Learning with Output Kernels and Latent Kernels

Dale Schuurmans

University of Alberta

PSL - 2013

Kernels

Enable **finite** computation with **unbounded** feature representations

objects → features → pairwise inner products

$$\begin{bmatrix} z_1 \\ \vdots \\ z_t \end{bmatrix} \rightarrow \begin{bmatrix} \phi_1(z_1) & \phi_2(z_1) & \cdots \\ \vdots & & \\ \phi_1(z_t) & \phi_2(z_t) & \cdots \end{bmatrix} \rightarrow \begin{bmatrix} \phi(z_1) \cdot \phi(z_1) & \cdots & \phi(z_1) \cdot \phi(z_t) \\ \vdots & & \vdots \\ \phi(z_t) \cdot \phi(z_1) & \cdots & \phi(z_t) \cdot \phi(z_t) \end{bmatrix}$$

↓

kernel matrix

$$\begin{bmatrix} \kappa(z_1, z_1) & \cdots & \kappa(z_1, z_t) \\ \vdots & & \vdots \\ \kappa(z_t, z_1) & \cdots & \kappa(z_t, z_t) \end{bmatrix}$$

iff training problem can be expressed strictly in terms of inner products

Pros: flexible modeling, finite model size

Cons: $\Theta(t^2)$ problem size, model size also depends on t

Question

When is kernelization possible?

Some New Opportunities

Consider learning a prediction function

$$\mathcal{X} \quad \longrightarrow \quad \mathcal{Y}$$

Some New Opportunities

Consider learning a prediction function

$$\mathcal{X} \quad \longrightarrow \quad \mathcal{Y}$$

Input kernels

- well understood

Some New Opportunities

Consider learning a prediction function

$$\mathcal{X} \quad \longrightarrow \quad \mathcal{Y}$$

Input kernels

- well understood

Output kernels?

- less is known
- open questions
- little attention

Some New Opportunities

Consider learning a prediction function

$$\mathcal{X} \quad \longrightarrow \quad \mathcal{Y}$$

Input kernels

- well understood

Latent kernels?

- less is known
- open questions
- little attention

Output kernels?

- less is known
- open questions
- little attention

Some New Opportunities

Input kernels

- reasonably well understood

Output kernels?

- little is known
- many open questions
- receiving minimal attention

Latent kernels?

- ditto

Recent Results

Input kernels

- *regularization for manifolds*

Output kernels?

- *convex clustering*
- **multi-label classification**

Latent kernels?

- **robust regression**
- **latent class classification**

Outline

1. Learning with Fixed Kernels
   - 1.1 Input Kernels
   - 1.2 Output Kernels
2. Learning while Adapting Kernels
   - 2.1 Adapting Input Kernels
   - 2.2 Adapting Output Kernels
3. Learning while Adapting **Latent** Kernels

Part 1: Learning with Fixed Kernels

Warm-up:
Learning with Fixed Input Kernels

The (Input) Kernel Trick

Recall support vector machines

- **Train**: Given input data X target outputs $\mathbf{y}$, solve

$$\begin{aligned}\min_{\mathbf{w}} \; & \mathbf{1}^\top(\mathbf{1} - \Delta(\mathbf{y})\phi(X)\mathbf{w})_+ \; + \; \tfrac{\alpha}{2}\|\mathbf{w}\|_2^2 \\ = \; & \max_{\mathbf{1}\geq\mathbf{a}\geq 0} \; \mathbf{1}^\top\mathbf{a} \; - \; \tfrac{1}{2\alpha}\mathbf{a}^\top\Delta(\mathbf{y})\,\phi(X)\phi(X)^\top\Delta(\mathbf{y})\mathbf{a} \\ & \text{where } \mathbf{w} = \phi(X)^\top\mathbf{a}\end{aligned}$$

- **Test**: Given test input $\mathbf{x}$, predict

$$\hat{y} = \operatorname{sign}(\mathbf{w}^\top\phi(\mathbf{x})) = \operatorname{sign}(\mathbf{a}^\top\phi(X)\phi(\mathbf{x}))$$

The (Input) Kernel Trick

Recall support vector machines

- **Train**: Given input data X target outputs $\mathbf{y}$, solve

$$\begin{aligned}\min_{\mathbf{w}} \; & \mathbf{1}^\top(\mathbf{1} - \Delta(\mathbf{y})\phi(X)\mathbf{w})_+ \; + \; \tfrac{\alpha}{2}\|\mathbf{w}\|_2^2 \\ = \; & \max_{\mathbf{1}\geq\mathbf{a}\geq 0} \; \mathbf{1}^\top\mathbf{a} \; - \; \tfrac{1}{2\alpha}\mathbf{a}^\top\Delta(\mathbf{y})\,\phi(X)\phi(X)^\top\Delta(\mathbf{y})\mathbf{a} \\ & \text{where } \mathbf{w} = \phi(X)^\top\mathbf{a}\end{aligned}$$

- **Test**: Given test input $\mathbf{x}$, predict

$$\hat{y} = \operatorname{sign}(\mathbf{w}^\top\phi(\mathbf{x})) = \operatorname{sign}(\mathbf{a}^\top\phi(X)\phi(\mathbf{x}))$$

Is this a special case?

- Need hinge loss? Need Lagrange dual?
- **No**!

The (Input) Kernel Trick

Recall support vector machines

- **Train**: Given input data X target outputs $\mathbf{y}$, solve

$$\begin{aligned}\min_{\mathbf{w}} \; & \mathbf{1}^\top(\mathbf{1} - \Delta(\mathbf{y})\phi(X)\mathbf{w})_+ \; + \; \tfrac{\alpha}{2}\|\mathbf{w}\|_2^2 \\ = \; & \max_{\mathbf{1}\geq\mathbf{a}\geq 0} \; \mathbf{1}^\top\mathbf{a} \; - \; \tfrac{1}{2\alpha}\mathbf{a}^\top\Delta(\mathbf{y})\,\phi(X)\phi(X)^\top\Delta(\mathbf{y})\mathbf{a} \\ & \text{where } \mathbf{w} = \phi(X)^\top\mathbf{a}\end{aligned}$$

- **Test**: Given test input $\mathbf{x}$, predict

$$\hat{y} = \operatorname{sign}(\mathbf{w}^\top\phi(\mathbf{x})) = \operatorname{sign}(\mathbf{a}^\top\phi(X)\phi(\mathbf{x}))$$

Is this a special case?

