Discriminative Embeddings
of Latent Variable Models for Structured Data

by Hanjun Dai, Bo Dai, Le Song

presentation by
Breandan Considine
McGill University

breandan.considine@mail.mcgill.ca

March 9, 2020

What is a kernel?

A feature map transforms the input space to a feature space:

$$\varphi : \overbrace{\mathbb{R}^n}^{\text{Input space}} \rightarrow \overbrace{\mathbb{R}^m}^{\text{Feature space}} \tag{1}$$

A kernel function k is a real-valued function with two inputs:

$$k : \Omega \times \Omega \rightarrow \mathbb{R} \tag{2}$$

Kernel functions generalize the notion of inner products to feature maps:

$$k(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^\intercal \varphi(\mathbf{y}) \tag{3}$$

Gives us $\varphi(x)^\intercal \varphi(y)$ without directly computing $\varphi(x)$ or $\varphi(y)$.

What is a kernel?

Consider the univariate polynomial regression algorithm:

$$\hat{f}(x;\boldsymbol{\beta}) = \beta\varphi(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_m x^m = \sum_{j=0}^{m} \beta_j x^j \tag{4}$$

Where $\varphi(\mathbf{x}) = [1, x, x^2, x^3, \ldots, x^m]$. We seek $\boldsymbol{\beta}$ minimizing the error:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} ||\mathbf{Y} - \hat{\mathbf{f}}(\mathbf{X};\boldsymbol{\beta})||^2 \tag{5}$$

Can solve for $\boldsymbol{\beta}^*$ using the normal equation or gradient descent:

$$\boldsymbol{\beta}^* = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{Y} \tag{6}$$

$$\boldsymbol{\beta}' \leftarrow \boldsymbol{\beta} - \alpha\nabla_{\boldsymbol{\beta}}||\mathbf{Y} - \hat{\mathbf{f}}(\mathbf{X};\boldsymbol{\beta})||^2 \tag{7}$$

What happens if we want to approximate a multivariate polynomial?

$$z(x, y) = 1 + \beta_x x + \beta_y y + \beta_{xy} xy + \beta_{x^2} x^2 + \beta_{y^2} y^2 + \beta_{xy^2} xy^2 + \ldots \tag{8}$$

What is a kernel?

Consider the polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T\mathbf{y})^2$ with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$.

$$k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T\mathbf{y})^2 = (1 + x_1\, y_1 + x_2\, y_2)^2 \tag{9}$$

$$= 1 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 \tag{10}$$

This gives us the same result as computing the 6 dimensional feature map:

$$k(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^\intercal \varphi(\mathbf{y}) \tag{11}$$

$$= [1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2]^\intercal \begin{bmatrix} 1 \\ y_1^2 \\ y_2^2 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ \sqrt{2}y_1 y_2 \end{bmatrix} \tag{12}$$

But does not require computing $\varphi(x)$ or $\varphi(y)$.

Examples of common kernels

Popular kernels

Polynomial	$k(\mathbf{x},\mathbf{y}) := (\mathbf{x}^T\mathbf{y}+r)^n$	$\mathbf{x},\mathbf{y}\in\mathbb{R}^d, n\in\mathbb{N}, r\geq 0$
Laplacian	$k(\mathbf{x},\mathbf{y}) := \exp\left(-\frac{\Vert\mathbf{x}-\mathbf{y}\Vert}{\sigma}\right)$	$\mathbf{x},\mathbf{y}\in\mathbb{R}^d, \sigma > 0$
Gaussian RBF	$k(\mathbf{x},\mathbf{y}) := \exp\left(-\frac{\Vert\mathbf{x}-\mathbf{y}\Vert^2}{2\sigma^2}\right)$	$\mathbf{x},\mathbf{y}\in\mathbb{R}^d, \sigma > 0$

