

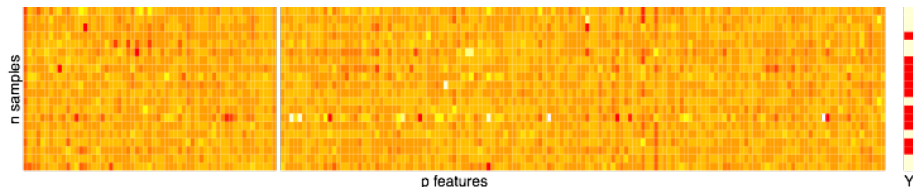
Learning on the symmetric group

Jean-Philippe Vert



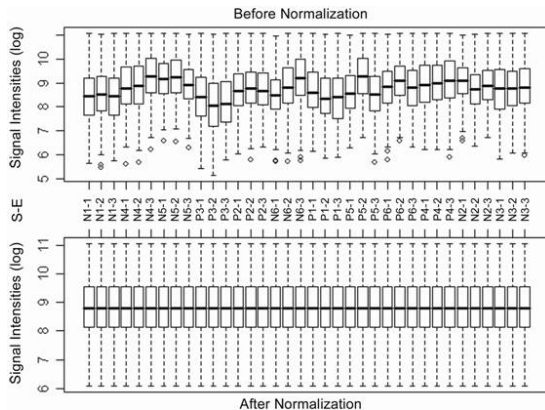
France-Japan Machine Learning workshop, ENS Paris,
Septembre 21, 2017

Motivation



- X gene expression profile of each patient
- Y survival information of each patient
- $n = 10^2 \sim 10^4$
- $p = 2 \times 10^4$
- Goal: learn to predict Y from X
- But... where does X come from?

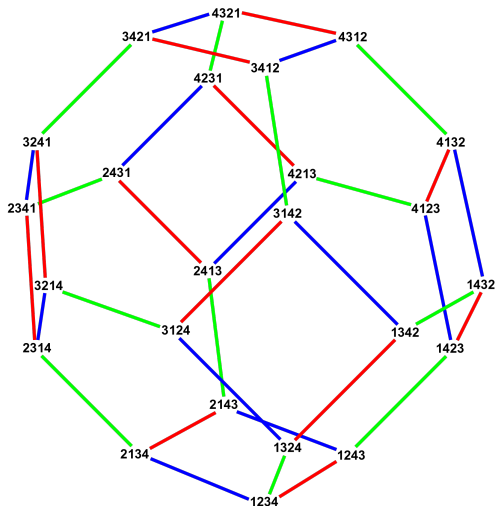
From raw data to X



- **Between-sample** variability: batch effect, drift over time, ...
- Typical pre-processing: **Quantile normalization** (per sample)
- Only the **relative ordering of features** within each sample is used

Learning on the symmetric group

- The symmetric group S_p is the set of permutations of $\{1, \dots, p\}$
- How to estimate $Y = f(X)$ where $X \in S_p$?



Outline

- 1 Supervised quantile normalization
- 2 The Kendall and Mallows kernels
- 3 Conclusion

Outline

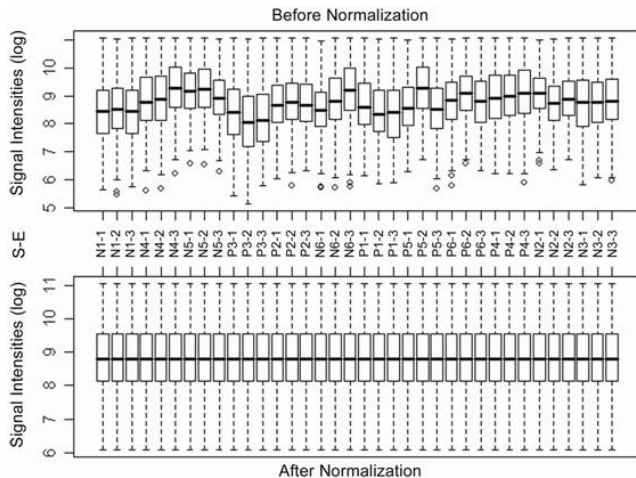
- 1 Supervised quantile normalization
- 2 The Kendall and Mallows kernels
- 3 Conclusion



Marine Le Morvan

<https://arxiv.org/abs/1706.00244>

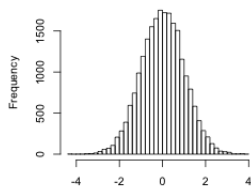
Standard full quantile normalization



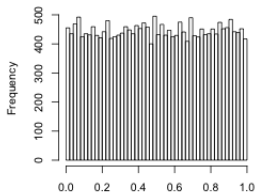
Typically followed by a predictive model $f(X)$ on the normalized data

How to choose a "good" target distribution?

