Learning in indefinite proximity spaces: Mathematical foundations, representations and models

Frank-Michael Schleif and Peter Tino
schleify@cs.bham.ac.uk
http://www.promos-science.blogspot.de/

IJCNN - 2015

Overview

First . . . some extras

Available material

Additional web resources (code, datasets, links to papers)
can be found at
http://www.techfak.uni-bielefeld.de/~fschleif/ijcnn_2015
very recent review paper
accepted and (available online in July)
Indefinite proximity learning - A review
Schleif/Tino, Neural Computation, MIT press, 2015

Motivation

Metric or Non-metric - this is the question

- The scientific world is widely metric, the reality not . . .
- Psychological studies - Colorspace is non-metric, perception is non-metric [22, 20]
- Image processing - Good recognition is non-metric [36]
- Life sciences - many effective proximity measures are indefinite
- Machine learning - asymmetry in graphs, ML in non-metric spaces [31]

Is non-metric representation the better one?

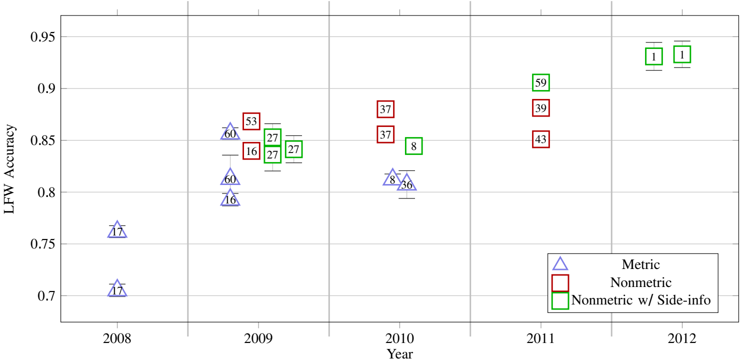


Figure : Recent study on Labeled Faces in the Wild (LFW) from [22]

. . . and where does it occur . . .

Some examples - Signal processing

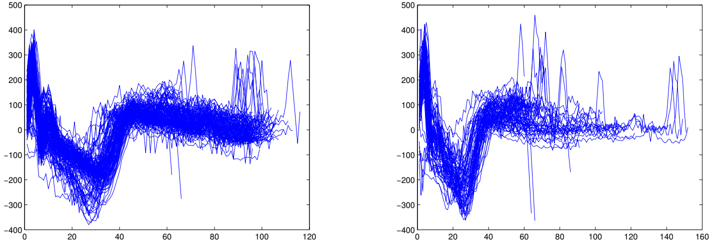


Figure : Normal and abnormal ecg data

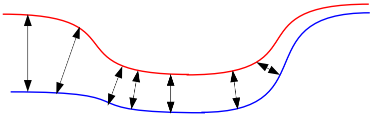


Figure : Dynamic time warping (DTW)[35]

Some examples - Audio processing

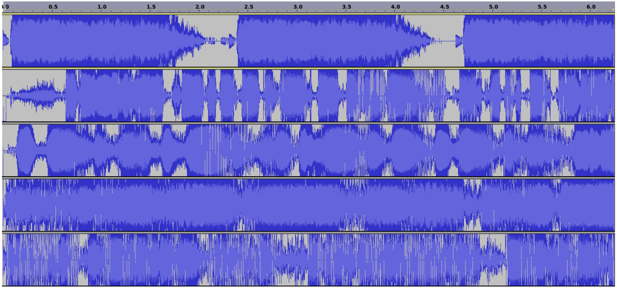


Figure : Search algorithms on audio files

Kullback-Leibler (or other) Divergence on Histogram features

Some examples - Image processing

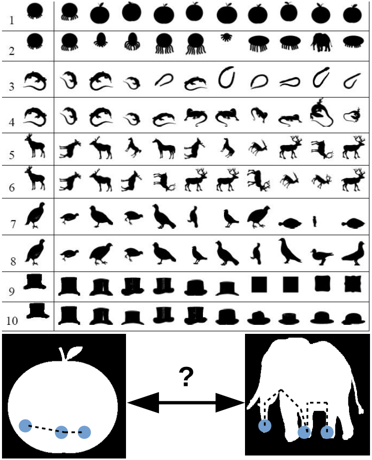


Figure : Shape retrieval using the inner distance[24]

Some examples - text processing

Ihr naht euch wieder, schwankende Gestalten, Die früh sich einst dem trüben Blick gezeigt. Versuch ich wohl, euch diesmal festzuhalten? Fühl ich mein Herz noch jenem Wahn geneigt? Ihr drängt euch zu! nun gut, so mögt ihr walten, Wie ihr aus Dunst und Nebel um mich steigt; Mein Busen fühlt sich jugendlich erschüttert Vom Zauberhauch, der euren Zug umwittert. (**from Faust I `http://www.projekt.gutenberg.de/`**)

Figure : Normalized compression distance [8]

Some examples - bioinformatics

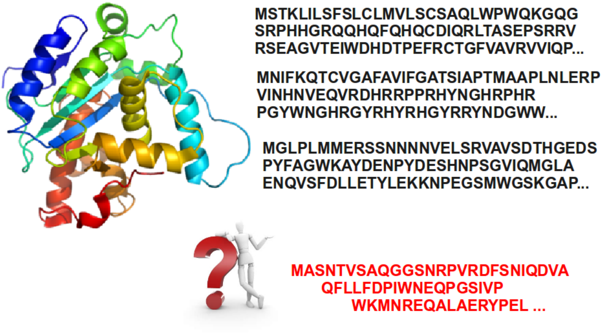


Figure : Smith-Waterman sequence alignment

Why should we care?

Challenges

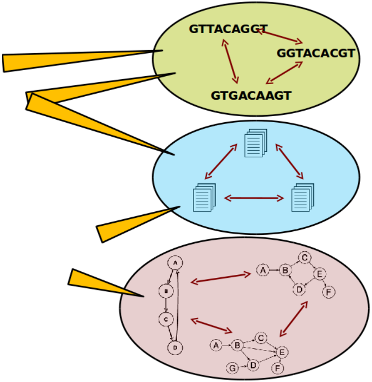
- for non-metric kernels - classical methods (e.g. SVM) fail
- often cheats are used and results do not link back to original data
- many effective optimization strategies e.g. for large scale approximation are inapplicable (psd assumtion)
- many algorithms (with psd requirement) show substantial numerical errors for non-psd data
- non-metric representations are often more natural
- enforcing metric properties can reduce efficiency [33]

A metric proximity function

- we can distinguish similarities $s(x, y)$ and dissimilarities $d(x, y)$
- (squared) dissimilarities $d(x, y) = \langle x, x\rangle + \langle y, y\rangle - 2\langle x, y\rangle$
- $\langle x, y\rangle$ is an inner product
- a metric proximity is symmetric, real, positive and obeys $\langle x, x\rangle = 0 \iff x = 0$
- it implies a norm $||x|| = \sqrt{\langle x, x\rangle}$ with the triangle inequality to hold
- a metric kernel gives raise to a reproducing kernel hilbert space
- indefinite, non-positive, non-metric, non-psd kernel (contains negative eigenvalues)

Indefinite proximity functions - are common . . .