Popular Graph Kernels

RW	$k_\times(G,H) := \sum_{i,j=1}^{\lvert V_\times\rvert}[\sum_{n=1}^{\infty}\lambda^n A_\times^n]_{ij} = \mathbf{e}^\intercal(\mathbf{I}-\lambda A_\times)^{-1}\mathbf{e}$	$\mathcal{O}(n^6)$
SP	$k_{SP}(G,H) := \sum_{s_1\in SD(G)}\sum_{s_2\in SD(H)} k(s_1,s_2)$	$\mathcal{O}(n^4)$
WL	$\mathrm{l}^{(i)}(G) := \begin{cases}\deg_v, \forall v\in G & \mathrm{i}=1\\ \mathrm{HASH}(\{\{\mathrm{l}^{(i-1)}(u), \forall u\in\mathcal{N}(v)\}\}) & \mathrm{i}>1\end{cases}$ $k_{WL}(G,H) := \langle\psi_{WL}(G),\psi_{WL}(H)\rangle$	$\mathcal{O}(hm)$

https://people.mpi-inf.mpg.de/~mehlhorn/ftp/genWLpaper.pdf

Positive definite kernels

Positive Definite Matrix

A symmetric matrix $\mathbf{K} \in \mathbb{R}^{N^2}$ is **positive definite** if $\mathbf{x}^\intercal \mathbf{K} \mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^N \setminus \mathbf{0}$.

Positive Definite Kernel

A symmetric kernel k is called positive definite on Ω if its associated kernel matrix $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=0}^N$ is positive definite $\forall N \in \mathbb{N}, \forall \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$.

http://www.math.iit.edu/~fass/PDKernels.pdf

What is an inner product space?

Linear function

Let X be a vector space over $\mathbb{R}$. A function $f : X \to \mathbb{R}$ is **linear** iff $f(\alpha x) = \alpha f(x)$ and $f(x+z) = f(x) + f(z)$ for all $\alpha \in \mathbb{R}, x, z \in X$.

Inner product space

X is an **inner product space** if there exists a symmetric bilinear map $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{R}$ if $\forall \mathbf{x} \in X, \langle \mathbf{x}, \mathbf{x} \rangle > 0$ (i.e. is positive definite).

Cauchy-Schwartz Inequality

If X is an inner product space, then $\forall \mathbf{u}, \mathbf{v} \in \mathcal{X}, |\langle \mathbf{u}, \mathbf{v} \rangle|^2 \leq \langle \mathbf{u}, \mathbf{u} \rangle \cdot \langle \mathbf{v}, \mathbf{v} \rangle$.

Scalar Product	**Vector Dot Product**	**Random Variable**
$\langle x, y \rangle := xy$	$\left\langle \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\rangle := x^\mathsf{T} y$	$\langle X, Y \rangle := \mathbb{E}(XY)$

What is a Hilbert space?

Let $d : X \times X \to \mathbb{R}^{\geq 0}$ be a metric on the space X.

Cauchy sequence

A sequence $\{x_n\}$ is called a **Cauchy sequence** if
$\forall \varepsilon > 0, \exists N \in \mathbb{N},$ such that $\forall n, m \geq N, d(x_n, x_m) \leq \varepsilon$.

Completeness

X is called **complete** if every Cauchy sequence converges to a point in X.

Separability

X is called **separable** if there exists a sequence $\{x_n\}_{n=1}^{\infty} \in X$ s.t. every nonempty open subset of X contains at least one element of the sequence.

Hilbert space

A Hilbert space $\mathcal{H}$ is an inner product space that is complete and separable.

Properties of Hilbert Spaces

Hilbert space inner products are kernels

The inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is a positive definite kernel:

$$\sum_{i,j=1}^{n} c_i c_j (x_i, x_j)_{\mathcal{H}} = \left(\sum_{i=1}^{n} c_i x_i, \sum_{j=1}^{n} c_j x_j \right)_{\mathcal{H}} = \left\| \sum_{i=1}^{n} c_i x_i \right\|_{\mathcal{H}}^{2} \geq 0$$

Reproducing Kernel Hilbert Space (RKHS)

Any continuous, symmetric, positive definite kernel $k : X \times X \to \mathbb{R}$ has a corresponding Hilbert space, which induces a feature map $\varphi : X \to \mathcal{H}$ satisfying $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$.

http://jmlr.csail.mit.edu/papers/volume11/vishwanathan10a/vishwanathan10a.pdf
https://marcocuturi.net/Papers/pdk_in_ml.pdf

Hilbert Space Embedding of Distributions

Maps distributions into potentially infinite dimensional feature spaces:

$$\mu_X := \mathbb{E}_X[\phi(X)] = \int_{\mathcal{X}} \phi(x)p(x)dx : \mathcal{P} \mapsto \mathcal{F} \tag{13}$$

By choosing the right kernel, we can make this mapping injective.

