Deep Canonical Correlation Analysis

Galen Andrew[1] Raman Arora[2]
Jeff Bilmes[1] Karen Livescu[2]

[1]University of Washington

[2]Toyota Technological Institute at Chicago

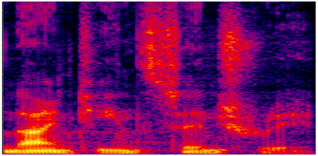
ICML, 2013

MELODI
MachinE Learning, Optimization, & Data Interpretation @ UW

TOYOTA TECHNOLOGICAL INSTITUTE CHICAGO

Outline

Data with multiple views

$x_1^{(i)}$	$x_2^{(i)}$
	
demographic properties	responses to survey
	
audio features at time i	video features at time i

Correlated representations

- CCA, KCCA, and DCCA all learn functions $f_1(x_1)$ and $f_2(x_2)$ that maximize

$$\mathrm{corr}(f_1(x_1), f_2(x_2)) = \frac{\mathrm{cov}(f_1(x_1), f_2(x_2))}{\sqrt{\mathrm{var}(f_1(x_1)) \cdot \mathrm{var}(f_2(x_2))}}$$

- Finding correlated representations can be used to
  - provide insight into the data
  - detect asynchrony in test data
  - remove noise that is uncorrelated across views
  - induce features that capture some of the information of the other view, if it is unavailable at test time
- Has been applied to problems in computer vision, speech, NLP, medicine, chemometrics, meterology, neurology, etc.

Canonical correlation analysis

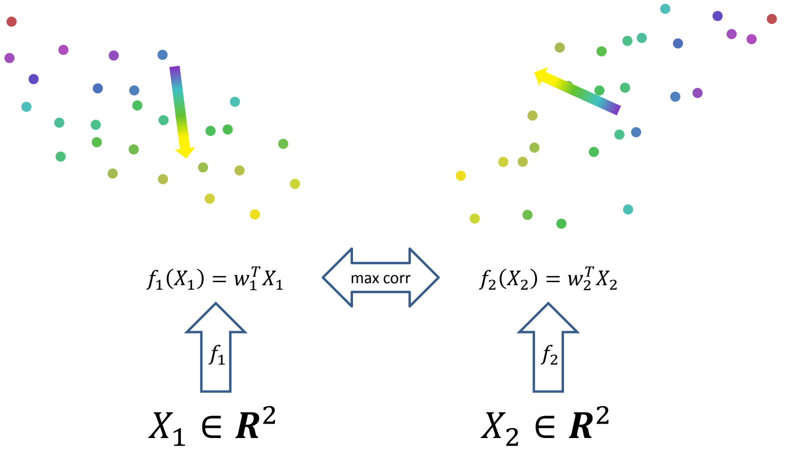
- CCA (Hotelling, 1936) is a classical technique to find linear relationships: $f_1(x_i) = W_1' x_1$ for $W_1 \in \mathbb{R}^{n_1 \times k}$ (and f_2).
- The first columns (w_1^1, w_2^1) of the matrices W_1 and W_2 are found to maximize the correlation of the projections

$$(w_1^1, w_2^1) = \operatorname*{argmax}_{w_1, w_2} \operatorname{corr}(w_1' X_1, w_2' X_2).$$

- Subsequent pairs (w_1^i, w_2^i) are constrained to be uncorrelated with previous components: For $j < i$,

$$\operatorname{corr}((w_1^i)' X_1, (w_1^j)' X_1)) = \operatorname{corr}((w_2^i)' X_2, (w_2^j)' X_2) = 0.$$

CCA Illustration



Two views of each instance have the same color

CCA: Solution

1. Estimate covariances, with regularization.

$$\Sigma_{11} = \frac{1}{m-1}\sum_{i=1}^{m}(x_1^{(i)} - \bar{x}_1)(x_1^{(i)} - \bar{x}_1)' + r_1 I \quad \text{(and } \Sigma_{22}\text{)}$$

$$\Sigma_{12} = \frac{1}{m-1}\sum_{i=1}^{m}(x_1^{(i)} - \bar{x}_1)(x_2^{(i)} - \bar{x}_2)'$$

2. Form normalized covariance matrix $T \triangleq \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2}$ and its singular value decomposition $T = UDV'$.
3. Total correlation at k is $\sum_{i=1}^{k} D_{ii}$.
4. The optimal projection matrices are

$$(W_1^*, W_2^*) = (\Sigma_{11}^{-1/2}U_k, \Sigma_{22}^{-1/2}V_k)$$

where U_k is the first k columns of U.