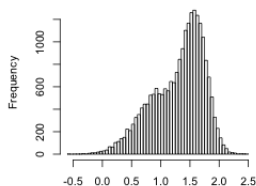
gaussian distribution (mean=0, sd=1)



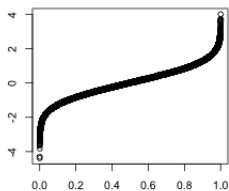
uniform distribution



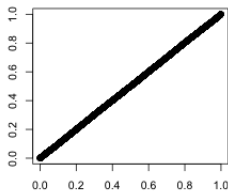
bigaussian distribution



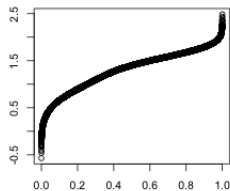
quantile function (->gaussian)



quantile function (-> uniform)

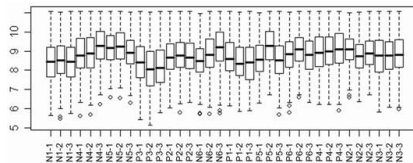


quantile function (->bigaussian)

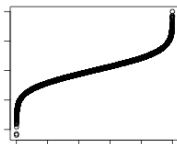


Notations

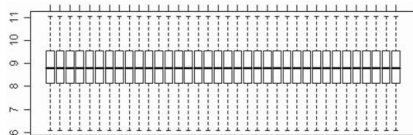
- $x_1, \dots, x_n \in \mathbb{R}^p$ a set of p -dimensional samples



- $f \in \mathbb{R}^p$ a non-decreasing target distribution (CDF)



- For $x \in \mathbb{R}^p$, let $\Phi_f(x) \in \mathbb{R}^p$ be the data after QN with target distribution f



From QN to supervised QN (SUQUAN)

Standard approaches: learn model **after** QN preprocessing:

- 1 **Fix** f arbitrarily
- 2 QN all samples to get $\Phi_f(x_1), \dots, \Phi_f(x_n)$
- 3 Learn a generalized linear model (w, b) on normalized data:

$$\min_{w,b} \left\{ \frac{1}{n} \sum_{i=1}^n \ell_i \left(w^\top \Phi_f(x_i) + b \right) + \lambda \Omega(w) \right\}$$

SUQUAN: **jointly** learn f and (w, b) :

$$\min_{w,b,f} \left\{ \frac{1}{n} \sum_{i=1}^n \ell_i \left(w^\top \Phi_f(x_i) + b \right) + \lambda \Omega(w) + \gamma \Omega_2(f) \right\}$$

SUQAN as matrix regression (1/2)

- For $x \in \mathbb{R}^p$, let $\Pi_x \in \mathbb{R}^{p \times p}$ the permutation matrix of x 's entries:

$$[\Pi_x]_{ij} = \mathbf{1}(x_j \text{ is the } i\text{-th smallest feature})$$

- Quantile normalized x with target distribution f is:

$$\Phi_f(x) = \Pi_x f$$

- Example:

$$x = \begin{pmatrix} 4.5 \\ 1.2 \\ 10.1 \\ 8.9 \end{pmatrix} \quad \Pi_x = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad f = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}$$

$$\Phi_f(x) = \Pi_x f = \begin{pmatrix} 1 \\ 0 \\ 4 \\ 3 \end{pmatrix}$$

SUQAN as matrix regression (2/2)

- SUQUAN solves

$$\begin{aligned} & \min_{w,b,f} \left\{ \frac{1}{n} \sum_{i=1}^n \ell_i \left(w^\top \Phi_f(x_i) + b \right) + \lambda \Omega(w) + \gamma \Omega_2(f) \right\} \\ &= \min_{w,b,f} \left\{ \frac{1}{n} \sum_{i=1}^n \ell \left(w^\top \Pi_{x_i} f + b \right) + \lambda \Omega(w) + \gamma \Omega_2(f) \right\} \\ &= \min_{w,b,f} \left\{ \frac{1}{n} \sum_{i=1}^n \ell \left(\langle w f^\top, \Pi_{x_i} \rangle_F + b \right) + \lambda \Omega(w) + \gamma \Omega_2(f) \right\} \end{aligned}$$

- A particular **rank-1 matrix optimization**, x is replaced by Π_x
- Non-convex
- Local optimum found by alternatively optimizing f and w