- Need hinge loss? Need Lagrange dual?
- **No**! Only need linearity and Euclidean regularization

The (Input) Kernel Trick

Simple representer theorem

For **any function** L and any increasing function R, if

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\phi(X)\mathbf{w}) + R(\|\mathbf{w}\|_2^2)$$

exists, then $\mathbf{w}^* = \phi(X)^\top \mathbf{a}^*$ for some $\mathbf{a}^*$

The (Input) Kernel Trick

Simple representer theorem

For **any function** L and any increasing function R, if

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\phi(X)\mathbf{w}) + R(\|\mathbf{w}\|_2^2)$$

exists, then $\mathbf{w}^* = \phi(X)^\top \mathbf{a}^*$ for some $\mathbf{a}^*$

Proof

Since $\mathbf{w}^* = \mathbf{w}_0^* + \mathbf{w}_1^*$ for $\mathbf{w}_1^* \in \text{rowspan}(\Phi)$ and $\mathbf{w}_0^* \perp \text{rowspan}(\Phi)$

$$\mathbf{w}_1^* = \Phi^\top \mathbf{a}^* \text{ for some } \mathbf{a}^*$$
$$\Phi\mathbf{w}^* = \Phi\mathbf{w}_0^* + \Phi\mathbf{w}_1^* = \Phi\mathbf{w}_1^*$$
$$\|\mathbf{w}^*\|_2^2 = \|\mathbf{w}_0^* + \mathbf{w}_1^*\|_2^2 = \|\mathbf{w}_0^*\|_2^2 + \|\mathbf{w}_1^*\|_2^2$$

If $\mathbf{w}_0^* \neq 0$ then

$$L(\Phi\mathbf{w}^*) + R(\|\mathbf{w}^*\|_2^2) = L(\Phi\mathbf{w}_1^*) + R(\|\mathbf{w}_0^*\|_2^2 + \|\mathbf{w}_1^*\|_2^2)$$
$$> L(\Phi\mathbf{w}_1^*) + R(\|\mathbf{w}_1^*\|_2^2) \text{ contradiction } \blacksquare$$

The (Input) Kernel Trick

Can easily extend theorem to multivariate predictors

- Vector output $\hat{\mathbf{y}}^\top = f(\phi(\mathbf{x})^\top W)$ for a parameter matrix W
- Can still equivalently reformulate training and testing in terms of an adjoint matrix A such that $W = \Phi^\top A$

Get equivalent adjoint and kernel formulations

Original training	$\min_W L(\Phi W) + R(\mathrm{tr}(W^\top W))$
Original prediction	$\mathbf{x} \mapsto \hat{\mathbf{y}}^\top = f(\phi(\mathbf{x})^\top W)$
Adjoint training	$\min_A L(\Phi\Phi^\top A) + R(\mathrm{tr}(A^\top \Phi\Phi^\top A))$
Adjoint prediction	$\mathbf{x} \mapsto \hat{\mathbf{y}}^\top = f(\phi(\mathbf{x})^\top \Phi^\top A)$
Kernel training	$\min_A L(KA) + R(\mathrm{tr}(A^\top KA))$
Kernel prediction	$\mathbf{x} \mapsto \hat{\mathbf{y}}^\top = f(\mathbf{k}^\top A)$

Question:
Is there an Output Kernel Trick

Is there an Output Kernel Trick?

Questions

- Can any loss for linear predictors be equivalently re-expressed in terms of **inner products** between output vectors?
- Is there an analog of the representer theorem in this case?

No. It is not that easy

Parameter regularization is irrelevant

$$\begin{aligned} & \min_W L(XW; Y) + R(\mathrm{tr}(W^\top W)) \\ = \; & \min_A L(XX^\top A; Y) + R(\mathrm{tr}(A^\top XX^\top A)) \\ = \; & \min_A L(KA; Y) + R(\mathrm{tr}(A^\top KA)) \end{aligned}$$

When can $L(KA; Y)$ be equivalently expressed in terms of $N = YY^\top$?

Positive Example: Squared Error Regression

Reformulate standard objective

by applying Fenchel duality

$$\min_W \|XW - Y\|_F^2 + \tfrac{\alpha}{2}\|W\|_F^2$$

$$= \max_\Lambda \; -\tfrac{1}{2\alpha}\mathrm{tr}(\Lambda^\top XX^\top \Lambda) - \tfrac{1}{2}\mathrm{tr}(\Lambda^\top \Lambda) + \mathrm{tr}(\Lambda^\top Y)$$

where $W = -\frac{1}{\alpha}X^\top \Lambda$

Let Ω be s.t. $\Omega Y = \Lambda$ (always exists if Y full rank)

$$= \max_\Omega \; -\tfrac{1}{2\alpha}\mathrm{tr}(\Omega^\top K\Omega N) - \tfrac{1}{2}\mathrm{tr}(\Omega^\top \Omega N) + \mathrm{tr}(\Omega^\top N)$$

where $K = XX^\top$ and $N = YY^\top$

- Both input and output kernelizable

Positive Example: Squared Error Regression

Now consider feature expansion and kernels

$$\begin{aligned} & \min_W \|\phi(X)W - \psi(Y)\|_F^2 + \tfrac{\alpha}{2}\|W\|_F^2 \\ = \quad & \max_\Omega \; -\tfrac{1}{2\alpha}\mathrm{tr}(\Omega^\top K\Omega N) - \tfrac{1}{2}\mathrm{tr}(\Omega^\top \Omega N) + \mathrm{tr}(\Omega^\top N) \end{aligned}$$

where

$$\begin{aligned} K_{ij} &= \kappa(X_{i:}, X_{j:}) &= \phi(X_{i:})\phi(X_{j:})^\top \\ N_{ij} &= \eta(Y_{i:}, Y_{j:}) &= \psi(Y_{i:})\psi(Y_{j:})^\top \end{aligned}$$

