do
    Compute Π_{x_i} (by sorting x_i)
    M_LDA ← M_LDA + (y_i / n_{y_i}) Π_{x_i}
end for
(σ, w, f) ← SVD(M_LDA, 1)
```

- Ridge penalty (no monotonicity constraint), equivalent to rank-1 regression problem
- SVD finds the closest rank-1 matrix to the LDA solution:

$$M_{LDA} = \frac{1}{n_+} \sum_{i\,:\,y_i=+1} \Pi_{x_i} - \frac{1}{n_-} \sum_{i\,:\,y_i=+1} \Pi_{x_i}$$

- Complexity $O(np\ln(p))$ (same as QN only)

Experiments: Simulations

- True distribution of X entries is normal
- Corrupt data with a cauchy, exponential, uniform or bimodal gaussian distributions.
- $p = 1000$, n varies, logistic regression.