Summary of NIPS (2012-2015) Embedding Papers

Muthu Chidambaram

Department of Computer Science, University of Virginia

https://qdata.github.io/deep2Read/

Topic-Partitioned Multinetwork Embeddings (2012)

- Authors: Peter Krafft, Juston Moore, Bruce Desmarais, Hanna M. Wallach
- Paper proposes Bayesian admixture model for analyzing communication networks
- Focuses specifically on email networks
- Goal is to find/summarize topic-specific subnetworks in email networks

Topic Partitioned Multinetwork Encodings (2012)

- Proposes unified network identification and visualization model
- Uses latent Dirichlet allocation (LDA) to find topics for networks
  - Topics are distributions over words
- Models network substructure using latent space model (LSM)

Topic Partitioned Multinetwork Encodings (2012)

- Defines representation for emails
  - Words, Topics, Actors, Recipients
- Communication patterns: matrix of probabilities of actor x sending an email with topic t and including actor y as a recipient
- Emails are modeled as a Dirichlet distribution over topics

Topic Partitioned Multinetwork Embeddings (2012)

- Discusses sampling methods for latent variables
  - Topic assignment
- New Hanover County email dataset used as opposed to classic Enron dataset

Topic Partitioned Multinetwork Embeddings (2012)

- 30,909 total emails in dataset
- Used model to predict recipients of “test” emails
  - Compared to baseline model that employed simple global statistics (proportion of emails sent from actor X to Y)
  - Also compared model against variant of model using Beta distribution
  - Also compared to existed network models

Topic Partitioned Multinetwork Embeddings (2012)

- Model provided better link prediction results than the other models it was compared to
- Compared topic coherence from model to that of LDA, produced similar results
- Assessed fit of model using network statistics
  - Generalized graph transitivity, dyad intensity distribution, the vertex degree distribution, and the geodesic distance distribution

Topic Partitioned Multinetwork Embeddings (2012)

- Fit was assessed by applying the aforementioned functions to 1000 synthesized networks generated by predictive distribution
- The recipients of an email are more likely to be close to the author of that email in the Euclidean space of that topic

Topic Partitioned Multinetwork Embeddings (2012)

- Model was used to conduct exploratory analysis; topic-specific communication patterns were identified and visualized
- Compared generated communication patterns to actual organizational structures (i.e. groups within the county)

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Authors: Angela Eigenstetter, Bjorn Ommer
- Proposes object representation based on curvature self-similarity
- Also proposes embedded feature selection methods for SVMs
- Discusses problems with more advanced methods of object representation
  - High dimensionality

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Embedded feature selection incorporates feature selection as part of the learning process
- Doubly regularized SVM instead of L1 or L2 regularized SVM
- Introduce an additional 0-1-encoded selection vector for features and use it while searching for best kernel function

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- SVM training is split into 3 sets
  - Optimize hyperplane for fixed feature selection parameter theta
  - Parameter theta is then optimized on validation data set
    - All features are initially included, and then reduced from there
  - Then evaluated on test set

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Bundle methods: iteratively add cutting hyperplanes to build lower bound for objective function
- Self-similarity: Measures the correlation of an image patch with a larger surrounding image region
- Computes all pairwise curvature self-similarities
  - Very high dimensional representation
- Uses 360 degree orientation in order to resolve curvature ambiguities

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Image is divided into non-overlapping 8x8 pixel cells and histograms are built over curvature values in each cell
  - Concatenated with 360 degree orientation of same histogram
- Histogram intersection used to compute similarities
- Superfluous dimensions are discarded using embedded feature selection

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Embedded feature selection model is compared to same models without embedded feature selection
- Dimensionality is reduced by 55% in the linear case and 40% in the non-linear case
  - Most cases leads to increase in performance, never leads to decrease

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Curvature self-similarity tested on the PASCAL dataset
- Since model is heavily reliant on curved object contours, it was not used on images with significant amounts of noise obscuring such contours
- Tested on all objects not marked as “difficult”

Visual Recognition using Embedded Feature Selection for Curvature Self-Similarity (2012)

- Results showed that self-similarity + feature selection marginally improved performance on most tested object categories
- Conclusion: embedded feature selection is effective in both increasing performance and reducing dimensionality, curvature self-similarity adds some information to object representations

References (2012)

- http://papers.nips.cc/paper/4799-visual-recognition-using-embedded-feature-selection-for-curvature-self-similarity.pdf
- http://papers.nips.cc/paper/4659-topic-partitioned-multinetwork-embeddings.pdf

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- Authors: Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, Bart Selman
- Sampling algorithm that embeds a set into a higher-dimensional space and then projects back to a lower dimensional subspace
- Sampling is used to approximate high-dimensional probability distributions
- Extend effectiveness of systematic search techniques such as branch and bound to sampling

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- Problem definition: probability distribution P over high-dimensional data set X proportional to some weight function W
  - Want to sample p(x) given some weight function w

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- New probability distribution p’ is derived from p using discretized weight function
- p’ is used to define uniform distribution p’’ over a discretized embedding of data set X in higher dimensional space
- Indirectly sample from p by sampling uniformly from p’’
- Weight discretization is done via the use of “buckets”, or uniform discretization of log-probability

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- Project to a configuration space that is randomly constrained by a universal family of hash functions
- Add constraints until P configurations survive, then output configuration using rejection sampling

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- Details PAWS algorithm for sampling configurations
- Sampling configurations used to obtain aforementioned distribution approximation

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- Discusses accuracy guarantees by bounding
- Appropriately choosing hyperparameters can lead to discretization errors being made arbitrarily small
- PAWS was evaluated on synthetic ising models

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- Ising grid model
  - Consists of n binary variables with single-node potentials and pairwise potentials
- PAWS outperformed Gibbs sampling, belief propagation, and WISH

Embed and Project: Discrete Sampling with Universal Hashing (2013)

- PAWS use case: software verification
- Main advantage over MCMC methods is strong accuracy guarantee
- Hyperparameters in PAWS can be tuned for runtime or for accuracy

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- Authors: Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, Tomas Mikolov
- Model proposed to identify images based on labeled image data as well as semantic information from unannotated text
- Explicitly maps images to a rich semantic embedding space

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- Deep convolutional neural network with softmax struggles to generalize as number of output classes grows
- WSABIE: algorithm that explored linear mappings from image features to the embedding space
- Zero-shot learning: train a deep network for images and a parallel deep network for text, then train a linear map between the two

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- DeViSE uses skip-gram to learn word vector embeddings
- Visual model architecture is based on deep convolutional net
- DeViSE is initialized using these two models

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- The core visual model is modified to have a projection layer that maps from the visual space to the word embedding space
- Dot product similarity and hinge rank loss used for loss function
- New inputs are transformed using visual model, and then nearest labels in embedding space are found

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- Performance gap between DeViSE model and softmax baseline on hierarchical metric can be attributed to learned word embeddings
- Model has ability to make reasonable generalizations due to proximity of labels in embedded space

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- DeViSE successfully predicts a wide range of labels not included in its training set
- As the number of output labels (k) allowed (“guesses”) increases, DeViSE outperforms baseline softmax
- On more difficult datasets, DeViSE outperformed baseline softmax for all k

DeViSE: A Deep Visual-Semantic Embedding Model (2013)

- Conclusion: DeViSE model performs on par with baseline softmax for flat object classification and better for hierarchical object classification
  - Flat is just percentage classified correctly
- Displays better effectiveness in generalizing to unlearned labels
- Promising for scaling from small, fixed label object classification to much larger label sets

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Authors: Andriy Mnih, Koray Kavukcuoglu
- Word embeddings learned through log-bilinear model with noise-contrastive estimation
- Word relationship information encoded into vector embeddings
- Word space models (based on co-occurrence statistics and word count) suffer from extremely high dimensionality

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Neural probabilistic language models (NPLM) specify the distribution for a target word based on several context words
  - Typically, scoring function used to estimate compatibility between context words and target word, then fed into softmax layer
- Uses log-bilinear model instead of NPLM, which operates by performing linear prediction on the word feature space

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Model predicts target word representation by taking a linear combination of the context vectors
  - Analogous to continuous bag of words
  - Has position-dependent weights for context words
- Scoring function then used to predict similarity between generated word representation and target word vectors
  - Vector log-bilinear language model (vLBL)
- Distributional hypothesis: words with similar meanings occur in similar contexts
  - vLBL can be adapted to inverse model (ivLBL) where context words are generated from target word (like skip-gram)