Note

The variable Ω is $t \times t$, so scaling is an issue

Positive Example: Squared Error Regression

Unfortunately prediction is hard

Given test $\mathbf{x}$, have to determine the prediction $\hat{\mathbf{y}}$ based on

$$\begin{aligned}\psi(\hat{\mathbf{y}})^\top &= \phi(\mathbf{x})^\top W \\ &= -\tfrac{1}{\alpha}\phi(\mathbf{x})^\top \phi(X)^\top \Omega \psi(Y) \\ &= -\tfrac{1}{\alpha}\mathbf{k}^\top \Omega \psi(Y)\end{aligned}$$

hence

$$\begin{aligned}\psi(\hat{\mathbf{y}})^\top \psi(Y)^\top &= -\tfrac{1}{\alpha}\mathbf{k}^\top \Omega \psi(Y)\psi(Y)^\top \\ &= -\tfrac{1}{\alpha}\mathbf{k}^\top \Omega N\end{aligned}$$

Requires a solution to pre-image problem

Have to find $\hat{\mathbf{y}}$ that satisfies $\eta(Y, \hat{\mathbf{y}}) = -\frac{1}{\alpha} N \Omega^\top \mathbf{k}$

Positive Example: Multi-class SVM

Reformulate primal objective

by applying Lagrange duality

$$\min_{W,\boldsymbol{\xi}} \tfrac{\alpha}{2}\|W\|_F^2 + \mathbf{1}^\top \boldsymbol{\xi} \text{ s.t. } \boldsymbol{\xi}\mathbf{1}^\top \geq \mathbf{1}\mathbf{1}^\top - Y + XW - \boldsymbol{\delta}(XWY^\top)\mathbf{1}^\top$$

$$= \max_{\Lambda \geq 0, \Lambda\mathbf{1}=\mathbf{1}} t - \operatorname{tr}(\Lambda^\top Y) - \tfrac{1}{2\alpha}\operatorname{tr}((Y-\Lambda)^\top XX^\top (Y-\Lambda))$$

Let Ω be s.t. $\Omega Y = \Lambda$ (always exists if Y full rank)

$$= \max_{\Omega \geq 0, \Omega\mathbf{1}=\mathbf{1}} t - \operatorname{tr}(\Omega^\top N) - \tfrac{1}{2\alpha}\operatorname{tr}((I-\Omega)^\top K(I-\Omega)N)$$

where $K = XX^\top$ and $N = YY^\top$

Equality holds provided $Y \geq 0,\ Y\mathbf{1} = \mathbf{1}$

- Both input and output kernelizable

Positive Example: Multi-class Logistic Regression

Reformulate standard training objective

by applying Fenchel duality

$$\min_W \tfrac{\alpha}{2}\|W\|_F^2 + A(XW) - \mathrm{tr}(XWY^\top)$$

where $A(XW) = \sum_i \log(\sum_j X_{i:} W_{:j})$

$$= \max_{\Lambda \geq 0, \Lambda \mathbf{1} = \mathbf{1}} -\mathrm{tr}(\Lambda \log \Lambda^\top) - \tfrac{1}{2\alpha}\mathrm{tr}((Y-\Lambda)^\top XX^\top (Y-\Lambda))$$

Let Ω be s.t. $\Omega Y = \Lambda$ (always exists if Y full rank)

$$= \max_{\Omega \geq 0, \Omega \mathbf{1} = \mathbf{1}} -\mathrm{tr}(\Omega \log \Omega^\top) - \mathbf{1}^\top \Omega \log(N\mathbf{1}) - \tfrac{1}{2\alpha}\mathrm{tr}((I-\Omega)^\top K(I-\Omega)N)$$

Equality holds provided $Y \in \{0,1\}^{t \times k}$, $Y\mathbf{1} = \mathbf{1}$

- Both input and output kernelizable

Multi-label Classification with Output Kernels

Multi-label classification

Multiple output classes per instance

$$\mathbf{x} \quad \mapsto \quad \mathbf{y} \in \{0,1\}^L$$

Try to capture correlation between labels

- Instead of learning a separate classifier for each label $y_\ell \in \{0,1\}$, exploit correlation between labels
- But retain tractability at training and test time (i.e. cannot enumerate every label subset)

Multi-label classification

Consider a simple linear classification model

Parameters

Feature by label weight matrix W, threshold vector $\mathbf{u}$

Classification

Given instance $\mathbf{x}$

- Compare label scores $s_\ell(\mathbf{x}) = \phi(\mathbf{x})^\top W_{:\ell}$ for $\ell \in \{1, ..., L\}$
- against threshold value $s_0(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{u}$
- Assign

$$y_\ell^* = \arg \max_{y_\ell \in \{0,1\}} y_\ell (s_\ell(\mathbf{x}) - s_0(\mathbf{x}))$$

for each label $\ell \in \{1, .., L\}$

Advantages

- Individual label classifications weakly coordinated via threshold
- Efficient multi-label classification

Learning Multi-label Classifiers

Consider a simple large margin training objective

Calibrated Separation Ranking Loss (CSRL)

Given input $X_{i:}$ and label indicator $Y_{i:} \in \{0,1\}^{1\times L}$, define

$$\begin{aligned} L_i(\phi(X_{i:})[W,\mathbf{u}];Y_{i:}) &= \max_{\ell\in Y_{i:}} (1+s_0(X_{i:})-s_\ell(X_{i:}))_+ \\ &+ \max_{\bar{\ell}\in \bar{Y}_{i:}} (1+s_{\bar{\ell}}(X_{i:})-s_0(X_{i:}))_+ \end{aligned}$$

where $\bar{Y}_{i:} = 1 - Y_{i:}$, $s_\ell(X_{i:}) = \phi(X_{i:})^\top W_{:\ell}$ and $s_0(X_{i:}) = \phi(X_{i:})^\top \mathbf{u}$

Idea

- For labels $\ell \in Y_{i:}$ tries to **increase** s_ℓ to a margin above s_0
- For labels $\bar{\ell} \in Y_{i:}$ tries to **decrease** s_ℓ to a margin below s_0

Learning Multi-label Classifiers

Training problem

With regularization, training problem is a quadratic program

$$\begin{aligned} \min_{W,\mathbf{u},\boldsymbol{\xi},\zeta} \quad & \frac{\alpha}{2}(\|W\|_F^2 + \|\mathbf{u}\|_2^2) + \mathbf{1}^\top \boldsymbol{\xi} + \mathbf{1}^\top \zeta \\ \text{s.t.} \quad & \boldsymbol{\xi} \geq 0, \;\; \boldsymbol{\xi}\mathbf{1}^\top \geq Y \circ (\mathbf{1}\mathbf{1}^\top + \phi(X)(\mathbf{u}\mathbf{1}^\top - W)) \\ & \zeta \geq 0, \;\; \zeta\mathbf{1}^\top \geq \bar{Y} \circ (\mathbf{1}\mathbf{1}^\top - \phi(X)(\mathbf{u}\mathbf{1}^\top - W)) \end{aligned}$$

Outcome

Achieves state of the art performance (Guo & Schuurmans, 2011)

Question

Can this multi-label training loss be output kernelized?

