Kernelization of matrix updates, when and how?

Manfred Warmuth[1] Wojciech Kotłowski[2] Shuisheng Zhou[3]

[1]University of California, Santa Cruz

[2]Poznań University of Technology, Poland

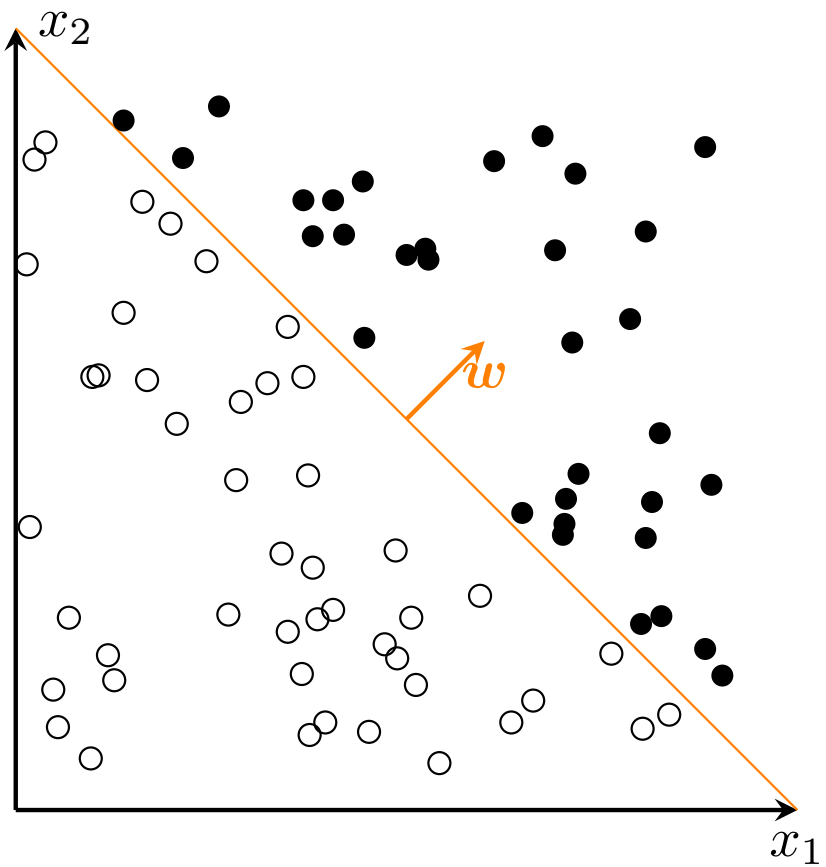
[3]Xidian University, Xian, China

ALT 2012, Lyon

Outline

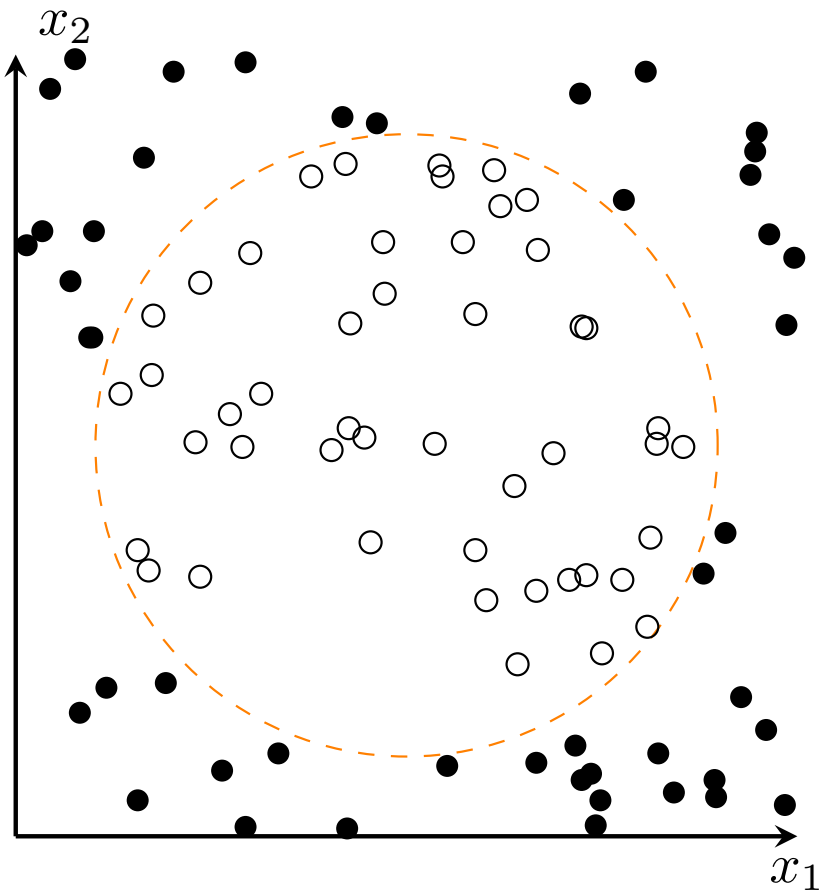
Outline

Simple start – linear binary classification



- Examples $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t, \ell_t)\}_{t=1}^{T}$.
- Linear hypothesis $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{\mathcal{S}})$