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Noise-contrastive estimation (NCE): train a logistic regression classifier to discriminate between valid data points and noise within the data
- NCE avoids explicit normalization, makes training time independent of vocabulary size
- Uses global unigram distribution for noise distribution

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Model is tested using analogy question data sets
- Analogy tasks of the form a:b -> c:(guess word)
- Used April 2013 dump of Wikipedia as well as the MSR sentence completion challenge data set
- All models were trained on a single core with no regularization

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Originally used halving learning rate, but led to poor representations
- Linear learning rate produced better results, but suffers from potentially undertraining some representations due to every representation sharing the same learning rate
- Opted to use adaptive gradient descent (AdaGrad), which yielded even better results
  - Sparse data has higher learning rate under AdaGrad

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Compared ivLBL against skip-gram to measure efficacy versus tree-based (hierarchical) algorithms
- 300-dimensional ivLBL outperformed 300-dimensional skip-gram by 3-9%
  - Same model did only marginally (2-4%) worse than 1000-dimensional skip-gram trained on 4 times as much data

Learning Word Embeddings Efficiently with Noise-Contrastive Estimation (2013)

- Surprisingly, position-independent versions of LBL models outperformed position-dependent counterparts
- Even lowest-dimensional (100D) LBL model performed with an accuracy of 51% on sentence completion dataset
- Conclusion: vLBL/ivLBL with NCE is a simpler, more scalable approach that also produces more effective embeddings than tree-based methods (i.e. skip-gram)

Translating Embeddings for Modeling Multi-Relational Data (2013)

- Authors: Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, Oksana Yakhnenko
- Considers problem of embedding relationships of multi-relational data in low-dimensional vector spaces
- Multi-relational data corresponds to directed graphs with entities and edges
  - i.e. social networks
- Modeling process boils down to collecting local or global connectivity patterns between entities

Translating Embeddings for Modeling Multi-Relational Data (2013)

- Most existing methods for modeling multi-relational data exist within the framework of relational learning from latent attributes
- Models that depend on tensor/collective matrix factorization suffer from potentially overfitting, more difficult interpretations, and higher computational costs
  - Potential overfitting due to difficulty of regularization
- TransE: Energy-based model for learning low-dimensional embeddings of entities

Translating Embeddings for Modeling Multi-Relational Data (2013)

- Relationships are expressed as translations in embedding space
  - (h, l, t) -> head vector is related to tail vector by some vector that depends on the relationship l (h+l=t)
- Energy of a triplet is equal to some dissimilarity function d(h+l, t) (either the L1 or L2 norm)
- Minimize a margin-based ranking criterion over the training set using stochastic gradient descent
  - Regular triplets are compared to “corrupted” triplets generated by using a random choice of tail for a given head and relationship

Translating Embeddings for Modeling Multi-Relational Data (2013)

- Structural Embedding (SE): embeds relationships into two matrices (L1, L2) such that d(L1, h, L2, t) is minimized for proper triplets and maximized for corrupt ones
- Neural Tensor Model: learns scores of the form s(h, l, t) = h^T*L*t+l1^T*h+l2^T*t
- TransE drawback: seen as encoding 2-way interactions, could fail for higher levels of dependency

Translating Embeddings for Modeling Multi-Relational Data (2013)

- TransE is evaluated on data extracted from Wordnet and Freebase
- Wordnet: knowledge base (KB) designed to produce an intuitively usable dictionary and thesaurus
  - Model evaluated by replacing heads in triplets with entities from dictionary
  - Similar process with replacing tails
- Freebase: huge and growing KB of general facts (1.2 billion+ triplets)

Translating Embeddings for Modeling Multi-Relational Data (2013)

- Compared to simplified version of TransE that only considers data as mono-relational
- Compared to RESCAL, a collective matrix factorization model
- Also compared to similar energy-based models SE and SME (linear and bilinear)
- TransE outperformed counterparts on all metrics for link prediction

Translating Embeddings for Modeling Multi-Relational Data (2013)

- TransE performance can be attributed to relative simplicity of model allowing more effective stochastic gradient descent
- Introduction of translation term seems to make a huge impact, as TransE significantly outperforms unstructured TransE (unstructured simply clusters entities without guessing based on relationships/translations)

Translating Embeddings for Modeling Multi-Relational Data (2013)

- TransE appears to learn much faster than comparable models, as its performance on predicting new relationships from few examples improves monotonically as new samples are introduced
- Conclusion: minimally parameterized models for learning knowledge base embeddings can work very well and are highly scalable

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Authors: Le Song, Bo Dai
- Proposes a hierarchical low rank decomposition of kernel embeddings
- Key idea is to map distributions to potentially infinite feature spaces and then perform analysis using kernel operations
- Current kernel embedding algorithms fail to take advantage of low rank structure in high-dimensional data

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Evaluation of a function f on any x can be viewed as an inner product in the embedding space
- A joint density can be embedded into a tensor product feature space by taking the expected value of the tensor product of the feature maps of all of the x_i

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Kernel embeddings can be generalized to include a conditional embedding operator that outputs an embedding after taking in a variable to condition on (z)
- Kernel embedding can be viewed as a multi-linear operator of order d

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Latent variables lead to low rank kernel embeddings
  - This is due to there being a decomposition proportional to the number of indicator values of the latent variables
- Conditional independence structure is a tree
  - Each edge corresponds to a pair of latent variables
  - Low rank reshapings corresponding to each edge in tree

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Considers case where latent tree structure has caterpillar shape
  - Leads to hierarchical tensor decomposition
- Low rank representation of the kernel embedding as a set of intermediate tensors

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Algorithm deals with infinite dimensions by using kernel singular value decomposition
- The resulting set of intermediate tensors can be applied to a set of elements and expressed as kernel operations
- Decomposition of empirical embeddings may suffer from sampling error

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Bounds the difference between true kernel embeddings and low rank kernel embeddings
- Proposed decomposition is robust and still provides good approximation when latent variable tree structure is misspecified

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Low rank embeddings provide best (or on par) negative log-likelihood
- Conclusion: robust, kernel embedding algorithm based on low rank structure of the data is effective
- Drawbacks: sequence of kernel singular decompositions is not efficient

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Low rank embeddings provide best (or on par) negative log-likelihood
- Conclusion: robust, kernel embedding algorithm based on low rank structure of the data is effective
- Drawbacks: sequence of kernel singular decompositions is not efficient

Robust Low Rank Kernel Embeddings of Multivariate Distributions (2013)

- Low rank embeddings provide best (or on par) negative log-likelihood
- Conclusion: robust, kernel embedding algorithm based on low rank structure of the data is effective
- Drawbacks: sequence of kernel singular decompositions is not efficient

References (2013)

- http://papers.nips.cc/paper/5080-robust-low-rank-kernel-embeddings-of-multivariate-distributions.pdf
- http://papers.nips.cc/paper/4965-embed-and-project-discrete-sampling-with-universal-hashing.pdf
- http://papers.nips.cc/paper/5204-devise-a-deep-visual-semantic-embedding-model.pdf
- http://papers.nips.cc/paper/5165-learning-word-embeddings-efficiently-with-noise-contrastive-estimation.pdf
- http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Authors: Sung Ju Hwang, Leonid Sigal
- Proposes a unified model for semantics in which semantic entities and super-categories are embedded in the same space
- Object categorization: drawing boundaries between objects in continuous space

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- A category can be represented as its super-category + its category-specific modifiers (attributes)
- Paper attempts to link category hierarchies and attributes
- Embedding-based methods perform classification in lower-dimensional shared embedding space

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Translate images and labels to same embedding space such that the similarity (S(image, label)) is maximized for labeled images
- Use large-margin constraints on distances between vectors in order to ensure that image is closest to its assigned label

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Data instances are mapped to be closer to their super category mappings than their sibling instance mappings
- Attributes can be considered as making up the basis of a semantic space representing a category
- Object class can be represented as its parent class plus a sparse combination of attributes

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Uses stochastic subgradient method to optimize embedding models
- Optimize data embedding and category embedding separately until convergence
- Method validated on 2 different data sets generated from publicly available image data

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Non-semantic baselines: ridge regression, nearest mean classifier, largest margin embedding
- Implicit semantic baselines: LMTE, ALE, HLE, AHLE
- Outperforms baselines in almost all categories
  - Performed slightly worse than AHLE and HLE in hierarchical precision (at k = 5)