Finding nonlinear relationships with Kernel CCA

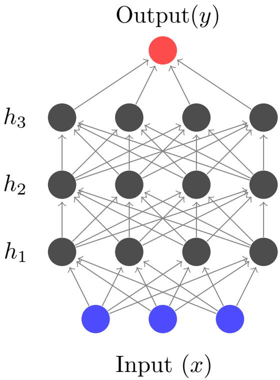
- There may be nonlinear functions f_1, f_2 that produce more highly correlated representations than linear maps.
- Kernel CCA is the principal method to detect such functions.
  - learns functions from any RKHS
  - may use different kernels for each view
- Using the RBF (Gaussian) kernel in KCCA is akin to finding sets of instances that form clusters in both views.

KCCA: Pros and Cons

- Advantages of KCCA over linear CCA
  - More complex function space can yield dramatically higher correlation with sufficient training data.
  - Can be used to produce features that improve performance of a classifier when second view is unavailable at test time (Arora & Livescu, 2013)
- Disadvantages
  - Slower to train
  - Training set must be stored and referenced at test time
  - Model is more difficult to interpret

Deep Networks

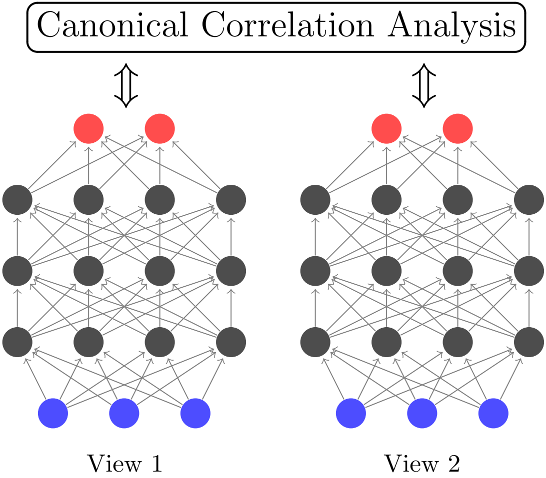
- Deep networks parametrize complex functions with many layers of transformation.
- In a typical architecture (MLP), $h_1 = \sigma(W_1'x + b_1)$, $h_2 = \sigma(W_2'h_1 + b_2)$, etc.
  - σ is nonlinear function (e.g., logistic sigmoid) applied componentwise
- Each layer detects higher-level features—well suited for tasks like vision, speech processing.



Training deep networks

- Until mid-2000s, little success with *deep* MLPs (>2 layers).
- Now, increasing performance with 10 or more layers due to pretraining methods like Contrastive Divergence, variants of autoencoders (Hinton et al. 2006, Bengio et al. 2007).
- Weights of each layer are initialized to optimize a *generative* criterion, to learn hidden layers that can in some sense reconstruct the input.
- After pretraining the network is “fine tuned” by adjusting the pretrained weights to reduce the error of the output layer.

Deep CCA



Deep CCA

- Advantages over KCCA:
  - May be better suited for natural, real-world data such as vision or audio, compared to standard kernels.
  - Parametric model
    - The training set can be discarded once parameters have been learned.
    - Computation of test representations is fast.
  - Does not require computing inner products.