- Prediction on instance $\boldsymbol{x}$: $\boldsymbol{w} \cdot \boldsymbol{x}$

What if data not close to linear?



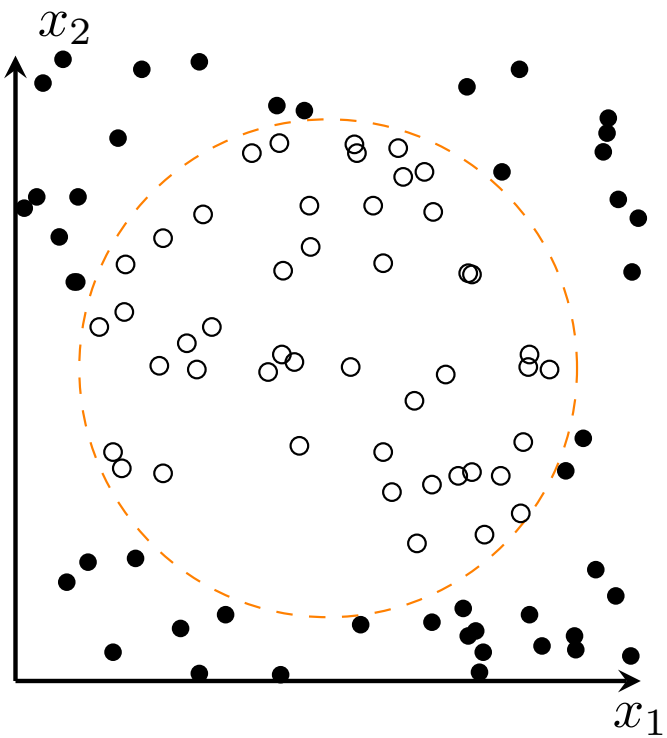
- Simply invent new features

Close to linear in feature space

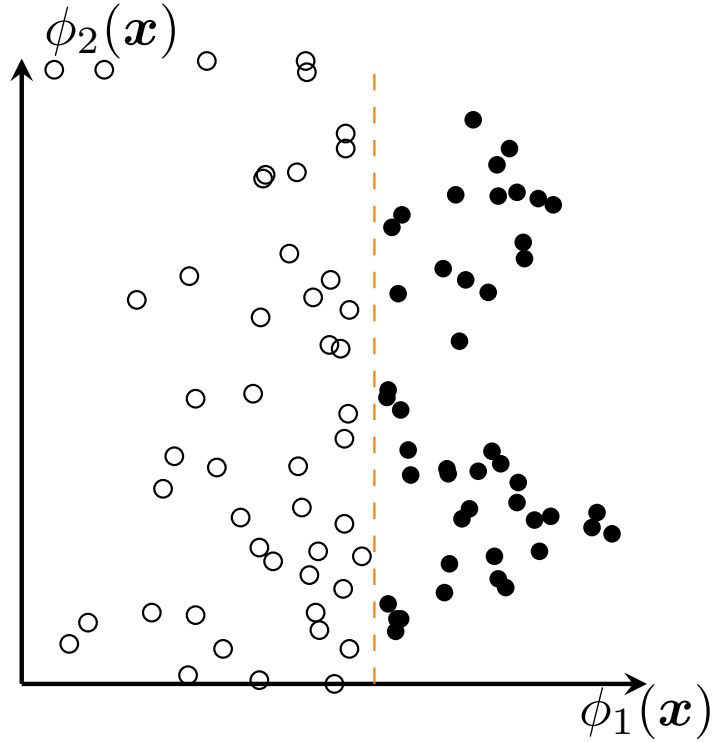
Embed instances into a feature space:

$$\phi\colon \mathbb{R}^n \to \mathbb{R}^m$$

original space



feature space



- $\boldsymbol{\phi}(x_1, x_2) = \left(\sqrt{x_1^2 + x_2^2}\ ,\ \ \arctan\frac{x_2}{x_1}\right).$