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Main advantage of method is ability to generate compact, semantic expressions for each category
  - Allows for human-readable explanations
- Rich generated descriptions should lead to better performance on few-shot learning (learning from very few data samples)

A Unified Semantic Embedding: Relating Taxonomies and Attributes (2014)

- Outlined method has greatest increase in performance as number of samples increase in few-shot learning
- Conclusion: proposed model has better flat-line performance and similar or better hierarchical precision in addition to semantically meaningful decompositions

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Authors: Andrej Karpathy, Armand Joulin, Fei Fei F. Li
- Model that embeds fragments of images and fragments of sentences into a common space
- Formulates a max-margin objective for a deep neural network that learns to embed both image and sentence fragments into a common, multimodal space

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Typical multimodal representation methods reason at the global level with a representation for the entire image/sentence
  - Proposed model reasons about objects that make up a more complex image/sentence
- Neural network that connects image pixels to 1-of-k word representations
- Inner product between vector representations is fragment compatibility score

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Sentence fragments extracted from dependency tree structure as opposed to n-grams
  - Edge in tree represents fragment
  - Words encoded as 1-of-k vectors from 400,000 word dictionary
- Sub-objects are detected in an image using region convolutional neural networks (RCNN)

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Objective function defined as sum of global ranking objective, fragment alignment objective, and a regularization term
- Fragment alignment objective encodes correspondence between sentence fragments and image fragments within sentence and image being considered
  - Assumes dense alignment between all pairs of fragments
- Multiple instance learning extension: infers latent alignment between fragments using negative sampling approach

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Global ranking objective: image-sentence alignment score is defined to be average threshold score of pairwise fragment scores
  - positive scores - correct alignments, negative scores - incorrect alignments
- Optimized using stochastic gradient descent with a mini-batch size of 100

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Image-sentence retrieval performance evaluated on Pascal1K, Flickr8K, and Flickr30K datasets
- Use Stanford CoreNLP processor to compute dependecy trees for sentences
- Used Caffe implementation of ImageNet RCNN model for image detection
- Compared against SDT-RNN and DeViSE

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Breaking down images into image fragments improves performance
- Dependency tree relations outperform continuous bag of words/n-grams
- Fine-tuning/preventing overfitting of the CNN improves performance on Flickr30K dataset

Deep Fragment Embeddings for Bidirectional Image Sentence Mapping (2014)

- Limitations: Edges from sentence dependency tree may oversimplify relationships, and RCNN can sometimes spuriously detect a single object as multiple ones
- Conclusion: Provided model improves upon previously proposed bidirectional image-sentence models and provides interpretable predictions

Subspace Embeddings for Polynomial Kernel (2014)

- Authors: Haim Avron, Huy L. Nguyen, and David P. Woodruff
- Proposes fast oblivious subspace embedding
  - Embed a space induced by a non-linear kernel without explicitly mapping the data to the high-dimensional space
- Oblivious subspace embedding: data-independent random transformation that produces an approximate isometry over the embedding space

Subspace Embeddings for Polynomial Kernel (2014)

- Kernel of the form k(x, y) = (<x, y>+c)^p
- Proposed transformation generates an approximate isometry and is data-independent
  - Brings together the characteristics of Kernel PCA and Random Fourier Features
- Transformation can be used to speed up various learning algorithms that employ polynomial kernels

Subspace Embeddings for Polynomial Kernel (2014)

- Prior work: TensorSketch and CountSketch
  - TensorSketch combines CountSketch with Fast Fourier Transform and can be used for statistical learning with a polynomial kernel
- Previous sketch methods do not provide provable guarantees for preserving entire subspace
- Similar work: random feature maps that approximate kernel function with inner product
  - More like a heuristic, hard to relate to metrics such as generalization error

Subspace Embeddings for Polynomial Kernel (2014)

- Prior work: TensorSketch and CountSketch
  - TensorSketch combines CountSketch with Fast Fourier Transform and can be used for statistical learning with a polynomial kernel
- Previous sketch methods do not provide provable guarantees for preserving entire subspace
- Similar work: random feature maps that approximate kernel function with inner product
  - More like a heuristic, hard to relate to metrics such as generalization error

Subspace Embeddings for Polynomial Kernel (2014)

- CountSketch is specified by a 2-wise independent hash function and a 2-wise independent sign function
- TensorSketch (for polynomial of order q) is specified by q 3-wise independent hash functions and q 4-wise independent sign functions
- TensorSketch is an oblivious subspace embedding

Subspace Embeddings for Polynomial Kernel (2014)

- Describes K-space algorithm that employs independent TensorSketches to construct oblivious subspace embedding
- Proves probabilistic bounds for approximate isometry
- Applications to approximate kernel PCA and low rank approximation

Subspace Embeddings for Polynomial Kernel (2014)

- Bounds complexity for computing embedding
- Proposes regularization via rank-k approximations to input matrix
- Methods that can be regularized using this approach:
  - Approximate kernel principal component regression
  - Approximate kernel canonical correlation analysis

Subspace Embeddings for Polynomial Kernel (2014)

- Compare ordinary l2 regression to approximate principal component l2 regression
- Experimented with feature extraction using only a subset of the training data in order to deal with k-space overhead
- K-space produces higher quality features than simply using TensorSketch

Subspace Embeddings for Polynomial Kernel (2014)

- Compare ordinary l2 regression to approximate principal component l2 regression
- Experimented with feature extraction using only a subset of the training data in order to deal with k-space overhead
- K-space produces higher quality features than simply using TensorSketch

Subspace Embeddings for Polynomial Kernel (2014)

- Compare ordinary l2 regression to approximate principal component l2 regression
- Experimented with feature extraction using only a subset of the training data in order to deal with k-space overhead
- K-space produces higher quality features than simply using TensorSketch

Subspace Embeddings for Polynomial Kernel (2014)

- Conclusion: paper describes first oblivious subspace embedding for non-linear kernel
- Proposes next step of designing oblivious subspace embeddings for non-finite kernels (i.e. expansion induced by Gaussian kernel)

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Authors: Omer Levy, Yoav Goldberg
- Proposes method using sparse shifted positive PMI word-context matrix
- Analyzes and attempts to broaden understanding of neural network based word embeddings
  - Specifically skip-gram with negative sampling

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Skip-Gram: Trains word embedding based on word-context pairs using neural network
- Negative Sampling: Maximize probability of observed word-context pairs and maximize complementary probability for random, “negative” samples of word-context pairs

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Views skim-gram with negative sampling as implicitly factoring some matrix M into word and context matrices
- Shows how optimization in skip-gram with negative sampling leads to factoring a shifted pointwise mutual information (PMI) matrix

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Casts the objective of skip-gram with negative sampling as a weighted matrix factorization problem
- Pointwise mutual information measures the association between a pair of discrete outcomes
- In this case, PMI is used to measure association between word and context

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Resulting matrices can be dense => can be made sparse using positive PMI (PPMI) metric
- Shifted PPMI derived from skip-gram with negative sampling objective function can be used to compute word embeddings
- Additional alternative: Singular Value Decomposition (SVD)

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Typical SVD-based factorization yields word and context matrices with very different properties
  - Proposes symmetric SVD factorization, which appears to work better empirically
- SVD versus skip-gram with negative sampling
  - SVD does not require learning rates/hyper-parameter tuning
  - Skip-gram with negative sampling differentiates between observed and unobserved events, SVD does not
  - Middle-ground: Stochastic Matrix Factorization (SMF)

Neural Word Embedding as Implicit Matrix Factorization (2014)

- All models being compared were trained on English Wikipedia
- Observed that sparse positive PMI (SPPMI) was a near-perfect approximation of the optimal solution computed via gradient descent
- Evaluated resulting word representations on word similarity and analogy datasets

Neural Word Embedding as Implicit Matrix Factorization (2014)

- Conclusion: SPPMI is a significant improvement on PMI methods for approximating skip-gram with negative sampling objective
  - Does not necessarily out-perform skip-gram, possibly due to over-weighting rare words
- SVD performed poorly in approximating objective, but had good performance on word similarity tasks
- Future work: Investigate weighted matrix factorizations of word-context matrices with PMI-based association matrices

References (2014)

- http://papers.nips.cc/paper/5289-a-unified-semantic-embedding-relating-taxonomies-and-attributes.pdf
- http://papers.nips.cc/paper/5281-deep-fragment-embeddings-for-bidirectional-image-sentence-mapping.pdf
- http://papers.nips.cc/paper/5240-subspace-embeddings-for-the-polynomial-kernel.pdf
- http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf

Space-Time Local Embeddings (2015)

- Authors: Ke Sun, Jun Wang, Alexandros Kalousis, Stephane Marchand-Maillet
- Proposes space-time representation as an alternative to traditional Euclidean space representation (Embedding in Minkowski space)
- Drawbacks of typical R^n embedding
  - Limited number of points that share a nearest neighbor
  - Hard to model pairwise similarities
  - Must admit transitive relationships (neighbor’s neighbor)

Space-Time Local Embeddings (2015)

- Defines space-time metric which has space component as first D_s elements of diagonal (identity) and time component as last D_t elements of diagonal (negative identity)
- Allows for the definition of a space-time interval between points to be a difference of sums of squared differences
  - Point is an “event”
- Is not necessarily transitive, as desired

Space-Time Local Embeddings (2015)

- Defines a linear mapping from a set of Gram matrices to a set of square distance matrices
- Events in space-time as well as their intervals are in the set of mapped square distance matrices
- Regular pairwise distance matrices in Euclidean space disregard directional information

Space-Time Local Embeddings (2015)

- Space-time embedding can represent any square matrix of positive pointwise similarities
- Similarities can be represented as accurately as desired in either space or space-time models, no reason to favor space-only
- Project a similarity matrix to a set of space-time events

Space-Time Local Embeddings (2015)

- Finds optimal embedding from similarity matrix by minimizing Kullback-Leibler divergence between input matrix p and output matrix Y(p)
  - Computed via stochastic gradient descent
- Compares SNE, t-SNE, and proposed method SNE^st (where SNE is stochastic neighbor embedding)
- Datasets: SCHOOL, NIPS22 (author-document matrix), GrQc (Arxiv co-authorship), W5000 (semantic similarities between 5000 english words)

Space-Time Local Embeddings (2015)

- Table 1: KL divergence of different embeddings. After repeated runs on different configurations for each embedding, the minimal KL that we have achieved within 5000 epochs is shown. The bold numbers show the winners among SNE, t-SNE and SNEST using the same number of parameters.

	SCHOOL	NIPS17	NIPS22	GrQc	W1000	W5000
SNE $\rightarrow \Re^2$	0.52	1.88	2.98	3.19	3.67	4.93
SNE $\rightarrow \Re^3$	0.36	0.85	1.79	1.82	3.20	4.42
SNE $\rightarrow \Re^4$	**0.19**	**0.35**	1.01	1.03	2.76	3.93
t-SNE $\rightarrow \Re^2$	0.61	**0.88**	**1.29**	**1.24**	**2.15**	**3.00**
t-SNE $\rightarrow \Re^3$	0.58	0.85	1.23	1.14	2.00	2.79
t-SNE $\rightarrow \Re^4$	0.58	0.84	1.22	1.11	1.96	2.74
SNEST $\rightarrow \Re^{1,1}$	**0.43**	0.91	1.62	2.34	2.59	3.64
SNEST $\rightarrow \Re^{2,1}$	**0.31**	**0.60**	**0.97**	**1.00**	**1.92**	**2.57**
SNEST $\rightarrow \Re^{3,1}$	0.29	0.54	**0.93**	**0.88**	**1.79**	**2.39**

Space-Time Local Embeddings (2015)

- Data visualization of proposed method: visually nearby points are similar, despite the introduction of time dimension
- If an input is larger than what can faithfully be modeled in a space-only model, it is pushed to a different time
  - Embedded points with large absolute times represent important points

Space-Time Local Embeddings (2015)

- Conclusion: Using the same number of dimensions, certain input data is better preserved using space-time embedding
- Proposed method is learning on a sub-manifold
- Future methods could employ different projections to sub-manifolds
  - Alternative measures to KL-divergence

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Authors:Miguel A. Carreira-Perpinan, Max Vladymyrov
- Uses auxiliary coordinates to alternate training of parametric input mapping and auxiliary embedding
- Given a high-dimensional dataset of N points, algorithm attempts to find low-dimensional projections
  - Similar algorithms: SNE, t-SNE, NeRV

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Optimizing nonlinear embeddings is difficult
  - There are many parameters
  - Objective function is nonconvex; gradient descent may require more iterations
  - Has a quadratic number of terms, so evaluating gradient can be slow
- Parametric embedding: restrict embeddings to only those realizable by a family of fast parametric mappings
- Paper focus: optimizing an unsupervised parametric embedding defined by a given objective and a given family of mapping

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Parametric embedding objective function: embedding objective evaluated on mappings of points to lower-dimensional space
  - Can thus be considered as a combination of embedding objective and mapping family
- Parametric embedding typically worsens free embeddings, with more powerful mapping families (i.e. neural nets vs. linear) being more effective at not worsening embeddings

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Parametric embedding can be optimized via computing gradient with respect to mapping parameters
  - Difficult to implement
  - Slow in practice, due to quadratic complexity
- Parametric embedding can be viewed as a nested function, where we apply F (mapping) and then E (objective)
- Nested problem can be represented as equivalent, constrained optimization problem by introducing auxiliary coordinates for each input pattern

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Optimize parametric embedding by alternating optimization between mapping and associated auxiliary coordinates
- Given auxiliary coordinates Z, we can optimize the family of mappings F over Z by least-squares regression
- Given a mapping family F, we can optimize auxiliary coordinates Z using regularized embedding
- Despite introducing new parameters, algorithm does not increase time complexity
- Allows for easy extension of existing N-body methods, due to not explicitly using chain-rule gradients

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Experiments compare proposed algorithm to conventional optimization based on chain-rule gradients for various embedding objectives and mapping families
- Uses COIL-20 dataset, which contains rotations of 20 physical objects
- Goal of experiment is to show that it is easy to derive convergent, efficient algorithms for various combinations of embeddings and mappings (universality)

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Applied proposed algorithm (MAC) to learn parametric embeddings for MNIST dataset of 60000 images
- Nonlinear (free) embedding on a dataset of this was previously very slow
  - MAC algorithm employing previous work on N-body methods allowed significant speed-up

A Fast, Universal Algorithm to Learn Parametric Nonlinear Embeddings (2015)

- Conclusion: using auxiliary coordinates to learn parametric embeddings simplifies algorithm development without sacrificing embedding quality
  - Particularly useful when able to employ high specialized/optimized work such as N-body methods
- Additionally, MAC can be quite faster than chain-rule gradient based optimization

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Authors: Dinesh Ramaswamy, Upamanyu Madhow
- Singular Value Decomposition (SVD), which is typically used for preprocessing/dimensionality reduction, becomes a bottleneck as problem size increases
- Sidesteps SVD by focusing on pairwise similarity metrics
- Problem setup: given an input matrix, we wish to compute a transformation on the rows of the matrix that succinctly describes the global structure of the matrix via similarity metrics

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Typically, a partial SVD is used to embed the rows of the matrix into a lower-dimensional space
  - Use of partial SVD can also have added benefit of “denoising” the data
- Bottlenecks result due to increasing number of singular vectors required to capture structure of MxN matrix as size increases
- Paper focuses on computing embedding that captures pairwise similarity metrics while sidestepping SVD

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Related work in exact/approximate SVD approximation attempts to minimize matrix reconstruction error
  - Approximation methods (i.e. Nystrom method) place constraints on computational budgets
- Algorithms that sidestep SVD computation are specialized
  - Graphs can be embedded based on diffusion of probability mass in random walks over the graph (specialized for probability transition matrices)
- Paper proposes framework for sidestepping SVD computation, first with NxN input matrix and then generalizes to MxN matrices

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Algorithm uses compressive embedding of N dimensions to O(LogN) dimensions, since only interested in pairwise similarity measures
- Approximates embedding function using polynomial in order to compute efficiently
- Proves performance bound for computing embedding function using polynomial approximation

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Algorithm uses compressive embedding of N dimensions to O(LogN) dimensions, since only interested in pairwise similarity measures
- Approximates embedding function using polynomial in order to compute efficiently
- Proves performance bound for computing embedding function using polynomial approximation

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Polynomial approximations are measured via spectral norm
  - Equivalent to seeing how well the function can be approximated at eigenvalues
- Overcomes inefficiency of computing eigenvalues by considering a uniform distribution of eigenvalues and then minimizing average error over the distribution
- Polynomials are then generated via Legendre polynomials