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

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

- Non-decreasing and smooth

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

SUQUAN-BND and SUQUAN-PAVA

Algorithm 2: SUQUAN-BND and SUQUAN-SPAV

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

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

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

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

3: **end for**

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

(standard linear model optimisation)

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

(isotonic optimisation problem using PAVA as prox)

OR

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

(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:

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

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

1: $M_{LDA} \leftarrow 0 \in \mathbb{R}^{p \times p}$

2: $n_{+1} \leftarrow |\{i : y_i = +1\}|$

3: $n_{-1} \leftarrow |\{i : y_i = -1\}|$

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

5: Compute Π_{x_i} (by sorting x_i)

6: $M_{LDA} \leftarrow M_{LDA} + \frac{y_i}{n_{y_i}} \Pi_{x_i}$

7: **end for**

8: $(\sigma, w, f) \leftarrow SVD(M_{LDA}, 1)$

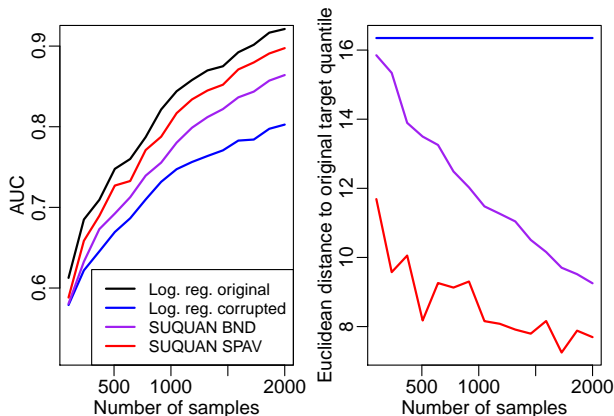
- Ridge penalty (no monotonicity constraint), equivalent to rank-1 regression problem
- SVD finds the closest rank-1 matrix to the LDA solution:

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

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

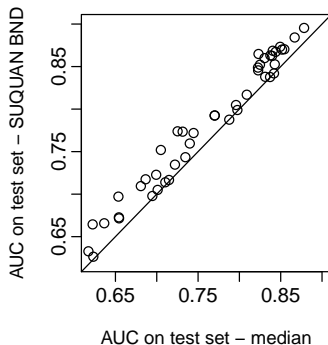
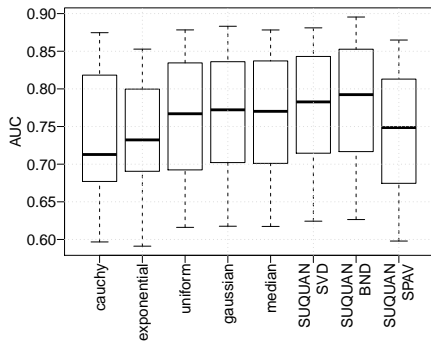
Experiments: Simulations

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



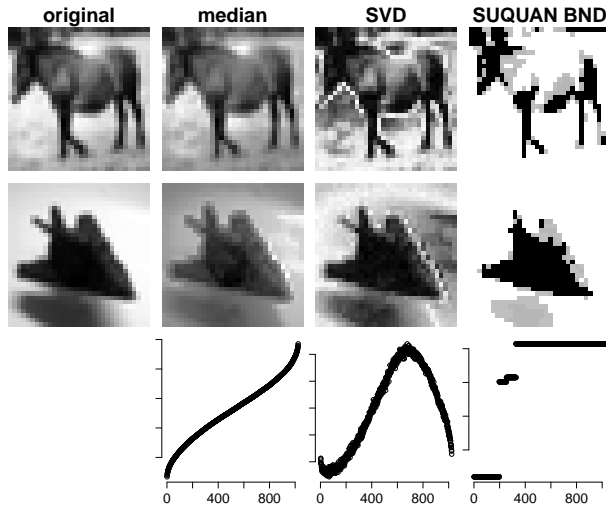
Experiments: CIFAR-10

- 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 distributions



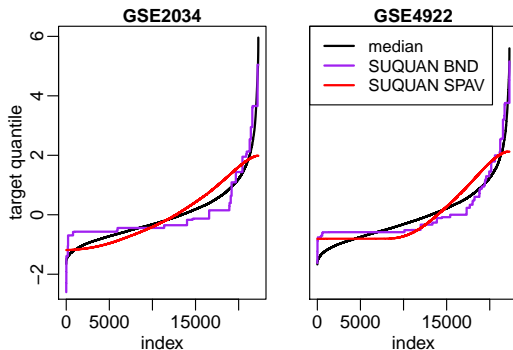
Experiments: gene expression data

- Breast cancer prognosis from gene expression data.
 - X = expression levels of 22,283 genes of the tumour at diagnosis
 - $Y = 1$ if cancer relapse within 6 years of diagnosis, 0 otherwise
- 4 datasets:

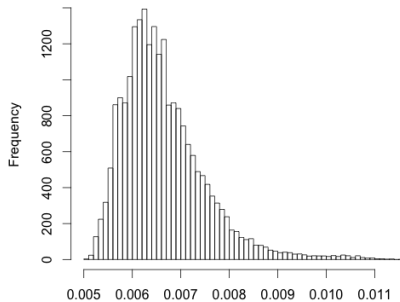
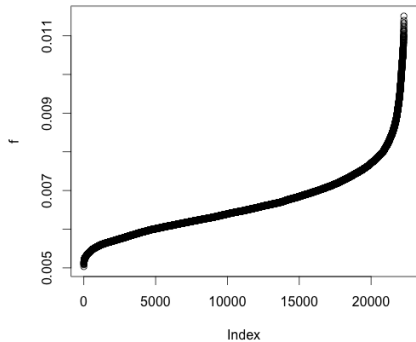
DATASET NAME	# PATIENTS	# POSITIVES	% POSITIVES
GSE1456	141	37	0.26
GSE2034	271	104	0.38
GSE2990	106	32	0.30
GSE4922	225	73	0.32

Results: gene expression data

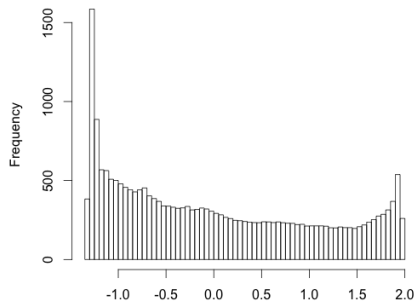
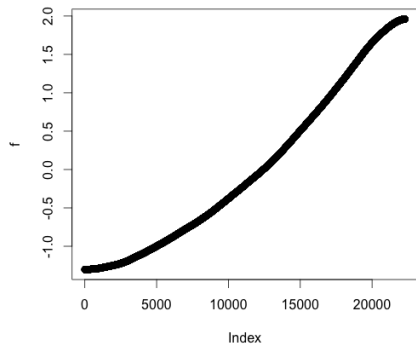
	LOGISTIC REGRESSION							SUQUAN		
	RAW	RMA	CAUCHY	EXP.	UNIF.	GAUS.	MEDIAN	SVD	BND	SPAV
GSE1456	65.94	68.73	59.56	68.86	68.72	69.00	69.06	57.60	71.44	69.60
GSE2034	74.52	75.42	61.91	74.53	75.22	76.45	74.92	52.61	70.50	76.11
GSE2990	57.01	60.43	54.72	61.25	56.25	58.66	59.72	52.51	59.22	59.94
GSE4922	58.52	58.86	55.24	58.81	55.66	60.01	59.18	52.39	61.82	61.41
AVERAGE	64.00	65.86	57.86	65.86	63.96	66.03	65.72	53.78	65.75	66.77



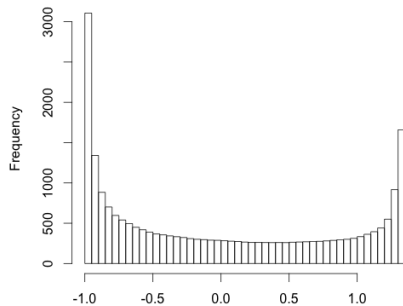
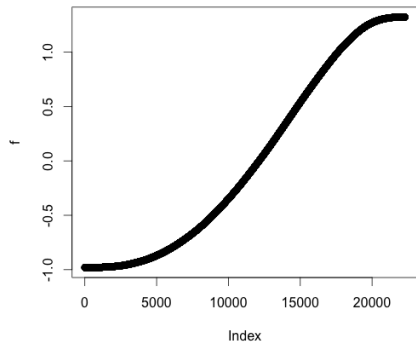
Estimated distribution: iteration=0



Estimated distribution: iteration=1



Estimated distribution: iteration=2



Outline

- 1 Supervised quantile normalization
- 2 The Kendall and Mallows kernels**
- 3 Conclusion



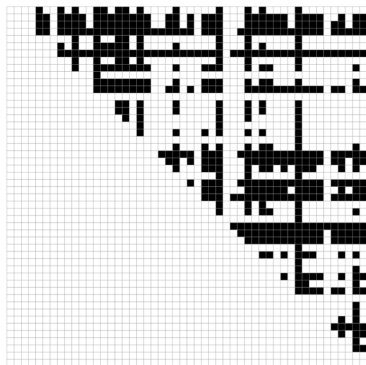
Yunlong Jiao

<https://hal.archives-ouvertes.fr/hal-01279273>

An idea: all pairwise comparisons

Replace $x \in \mathbb{R}^p$ by $\Phi(x) \in \{0, 1\}^{p(p-1)/2}$:

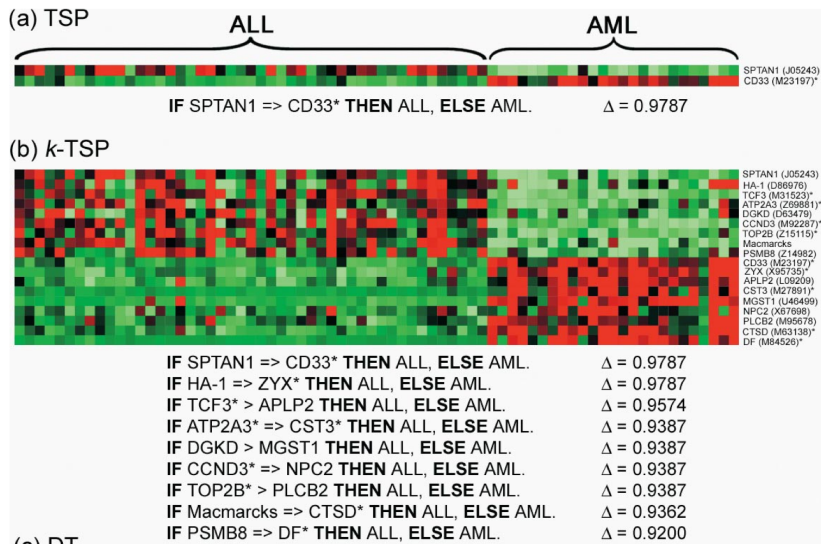
$$\Phi_{i,j}(x) = \begin{cases} 1 & \text{if } x_i \leq x_j, \\ 0 & \text{otherwise.} \end{cases}$$



**One sample x
 p features**

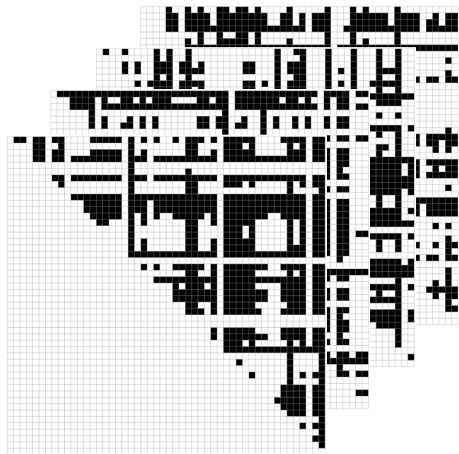
**Mapping $f(x)$
 $p(p-1)/2$ bits**

Related work: Top scoring pairs (TSP)



(Geman et al., 2004; Tan et al., 2005; Leek, 2009)

Practical challenge



- Need to store $O(p^2)$ bits per sample
- Need to train a model in $O(p^2)$ dimensions

Theorem (Wahba, Schölkopf, ...)

Training a linear model over a representation $\Phi(x) \in \mathbb{R}^Q$ of the form:

$$\min_{w \in \mathbb{R}^Q} \frac{1}{n} \sum_{i=1}^n \ell(w^\top \Phi(x_i), y_i) + \lambda \|w\|^2$$

can be done efficiently, independently of Q , if the kernel

$$K(x, x') = \Phi(x)^\top \Phi(x')$$

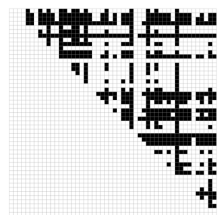
can be computed efficiently.

Ex: ridge regression, $O(Q^3 + nQ^2)$ becomes $O(n^3 + n^2 T)$

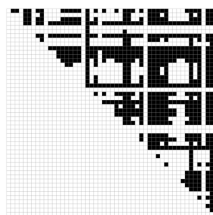
Other: SVM, logistic regression, Cox model, survival SVM, ...

Kernel trick for us: Kendall's τ

$$\Phi(x)^\top \Phi(x') = \tau(x, x') \quad (\text{up to a scaling})$$



\times



$$= \tau \left(\begin{array}{c} \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \end{array} , \begin{array}{c} \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \\ \text{red} \\ \text{green} \end{array} \right)$$

$O(p^2)$

$O(p \log(p))$

Good news for SVM and kernel methods!