The Kernel Trick [BGV92]

- If $\boldsymbol{w}$ linear combination of expanded instances, then

$$\boldsymbol{w} \cdot \boldsymbol{x} = \underbrace{\sum_{t=1}^{T} c_t \, \phi(\boldsymbol{x}_t)}_{\boldsymbol{w}} \cdot \phi(\boldsymbol{x}) = \sum_{t=1}^{T} c_t \, \underbrace{\phi(\boldsymbol{x}_t) \cdot \phi(\boldsymbol{x})}_{K(\boldsymbol{x}_t, \boldsymbol{x})}$$

- Kernel function $K(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$ often efficient to compute.

Our contribution

Which algorithms can be kernelized?

- Necessary and sufficient conditions for kernelizability.
  - Vectors [Warmuth & Vishwanathan, 2005]
  - Asymmetric matrices.
  - Symmetric matrices.
- Prove new versions of the Representer Theorem.
- Build kernelizable algorithms with matrix parameters.
  - Multiplicative updates.

Outline

Vector data

- Labeled sequence of examples $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t, \ell_t)\}_{t=1}^{T}$.
- Instance matrix $\boldsymbol{X}$.
  Contains instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ as columns.
- Kernel matrix $\boldsymbol{X}'\boldsymbol{X}$.
  Contains pairwise dot products between the instances $\boldsymbol{x}_t$:

$$(\boldsymbol{X}'\boldsymbol{X})_{ij} = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

  (we assume instances $\boldsymbol{x}_t$ are already expanded).
- Augmented kernel matrix $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$, where $\widehat{\boldsymbol{X}}$ contains instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ and an unlabeled instance $\boldsymbol{x}$ as columns.

Kernelizable algorithms

Learning algorithm:

$$\mathcal{A}\colon \boldsymbol{\mathcal{S}} \times \boldsymbol{x} \to \mathbb{R}.$$

- A mapping from example sequence $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t, \ell_t)\}_{t=1}^{T}$ followed by a next instance $\boldsymbol{x}$ to some output range.

Definition: kernelizable algorithm

$\mathcal{A}$ is kernelizable, if for all $\boldsymbol{\mathcal{S}}, \boldsymbol{x}$ with the same labels and the same augmented kernel matrix $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$, algorithm $\mathcal{A}$ maps to the same output.

Kernelizable algorithms: rotational invariance

- For any orthogonal matrix $\boldsymbol{U}$, let:

$$\boldsymbol{U}\boldsymbol{\mathcal{S}} = \{(\boldsymbol{U}\boldsymbol{x}_t, \ell_t)\}_{t=1}^{T}.$$

- Transformation does not affect the kernel matrix:

$$(\boldsymbol{U}\boldsymbol{X})'\,\boldsymbol{U}\boldsymbol{X} = \boldsymbol{X}'\boldsymbol{U}'\boldsymbol{U}\boldsymbol{X} = \boldsymbol{X}'\boldsymbol{X}.$$

- Lemma:
  Two sequences are orthogonal transformations of each other if and only if the kernel matrices associated with the sequences are the same.

Kernalizable algorithms: definition

Original definition

$\mathcal{A}$ is kernelizable, if for all $\boldsymbol{\mathcal{S}}, \boldsymbol{x}$ with the same labels and the same augmented kernel matrix $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$, algorithm $\mathcal{A}$ maps to the same output.

Rewritten definition (based on Lemma)

$\mathcal{A}$ is kernelizable if and only if for all sequnences $\boldsymbol{\mathcal{S}}$, next instances $\boldsymbol{x}$, and orthogoal matrices $\boldsymbol{U}$, $\mathcal{A}(\boldsymbol{S}, \boldsymbol{x}) = \mathcal{A}(\boldsymbol{US}, \boldsymbol{Ux})$.