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Generalizes approach from symmetric NxN matrix to any MxN matrix (with determinant less than 1) by considering (M+N)x(M+N) matrix
  - First M rows correspond to rows of MxN matrix, last N rows correspond to columns of MxN matrix
- Implementation considerations: use of spectral norm, use of polynomial approximations, use of legendre polynomials, denoising by cascading

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Exact embedding compared to compressive embedding generated by algorithm
- Considers real-world undirected graph data
  - DBLP Collaboration Network
  - Amazon Co-purchasing Network
- Significant time improvements, at the cost of some inference quality
- Yields better clustering quality due to being able to more concisely capture eigenvectors

Compressive Spectral Embedding: Sidestepping the SVD (2015)

- Conclusion: combination of random projections and polynomial expansions is effective in approximating pairwise distances in a spectral embedding
- Method can be used to approximate spectral embeddings dependent on entire SVD (as opposed to just partial), as it is independent of number of dominant vectors used in model
- Future work concerning extending method to improving downstream inference tasks in various large-scale problems

Sparse Local Embeddings for Multi-label Classification (2015)

- Authors: Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, Prateek Jain
- Motivation: leading embedding approaches have been unable to deliver high prediction accuracies or scale to large datasets
- Proposes SLEEC (Sparse Local Embeddings for Extreme Classification), which learns a small ensemble of local distance preserving embeddings
- Problem (XML): creating a classifier that can accurately tag a point with the most relevant subset of labels

Sparse Local Embeddings for Multi-label Classification (2015)

- 1-vs-All techniques (classifier per label) infeasible for XML (extreme multi-label learning) problem due to the sheer number of labels
- Standard approach is to reduce label dimensionality by low rank embedding
  - Assumes that the training-label matrix is low rank
  - Slow at training and prediction, even for small label sets
- SLEEC learns non-linear embeddings that preserve pairwise distances between only the closest label vectors
  - Prediction is done via kNN instead of matrix decompression

Sparse Local Embeddings for Multi-label Classification (2015)

- SLEEC clusters training data into C clusters and learns an embedding per cluster, so as to perform a localized kNN
  - Tackles instability of high-dimensional clustering by learning a small ensemble of embeddings for randomized clusters
  - Significant speedup, can be attributed to corresponding embeddings
  - Outperforms tree-based methods
- On WikiLSHTC, had a 55% classification accuracy with an 8 ms prediction time vs. LEML's 20% classification accuracy on 300 ms

Sparse Local Embeddings for Multi-label Classification (2015)

- Label matrix Y cannot be well-approximated using a low-dimensional linear subspace
  - However, can be approximated using a low-dimensional non-linear manifold
- Focus on preserving distances only between closest neighbors
  - Modify objective function to consider set of neighbors to be preserved
  - Adds L1 and L2 regularizations to maintain sparse embeddings

Sparse Local Embeddings for Multi-label Classification (2015)

- Optimization is split into two phases: learning embeddings and then learning regressors
  - Regressors are used to predict embeddings using input features
- Uses Singular Value Projection (SVP) to perform optimization
- Defines a loss function whose optima yields the neighborhood selection criteria

Sparse Local Embeddings for Multi-label Classification (2015)

- Proves that generalization error bound is independent of dimension of input/label spaces
- Train method over local clusters so as to combat high-dimensionality of label space
  - Even with clustering into homogenous regions, the data is still not low-rank
- Using ensembles of classifiers based on different clusterings yields significantly better performance (deals with instability of clustering)

Sparse Local Embeddings for Multi-label Classification (2015)

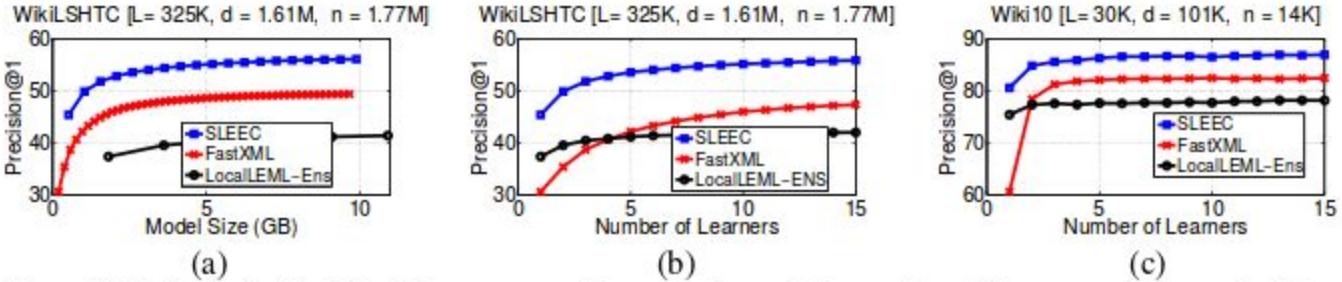

Figure 2: Variation in Precision@1 accuracy with model size and the number of learners on large-scale data sets. Clearly, SLEEC achieves better accuracy than FastXML and LocalLEML-Ensemble at every point of the curve. For WikiLSTHC, SLEEC with a single learner is more accurate than LocalLEML-Ensemble with even 15 learners. Similarly, SLEEC with 2 learners achieves more accuracy than FastXML with 50 learners.

- Datasets used: Ads1M, Amazon, WikiLSHTC, DeliciousLarge, Wiki10
  - Also uses smaller datasets due to inability to scale other techniques for comparison

Sparse Local Embeddings for Multi-label Classification (2015)

Table 1: **Precision Accuracies** (a) Large-scale data sets : Our proposed method SLEEC is as much as 35% more accurate in terms of P@1 and 22% in terms of P@5 than LEML, a leading embedding method. Other embedding based methods do not scale to the large-scale data sets; we compare against them on small-scale data sets in Table 3. SLEEC is also 6% more accurate (w.r.t. P@1 and P@5) than FastXML, a state-of-the-art tree method. '-' indicates LEML could not be run with the standard resources. (b) Small-scale data sets : SLEEC consistently outperforms state of the art approaches. WSABIE, which also uses kNN classifier on its embeddings is significantly less accurate than SLEEC on all the data sets, showing the superiority of our embedding learning algorithm.

(a)

Data set		SLEEC	LEML	FastXML	LPSR-NB
Wiki10	P@1	**85.54**	73.50	82.56	72.71
	P@3	**73.59**	62.38	66.67	58.51
	P@5	**63.10**	54.30	56.70	49.40
Delicious-Large	P@1	**47.03**	40.30	42.81	18.59
	P@3	**41.67**	37.76	38.76	15.43
	P@5	**38.88**	36.66	36.34	14.07
WikiLSHTC	P@1	**55.57**	19.82	49.35	27.43
	P@3	**33.84**	11.43	32.69	16.38
	P@5	**24.07**	8.39	24.03	12.01
Amazon	P@1	**35.05**	8.13	33.36	28.65
	P@3	**31.25**	6.83	29.30	24.88
	P@5	**28.56**	6.03	26.12	22.37
Ads-1m	P@1	21.84	-	**23.11**	17.08
	P@3	**14.30**	-	13.86	11.38
	P@5	**11.01**	-	10.12	8.83

(b)

Data set		SLEEC	LEML	FastXML	WSABIE	OneVsAll
BibTex	P@1	**65.57**	62.53	63.73	54.77	61.83
	P@3	**40.02**	38.4	39.00	32.38	36.44
	P@5	**29.30**	28.21	28.54	23.98	26.46
Delicious	P@1	68.42	65.66	**69.44**	64.12	65.01
	P@3	61.83	60.54	**63.62**	58.13	58.90
	P@5	56.80	56.08	**59.10**	53.64	53.26
MediaMill	P@1	**87.09**	84.00	84.24	81.29	83.57
	P@3	**72.44**	67.19	67.39	64.74	65.50
	P@5	**58.45**	52.80	53.14	49.82	48.57
EurLEX	P@1	**80.17**	61.28	68.69	70.87	74.96
	P@3	**65.39**	48.66	57.73	56.62	62.92
	P@5	**53.75**	39.91	48.00	46.2	53.42

Sparse Local Embeddings for Multi-label Classification (2015)

- Conclusion: SLEEC scales better than all other compared embedding methods, and has better or comparable performance to all methods
  - Only real competition is FastXML (tree-based methods)