More formally

- For two permutations σ, σ' let $n_c(\sigma, \sigma')$ (resp. $n_d(\sigma, \sigma')$) the number of **concordant** (resp. **discordant**) pairs.
- The **Kendall kernel** (a.k.a. **Kendall tau coefficient**) is defined as

$$K_\tau(\sigma, \sigma') = \frac{n_c(\sigma, \sigma') - n_d(\sigma, \sigma')}{\binom{p}{2}}.$$

- The **Mallows kernel** is defined for any $\lambda \geq 0$ by

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

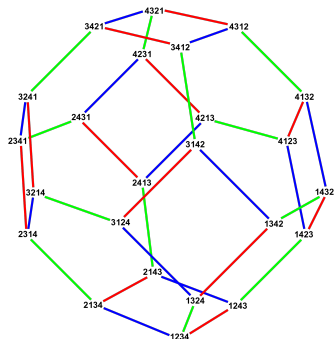
Theorem (Jiao and V., 2015)

*The Kendall and Mallows kernels are **positive definite**.*

Theorem (Knight, 1966)

These two kernels for permutations can be evaluated in $O(p \log p)$ time.

Related work



Cayley graph of S_4

- Kondor and Barbarosa (2010) proposed the **diffusion kernel** on the Cayley graph of the symmetric group generated by adjacent transpositions.
- Computationally intensive ($O(p^p)$)
- 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 $O(p \log p)$

Application: supervised classification

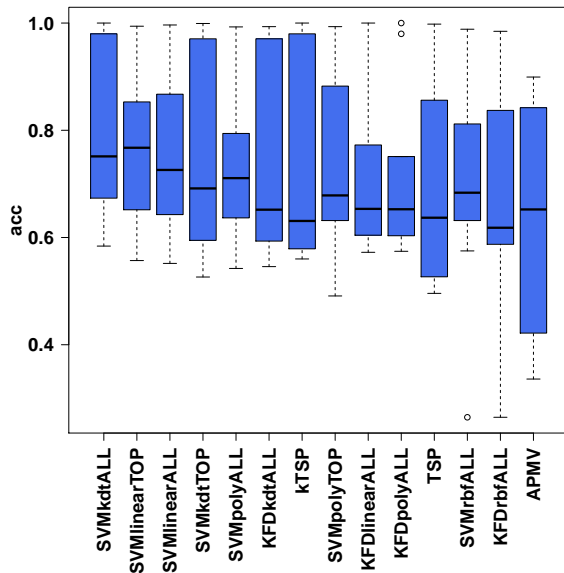
Datasets

Dataset	No. of features	No. of samples (training/test)	
		C_1	C_2
Breast Cancer 1	23624	44/7 (Non-relapse)	32/12 (Relapse)
Breast Cancer 2	22283	142 (Non-relapse)	56 (Relapse)
Breast Cancer 3	22283	71 (Poor Prognosis)	138 (Good Prognosis)
Colon Tumor	2000	40 (Tumor)	22 (Normal)
Lung Cancer 1	7129	24 (Poor Prognosis)	62 (Good Prognosis)
Lung Cancer 2	12533	16/134 (ADCA)	16/15 (MPM)
Medulloblastoma	7129	39 (Failure)	21 (Survivor)
Ovarian Cancer	15154	162 (Cancer)	91 (Normal)
Prostate Cancer 1	12600	50/9 (Normal)	52/25 (Tumor)
Prostate Cancer 2	12600	13 (Non-relapse)	8 (Relapse)

Methods

- Kernel machines Support Vector Machines (SVM) and Kernel Fisher Discriminant (KFD) with Kendall kernel, linear kernel, Gaussian RBF kernel, polynomial kernel.
- Top Scoring Pairs (TSP) classifiers Tan et al. (2005).
- Hybrid scheme of SVM + TSP feature selection algorithm.

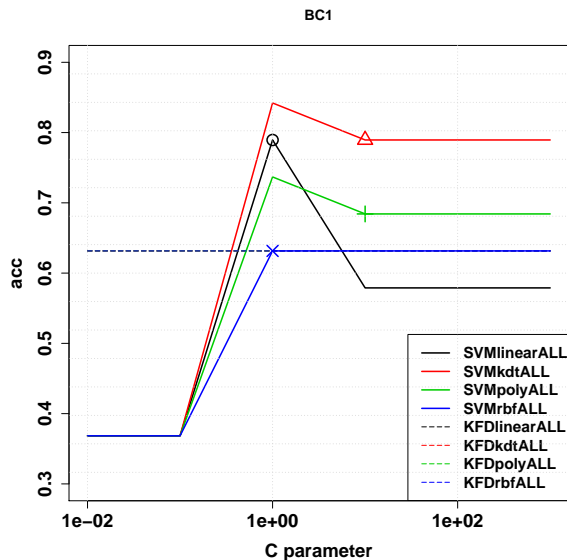
Results



Kendall kernel SVM

- **Competitive accuracy!**
- Less sensitive to regularization parameter!
- No need for feature selection!