Linear algorithm

1. $\mathcal{A}$ produces a weight vector:

$$w \colon \boldsymbol{\mathcal{S}} \to \boldsymbol{R}^n.$$

2. $\mathcal{A}$ predicts with:

$$\mathcal{A}(\boldsymbol{\mathcal{S}}, \boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}.$$

Kernalizable linear algorithm

Theorem

A linear algorithm $\mathcal{A}$ is kernelizable if and only if for every input sequence $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t, \ell_t)\}_{t=1}^T$:

1. $\boldsymbol{w}$ is a linear combination of the instances of $\boldsymbol{\mathcal{S}}$,

$$\boldsymbol{w} = \sum_{t=1}^{T} c_t \boldsymbol{x}_t,$$

2. The coefficients $\{c_t\}_{t=1}^T$ depend on $\boldsymbol{\mathcal{S}}$ only via the kernel matrix $\boldsymbol{X}'\boldsymbol{X}$.

Which algorithms are kernelizable?

- Algorithm that makes linear predictions with the parameter vector which is a linear combination of instances?

- Algorithms to which the Representer Theorem [Kimeldorf & Wahba, 71] applies?

$$\boldsymbol{w} = \arg\min_{\tilde{\boldsymbol{w}}} \left(||\tilde{\boldsymbol{w}}||^2 + \eta \sum_t \text{loss}(\tilde{\boldsymbol{w}} \cdot \boldsymbol{x}_t) \right)$$

  Solution $\boldsymbol{w}$ is a linear combination of the $\boldsymbol{x}_t$.

Which algorithms are kernelizable?

- Algorithm that makes linear predictions with the parameter vector which is a linear combination of instances?
  $\Longrightarrow$ Necessary, but not sufficient.

- Algorithms to which the Representer Theorem [Kimeldorf & Wahba, 71] applies?

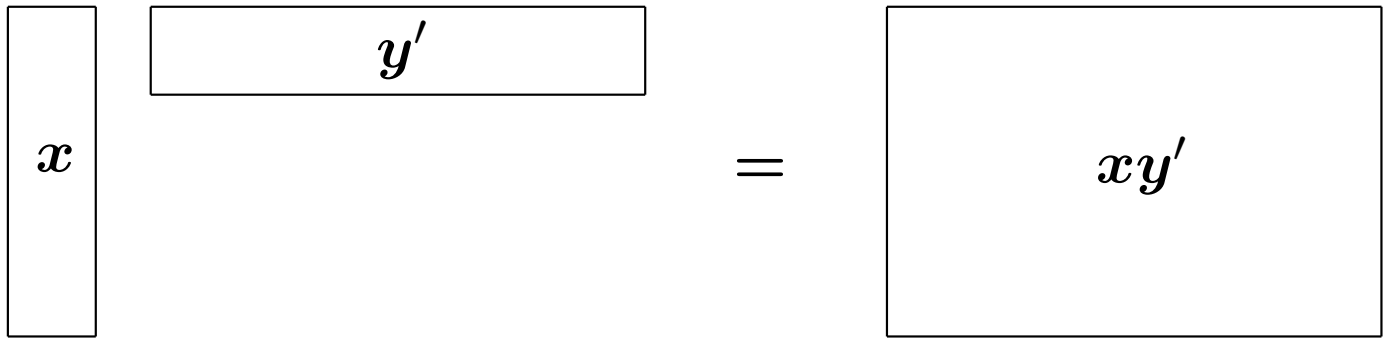
$$\boldsymbol{w} = \arg\min_{\tilde{\boldsymbol{w}}} \left(||\tilde{\boldsymbol{w}}||^2 + \eta \sum_t \text{loss}(\tilde{\boldsymbol{w}} \cdot \boldsymbol{x}_t) \right)$$

  Solution $\boldsymbol{w}$ is a linear combination of the $\boldsymbol{x}_t$.
  $\Longrightarrow$ Sufficient, but not necessary.

Outline

Asymmetric matrix data

- Instances are outer products $\boldsymbol{x}_t\boldsymbol{y}_t'$, where $\boldsymbol{x}_t \in \boldsymbol{R}^n$, $\boldsymbol{y}_t \in \boldsymbol{R}^m$



  $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t\boldsymbol{y}_t', \ell_t)\}_{t=1}^T$.

- Instance matrices:
  $\boldsymbol{X}$ containinig instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ as columns,
  $\boldsymbol{Y}$ containinig instances $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$ as columns.
- Two kernel matrices $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{Y}'\boldsymbol{Y}$.
- Corresponding augmented kernel matrices $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{Y}}'\widehat{\boldsymbol{Y}}$.
- Application: collaborative filtering:
  $\boldsymbol{x}_t$ describes user,
  $\boldsymbol{y}_t$ describes movie.