- I.e., can the training problem be equivalently re-expressed in terms of $YY^\top$?
- Would allow more complex label correlations to be modeled

Is Output Kernelization Possible?

First step

Consider the **dual training problem**

$$\begin{array}{ll} \max\limits_{M,N} & \mathbf{1}^\top (M+N)\mathbf{1} - \frac{1}{2\alpha}\mathrm{tr}((M-N)^\top K (M-N)(I+\mathbf{1}\mathbf{1}^\top) \\ \text{s.t.} & M \geq 0,\ M\mathbf{1} \leq \mathbf{1},\ tr(M^\top \bar{Y}) = 0 \\ & N \geq 0,\ N\mathbf{1} \leq \mathbf{1},\ tr(N^\top Y) = 0 \end{array}$$

where

$$\begin{aligned} K &= \phi(X)\phi(X)^\top \\ W &= \tfrac{1}{\alpha}\phi(X)^\top (M-N) \\ \mathbf{u} &= \tfrac{1}{\alpha}\phi(X)^\top (N-M)\mathbf{1} \end{aligned}$$

Input kernelized, but still not output kernelized

Is Output Kernelization Possible?

Second step

Key idea: consider an expanded set of inner products

$$\text{Let } S = \begin{bmatrix} I \\ Y \\ \bar{Y} \end{bmatrix} \quad \begin{matrix} \textit{canonical singleton labels} \\ \textit{original labeling} \\ \textit{complement labeling} \end{matrix}$$

Use the expanded kernel matrix

$$Q \;=\; SS^\top \;=\; \begin{bmatrix} I & Y^\top & \bar{Y}^\top \\ Y & YY^\top & Y\bar{Y}^\top \\ \bar{Y} & \bar{Y}Y^\top & \bar{Y}\bar{Y}^\top \end{bmatrix}$$

Is Output Kernelization Possible?

Third step

Lemma
For any S (defined as above), $M \geq 0$ and $N \geq 0$, there must exist two matrices $\Omega \geq 0$ and $\Gamma \geq 0$ such that

$$M = \Omega S \text{ and } N = \Gamma S$$

Is Output Kernelization Possible?

Putting the pieces together

Proposition

The training problem can be equivalently re-expressed as

$$\begin{aligned}
\max_{\Omega,\Gamma} \quad & \mathbf{1}^\top(\Omega+\Gamma)Q\mathbf{1} - \tfrac{1}{2\alpha}\mathrm{tr}\left\{(\Omega-\Gamma)^\top K(\Omega-\Gamma)\left((t+1)Q + \tfrac{1}{t+1}Q\mathbf{1}\mathbf{1}^\top Q\right)\right\} \\
\text{s.t.} \quad & \Omega \geq 0,\ \Omega Q\mathbf{1} \leq (t+1)\mathbf{1},\ tr(\Omega Q_{\{:,L+t+1:L+2t\}}) = 0 \\
& \Gamma \geq 0,\ \Gamma Q\mathbf{1} \leq (t+1)\mathbf{1},\ tr(\Gamma Q_{\{:,L+1:L+t\}}) = 0
\end{aligned}$$

- Both input and output kernelized

Extension to Output Kernels

Consider output feature expansion/kernel

$$\begin{aligned} \mathbf{y} &\mapsto \boldsymbol{\psi}(\mathbf{y}) \\ \boldsymbol{\psi}(\mathbf{y})^\top \boldsymbol{\psi}(\tilde{\mathbf{y}}) &= \eta(\mathbf{y}, \tilde{\mathbf{y}}) \end{aligned}$$

Get constraints on admissible kernel matrix

- To preserve proposition, Q must be doubly non-negative ($Q \succeq 0$, and $Q \geq 0$)
- To preserve Lemma 1, Q must also preserve orthogonality (if $Y_{i:} Y_{j:}^\top = 0$ then $Q_{ij} = 0$)

Therefore constraints on admissible kernel function

- Positive semi-definiteness: $\eta(S, S) \succeq 0$ for any finite S
- Non-negativity: $\eta(\mathbf{y}, \tilde{\mathbf{y}}) \geq 0$ for all $\mathbf{y}, \tilde{\mathbf{y}} \in \{0, 1\}^L$
- Orthogonality: $\mathbf{y}^\top \tilde{\mathbf{y}} = 0$ must imply $\eta(\mathbf{y}, \tilde{\mathbf{y}}) = 0$

Extension to Output Kernels

Are there any useful admissible output kernels?