Deep CCA training

- To train a DCCA model
  1. Pretrain the layers of each side individually.
     - We use denoising autoencoder pretraining in this work. (Vincent et al., 2008)
  2. Jointly fine-tune all parameters to maximize the total correlation of the output layers H_1, H_2. Requires computing correlation gradient:
     1. Forward propagate activations on both sides.
     2. Compute correlation and its gradient w.r.t. output layers.
     3. Backpropagate gradient on both sides.
- Correlation is a population objective, but typical stochastic training methods use one instance (or minibatch) at a time
  - Instead, we use L-BFGS second-order method (full-batch)

DCCA Objective Gradient

- To fine-tune all parameters via backpropagation, we need to compute the gradient $\partial \text{corr}(H_1, H_2)/\partial H_1$.
- Let $\Sigma_{11}, \Sigma_{22}, \Sigma_{12}$, and $T = \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2} = UDV'$. Then,

$$\frac{\partial \text{corr}(H_1, H_2)}{\partial H_1} = \frac{1}{m-1}\left(\nabla_{12}(H_2 - \bar{H}_2) - \nabla_{11}(H_1 - \bar{H}_1)\right)$$

where

$$\nabla_{12} = \Sigma_{11}^{-1/2} U V' \Sigma_{22}^{-1/2}$$

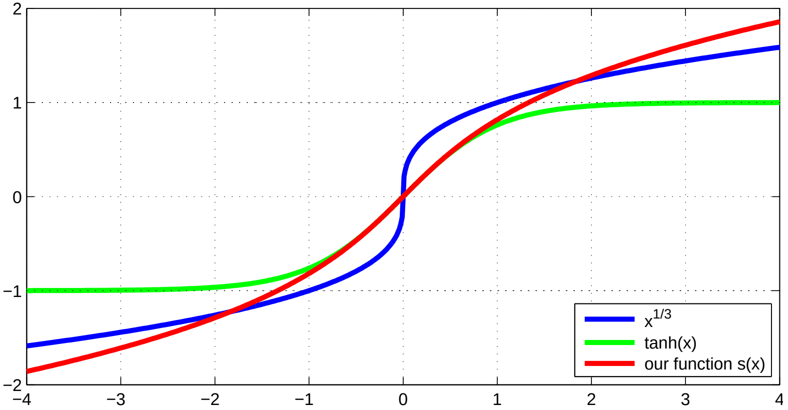
and

$$\nabla_{11} = \Sigma_{11}^{-1/2} U D U' \Sigma_{11}^{-1/2}.$$

Nonsaturating nonlinearity

- Standard, saturating sigmoid nonlinearities (logistic, tanh) sometimes cause problems for optimization (plateaus, ill-conditioning).
- We obtained better results with a novel nonsaturating sigmoid related to the cube root.

Nonsaturating nonlinearity

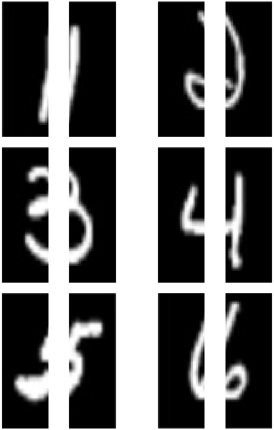


Nonsaturating nonlinearity

- If $g : \mathbb{R} \mapsto \mathbb{R}$ is the function $g(y) = y^3/3 + y$, then our function is $s(x) = g^{-1}(x)$.
- Unlike σ and $\tanh$, does not saturate, derivative decays slowly.
- Unlike cube root, differentiable at $x = 0$ (with unit slope).
- Like σ and $\tanh$, derivative is expressible in terms of function value: $s'(x) = (s^2(x) + 1)^{-1}$.
- Efficiently computable with Newton's method.

Split MNIST data

- Left and right halves of MNIST handwritten digits.
- Deep MLPs have done extremely well at MNIST digit classification.
- Two views have a high mutual information, but mostly in terms of "deeper" features than pixels.
- Each half-image is 28x14 matrix of grayscale values (392 features).
- 60k train instances, 10k test.