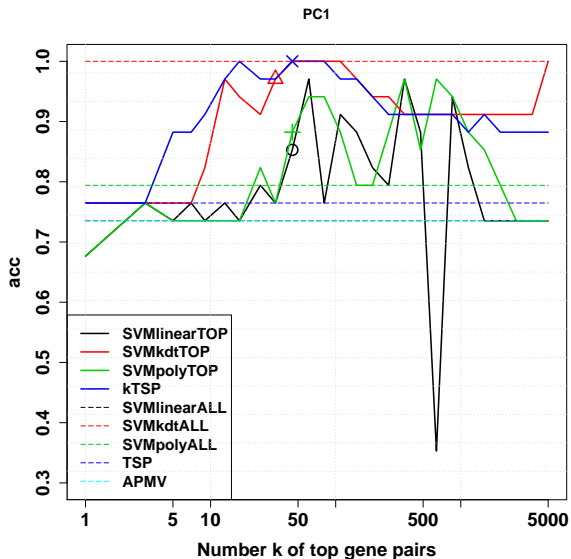
Results



Kendall kernel SVM

- Competitive accuracy!
- **Less sensitive to regularization parameter!**
- No need for feature selection!

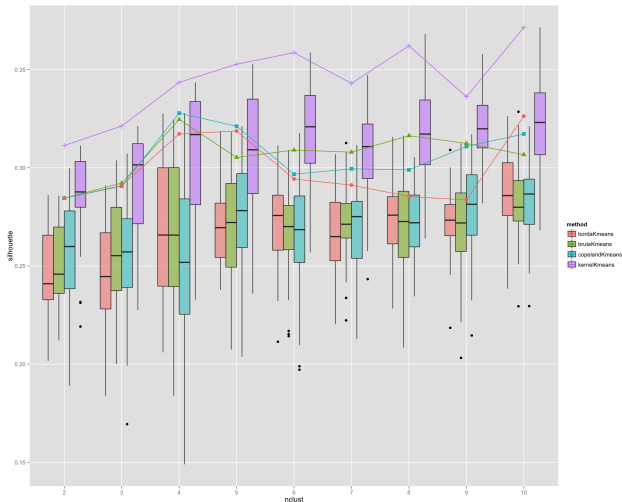
Results



Kendall kernel SVM

- Competitive accuracy!
- Less sensitive to regularization parameter!
- **No need for feature selection!**

Application: clustering



- APA data (full rankings)
- $n = 5738$, $p = 5$
- (new) Kernel k-means vs (standard) k-means in \mathbb{S}_5
- Show silhouette as a function of number of clusters (higher better)

Extension to partial rankings

- Two interesting types of partial rankings are **interleaving partial ranking**

$$x_{i_1} \succ x_{i_2} \succ \dots \succ x_{i_k}, \quad k \leq n.$$

and **top-k partial ranking**

$$x_{i_1} \succ x_{i_2} \succ \dots \succ x_{i_k} \succ X_{\text{rest}}, \quad k \leq n.$$

- Partial rankings can be **uniquely represented** by a set of permutations compatible with all the observed partial orders.

Theorem

For these two particular types of partial rankings, the convolution kernel (Haussler, 1999) induced by Kendall kernel

$$K_{\tau}^*(R, R') = \frac{1}{|R||R'|} \sum_{\sigma \in R} \sum_{\sigma' \in R'} K_{\tau}(\sigma, \sigma')$$

can be evaluated in $O(k \log k)$ time.

Extension to partial rankings

- Two interesting types of partial rankings are **interleaving partial ranking**

$$x_{i_1} \succ x_{i_2} \succ \dots \succ x_{i_k}, \quad k \leq n.$$

and **top-k partial ranking**

$$x_{i_1} \succ x_{i_2} \succ \dots \succ x_{i_k} \succ X_{\text{rest}}, \quad k \leq n.$$

- Partial rankings can be **uniquely represented** by a set of permutations compatible with all the observed partial orders.