Yes, polynomial kernel

$$\eta_{poly}(\mathbf{y}, \tilde{\mathbf{y}}) \quad = \quad \sum_{d=1}^{k} \mu_d (\mathbf{y}^\top \tilde{\mathbf{y}})^d, \qquad \text{for } \boldsymbol{\mu} \geq 0$$

- Admissible
- Can capture *nonlinear* label correlations

Admissible for multi-label classification

- Note: RBF kernel is **not** admissible

Classification with Output Kernels

Given instance $\mathbf{x}$

- Compare label scores $\quad s_\ell(\mathbf{x}) = \phi(\mathbf{x})^\top W_{:\ell}$

$$= \tfrac{1}{\alpha}\kappa(\mathbf{x}, X)(\Omega - \Gamma)\eta(S, \mathbf{1}_\ell)$$

$$\text{for } \ell \in \{1, ..., L\}$$

- against threshold value $\quad s_0(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{u}$

$$= \tfrac{1}{\alpha(t+1)}\kappa(\mathbf{x}, X)(\Gamma - \Omega)Q\mathbf{1}$$

- Assign

$$y_\ell^* \quad = \quad \arg \max_{y_\ell \in \{0,1\}} y_\ell(s_\ell(\mathbf{x}) - s_0(\mathbf{x}))$$

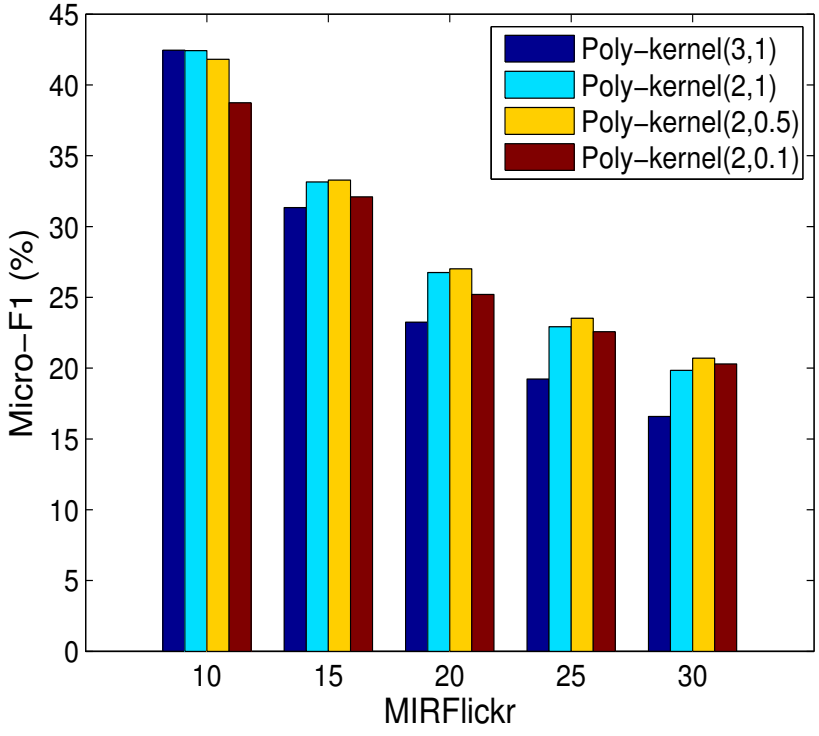
$$\text{for each label } \ell \in \{1, .., L\}$$

Experimental Results

Table : Results on a multi-label image classification problem
Top: micro-F1; Bottom: macro-F1

Data set	PairRank	M3L	CSRL	LS-K	LM-K
MIRFlickr-10	30.1±0.8	32.5±0.9	38.7±0.9	33.8±0.5	**42.4**±0.4
MIRFlickr-15	22.3±0.7	23.1±0.1	31.3±0.5	25.0±0.6	**33.2**±0.1
MIRFlickr-20	16.7±0.3	17.3±0.7	23.2±0.8	17.4±0.6	**26.8**±0.2
MIRFlickr-25	14.8±0.5	15.8±0.8	21.2±0.9	14.8±0.5	**22.9**±0.0
MIRFlickr-30	12.8±0.5	14.4±0.7	17.9±0.6	10.8±0.4	**19.8**±0.2
MIRFlickr-10	27.5±0.6	26.7±0.7	35.5±1.1	30.4±0.7	**40.5**±0.5
MIRFlickr-15	18.5±0.8	17.1±0.7	26.4±0.5	19.5±0.5	**31.3**±0.3
MIRFlickr-20	14.0±0.5	12.2±0.5	19.1±0.8	13.1±0.4	**25.5**±0.2
MIRFlickr-25	12.3±0.6	11.0±0.4	16.8±0.8	10.4±0.4	**21.7**±0.1
MIRFlickr-30	10.9±0.6	9.1±0.5	14.2±0.6	7.6±0.2	**19.1**±0.2

Experimental Results



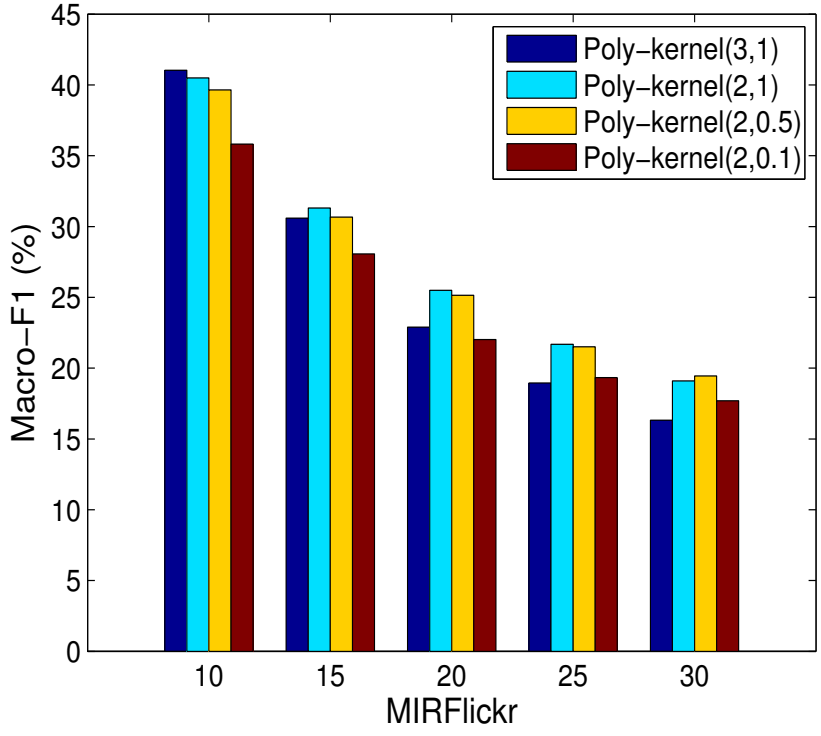


Figure : Performance of different polynomial kernels across the data sets

Open Question

When is output kernelization possible?