- alignment (bioinformatics)
- cosinus measure (information retrieval)
- Hamming (information theory)
- geodesic distance (geometry)
- Jaccard index (statistics)
- compression distance
- graph structure kernels
- dynamic time warping (time-series)
- shape matching distance
- earth mover distance
- manhattan kernel
- divergence measures [7]
- tangential distance [17]
- . . .

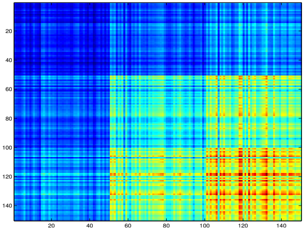


Basic formalisms

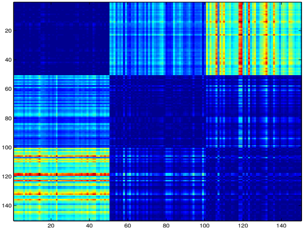
Basic formalisms

- $\mathcal{X}$ is a *collection* of N objects x_i, $i = 1, 2, ..., N$, in some input space Ω
- Ω may not be an explicit vector space
- a similarity function $\Omega \times \Omega \rightarrow \mathbf{R}$ (maybe not explicit)
- $\mathbf{Y}$ is an (optional) label space
- a proximity matrix $S = X \times X$, in general S is symmetric
- a test point x is a vector of N similarities obtained by $x \times X$

- similarity matrices (kernels) inner products
- dissimilarities distances
- conversion between (symmetric) proximities



- similarity matrices (kernels) inner products
- dissimilarities distances
- conversion between (symmetric) proximities



- similarity matrices (kernels) inner products
- dissimilarities distances
- conversion between (symmetric) proximities

double centering

$$
\begin{aligned}
S &= -\frac{1}{2}JDJ \quad J = I - 11^{\top}/N \\
D &= \sqrt{s_{ii} + s_{jj} - 2s_{ij}}
\end{aligned}
$$

- similarity matrices (kernels) inner products
- dissimilarities distances
- conversion between (symmetric) proximities

. . . proximity matrices can become huge $\mathcal{O}(N^2)$ complexity

Overview

Krein space and pseudo-Euclidean space I

- A Krein space is an *indefinite* inner product space endowed with a Hilbertian topology
- let $\mathcal{K}$ be a real vector space.
- A vector space $\mathcal{K}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ is called an inner product space.
- an inner product space with an *indefinite* inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ on $\mathcal{K}$ is a bi-linear form where all $f, g, h \in \mathcal{K}$ and $\alpha \in \mathbb{R}$ obey the following conditions.
  - Symmetry: $\langle f, g \rangle_{\mathcal{K}} = \langle g, f \rangle_{\mathcal{K}}$
  - linearity: $\langle \alpha f + g, h \rangle_{\mathcal{K}} = \alpha \langle f, h \rangle_{\mathcal{K}} + \langle g, h \rangle_{\mathcal{K}}$;
  - $\langle f, g \rangle_{\mathcal{K}} = 0$ implies $f = 0$.

Krein space and pseudo-Euclidean space II

- An inner product is positive definite if $\forall f \in \mathcal{K}, \langle f, f\rangle_{\mathcal{K}} \geq 0$, negative definite if $\forall f \in \mathcal{K}, \langle f, f\rangle_{\mathcal{K}} \leq 0$, otherwise it is indefinite.
- An inner product space $(\mathcal{K}, \langle\cdot,\cdot\rangle_{\mathcal{K}})$ is a Krein space if we have two Hilbert spaces $\mathcal{H}_+$ and $\mathcal{H}_-$ spanning $\mathcal{K}$ such that $\forall f \in \mathcal{K}$ we have $f = f_+ + f_-$ with $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$ and $\forall f, g \in \mathcal{K}$, $\langle f, g\rangle_{\mathcal{K}} = \langle f_+, g_+\rangle_{\mathcal{H}_+} - \langle f_-, g_-\rangle_{\mathcal{H}_-}$.
- A finite-dimensional Krein-space is a so called pseudo Euclidean space

Krein space and pseudo-Euclidean space III

- we can have negative squared ”norm”, negative squared ”distances” and the concept of orthogonality is different
- given a symmetric *dissimilarity* matrix with zero diagonal, an embedding of the data in a pseudo-Euclidean vector space determined by the eigenvector decomposition of the associated similarity matrix **S** is always possible [12]
- so in principle we can have an embedding (maybe into high dimensions) but it is very costly

Krein space and pseudo-Euclidean space IV

- Given the eigendecomposition of $\mathbf{S}$, $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, we can compute the corresponding vectorial representation $\mathbf{V}$ in the pseudo-Euclidean space by

$$\mathbf{V} = \mathbf{U}_{p+q+z} \left|\mathbf{\Lambda}_{p+q+z}\right|^{1/2} \tag{1}$$

  where $\mathbf{\Lambda}_{p+q+z}$ consists of p positive, q negative non-zero eigenvalues and z zero eigenvalues. $\mathbf{U}_{p+q+z}$ consists of the corresponding eigenvectors.
- The triplet (p, q, z) is also referred to as the signature of the Pseudo-Euclidean space.
- details provided in [31, 9, 30].

Sources of indefiniteness

- Distance-based kernels: non-Hilbertian, non-metric
- Prior knowledge in kernel construction
- Invariant kernels (e.g. tangential kernel)
- Robust or approximate (dis)similarities
- Kernel combination (not all combinations lead to psd kernels)
- Noise

Take home message

- for indefinite spaces we speak about a Krein space
- a discrete Krein space is a Pseudo Euclidean space
- a Pseudo-Euclidean space basically consists of a positive *and* a negative Euclidean space
- for real problems we observe the Pseudo-Euclidean space as a *generalization* of the Euclidean space
- the positive Euclidean space is what we all know
- the negative Euclidean space can have many sources (noise, extended objects, . . .)