Kernelizable algorithm

- Definition: $\mathcal{A}$ is kernelizable, if for all $\boldsymbol{\mathcal{S}}, \boldsymbol{xy}'$ with the same labels and the same augmented kernel matrices $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$, $\widehat{\boldsymbol{Y}}'\widehat{\boldsymbol{Y}}$, $\mathcal{A}$ maps to the same output.

- For any orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{n\times n}$, $\boldsymbol{V} \in \mathbb{R}^{m\times m}$, let:

$$\boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{V}' = \{(\boldsymbol{U}\boldsymbol{x}_t\boldsymbol{y}_t'\boldsymbol{V}', \ell_t)\}_{t=1}^{T}.$$

- Rewritten definition: $\mathcal{A}$ is kernelizable if and only if for all sequences $\boldsymbol{\mathcal{S}}$, next instances $\boldsymbol{x}$, and orthogoal matrices $\boldsymbol{U}$ $\boldsymbol{V}$,

$$\mathcal{A}(\boldsymbol{\mathcal{S}}, \boldsymbol{xy}') = \mathcal{A}(\boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{V}', \boldsymbol{U}\boldsymbol{xy}'\boldsymbol{V}').$$

Linear algorithm

1. $\mathcal{A}$ produces a weight matrix:

$$\boldsymbol{W} : \boldsymbol{\mathcal{S}} \to \boldsymbol{R}^{m \times n}.$$

2. $\mathcal{A}$ predicts with:

$$\mathcal{A}(\boldsymbol{\mathcal{S}}, \boldsymbol{x}\boldsymbol{y}') = \operatorname{tr}(\boldsymbol{W}' \, \boldsymbol{x}\boldsymbol{y}') = \boldsymbol{y}'\boldsymbol{W}'\boldsymbol{x}.$$

Kernalizable linear algorithm

Theorem

A linear algorithm $\mathcal{A}$ is kernelizable if and only if for every input sequence $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t \boldsymbol{y}_t', \ell_t)\}_{t=1}^T$:

1. $\boldsymbol{W}$ is a linear combination of the instances of $\boldsymbol{\mathcal{S}}$,

$$\boldsymbol{W} = \sum_{i,j=1}^{T} c_{ij} \boldsymbol{x}_i \boldsymbol{y}_j' = \boldsymbol{X} \boldsymbol{C} \boldsymbol{Y}',$$

2. The coefficients $\{c_{ij}\}_{i,j=1}^T$ depend on $\boldsymbol{\mathcal{S}}$ only via the kernel matrices $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{Y}'\boldsymbol{Y}$.

Kernalizable linear algorithm

Theorem

A linear algorithm $\mathcal{A}$ is kernelizable if and only if for every input sequence $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t \boldsymbol{y}_t', \ell_t)\}_{t=1}^T$:

1. $\boldsymbol{W}$ is a linear combination of the instances of $\boldsymbol{\mathcal{S}}$,

$$\boldsymbol{W} = \sum_{i,j=1}^{T} c_{ij} \boldsymbol{x}_i \boldsymbol{y}_j' = \boldsymbol{X} \boldsymbol{C} \boldsymbol{Y}',$$

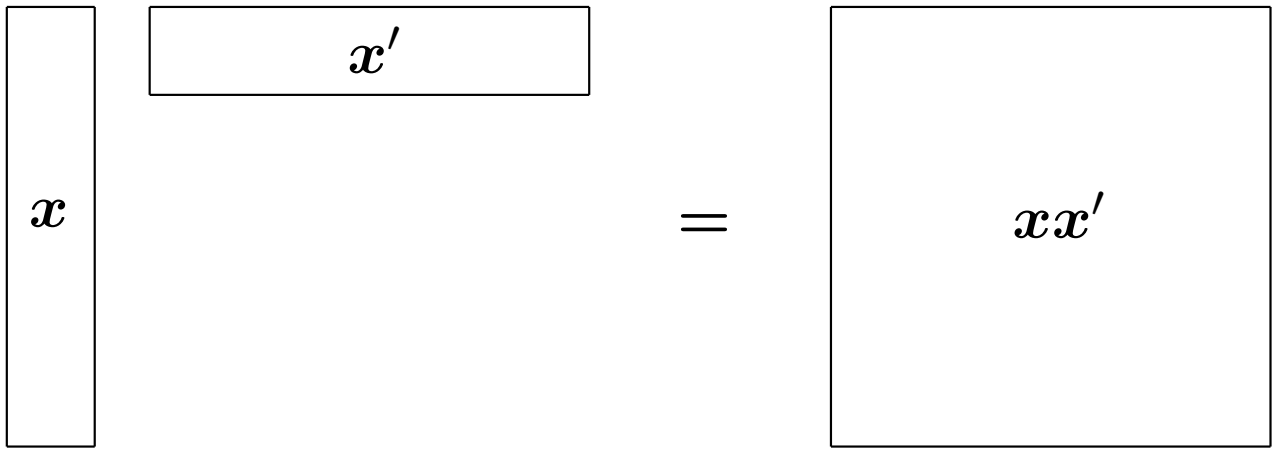
2. The coefficients $\{c_{ij}\}_{i,j=1}^T$ depend on $\boldsymbol{\mathcal{S}}$ only via the kernel matrices $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{Y}'\boldsymbol{Y}$.