What losses can be equivalently expressed in terms of inner products between output vectors?

- So far, results are obtained on a case by case basis
- Almost no one is working on this

Part 2: Adapting Input-Output Kernels

Adapting Input Kernels

Multiple Kernel Learning

Can solve for optimal linear combination of kernels

- Let $\boldsymbol{\mu}$ be vector of kernel combination weights
- Add regularizer $R(\boldsymbol{\mu})$

Training problem

$$\min_{0 \leq \boldsymbol{\mu} \leq 1} \min_{\mathbf{w}} L(\Phi \Delta(\boldsymbol{\mu})^{1/2} \mathbf{w}; \mathbf{y}) + \frac{\beta}{2} \|\mathbf{w}\|_2^2 + R(\boldsymbol{\mu})$$

$$= \min_{0 \leq \boldsymbol{\mu} \leq 1} \max_{\mathbf{a}} -L^*(\mathbf{a}; \mathbf{y}) - \frac{1}{2\beta} \sum_{j=1}^{k} \mu_j \mathbf{a}^\top K_j \mathbf{a} + R(\boldsymbol{\mu})$$

- Get concave-convex program—no local minima

Manifold learning

Extension to kernelized setting

- Learning a low rank kernel $\hat{K} = \hat{X}^\top \hat{X}$ (nonlinear manifold)
- Given data $K = X^\top X$ solve

$$\min_{\hat{K} \succeq 0} L(\hat{K}; K) \text{ subject to } \operatorname{rank}(\hat{K}) = m \quad \text{(intractable)}$$

Existing approaches

- Isomap, MVU, SNE, tSNE
- Most use convex loss on $\hat{K}$ (not previously realized for SNE)
- But heuristic: – ignore rank, solve relaxed, project back
  – or just locally optimize in $\hat{X}$

New approach: convex relaxation of

$$\min_{\hat{K} \succeq 0} L(\hat{K}; K) + \alpha \lambda_{t-m}^{min}(\hat{K}) - \alpha \lambda_{m}^{max}(\hat{K})$$

Adapting Output Kernels

Convex relaxation of clustering

Use output kernels

Recall: Multiclass SVM, logistic regression output kernelizable

Convert classification to clustering

- Optimize output equivalence relation

$$\min_{N \in \{0,1\}^{t \times k},\, \boldsymbol{\delta}(N)=\mathbf{1},\, N \succcurlyeq 0,\, balanced(N)} f(N)$$

$$\geq \min_{N \geq 0,\, \boldsymbol{\delta}(N)=\mathbf{1},\, N \succcurlyeq 0,\, balanced(N)} f(N)$$

Leads to

- Large margin clustering
- Viterbi-conditional EM

Can recover a (relaxed) clustering and prediction model
jointly, globally, and tractably

Part 3: Adapting Latent Kernels

Tractable Robust Regression

Regression

Supervised recognition model

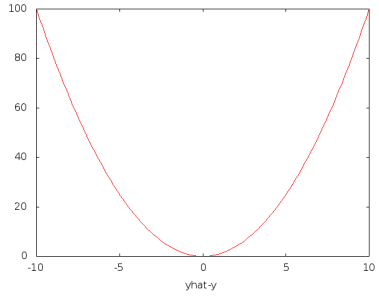
Given X and $\mathbf{y}$, recover linear predictor $\mathbf{u}$ via

$$\min_{\mathbf{u}} \frac{\alpha}{2}\|\mathbf{u}\|_2^2 + L(X\mathbf{u}, \mathbf{y})$$

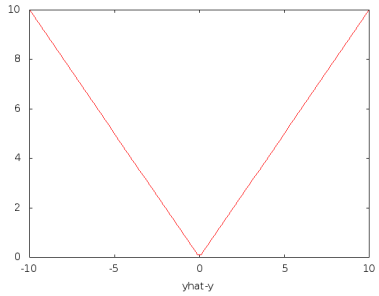
Normally, use a convex loss function L, to ensure tractability

Examples

$$L(X\mathbf{u}, \mathbf{y}) = \|X\mathbf{u} - \mathbf{y}\|_2^2$$



$$L(X\mathbf{u}, \mathbf{y}) = \|X\mathbf{u} - \mathbf{y}\|_1$$



Problem

For any convex loss

A single data point can perturb the solution arbitrarily

Demo

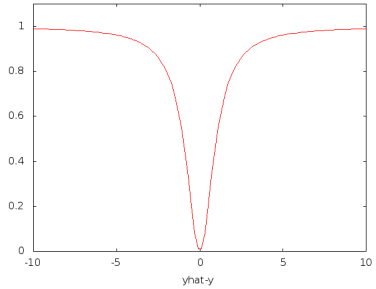
Robust regression

Standard approach: "M-estimation"

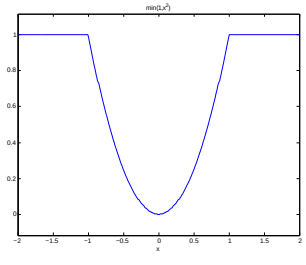
- Use a bounded loss function

Examples

$$L(X\mathbf{u}, \mathbf{y}) = \sum_i \frac{(X_{i:}\mathbf{u} - y_i)^2}{(X_{i:}\mathbf{u} - y_i)^2 + \sigma^2}$$



$$L(X\mathbf{u}, \mathbf{y}) = \sum_i \max(1, (X_{i:}\mathbf{u} - y_i)^2)$$



Problem

Minimizing any bounded (nontrivial) loss is NP-hard

Just use local minimization?

Demo

Robust regression

State of the art

Denial

Robust statistics

- propose *intractable* estimators
- prove they would work if you could have them
- bait and switch: declare satisfaction with weak heuristics
- ignore inconvenient truth: estimators will never exist

Machine learning

- assume outliers never exceed bounds
- achieve satisfaction by not realizing there's a problem

Robust regression

All we want

An estimator that is

- tractable
  - polynomial time
- robust
  - bounded influence from any one data point
- consistent
  - converges to optimal model in limit

Doesn't seem to exist!

Robust regression

Fundamental dilemma

convex loss $\Rightarrow$ tractable & not robust
bounded loss $\Rightarrow$ robust & not tractable

Can have tractability or robustness

- but apparently not both

Really?