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Authors: Rie Johnson, Tong Zhang
- Proposes semi-supervised network with convolutional neural networks for text categorization
- Method learns embeddings of small text regions as opposed to words
  - Where ‘embedding’ signifies structure-preserving function

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Framework learns a region embedding from unlabeled data, which is used as input to a supervised CNN
- Learns tv-embeddings: two-view embeddings based on predicting context of unlabeled data
  - Goal is to learn tv-embeddings specific to tasks of interest, as opposed to general word embeddings
  - Map text regions to high-level concepts relevant to the task
- Preliminary work: bag-of-words CNN
  - Co-presence and absence of words to produce predictive features

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Procedure: learn embeddings of unlabeled data and then use embeddings in supervised training
- F is a tv-embedding of X1 with respect to X2 is there exists G such that P(X2 | X1) = G(F(X1), X2))
  - Preserves everything required from a view (X1) to predict another view (X2)
- Focuses on nonlinear learning of region embedding
- First train neural network to produce tv-embeddings, then integrate tv-embeddings into base CNN

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Considers sub-task of labeling to individual text regions, and then builds predictions based on these regions
- Unsupervised target representation encodes target/context as BOW vectors of regions to the left and right of X1 (considered region) with vocabulary-control
- Partially supervised target representation uses CNN trained on labeled data to produce context vectors

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Tv-embeddings are used as additional input to base CNN’s convolutional layer to generate a supervised embedding
- Datasets used: IMDB, Elec, RCV1
- Implementation: Used one-layer CNN models as base, and then fed in tv-embeddings
- Tuning of meta-parameters was done via cross-validation

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

				IMDB	Elec	RCV1
1	linear SVM with 1-3grams [11]			10.14	9.16	10.68
2	linear TSVM with 1-3grams			9.99	16.41	10.77
3	[13]'s CNN			9.17	8.03	10.44
4	One-hot CNN (simple) [11]			8.39	7.64	9.17
5	One-hot CNN (simple) co-training best			(8.06)	(7.63)	(8.73)
6	Our CNN	unsup-tv.	100-dim	7.12	6.96	8.10
7			200-dim	6.81	6.69	7.97
8		parsup-tv.	100-dim	7.12	6.58	8.19
9			200-dim	7.13	6.57	7.99
10		unsup3-tv.	100-dim	7.05	6.66	8.13
11			200-dim	6.96	6.84	8.02
12		all three	100×3	**6.51**	**6.27**	**7.71**

Table 3: Error rates (%). For comparison, all the CNN models were constrained to have 1000 neurons. The parentheses around the error rates indicate that co-training meta-parameters were tuned on test data.

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Results: region tv-embeddings shown to be more effective than simply manipulating word vectors
- Compared tv-embeddings to traditional word vector concatenation and word vector average embeddings
  - Regional embedding learns co-presence and absence effectively, is more expressive than other embeddings for similar tasks

Semi-supervised Convolutional Networks for Text Categorization via Region Embedding (2015)

- Conclusion: Region embeddings capture more information than general word embeddings, which can be isolated
- Models based on proposed method employing tv-embeddings had better performance on sentiment and topic classification tasks when compared to previous best models

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Authors: Frederik D. Johansson, Ankani Chattoraj, Chiranjib Bhattacharyya, Devdatt Dubhashi
- Proposes a unifying generalization of the Lovasz theta function, and the associated geometric embedding
  - Extends from unweighted graphs to graphs with weighted edges and nodes

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Exploits connection between SVM and theta function to produce equivalent characterization of Delsarte version of Lovasz number
  - Kernel characterization used as approximation to semidefinite program formulation of theta function
- Resulting weighted theta function is a measure of diversity in graphs
- Key observation: set of orthonormal representations (embeddings) is equivalent to the set of kernels K

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Delsarte version of the Lovasz number relaxes orthogonality constraint; introduces obtuse labelings
  - Non-adjacent nodes correspond to vectors meeting at obtuse angles on the unit sphere
- Extends Delsarte version of Lovasz by introducing weight vector into optimization
  - Taking uniform edge weights of 1 reduces to original formulation for unweighted graphs (strict generalization)

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

Table 1: Characterizations of weighted theta functions. In the first row are characterizations following the original definition. In the second are kernel characterizations. The bottom row are versions of the LS-labelling [14]. In all cases, $\|\mathbf{u}_i\| = \|\mathbf{c}\| = 1$. A refers to the adjacency matrix of G.

Unweighted	**Node-weighted**	**Edge-weighted**
$\min_{\{\mathbf{u}_i\}} \min_{\mathbf{c}} \max_{i} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2}$ $\mathbf{u}_i^\top \mathbf{u}_j \leq 0,\ \forall (i,j) \notin E$	$\min_{\{\mathbf{u}_i\}} \min_{\mathbf{c}} \max_{i} \frac{\sigma_i}{(\mathbf{c}^\top \mathbf{u}_i)^2}$ $\mathbf{u}_i^\top \mathbf{u}_j = 0,\ \forall (i,j) \notin E$	$\min_{\{\mathbf{u}_i\}} \min_{\mathbf{c}} \max_{i} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2}$ $\mathbf{u}_i^\top \mathbf{u}_j \leq S_{ij},\ i \neq j$
$\mathcal{K}_G = \{K \succeq 0 \mid K_{ii} = 1,$ $K_{ij} = 0, \forall (i,j) \notin E\}$	$\mathcal{K}_{G,\boldsymbol{\sigma}} = \{K \succeq 0 \mid K_{ii} = 1/\sigma_i,$ $K_{ij} = 0, \forall (i,j) \notin E\}$	$\mathcal{K}_{G,S} = \{K \succeq 0 \mid K_{ii} = 1,$ $K_{ij} \leq S_{ij}, i \neq j\}$
$K_{LS} = \frac{A}{\lvert\lambda_n(A)\rvert} + I$	$K_{LS}^{\boldsymbol{\sigma}} = \frac{A}{\sigma_{max}\lvert\lambda_n(A)\rvert} + \mathrm{diag}(\boldsymbol{\sigma})^{-1}$	$K_{LS}^{S} = \frac{S}{\lvert\lambda_n(S)\rvert} + I$

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Computing weighted theta function can be done via SDP (semidefinite programming), but too slow in many cases
  - Extends fast approximation known as SVM-theta introduced by Jethava et al. to weighted graphs
- Using a truncated SVD for low rank approximation combined with a one-class SVM, almost quadratic time complexity can be achieved
- Maximizing the expression used in the weighted theta function can be viewed as finding a subset of nodes that is large and diverse

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Applies weighted theta function to clustering problems
  - The weighted Lovasz number is used as the number of clusters, which handles the problem of having to guess the optimal number of clusters
  - The support vectors calculated during optimization are used as initialization parameters
- Handles the max-cut graph problem through the use of the geometric embedding resulting from the proposed method

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Also applies proposed method as well as variants to the problem of correlation clustering
  - Theta-means, theta-means with random initialization, and k-means with theta initialization
- Theta-means achieves the highest F1 score, followed by theta-means with random initialization
  - Also significant speedups in time complexity when compared to other approaches

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

Table 3: Clustering of the (mini) newsgroup dataset. Average (and std. deviation) over 5 splits. $\hat{k}$ is the average number of clusters predicted. The true number is $k = 16$.