- Compare with vector case: $\boldsymbol{w} = \sum_{t=1}^T c_t \boldsymbol{x}_t$.

Outline

Symmetric matrix data

- Instances are outer products $\boldsymbol{x}_t\boldsymbol{x}_t'$, where $\boldsymbol{x}_t \in \boldsymbol{R}^n$.



  $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t\boldsymbol{x}_t', \ell_t)\}_{t=1}^T$.

- Instance matrix:
  $\boldsymbol{X}$ containinig instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ as columns,
- Kernel matrix $\boldsymbol{X}'\boldsymbol{X}$.
- Corresponding augmented kernel matrix $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$.
- Application: PCA.

Kernelizable algorithm

- Definition: $\mathcal{A}$ is kernelizable, if for all $\boldsymbol{\mathcal{S}}, \boldsymbol{x}\boldsymbol{x}'$ with the same labels and the same augmented kernel matrix $\widehat{\boldsymbol{X}}'\widehat{\boldsymbol{X}}$, $\mathcal{A}$ maps to the same output.

- For any orthogonal matrix $\boldsymbol{U}$, let:

$$\boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{U}' = \{(\boldsymbol{U}\boldsymbol{x}_t\boldsymbol{x}_t'\boldsymbol{U}', \ell_t)\}_{t=1}^{T}.$$

- Rewritten definition: $\mathcal{A}$ is kernelizable if and only if for all sequences $\boldsymbol{\mathcal{S}}$, next instances $\boldsymbol{x}\boldsymbol{x}'$, and orthogoal matrices $\boldsymbol{U}$,

$$\mathcal{A}(\boldsymbol{S}, \boldsymbol{x}\boldsymbol{x}') = \mathcal{A}(\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}', \boldsymbol{U}\boldsymbol{x}\boldsymbol{x}'\boldsymbol{U}').$$

Linear algorithm

1. $\mathcal{A}$ produces a symmetric weight matrix:

$$\boldsymbol{W} : \boldsymbol{\mathcal{S}} \to \boldsymbol{R}^{n \times n}.$$

2. $\mathcal{A}$ predicts with:

$$\mathcal{A}(\boldsymbol{\mathcal{S}}, \boldsymbol{x}\boldsymbol{x}') = \mathrm{tr}(\boldsymbol{W}\ \boldsymbol{x}\boldsymbol{x}') = \boldsymbol{x}'\boldsymbol{W}\boldsymbol{x}.$$

Kernalizable linear algorithm

Theorem

A linear algorithm $\mathcal{A}$ is kernelizable if and only if for every input sequence $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t \boldsymbol{x}'_t, \ell_t)\}_{t=1}^T$:

1. $\boldsymbol{W}$ is a linear combination of the instances of $\boldsymbol{\mathcal{S}}$ and the identity matrix,

$$\boldsymbol{W} = \sum_{i,j=1}^{T} c_{ij} \boldsymbol{x}_i \boldsymbol{x}'_j + c\boldsymbol{I} = \boldsymbol{X}\boldsymbol{C}\boldsymbol{X}' + c\boldsymbol{I}.$$

2. The coefficients $\{c_{ij}\}_{i,j=1}^T, c$ depend on $\boldsymbol{\mathcal{S}}$ only via the kernel matrix $\boldsymbol{X}'\boldsymbol{X}$.