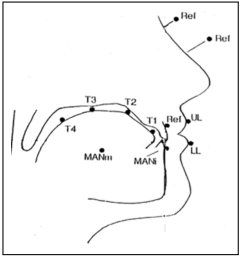
Split MNIST results

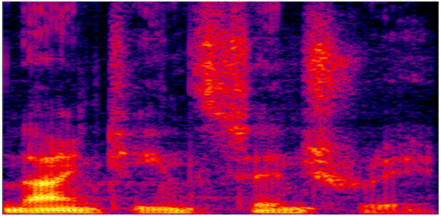
- Compare total correlation on test data after applying transformations f_1, f_2 learned by each model.
- Output dimensionality is 50 for all models.
  - Maximum possible correlation is 50.
- Hyperparameters of all models fit on random 10% of training data.
- DCCA model has two layers; hidden layer widths chosen on development set as 2038 and 1608.

	CCA	KCCA (RBF)	DCCA (50-2)	max
Dev	28.1	33.5	**39.4**	50
Test	28.0	33.0	**39.7**	50

Acoustic and articulatory views

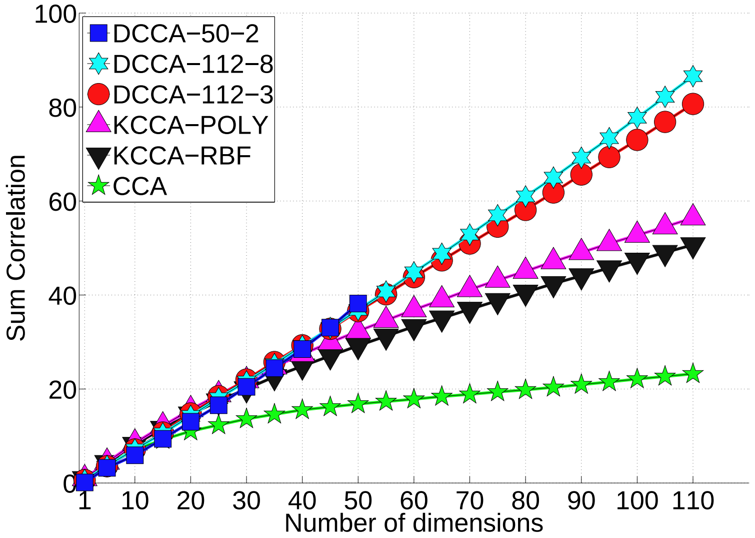
- Wisconsin XRMB database of simultaneous acoustic and articulatory recordings
  - Articulatory view: horizontal and vertical displacements of eight pellets on speaker's lips, tongue and jaws concatenated over seven frames (112 features)
  - Acoustic view: 13 MFCCs + first and second derivatives, concatenated over seven frames (273 features)





Comparing top k components

- We compare the total correlation of the top k components of each model, for all $k \leq o$ (DCCA output size).
- CCA and KCCA order components by training correlation, but the output of a DCCA model has no inherent ordering.
- To evaluate at $k < o$
  - Perform linear CCA over DCCA representations of training data to obtain linear transformations W_1, W_2.
  - Map DCCA representations of test data by W_1 and W_2, then compare total correlation of top k components.


DCCA−50−2
DCCA−112−8
DCCA−112−3
KCCA−POLY
KCCA−RBF
CCA
Sum Correlation
Number of dimensions
0
20
40
60
80
100
1
10
20
30
40
50
60
70
80
90
100
110

Correlation as a function of depth

- Explore relative contribution of depth/width
- Vary depth from three to eight layers, reducing the width to keep the total number of parameters constant
- Total correlation increases monotonically with depth, and at eight layers has still not reached saturation

layers	3	4	5	6	7	8	max
Dev set	66.7	68.1	70.1	72.5	76.0	**79.1**	112
Test set	80.4	81.9	84.0	86.1	88.5	**88.6**	112

Conclusions

- DCCA learns complex nonlinear transformations to discover latent relationships in two views of data.
- Unlike KCCA, DCCA is a parametric method.
    - does not require an inner product
    - representations of unseen instances can be computed without reference to the training set
- In experiments, DCCA finds much more highly correlated representations than KCCA or linear CCA.
- Tall skinny networks are better than short fat ones.