Overview

Approaches for processing indefinite proximities

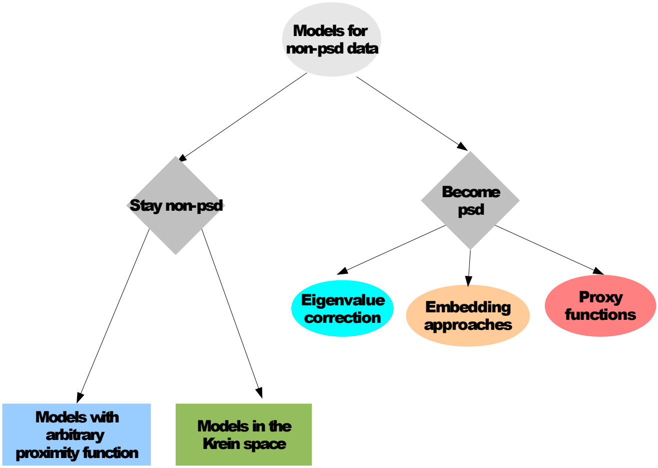
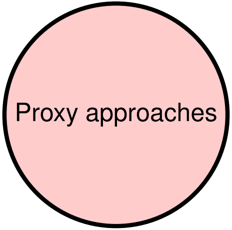


Figure : Schematic view of different approaches to analyze non-psd data


Proxy approaches

Back to metric - optimizing an alternative *metric* matrix

Indefinite proximity due to noise

- Optimization problem $\max_\alpha E(\alpha)$ s.t. $C(\alpha)$
- for SVM: $\max_\alpha \alpha^\top e - \frac{1}{2}\alpha^\top YK_0Y\alpha$ s.t. $\alpha^\top y = 0, 0 \leq \alpha \leq C$
- try to learn a psd proxy kernel K which is close to K_0
- Optimization problem $\max_\alpha \min_K E(\alpha) + \rho||K - K_0||_F$ s.t. $C(\alpha), K \succeq 0$
- for SVM: $\max_\alpha \min_K \alpha^\top e - \frac{1}{2}\alpha^\top YKY\alpha + \rho||K - K_0||_F$ s.t. $\alpha^\top y = 0, 0 \leq \alpha \leq C, K \succeq 0$

Work in this line e.g. [6, 26, 13]

Exemplary code

Some (matlab / c code) examples for proxy approaches

Most code can be found here
`http://www.techfak.uni-bielefeld.de/~fschleif/ijcnn_2015/`

In parts you will need to download extra optimizers like MOSEK https://www.mosek.com/ (Mosek provides renewable licenses - free of charge - for academic use - just contact them)

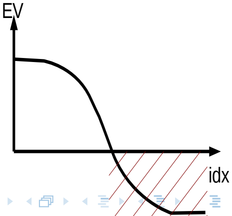
Sometimes you may need an older matlab to get the code running (without to much effort)

Eigenspectrum approaches

Back to metric - via Eigenvalue correction

$$\mathbf{S} = U \Lambda U^\top, \quad \text{U - eigenvectors, } \Lambda \text{ - eigenvalues}$$

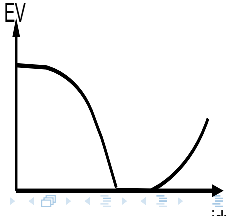
- **Clip**: negative eigenvalues in Λ are set to 0 - nearest psd matrix **S** in terms of the Frobenius norm [19].
- **Flip**: all negative eigenvalues in Λ are set to $\Lambda_i := |\Lambda_i| \; \forall i$ keeps the absolute values of the negative eigenvalues - information preserved [33].
- **Shift**: [23, 10] $\Lambda := \Lambda - \min_{ij} \Lambda$ Spectrum shift enhances all the self-similarities by ν and does not change the similarity between any two different data points.
- **Square**: Λ is changed to $\Lambda := \Lambda^2$ (elementwise)
- others (mixed schemes) see e.g. [28]



Back to metric - via Eigenvalue correction

$$\mathbf{S} = U \Lambda U^\top, \quad \text{U - eigenvectors, } \Lambda \text{ - eigenvalues}$$

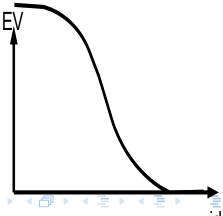
- **Clip**: negative eigenvalues in Λ are set to 0 - nearest psd matrix **S** in terms of the Frobenius norm [19].
- **Flip**: all negative eigenvalues in Λ are set to $\Lambda_i := |\Lambda_i| \; \forall i$ keeps the absolute values of the negative eigenvalues - information preserved [33].
- **Shift**: [23, 10] $\Lambda := \Lambda - \min_{ij} \Lambda$ Spectrum shift enhances all the self-similarities by ν and does not change the similarity between any two different data points.
- **Square**: Λ is changed to $\Lambda := \Lambda^2$ (elementwise)
- others (mixed schemes) see e.g. [28]



Back to metric - via Eigenvalue correction

$$\mathbf{S} = U\Lambda U^{\top}, \quad \text{U - eigenvectors, } \Lambda \text{ - eigenvalues}$$

- **Clip**: negative eigenvalues in Λ are set to 0 - nearest psd matrix **S** in terms of the Frobenius norm [19].
- **Flip**: all negative eigenvalues in Λ are set to $\Lambda_i := |\Lambda_i| \; \forall i$ keeps the absolute values of the negative eigenvalues - information preserved [33].
- **Shift**: [23, 10] $\Lambda := \Lambda - \min_{ij} \Lambda$ Spectrum shift enhances all the self-similarities by ν and does not change the similarity between any two different data points.
- **Square**: Λ is changed to $\Lambda := \Lambda^2$ (elementwise)
- others (mixed schemes) see e.g. [28]



Back to metric - via Eigenvalue correction

$$\mathbf{S} = U \Lambda U^\top, \quad \text{U - eigenvectors, } \Lambda \text{ - eigenvalues}$$

- **Clip**: negative eigenvalues in Λ are set to 0 - nearest psd matrix **S** in terms of the Frobenius norm [19].
- **Flip**: all negative eigenvalues in Λ are set to $\Lambda_i := |\Lambda_i| \; \forall i$ keeps the absolute values of the negative eigenvalues - information preserved [33].
- **Shift**: [23, 10] $\Lambda := \Lambda - \min_{ij} \Lambda$ Spectrum shift enhances all the self-similarities by ν and does not change the similarity between any two different data points.
- **Square**: Λ is changed to $\Lambda := \Lambda^2$ (elementwise)
- others (mixed schemes) see e.g. [28]

Back to metric - via Eigenvalue correction

$$\mathbf{S} = U \Lambda U^\top, \quad \text{U - eigenvectors, } \Lambda \text{ - eigenvalues}$$