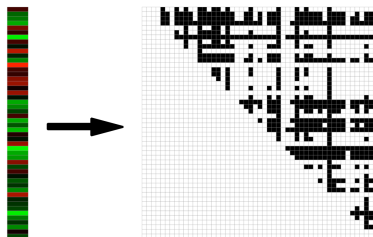
Theorem

For these two particular types of partial rankings, the convolution kernel (Haussler, 1999) induced by Kendall kernel

$$K_{\tau}^*(R, R') = \frac{1}{|R||R'|} \sum_{\sigma \in R} \sum_{\sigma' \in R'} K_{\tau}(\sigma, \sigma')$$

can be evaluated in $O(k \log k)$ time.

Extension to smoother, continuous representations



One sample x
 p features

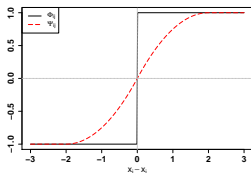
Mapping $f(x)$
 $p(p-1)/2$ bits

- Instead of $\Phi : \mathbb{R}^p \rightarrow \{0, 1\}^{p(p-1)/2}$, consider the continuous mapping $\Psi_a : \mathbb{R}^p \rightarrow \mathbb{R}^{p(p-1)/2}$:

$$\Psi_a(x) = \mathbb{E}\Phi(x + \epsilon) \quad \text{with} \quad \epsilon \sim (\mathcal{U}[-\frac{a}{2}, \frac{a}{2}])^n$$

- Corresponding kernel $G_a(x, x') = \Psi_a(x)^\top \Psi_a(x')$

Computation of $G(x, x')$



- $G_a(x, x')$ can be computed **exactly** in $O(p^2)$ by explicit computation of $\Psi_a(x)$ in $\mathbb{R}^{p(p-1)/2}$

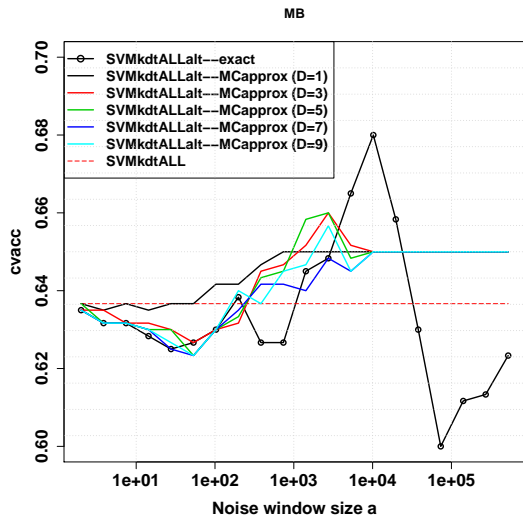
- $G_a(x, x')$ can be computed **approximately** in $O(D^2 p \log p)$ by Monte-Carlo approximation:

$$\tilde{G}_a(x, x') = \frac{1}{D^2} \sum_{i,j=1}^D K(x + \epsilon_i, x' + \epsilon'_j)$$

- Theorem: for supervised learning, Monte-Carlo approximation is better¹ than exact computation when $n = o(p^{1/3})$

¹ faster for the same accuracy

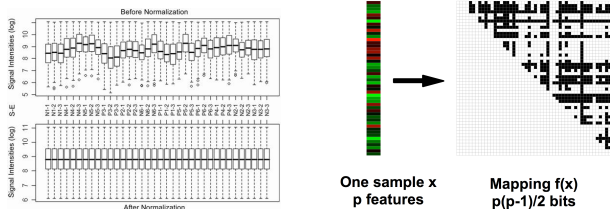
Performance of $G_a(x, x)$



Outline

- 1 Supervised quantile normalization
- 2 The Kendall and Mallows kernels
- 3 Conclusion**

Conclusion



- Representing omics data as **permutations** has some potential
 - **Kendall and Mallows** kernel in $O(p \ln(p))$
 - **SUQUAN** supervised quantile normalization as matrix regression
- Ongoing work:
 - Extension of SUQUAN to nonlinear models
 - Extension of SUQUAN to Kendall representation

Thanks



Inserm

Institut national
de la santé et de la recherche médicale



ENS
ÉCOLE NORMALE
SUPÉRIEURE



The Adolph C. and Mary Sprague
Miller Institute for Basic
Research in Science
University of California, Berkeley



SIMONS
INSTITUTE
for the Theory of Computing

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>.
- M. Le Morvan and J.-P. Vert. Supervised quantile normalisation. Technical Report 1706.00244, arXiv, 2017.
- 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>.
- A. C. Tan, D. Q. Naiman, L. Xu, R. L. Winslow, and D. Geman. Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics*, 21(20):3896–3904, Oct 2005. doi: 10.1093/bioinformatics/bti631. URL <http://dx.doi.org/10.1093/bioinformatics/bti631>.