Kernalizable linear algorithm

Theorem

A linear algorithm $\mathcal{A}$ is kernelizable if and only if for every input sequence $\boldsymbol{\mathcal{S}} = \{(\boldsymbol{x}_t \boldsymbol{x}'_t, \ell_t)\}_{t=1}^T$:

1. $\boldsymbol{W}$ is a linear combination of the instances of $\boldsymbol{\mathcal{S}}$ and the identity matrix,

$$\boldsymbol{W} = \sum_{i,j=1}^{T} c_{ij} \boldsymbol{x}_i \boldsymbol{x}'_j + c\boldsymbol{I} = \boldsymbol{X}\boldsymbol{C}\boldsymbol{X}' + c\boldsymbol{I}.$$

2. The coefficients $\{c_{ij}\}_{i,j=1}^T, c$ depend on $\boldsymbol{\mathcal{S}}$ only via the kernel matrix $\boldsymbol{X}'\boldsymbol{X}$.

- Compare with asymmetric case: $\boldsymbol{W} = \sum_{i,j=1}^{T} c_{ij} \boldsymbol{x}_i \boldsymbol{y}'_j$.

	vectors	asym. matrices	sym. matrices
instances	$\boldsymbol{x}_t$	$\boldsymbol{x}_t \boldsymbol{y}_t'$	$\boldsymbol{x}_t \boldsymbol{x}_t'$
kernel matrix	$\boldsymbol{X}'\boldsymbol{X}$,	$\boldsymbol{X}'\boldsymbol{X}$, $\boldsymbol{Y}'\boldsymbol{Y}$	$\boldsymbol{X}'\boldsymbol{X}$
transformation	$\boldsymbol{\mathcal{S}} \mapsto \boldsymbol{U}\boldsymbol{\mathcal{S}}$	$\boldsymbol{\mathcal{S}} \mapsto \boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{V}'$	$\boldsymbol{\mathcal{S}} \mapsto \boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{U}'$
weights	$\boldsymbol{w} \in \mathbb{R}^n$	$\boldsymbol{W} \in \mathbb{R}^{n \times m}$	$\boldsymbol{W} \in \mathbb{R}^{n \times n}, \boldsymbol{W}' = \boldsymbol{W}$
prediction	$\boldsymbol{w} \cdot \boldsymbol{x}$	$\mathrm{tr}(\boldsymbol{W}'\, \boldsymbol{x}\boldsymbol{y}')$	$\mathrm{tr}(\boldsymbol{W}\, \boldsymbol{x}\boldsymbol{x}')$
kernelizability	$\boldsymbol{w} = \sum_i c_i \boldsymbol{x}_i$	$\boldsymbol{W} = \sum_{ij} c_{ij} \boldsymbol{x}_i \boldsymbol{y}_j'$	$\boldsymbol{W} = \sum_{ij} c_{ij} \boldsymbol{x}_i \boldsymbol{x}_j' + c\boldsymbol{I}$

Outline

Representer theorem for matrices

$$\inf_{\boldsymbol{W}} \quad \Omega(\boldsymbol{W}) + \eta \sum_t \text{loss}(\text{tr}(\boldsymbol{W}\ \boldsymbol{x}_t \boldsymbol{y}_t'))$$

- [Abernethy et al, 09] Kernelizable, if $\Omega(\boldsymbol{W})$ is:
  - spectrally invariant,
  - non-decreasing in singular values of $\boldsymbol{W}$.

Our version of the Representer Theorem

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\mathcal{S}}).$$

- Kernelizable, if:
  - $\boldsymbol{W}^*$ is unique,
  - $\mathcal{L}(\boldsymbol{W}, \boldsymbol{\mathcal{S}})$ is rotationally invariant in a sense that for any orthogonal $\boldsymbol{U}, \boldsymbol{V}$:

    $$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\mathcal{S}}) \implies \boldsymbol{U}\boldsymbol{W}^*\boldsymbol{V}' = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{V}')$$

- Conditions incomparable to [Abernethy et al, 09].

- E.g.,

  $$\inf_{\boldsymbol{W} \in \mathcal{W}} \quad \Omega(\boldsymbol{W}) + \eta \sum_t \mathrm{loss}(\mathrm{tr}(\boldsymbol{W}\ \boldsymbol{x}_t \boldsymbol{y}_t')),$$

  for some spectrally invariant Ω and $\mathcal{W}$.