- **Clip**: negative eigenvalues in Λ are set to 0 - nearest psd matrix **S** in terms of the Frobenius norm [19].
- **Flip**: all negative eigenvalues in Λ are set to $\Lambda_i := |\Lambda_i| \; \forall i$ keeps the absolute values of the negative eigenvalues - information preserved [33].
- **Shift**: [23, 10] $\Lambda := \Lambda - \min_{ij} \Lambda$ Spectrum shift enhances all the self-similarities by ν and does not change the similarity between any two different data points.
- **Square**: Λ is changed to $\Lambda := \Lambda^2$ (elementwise)
- others (mixed schemes) see e.g. [28]

Back to metric - via Eigenvalue correction

$$\mathbf{S} = U \Lambda U^\top, \quad \text{U - eigenvectors, } \Lambda \text{ - eigenvalues}$$

- **Clip**: negative eigenvalues in Λ are set to 0 - nearest psd matrix **S** in terms of the Frobenius norm [19].
- **Flip**: all negative eigenvalues in Λ are set to $\Lambda_i := |\Lambda_i| \ \forall i$ keeps the absolute values of the negative eigenvalues - information preserved [33].
- **Shift**: [23, 10] $\Lambda := \Lambda - \min_{ij} \Lambda$ Spectrum shift enhances all the self-similarities by ν and does not change the similarity between any two different data points.
- **Square**: Λ is changed to $\Lambda := \Lambda^2$ (elementwise)
- others (mixed schemes) see e.g. [28]

If input is a dissimilarity matrix, double centering [31] is needed first

Exemplary code

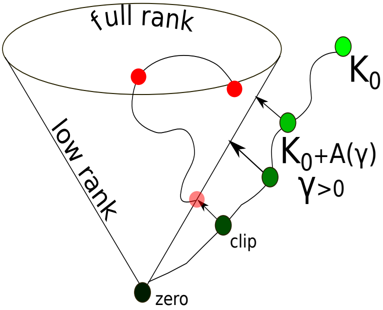
Some (matlab / c code) examples for eigenvalue correction approaches

An eigenvalue correction is fairly simple - but, can be costly for large scale or if you start with a dissimilarity matrix.

At
`http://www.techfak.uni-bielefeld.de/~fschleif/ijcnn_2015/`
you can find an archive with some extra code also for eigenvalue corrections with low rank matrices.

Take home message

- Eigenvalue correction is a simple way to make the data psd
- Clip is perfect if the indefiniteness is due to noise
- Flip / Square appear to be good if indefiniteness is meaningful
- Eigenvalue corrections are costly (with exceptions - see later)



$\boxed{\text{Native methods in the Krein space}}$

Machine learning in another world ?

- Some learning algorithms (e.g. Fisher Discriminant) remain valid [32]
- Support Vector Machine with SMO reaches a local optimum [40]
- Core Vector Machine will in general not converge (due to strong geometric assumptions)
- Alternatives: empirical feature / similarity / dissimilarity space representation

Indefinite Kernel Methods

- Nearest Mean Classifier [31]
- Regression [30]
- Indefinite Support Vector Machine [15]
- **Indefinite Fisher Discriminant** [16]
- Indefinite Kernel Quadratic Discriminant [32]
- Kernel Mahalanobis Distances [16, 18]
- Indefinite Slow Feature Analysis [25]
- Non-metric Locality Sensitive Hashing [27]
- Relevance Vector Machine [41]
- Probabilistic Classification Vector Machine [5]

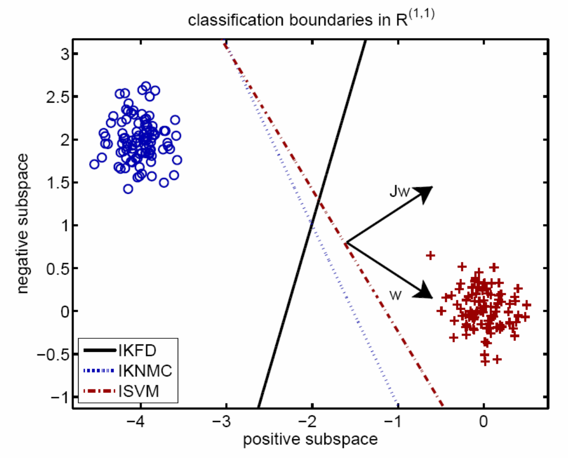
Indefinite Fisher Discriminant (Pseudo Euclidean Fisher Discriminant)

- class means $\mu_\pm := \frac{1}{n_\pm} \sum_{i \in I_\pm} \phi(x_i)$
- Between-class scatter projection: $\sum_{pE}^{B} w = (\mu_+ - \mu_-)\langle \mu_+ - \mu_-, w \rangle_{pE}$
- Within-class scatter projection: $\sum_{pE}^{W} w = \sum_{pE+}^{W} w + \sum_{pE-}^{W} w$
- $\sum_{pE,\pm}^{W} w = \sum_{i \in I_\pm} (\phi(x_i) - \mu_\pm)\langle \phi(x_i) - \mu_\pm, w \rangle_{pE}$
- Maximize Fisher Criterion:

$$J(w) = \frac{\langle w, \sum_{pE}^{B} w \rangle_{pE}}{\langle w, \sum_{pE}^{W} w \rangle_{pE}}$$

- Fisher Discriminant (decision function)
  $f(z) = \langle w, z \rangle_{pE} + b \qquad b = \frac{-1}{2}\langle \mu_+ + \mu_-, w \rangle_{pE}$

Geometric interpretation of the iKFD



Indefinite *kernel* fisher discriminant

Kernelization

- Normal: $w = \sum_{i=1}^{n} \alpha_i \phi(x_i)$
- Between scatter: $\langle w, \sum_{pE}^{B} w \rangle_{pE} = \alpha^\top K(c_+ - c_-)(c_+ - c_-)^\top K\alpha$
- Within-scatter: $\langle w, \sum_{pE}^{W} w \rangle_{pE} = \alpha^\top (K_+ H_+ K_+^\top + K_- H_- K_-^\top)\alpha$
- $M = KCK$ and $N = K_+ H_+ K_+ + K_- H_- K_-$ and C a coefficient matrix and H a centering matrix (see [32])
- Maximization of regularized fisher criterion

$$J(\alpha) = \frac{\alpha^\top M\alpha}{\alpha^\top N\alpha} \qquad \alpha = N^{-1}K(c_+ - c_-)$$

- Indefinite KFD
  $f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b \qquad b = \frac{-1}{2}\alpha^\top(\frac{1}{n_+}K_+ 1_{n_+} + \frac{1}{n_-}K_- 1_{n_-})$