Reformulate as representation learning

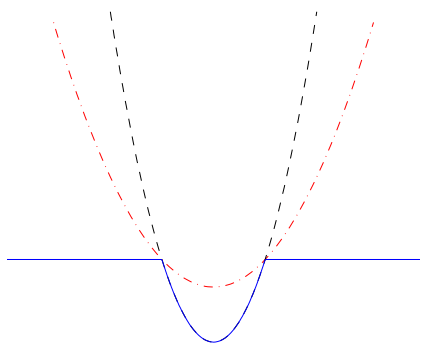
Introduce outlier indicator ϕ

- Represents a latent property of each data point
- Try to identify outliers via ϕ while optimizing model $\mathbf{u}$

$$\min_{\phi \in \{0,1\}^{t \times 1}} \min_{\mathbf{u}} \underbrace{\frac{\alpha}{2}\|\mathbf{u}\|_2^2 \|\phi\|_1}_{\text{regularize}} + \underbrace{\mathbf{1}^\top(\mathbf{1}-\phi)}_{\text{penalize outliers}} + \underbrace{\phi^\top L(X\mathbf{u}, \mathbf{y})}_{\text{loss on inliers}} \quad (1)$$

Chicken and egg problem

- given ϕ, can optimize $\mathbf{u}$
- given $\mathbf{u}$, can infer ϕ



Problem

Not jointly convex in $\mathbf{u}$ and ϕ

Note:

$$\min_{0 \le \phi_i \le 1} 1 - \phi_i + \phi_i L(X_{i:}\mathbf{u}, _i) = \max(1, L(X_{i:}\mathbf{u}, y_i))$$

Reformulate as representation learning

Reformulate

$$(1) \quad = \quad \min_{0 \le \phi \le 1} \min_{\mathbf{a}} \quad \tfrac{\alpha}{2} \|\phi\|_1 \mathbf{a}^\top K \mathbf{a} + \mathbf{1}^\top (\mathbf{1} - \phi) + \phi^\top L(K\mathbf{a}, \mathbf{y}) \quad (2)$$

$$= \quad \min_{0 \le \phi \le 1} \max_{\nu} \quad \mathbf{1}^\top (\mathbf{1} - \phi) - \phi^\top (L^*(\nu, \mathbf{y}) - \Delta(\mathbf{y})\nu)$$

$$- \tfrac{1}{2\alpha} \nu^\top \Big(K \circ \underbrace{(\phi \|\phi\|_1^{-1} \phi^\top)}_{\text{latent kernel } N} \Big) \nu \qquad (3)$$

Relax

$$N \in \mathcal{N}_\phi \quad = \quad \{N : N \succcurlyeq 0, N\mathbf{1} = \phi, \operatorname{rank}(N) = 1\}$$

$$\mathcal{M}_\phi \quad = \quad \{M : M \succcurlyeq 0, M\mathbf{1} = \phi, \operatorname{tr}(M) \le 1\}$$

$$(3) \quad \ge \quad \min_{0 \le \phi \le 1} \min_{M \in \mathcal{M}_\phi} \max_{\nu} \mathbf{1}^\top (\mathbf{1} - \phi) - \phi^\top (L^*(\nu, \mathbf{y}) - \Delta(\mathbf{y})\nu)$$

$$- \tfrac{1}{2\alpha} \nu^T (K \circ M) \nu \qquad (4)$$

Round

- Fix ϕ and re-solve for $\mathbf{a}$ in (2), recover estimator

Theorem

Under some conditions, can prove that this estimator

- is tractable
- has bounded response to any single data point
- is loss consistent

Demo

Some experiments

RMSE (stdev)

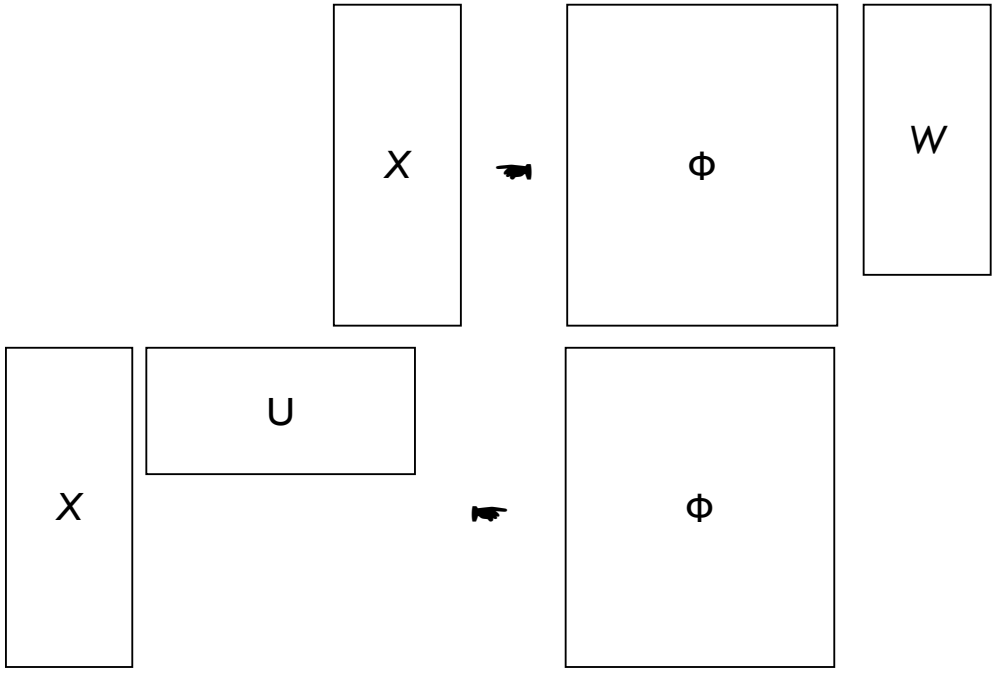
Seeded with 5% outliers

Methods	Datasets							
	cal-housing		abalone		pumadyn		bank-8fh	
L2	1185	(124.59)	7.93	(0.67)	1.24	(0.42)	18.21	(6.57)
L1	1303	(244.85)	7.30	(0.40)	1.29	(0.42)	6.54	(3.09)
Huber	1221	(119.18)	7.73	(0.49)	1.24	(0.42)	7.37	(3.18)
LTS	533	(398.92)	755.1	(126)	0.32	(0.41)	10.96	(6.67)
LocBndL2	967	(522.40)	8.39	(0.54)	0.81	(0.77)	7.74	(9.40)
CvxBndL2	9	(0.64)	7.60	(0.86)	0.07	(0.07)	0.20	(0.05)