	$\mathbf{F}_1$	$\hat{k}$	**Time**
VOTE/BOEM	31.29 ± 4.0	124	8.7m
PIVOT/BOEM	30.07 ± 3.4	120	14m
BEST/BOEM	29.67 ± 3.4	112	13m
FIRST/BOEM	26.76 ± 3.8	109	14m
k-MEANS+RAND	17.31 ± 1.3	2	15m
k-MEANS+INIT	20.06 ± 6.8	3	5.2m
ϑ-MEANS+RAND	35.60 ± 4.3	25	45s
ϑ-MEANS	36.20 ± 4.9	25	11s

Weighted Theta Functions and Embeddings with Applications to Max-Cut, Clustering, and Summarization

---

- Also applied method to overlapping correlation clustering and document summarization
  - In document summarization, the weighted theta expression optimization is viewed as the trade-off between brevity and relevance (analogous to size and diversity)
- Conclusion: Extension of Lovasz theta function to weighted graphs can be applied to various machine learning problems, with the SVM approximation yielding significant speedups

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Authors: Yuya Yoshikawa, Tomoharu Iwata, Hiroshi Sawada, Takeshi Yamada
- Proposes method for finding relationships between documents across domains via embedding into a shared latent space
- Given an instance in a source domain, objective is to find most related instance in a target domain

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Previous work: Canonical Correspondence Analysis (CCA) and kernel CCA
  - Linear (or other kernel-based) projection into latent space that maximizes correlation between instance pairs
  - Kernel CCA struggles with different words that are semantically equivalent (i.e. “PC” vs “computer”)
- Proposed method associates each feature with a vector in the latent projection space, and then instances are represented as distributions over latent vectors

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Proposed method differs from CCA and kernel CCA in that it represents each instance as a set of latent vectors (as opposed to a single mapping), and can thus learn more complex representations
  - Also method is discriminative as opposed to generative
- Kernel embedding embeds probability distribution P on space X into a reproducing kernel Hilbert space (RKHS) specified by the chosen kernel
- Difference between embeddings of samples can be calculated via square of maximum mean discrepancy (MMD)

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Training is done via considering instance pairs of the form (source, target)
- Each feature of each instance is translated into a vector in the projection space
  - Goal is to reflect co-occurrence of different but related features via kernel calculations between instances
- Kernel embedding of distribution of features is used to represent distribution of latent vectors for instances
  - Difference between instance distributions computed via MMD

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Proposed model assumes that related instances have similar distributions of latent vectors
- Latent vectors are estimated via maximizing the posterior probability, or equivalently minimizing the negative log of the posterior probability
  - Gradient-based optimization
- Instance matching is then done via finding the minimal distance between the source instance and a target instance via MMD

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Setup of proposed method: used Gaussian embedding kernel and selected optimal hyper-parameters via validation data
- Compared proposed method to: K-nearest neighbors (KNN), CCA, kernel CCA, bilingual latent dirichlet allocation, and kernel CCA with kernel embeddings of distributions
- Evaluated via precision@k
- Tested on Wikipedia document dataset over German, English, Finnish, French, Italian, and Japanese

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

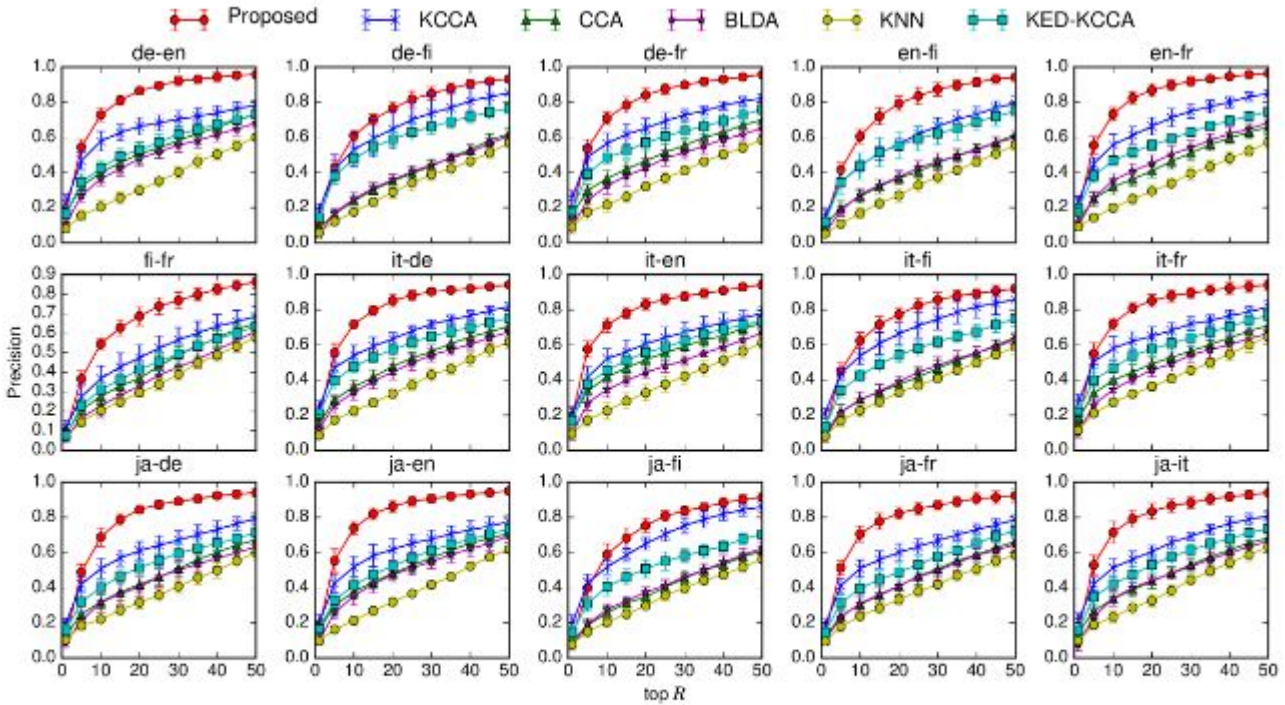

Figure 2: Precision of matching prediction and its standard deviation on multi-lingual Wikipedia datasets.

Cross-Domain Matching for Bag-of-Words Data via Kernel Embeddings of Latent Distributions (2015)

- Also tested proposed method on matching documents and tags, and images and tags
  - Had similar results to previously shown data, proposed method outperformed compared methods
- Conclusion: Representing instances as distributions over latent vectors in a projected space can capture more semantic information and can lead to better performance on relationship identification tasks

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Authors: Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, Martin Riedmiller
- Introduces Embed to Control (E2C), a deep generative model that learns to generate image trajectories from a latent space in which dynamics are constrained to be locally linear
- Considers problem of nonlinear dynamic system control for robots and autonomous agents
- Use of Stochastic Optimal Control (SOC) methods would be computationally infeasible considering the high-dimensionality of sensory data

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Problem formulation: goal is to map high-dimensional images to low-dimensional vectors, and then solve optimal control using the mapped vectors
- Proposes SOC formulation of problem using latent vectors
  - Optimal controls for trajectories of a specified length T can be computed via minimizing a cost function for expected future cost
  - Yields a locally linear-quadratic-Gaussian formulation at each time step that can be solved via existing SOC algorithms

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Must learn a low-dimensional latent representation with specific properties
  - Representation must capture sufficient information about image
  - Must allow for accurate prediction of the next latent state
  - The prediction of the next latent state must be locally linearizable
- Proposes an inference model for sampling latent states
  - Based on a diagonal Gaussian distribution whose properties are computed via a neural network
- A generative model is built from inference model to generate image samples from latent samples

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Transformation parameters are predicted from latent samples via a neural network (transformation network)
  - Requires that latent state transition distribution be similar to actual state transition distribution
- Model is trained via data set consisting of observation tuples and corresponding controls
  - Parameters for inference, transition, and generation are learned via minimizing a variational bound on negative log-likelihood

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Model is learned via minimizing loss function based on log-likelihood and KL divergence
  - Computed via stochastic gradient descent
- Model training: considered both a standard (max 3 layer, fully connected) neural network as well as deep convolutional neural networks
- Variant: model estimating dynamics as a non-linear function
- Baselines: standard variational autoencoder (VAE) and deep autoencoder (AE)

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Applies iterative Linear Quadratic Regulation (iLQR) and Approximate Inference Control (AICO) to perform optimal control in latent space
- Control in planar systems: autoencoders fail to discover underlying structure of the state space

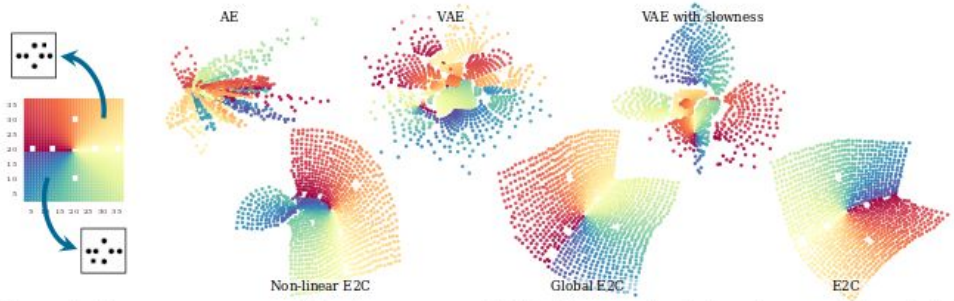

Figure 2: The true state space of the planar system (left) with examples (obstacles encoded as circles) and the inferred spaces (right) of different models. The spaces are spanned by generating images for every valid position of the agent and embedding them with the respective encoders.