Correspondence to KFD with indefinite kernel

Indefinite Kernel Methods

- Nearest Mean Classifier [31]
- Regression [30]
- Indefinite Support Vector Machine [15]
- Indefinite Fisher Discriminant [16]
- Indefinite Kernel Quadratic Discriminant [32]
- Kernel Mahalanobis Distances [16, 18]
- Indefinite Slow Feature Analysis [25]
- Non-metric Locality Sensitive Hashing [27]
- Relevance Vector Machine [41]
- **Probabilistic Classification Vector Machine** [5]

Probabilistic Classification VM I

- Similar to Relevance Vector Machine by M. Tipping (JMLR'01)
- Decision function looks like:

$$f(\mathbf{x}) = \Psi(\Phi_\theta(\mathbf{x})\mathbf{w} + b)$$

- **sparse probabilistic** kernel classifier
- unused basis functions in Φ_θ are pruned during training

Probabilistic Classification VM I

- Similar to Relevance Vector Machine by M. Tipping (JMLR'01)
- Decision function looks like:

$$f(\mathbf{x}) = \Psi(\Phi_\theta(\mathbf{x})\mathbf{w} + b)$$

- **sparse probabilistic** kernel classifier
- unused basis functions in Φ_θ are pruned during training

Probabilistic Classification VM I

- Similar to Relevance Vector Machine by M. Tipping (JMLR'01)
- Decision function looks like:

$$f(\mathbf{x}) = \Psi(\Phi_\theta(\mathbf{x})\mathbf{w} + b)$$

- **sparse probabilistic** kernel classifier
- unused basis functions in Φ_θ are pruned during training

Probabilistic Classification VM I

- Similar to Relevance Vector Machine by M. Tipping (JMLR'01)
- Decision function looks like:

$$f(\mathbf{x}) = \Psi(\Phi_\theta(\mathbf{x})\mathbf{w} + b)$$

- **sparse probabilistic** kernel classifier
- unused basis functions in Φ_θ are pruned during training

Probabilistic Classification VM II

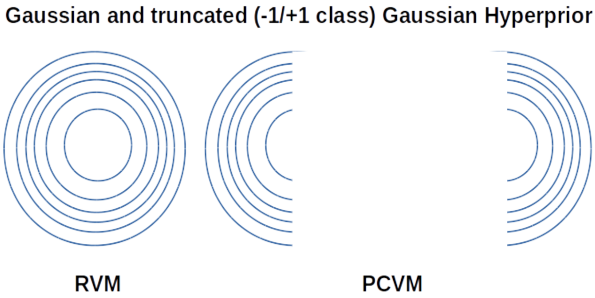
$$f(\mathbf{x}) = \Psi(\Phi_\theta(\mathbf{x})\mathbf{w} + b)$$

- $\Phi_\theta(\mathbf{x})$ is a vector of basis function evaluations for the data point x (e.g. the similarities of $\mathbf{x}$ w.r.t. all other points)
- $\Psi(z) = \int_{-\infty}^{z} \mathcal{N}(t|0,1)dt$ is the probit link function
- parameters are:
  - $\mathbf{w}$ - weights with hyper parameters α_i
  - b - bias with hyper parameter β
- learning by modified Expectation Maximization (EM)
- but classical RVM has various issues due to an inappropriate model for the hyperparameter priors

Probabilistic Classification VM III

Instead: PCVM hyperparameter priors are **truncated** Gaussian priors:

- negative weight for class 1
- positive for class 2



... but both approaches have $\mathcal{O}(N^3)$ complexity at the beginning.

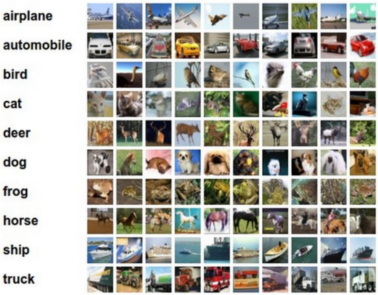
Indefinite Kernel Methods

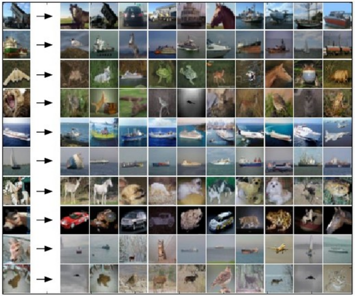
- Nearest Mean Classifier [31]
- Regression [30]
- Indefinite Support Vector Machine [15]
- Indefinite Fisher Discriminant [16]
- Indefinite Kernel Quadratic Discriminant [32]
- Kernel Mahalanobis Distances [16, 18]
- Indefinite Slow Feature Analysis [25]
- **Non-metric Locality Sensitive Hashing** [27]
- Relevance Vector Machine [41]
- Probabilistic Classification Vector Machine [5]

Non-metric Locality Sensitive Hashing

- Hashing is used to organize large datasets by small codes
- Locality sensitive hashing (LSH) by Indyk provides hash functions such that objects in close proximity share similar hash codes
- A hash function family $\mathcal{H}$ is called locality sensitive if $P_{\mathcal{H}}[h(p) = h(q)] = Sim(p, q)$ with $h \in \mathcal{H}$ (originally with $Sim(p, q)$ - metric)
- We can obtain $K = K_+ - K_-$ by an SVD on Sim (K)
- Now in [27] LSH hash functions are constructed for h_+ and h_- - the two Euclidean spaces in the Krein space
- This can be done on a few training points and using the kernel trick (details in [27])

Non-metric LSH - image retrieval





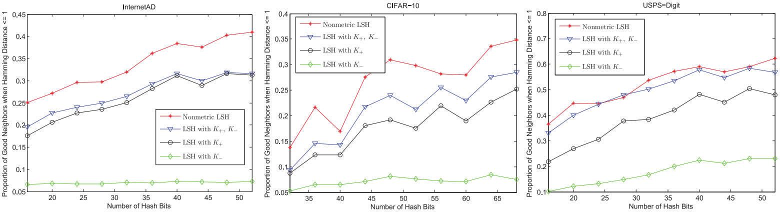


Figure : In general the retrieval accuracy (proportion of good neighbors) is better with non-metric LSH than using K_+ or K_- alone.