The Representer Theorem for symmetric matrices

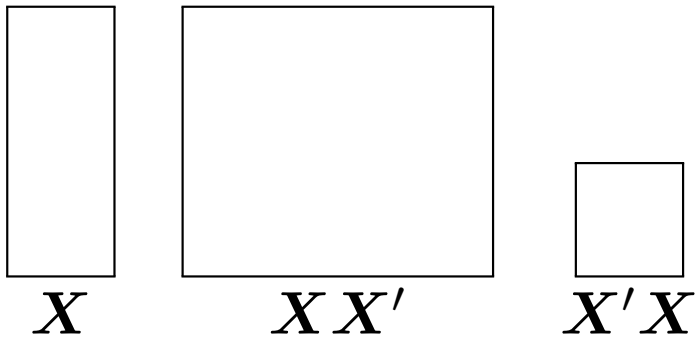
$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\mathcal{S}}).$$

- Kernelizable, if:
  - $\boldsymbol{W}^*$ is unique,
  - $\mathcal{L}(\boldsymbol{W}, \boldsymbol{\mathcal{S}})$ is rotationally invariant in a sense that for any orthogonal $\boldsymbol{U}$:

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\mathcal{S}}) \implies \boldsymbol{U}\boldsymbol{W}^*\boldsymbol{U}' = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{U}\boldsymbol{\mathcal{S}}\boldsymbol{U}')$$

Outline

Standard method



- Eigendecomposition of large Gram Matrix $\boldsymbol{X}\boldsymbol{X}'$ can be computed from eigendecomposition of small kernel matrix $\boldsymbol{X}'\boldsymbol{X}$.

- Key to kernelization of PCA and Fisher Linear Discriminant Functions. [Schölkopf et al., 98; Mika et al., 99]

Generalization

- Need decomposition of large Gram matrix $\boldsymbol{XCY}' \in \boldsymbol{R}^{n \times m}$ in terms of small kernel matrices $\boldsymbol{X}'\boldsymbol{X}$ and $\boldsymbol{Y}'\boldsymbol{Y} \in \boldsymbol{R}^{T \times T}$

Lemma

SVD of generalized Gram Matrix $\underset{n \times T}{\boldsymbol{X}} \; \underset{T \times T}{\boldsymbol{C}} \; \underset{T \times m}{\boldsymbol{Y}}{}'$ can be computed from the SVD of $\sqrt{\boldsymbol{X}'\boldsymbol{X}}\, \boldsymbol{C}\, \sqrt{\boldsymbol{Y}'\boldsymbol{Y}}$

- Symmetric case covered by setting $\boldsymbol{Y} = \boldsymbol{X}$.

Outline

Matrix Exponentiated Gradient

- Offline problem:

$$\min_{\boldsymbol{W}\in\mathcal{W}} \sum_t \ell_t(\mathrm{tr}(\boldsymbol{W}\boldsymbol{x}_t\boldsymbol{x}_t')),$$

  where $\mathcal{W}$ is a set of positive definite matrices with unit trace.

- Online algorithm derived from:

$$\boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W}\in\mathcal{W}} \ \mathrm{tr}\left(\boldsymbol{W}\left(\log \boldsymbol{W} - \log \boldsymbol{W}_t\right)\right) + \eta \mathrm{tr}\left(\frac{\partial \ell(\mathrm{tr}(\boldsymbol{W}_t'\boldsymbol{x}_t\boldsymbol{x}_t'))}{\partial \boldsymbol{W}_t}\boldsymbol{W}\right).$$

  Rotationally invariant problem with unique solution
  $\Longrightarrow$ kernelizable.

Matrix Exponentiated Gradient

- MEG update:

$$\boldsymbol{W}_{t+1} = \frac{\mathbf{exp}\left(-\eta \sum_{i=1}^{t} \partial_i(\boldsymbol{W}_i)\boldsymbol{x}_i\boldsymbol{x}_i'\right)}{Z_t},$$

  where $\partial_i = \frac{\partial \ell_i(z)}{\partial z}\Big|_{z=\boldsymbol{W}_i\boldsymbol{x}_i\boldsymbol{x}_i'}$ and Z_t is such that $\mathrm{tr}(\boldsymbol{W}_{t+1}) = 1$.

- has the form $\boldsymbol{X}\boldsymbol{C}\boldsymbol{X}' + c\boldsymbol{I}$ where $\boldsymbol{C}$ is diagonal.
  $\Longrightarrow$ SVD of $\boldsymbol{X}\boldsymbol{C}\boldsymbol{X}'$ can be computed efficiently.
- Applications: online PCA.

Conclusions

- Necessary and sufficient conditions for kernelizability for asymmetric and symmetric matrix data.
- New versions of the Representer Theorem.
- Kernelization of multiplicative updates and other online algorithms.

Conclusions

- Necessary and sufficient conditions for kernelizability for asymmetric and symmetric matrix data.
- New versions of the Representer Theorem.
- Kernelization of multiplicative updates and other online algorithms.

Thank you!