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Swing-up pendulum task: Swing-up and balance an underactuated pendulum from resting position
- More complex dynamical tasks: Cart-pole trajectory and robot arm trajectory

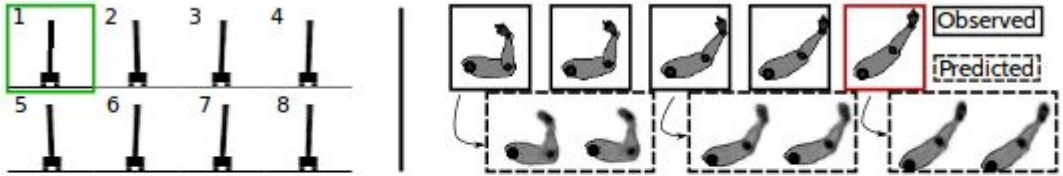

Figure 4: Left: Trajectory from the cart-pole domain. Only the first image (green) is "real", all other images are "dreamed up" by our model. Notice discretization artifacts present in the real image. Right: Exemplary observed (with history image omitted) and predicted images (including the history image) for a trajectory in the visual robot arm domain with the goal marked in red.

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

Table 1: Comparison between different approaches to model learning from raw pixels for the planar and pendulum system. We compare all models with respect to their prediction quality on a test set of sampled transitions and with respect to their performance when combined with SOC (trajectory cost for control from different start states). Note that trajectory costs in latent space are not necessarily comparable. The "real" trajectory cost was computed on the dynamics of the simulator while executing planned actions. For the true models for $\mathbf{s}_t$, real trajectory costs were 20.24 ± 4.15 for the planar system, and 9.8 ± 2.4 for the pendulum. Success was defined as reaching the goal state and staying ϵ-close to it for the rest of the trajectory (if non terminating). All statistics quantify over 5/30 (plane/pendulum) different starting positions. A † marks separately trained dynamics networks.

Algorithm	State Loss $\log p(\mathbf{x}_t \mid \hat{\mathbf{x}}_t)$	Next State Loss $\log p(\mathbf{x}_{t+1} \mid \hat{\mathbf{x}}_t, \mathbf{u}_t)$	Trajectory Cost Latent	Trajectory Cost Real	Success percent
Planar System					
AE†	11.5 ± 97.8	3538.9 ± 1395.2	1325.6 ± 81.2	273.3 ± 16.4	0 %
VAE†	3.6 ± 18.9	652.1 ± 930.6	43.1 ± 20.8	91.3 ± 16.4	0 %
VAE + slowness†	10.5 ± 22.8	104.3 ± 235.8	47.1 ± 20.5	89.1 ± 16.4	0 %
Non-linear E2C	8.3 ± 5.5	11.3 ± 10.1	19.8 ± 9.8	42.3 ± 16.4	96.6 %
Global E2C	**6.9 ± 3.2**	**9.3 ± 4.6**	12.5 ± 3.9	27.3 ± 9.7	**100** %
E2C	7.7 ± 2.0	9.7 ± 3.2	10.3 ± 2.8	**25.1 ± 5.3**	**100** %
Inverted Pendulum Swing-Up					
AE†	8.9 ± 100.3	13433.8 ± 6238.8	1285.9 ± 355.8	194.7 ± 44.8	0 %
VAE†	7.5 ± 47.7	8791.2 ± 17356.9	497.8 ± 129.4	237.2 ± 41.2	0 %
VAE + slowness†	26.5 ± 18.0	779.7 ± 633.3	419.5 ± 85.8	188.2 ± 43.6	0 %
E2C no latent KL	64.4 ± 32.8	87.7 ± 64.2	489.1 ± 87.5	213.2 ± 84.3	0 %
Non-linear E2C	**59.6 ± 25.2**	**72.6 ± 34.5**	313.3 ± 65.7	37.4 ± 12.4	63.33 %
Global E2C	115.5 ± 56.9	125.3 ± 62.6	628.1 ± 45.9	125.1 ± 10.7	0 %
E2C	84.0 ± 50.8	89.3 ± 42.9	275.0 ± 16.6	**15.4 ± 3.4**	**90** %

Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images (2015)

- Conclusion: Embed to Control (E2C) model for stochastic optimal control on high-dimensional image streams can find embeddings that can produce performance that is competitive with performance achieved by performing optimal control on the real system model itself
- Method extends similar work concerning deep autoencoders for control tasks and enforcing desired transformations during learning

Community Detection via Measure Space Embedding (2015)

- Authors: Mark Kozdoba, Shie Mannor
- Proposes algorithm to detect communities via embedding graphs in a measure space using random walks, and then applying k-means
- Problem: identifying subsets of vertices in graphs in which there is dense connectivity (communities)
- Presents Diffusion Entropy Reducer (DER) algorithm for non-overlapping community detection

Community Detection via Measure Space Embedding (2015)

- Algorithm is evaluated on random graph benchmarks (LFR models)
- Algorithm can be modified to detect overlapping communities in specific cases
- DER also reconstructs (with high probability) the partition of the p,q-stochastic block model (the generative model for the considered random graphs)

Community Detection via Measure Space Embedding (2015)

- Related work: stochastic block model (p,q-SBM) as a generative model for non-overlapping communities
  - Distribution on the graphs over a vertex set V
- Consider a graph G with set of vertices V and define pi(i) to be the stationary random walk based on the degree of vertex i (sum of outgoing edge weights)
- Choose a start vertex randomly from pi and perform a random walk of L steps yielding a sequence of L+1 vertices
  - Perform this N times to obtain N such sequences

Community Detection via Measure Space Embedding (2015)

- The measure of a vertex i is the average of the distributions of the random walks of lengths 1 up to L from i
- Perform k-means using the computed measures of the vertices
  - Finds optimal partition of vertices of the graph
- DER can be interpreted as seeking a partition that maximizes information between current state and the next step from it

Community Detection via Measure Space Embedding (2015)

- Since DER is a k-means algorithm, its results are somewhat contingent on the random initialization of partitions
- Walktrap algorithm is similar to DER in that it also calculates measures, but employs and optimizes them differently
- Infomap algorithm attempts to minimize information required to transmit a random walk across a graph through a channel (coding is done via clusters)

Community Detection via Measure Space Embedding (2015)

- LFR benchmark model: node degrees and community sizes have power law distribution
- Given a set of computed communities and a ground truth set of communities, “closeness” can be measured using normalized mutual information (NMI)
  - Only works for non-overlapping communities, although extensions exist
- DER tested on graphs with N = 1000 nodes and N = 5000 nodes with 10 to 50 communities or 20 to 100 communities

Community Detection via Measure Space Embedding (2015)

- DER can be extended to overlapping communities by defining a function corresponding to the probability that a walk started in a specific partition given that it ended at a specific vertex
  - Can be used to compute groups of likely overlapping communities
- Results showed that DER outperformed other models for overlapping community detection in graphs with sparse overlaps

Community Detection via Measure Space Embedding (2015)

- Proves analytic bounds on the probability that DER recovers partitions of a graph after one iteration from a given random initialization
  - Uses the non-ideal nature of the random initialization in tandem with linearization argument

References (2015)

- http://papers.nips.cc/paper/5971-space-time-local-embeddings.pdf
- http://papers.nips.cc/paper/5972-a-fast-universal-algorithm-to-learn-parametric-nonlinear-embeddings.pdf
- http://papers.nips.cc/paper/5992-compressive-spectral-embedding-sidestepping-the-svd.pdf
- http://papers.nips.cc/paper/5969-sparse-local-embeddings-for-extreme-multi-label-classification.pdf

References (2015) (continued)

- http://papers.nips.cc/paper/5849-semi-supervised-convolutional-neural-networks-for-text-categorization-via-region-embedding.pdf
- http://papers.nips.cc/paper/5902-weighted-theta-functions-and-embeddings-with-applications-to-max-cut-clustering-and-summarization.pdf
- http://papers.nips.cc/paper/5959-cross-domain-matching-for-bag-of-words-data-via-kernel-embeddings-of-latent-distributions.pdf

References (2015) (continued)

- http://papers.nips.cc/paper/5964-embed-to-control-a-locally-linear-latent-dynamics-model-for-control-from-raw-images.pdf
- http://papers.nips.cc/paper/5808-community-detection-via-measure-space-embedding.pdf