Take home message

- tailored methods to indefinite problems beneficial
- available e.g. for classification, regression, variance analysis (PCA), retrieval (hashing)
- classical implementations are costly (typically $\mathcal{O}(N^3)$)
- Efficient implementations possible if input matrix has low rank (next slides)
- A more comprehensive overview is available in our Neural Computation paper *Indefinite proximity learning - A review*, Schleif/Tino, Neural Computation, MIT press, 2015

Overview

Computational effort

n	size
5000	190 MB
10.000	763 MB
20.000	3.0 GB
50.000	18.6 GB
200.000	300.0 GB

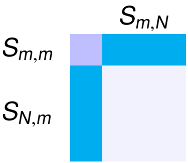
Table : **Size of a matrix (double precision)**

Computational effort

Dissimilarity calculation to a parameter vector w_j based on similarities S

$$
\begin{aligned}
\|x_i - w_j\|^2 &= S_{i,i} - 2\sum_{l} \alpha_{j,l} S_{i,l} + \sum_{l,l'} \alpha_{j,l}\alpha_{j,l'} S_{l,l'} \\
&= e_i^\top S e_i - 2 e_i S \alpha_j + \alpha_j^\top S \alpha_j
\end{aligned}
$$

Nyström approximation (low rank approach) [43]
Sample m landmarks only: approximate
$S \approx S_{N,m} S_{m,m}^{-1} S_{m,N}$



This approximation can be done for dissimilarities and similarities psd or non-psd [39].

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of $\mathbf{x}$)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1}) [K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\mathrm{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)}((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector $\mathbf{x}$)

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of $\mathbf{x}$)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1}) [K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\mathrm{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)} ((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector $\mathbf{x}$)

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of $\mathbf{x}$)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1})[K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\operatorname{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)}((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector $\mathbf{x}$)

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of $\mathbf{x}$)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1}) [K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\mathrm{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)} ((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector $\mathbf{x}$)

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of $\mathbf{x}$)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1}) [K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\text{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)}((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector $\mathbf{x}$)

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of $\mathbf{x}$)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1})[K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\mathrm{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)}((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector $\mathbf{x}$)

Practical benefits of the Nyström approximation I

K is a (symmetric) proximity matrix (similarities or dissimilarities)

- $\hat{K} = K_{N,(q)} K_q^{-1} K_{(q),N}$ (Kernel reconstruction)
- $[\hat{K}]_{i,j} = [K_{N,(q)}]_{i,\cdot} K_q^{-1} [K_{(q),N}]_{\cdot,j}$ (single value evaluation)
- $\hat{\mathbf{x}} = K_{N,(q)} K_q^{-1} \mathbf{x}$ (Extension of **x**)
- $[\hat{K}]_{1,\cdot} = K_{N,(q)} K_q^{-1} [K_{(q),N}]_{\cdot,1}$ (Kernel evaluation idx 1 vs all)
- $\sum_i [\hat{K}]_{k,i} = (\sum K_{N,(q)} K_q^{-1})[K_{(q),N}]_{\cdot,k}$ (k-th Row/Column sum of K)
- $\operatorname{diag}(K) = \sum (K_q^{-1} K'_{N,(q)}) \odot K'_{N,(q)}$ (Diagonal elements of K)
- $K_{N,(q)}((K_{N,(q)}^\top \mathbf{x})^\top K_q^{-1})^\top$ (Matrix times vector **x**)

with linear costs (and accurate given the matrix is low rank)
→ replace full matrix operations in the corresponding algorithms

Practical benefits of the Nyström approximation II

Pseudo-Inverse (PINV), Singular Value Decomposition (SVD), Eigenvalue Decomposition (EVD)

- to calculate the pseudo-inverse we need a singular value decomposition
- for the SVD we need the eigenvectors of $\tilde{K}^\top \tilde{K}$ and $\tilde{K}\tilde{K}^\top$
- due to symmetry we approximate $\zeta = \tilde{K}^\top \tilde{K}$ by Nyström
- now we only need to calculate eigenvectors / eigenvalues of ζ
- this can be done (exact) in linear time also for indefinite kernels

Details in [11, 37, 38]

Runtime analysis employing the Nyström approximatiom

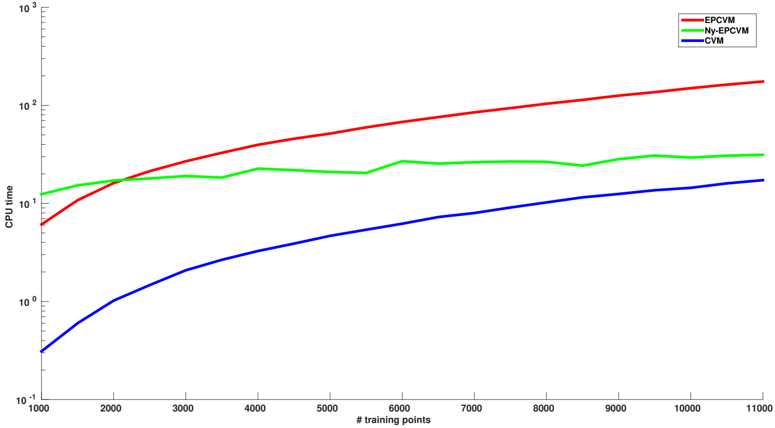


Figure : CPU time at logarithmic scale for a larger dataset for EPCVM, CVM and Ny-EPCVM. For Ny-EPCVM

Further approximation concepts

Locality and nearness

- often local metric preservation sufficient
- $\rightarrow$ enforced by local correction approach [3, 4]
- Barnes-Hut can be heuristically applied (e.g. almost metric) [1]
- sparsity and feature selection strategies can be used for proximity / empirical feature space representation

Take home message

- if low rank, proximity matrices can be approximated with linear costs
- proximity matrices can be effectively converted between each other see [11][1]
- various calculations (EVD,SVD,PINV) can be based on the approximation
- locality / nearness concepts can help as well
- but still a lot of work todo for non-heuristic approaches

[1] *Metric and non-metric proximity transformations at linear costs*, Gisbrecht / Schleif, Neurocomputing, currently open access online.

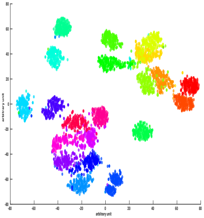
Overview

Life science data sets

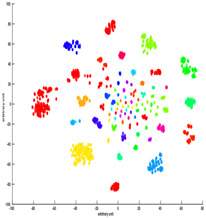
Dataset description

- *Copenhagen Chromosomes* 4,200 human chromosomes from 21 classes, given as grey-valued images and encoded as strings measuring the thickness of their silhouettes. Compared using the edit distance [29]. Signature of $(2258, 1899, 43)$.
- *ProDom* dataset with signature $(1502, 680, 422)$ consists of 2604 protein sequences with 53 labels [34]. The pairwise structural alignments are computed by [34]. Each sequence belongs to a group labeled by experts
- the Protein data set has sequence-alignment similarities for 213 proteins from 4 classes [21]. The signature is $(170, 40, 3)$.
- the *SwissProt* data set with a signature $(8487, 2500, 1)$, consists of 5,791 points of protein sequences in 10 classes as a subset from the SwissProt database [2]. (release 37, 10 most frequent classes) compared using Smith-Waterman[14].