Learning Two-layer Models

Learning two-layer models

Combined model



- two-layer neural network
- auto-encoder network

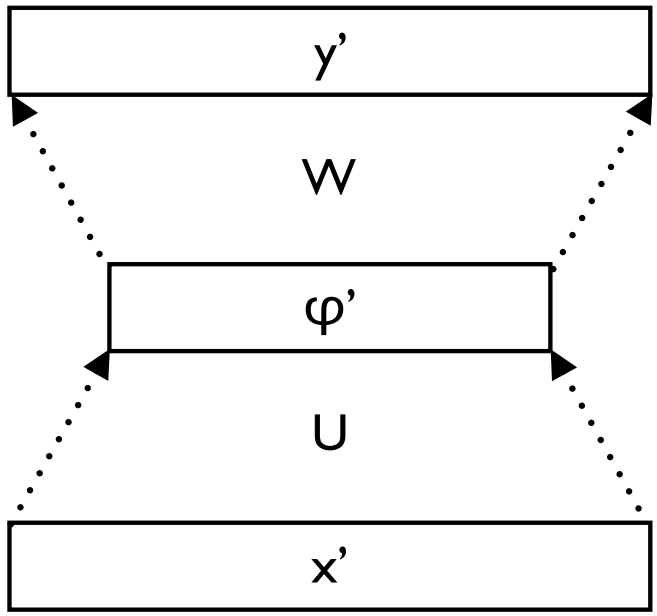
Key subproblem in local methods for "deep" learning

Does deep learning have to be intractable?

No!

Idea

- Use latent kernel trick



$$\min_{\Phi, U \in \mathcal{U}, W \in \mathcal{W}} L(XU, \Phi) + L(\Phi W, Y) + R(\Phi)$$

Example

Nested SVMs

$$\min_{\Phi \in \mathcal{C}} \quad \max_{\Omega \geq 0, \Omega \mathbf{1} = \mathbf{1}} \quad \max_{\Upsilon \geq 0, \Upsilon \mathbf{1} = \mathbf{1}}$$
$$t - \operatorname{tr}(\Omega^\top \Phi \Phi^\top) - \tfrac{1}{2\alpha} \operatorname{tr}((I - \Omega)^\top X X^\top (I - \Omega) \Phi \Phi^\top) +$$
$$t - \operatorname{tr}(\Upsilon^\top X X^\top) - \tfrac{1}{2\alpha} \operatorname{tr}((I - \Upsilon)^\top \Phi \Phi^\top (I - \Upsilon) Y Y^\top) +$$
$$\tilde{R}(\Phi \Phi^\top)$$

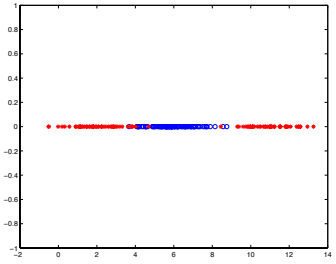
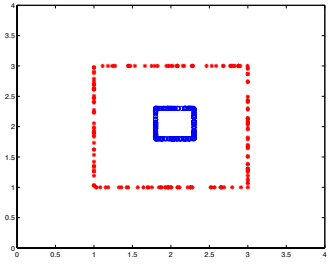
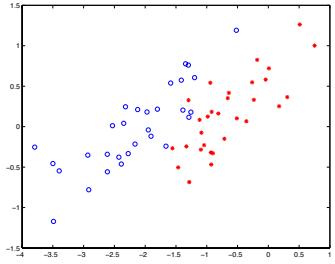
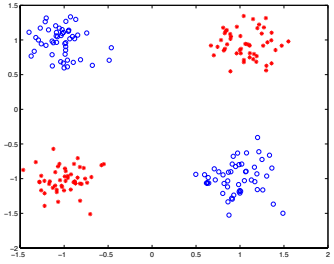
Convex reformulation

$$\min_{N \in \mathcal{M}} \quad \max_{\Omega \geq 0, \Omega \mathbf{1} = \mathbf{1}} \quad \max_{\Upsilon \geq 0, \Upsilon \mathbf{1} = \mathbf{1}}$$
$$t - \operatorname{tr}(\Omega^\top N) - \tfrac{1}{2\alpha} \operatorname{tr}((I - \Omega)^\top K (I - \Omega) N) +$$
$$t - \operatorname{tr}(\Upsilon^\top X X^\top) - \tfrac{1}{2\alpha} \operatorname{tr}((I - \Upsilon)^\top N (I - \Upsilon) Y Y^\top) +$$
$$\tilde{R}(N)$$

Learn a **latent** kernel between input and output
jointly with prediction models

Proof of Concept

Data sets

Test misclassification error %

1 layer linear SVM	51.5	29.7	40.0	50.0
2 layer linear SVM	25.5	11.6	20.0	0.0

Proof of Concept

Real data sets

- liver, vowel, MNIST

Transductive misclassification error %

data sets	liver	vowel	MNIST
1 layer linear SVM	32.9	19.0	14.1
1 layer trans SVM (SVMlight)	30.0	19.2	13.4
1 layer trans SVM (SVMlin)	29.4	20.7	12.5
2 layer linear SVM	25.8	2.5	8.5

Conclusion

Conclusion

Output kernels:

- Allow effective multi-label classification methods
- **Open question**: when is output kernelization achievable?

Latent kernels:

- Allow tractable robust regression
- Allow convex formulation of latent subclass classification

Challenges:

- Scaling ($\Theta(t^2)$ problem and *model* size)
- Understand when/if kernelization can be applied

Related possibilities

Other tractable forms of latent learning

- induced matrix norms
- moment based estimators ("spectral methods")

Still plenty of opportunity to do more!

Thanks