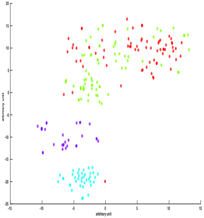
Embeddings of the similarity matrices

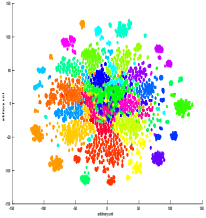


(a) Chromosom

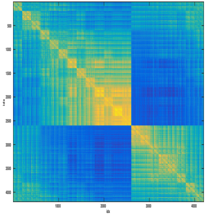


(b) Prodom

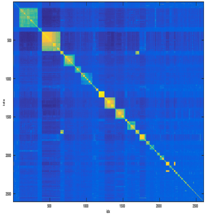




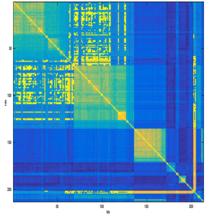
Visualization of the proxy kernel matrices

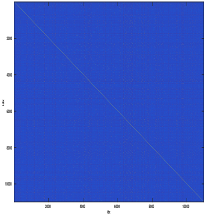


(e) Chromosom

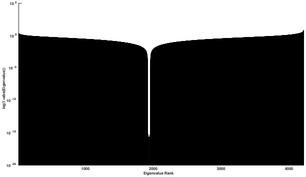


(f) Prodom

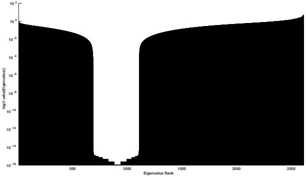




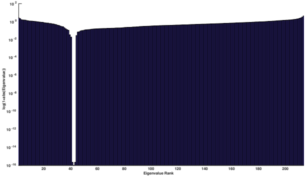
Eigenspectra of the proxy kernels



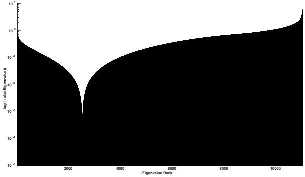
(i) Chromosom



(j) Prodom



(k) Protein



(l) Swissprot

Figure : Eigenspectra of the proxy kernel matrices of Aural sonar, Chromosom, Delft and Prodom.

A classification task - I

Table : Comparison of different priorly discussed methods for various non-psd data sets.

Method	PCVM	IKFD	kNN	SVM
Chromosoms	85.48 ± 3.65	97.36 ± 1.09	95.11 ± 0.88	97.10 ± 1.00
Prodom	99.62 ± 0.60	99.46 ± 0.55	99.87 ± 0.21	not converged
Protein	95.76 ± 4.17	**99.05** ± **2.01**	59.13 ± 12.44	61.50 ± 10.64
SwissProt	97.78 ± 0.48	96.81 ± 0.79	**98.59** ± **0.35**	97.38 ± 0.36

A classification task - II

Table : Comparison of different priorly discussed methods for various non-psd data sets.

Method	SVM-Flip	SVM-Clip	SVM-Squared	SVM-Shift
Chromosoms	**97.64** ± **0.79**	97.48 ± 0.72	96.81 ± 0.68	97.10 ± 0.92
Prodom	99.65 ± 0.56	99.65 ± 0.56	**99.92** ± **0.22**	98.96 ± 0.99
Protein	98.59 ± 2.30	89.67 ± 9.75	98.59 ± 3.21	61.97 ± 9.83
SwissProt	97.33 ± 0.42	97.38 ± 0.37	98.37 ± 0.33	97.37 ± 0.38

Effect of negativity in the protein data

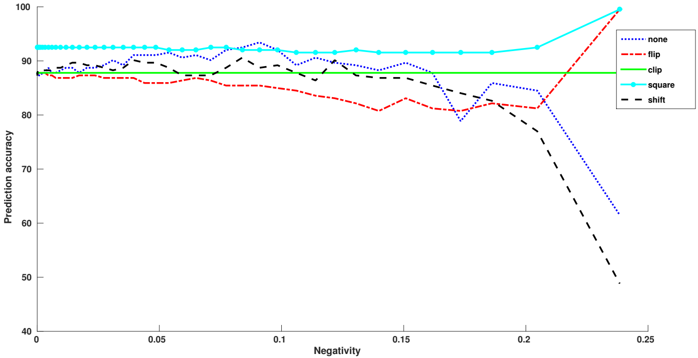


Figure : Analysis of eigenvalue correction approaches using the Protein data with varying negativity. The prediction accuracies have been obtained by using SVM.

Visualization of non metric data relations

- t-distributed stochastic Neighbor Embedding (t-SNE) with multiple maps (mm-tsne) [42]
- classical embedding in general restricted to Euclidean embeddings
- intransitive similarities and central objects can be visualized within multiple maps visualizations

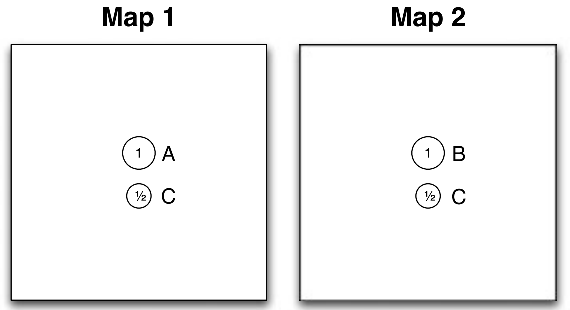


Figure : A maybe close to C and B maybe close to C. But A may not be close to B

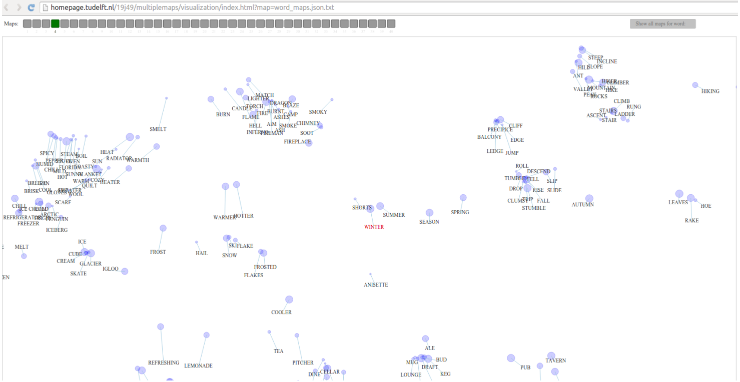


Figure : With MM-tsne rather complex indefinite similarity relations can be represented.

Take home message

- indefinite proximities can be very useful
- many classical methods can be non-heuristically applied with extra effort
- native methods for indefinite proximities are available for many learning tasks
- no need to restrict yourself to Euclidean proximities

Thank you, for your attention!

References I

J. Barnes and P. Hut.
A hierarchical O(N log N) force-calculation algorithm.
Nature, 324(4):446–449, 1986.

B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider.
The swiss-prot protein knowledgebase and its supplement trembl in 2003,.
Nucleic Acids Research, 31:365–370, 2003.

Justin Brickell, Inderjit S. Dhillon, Suvrit Sra, and Joel A. Tropp.
The metric nearness problem.
SIAM J. Matrix Analysis Applications, 30(1):375–396, 2008.

Benjamin Bustos and Tomás Skopal.
Non-metric similarity search problems in very large collections.
In Serge Abiteboul, Klemens Böhm, Christoph Koch, and Kian-Lee Tan, editors, *ICDE*, pages 1362–1365. IEEE Computer Society, 2011.

Huanhuan Chen, Peter Tino, and Xin Yao.
Probabilistic classification vector machines.
IEEE Transactions on Neural Networks, 20(6):901–914, 2009.

J. Chen and J. Ye.
Training svm with indefinite kernels.
pages 136–143, 2008.

A. Cichocki and S.-I. Amari.
Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities.
Entropy, 12(6):1532–1568, 2010.

References II

Rudi Cilibrasi and Paul M. B. Vitányi.
Clustering by compression.
IEEE Transactions on Information Theory, 51(4):1523–1545, 2005.

M.M. Deza and E. Deza.
Encyclopedia of Distances.
Encyclopedia of Distances. Springer, 2009.

M. Filippone.
Dealing with non-metric dissimilarities in fuzzy central clustering algorithms.
International Journal of Approximate Reasoning, 50(2):363–384, 2009.

A. Gisbrecht and F.-M. Schleif.
Metric and non-metric proximity transformations at linear costs.
Neurocomputing, to appear, 2015.

L. Goldfarb.
A unified approach to pattern recognition.
Pattern Recognition, 17(5):575 – 582, 1984.

S. Gu and Y. Guo.
Learning svm classifiers with indefinite kernels.
volume 2, pages 942–948, 2012.

Dan Gusfield.
Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.
Cambridge University Press, 1997.

References III

B. Haasdonk.
Feature space interpretation of svms with indefinite kernels.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(4):482–492, 2005.

B. Haasdonk and E. Pkalska.
Indefinite kernel fisher discriminant.
2008.

Bernard Haasdonk and Daniel Keysers.
Tangent distance kernels for support vector machines.
In *ICPR (2)*, pages 864–868, 2002.

Bernard Haasdonk and ElÅbieta Pkalska.
Indefinite kernel discriminant analysis.
In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 221–230. Physica-Verlag HD, 2010.

N.J. Higham.
Computing a nearest symmetric positive semidefinite matrix.
Linear Algebra and Its Applications, 103(C):103–118, 1988.

C.J. Hodgetts and U. Hahn.
Similarity-based asymmetries in perceptual matching.
Acta Psychologica, 139(2):291–299, 2012.

Thomas Hofmann and Joachim M. Buhmann.
Pairwise data clustering by deterministic annealing.
IEEE Trans. Pattern Anal. Mach. Intell., 19(1):1–14, 1997.

References IV

T. Kinsman, M. Fairchild, and J. Pelz.
Color is not a metric space implications for pattern recognition, machine learning, and computer vision.
pages 37–40, 2012.

Julian Laub.
Non-metric pairwise proximity data.
PhD thesis, 2004.

Haibin Ling and David W. Jacobs.
Using the inner-distance for classification of articulated shapes.
In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 719–726. IEEE Computer Society, 2005.

S. Liwicki, S. Zafeiriou, and M. Pantic.
Incremental slow feature analysis with indefinite kernel for online temporal video segmentation.
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7725 LNCS(PART 2):162–176, 2013.

R. Luss and A. d'Aspremont.
Support vector machine classification with indefinite kernels.
Mathematical Programming Computation, 1(2-3):97–118, 2009.

Yadong Mu and Shuicheng Yan.
Non-metric locality-sensitive hashing.
In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

A. Muoz and I.M. De Diego.
From indefinite to positive semi-definite matrices.
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4109 LNCS:764–772, 2006.

References V

M. Neuhaus and H. Bunke.
Edit distance based kernel functions for structural pattern classification.
Pattern Recognition, 39(10):1852–1863, 2006.

C.S. Ong, X. Mary, S. Canu, and A.J. Smola.
Learning with non-positive kernels.
pages 639–646, 2004.

E. Pekalska and R. Duin.
The dissimilarity representation for pattern recognition.
World Scientific, 2005.

Elsbieta Pekalska and Bernard Haasdonk.
Kernel discriminant analysis for positive definite and indefinite kernels.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(6):1017–1032, 2009.

Elzbieta Pekalska, Robert P. W. Duin, Simon Günter, and Horst Bunke.
On not making dissimilarities euclidean.
In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004 Proceedings*, pages 1145–1154, 2004.

Volker Roth, Julian Laub, Joachim M. Buhmann, and Klaus-Robert Müller.
Going metric: Denoising pairwise data.
In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *NIPS*, pages 817–824. MIT Press, 2002.

H. Sakoe and S. Chiba.
Dynamic programming algorithm optimization for spoken word recognition.
Acoustics, Speech and Signal Processing, IEEE Transactions on, 26(1):43–49, Feb 1978.

References VI

Walter J. Scheirer, Michael J. Wilber, Michael Eckmann, and Terrance E. Boult.
Good recognition is non-metric.
Pattern Recognition, 47(8):2721–2731, 2014.

F.-M. Schleif, A.Gisbrecht, and P. Tino.
Probabilistic classification vector machine at large scale.
In *Proceedings of ESANN 2015*, page to appear, 2015.

F.-M. Schleif, H. Chen, and P. Tino.
Incremental probabilistic classification vector machine with linear costs.
In *Proceedings of IJCNN 2015*, page to appear, 2015.

F.-M. Schleif and A. Gisbrecht.
Data analysis of (non-)metric proximities at linear costs.
In *Proceedings of SIMBAD 2013*, pages 59–74, 2013.

Hsuan tien Lin and Chih-Jen Lin.
A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods.
Technical report, 2003.

M.E. Tipping.
Sparse bayesian learning and the relevance vector machine.
Journal of Machine Learning Research, 1(3):211–244, 2001.

L. Van Der Maaten and G. Hinton.
Visualizing non-metric similarities in multiple maps.
Machine Learning, 87(1):33–55, 2012.

References VII

Christopher K. I. Williams and Matthias Seeger.
Using the nyström method to speed up kernel machines.
In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 682–688, 2000.