$$f(p(x)) = \tilde{f}(\mu_x), f : \mathcal{P} \mapsto \mathbb{R} \tag{14}$$

$$\mathcal{T} \circ p(x) = \tilde{\mathcal{T}} \circ \mu_x, \tilde{\mathcal{T}} : \mathcal{F} \mapsto \mathbb{R}^d \tag{15}$$

Hilbert Space Embedding of Distributions

Maps distributions into potentially infinite dimensional feature spaces:

$$\mu_X := \mathbb{E}_X[\phi(X)] = \int_{\mathcal{X}} \phi(x) p(x) dx : \mathcal{P} \mapsto \mathcal{F} \tag{16}$$

By choosing the right kernel, we can make this mapping injective.

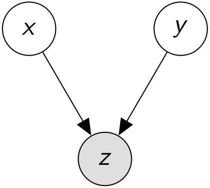
$$f(p(x)) = \tilde{f}(\mu_x), f : \mathcal{P} \mapsto \mathbb{R} \tag{17}$$

$$\mathcal{T} \circ p(x) = \tilde{\mathcal{T}} \circ \mu_x, \tilde{\mathcal{T}} : \mathcal{F} \mapsto \mathbb{R}^d \tag{18}$$

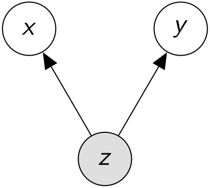
Belief Networks

Belief network is a distribution of the form:

$$P(x_1, \ldots, x_D) = \prod_{i=1}^{D} P(x_i | pa(x_i)) \tag{19}$$

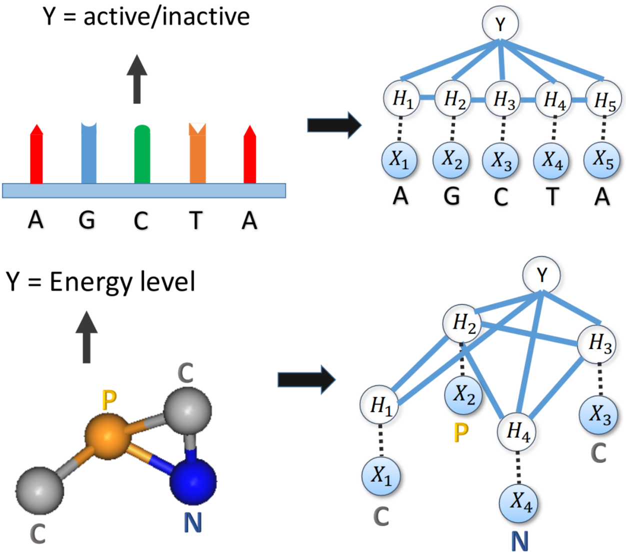


$P(X, Y|Z) \propto P(Z|X, Y)P(X)P(Y)$



$P(X, Y|Z) = P(X|Z)P(Y|Z)$

Latent Variable Models



Embedded mean field

Algorithm 1 Embedded Mean Field

Input: parameter $\mathbf{W}$ in $\widetilde{\mathcal{T}}$
Initialize $\widetilde{\mu}_i^{(0)} = \mathbf{0}$, for all $i \in \mathcal{V}$
for $t = 1$ **to** T **do**
    for $i \in \mathcal{V}$ **do**
        $l_i = \sum_{j \in \mathcal{N}(i)} \widetilde{\mu}_j^{(t-1)}$
        $\widetilde{\mu}_i^{(t)} = \sigma(W_1 x_i + W_2 l_i)$
    end for
end for{fixed point equation update}
return $\{\widetilde{\mu}_i^T\}_{i \in \mathcal{V}}$

Algorithm 2 Embedding Loopy BP

```
Input: parameter W in T̃_1 and T̃_2
Initialize ν̃_ij^(0) = 0, for all (i, j) ∈ ℰ
for t = 1 to T do
    for (i, j) ∈ ℰ do
        ν̃_ij^t = σ(W_1 x_i + W_2 Σ_{k∈N(i)\j} ν̃_ki^(t−1))
    end for
end for
for i ∈ V do
    μ̃_i = σ(W_3 x_i + W_4 Σ_{k∈N(i)\j} ν̃_ki^(T))
end for
return {μ̃_i}_{i∈V}
```

Input: parameter $\mathbf{W}$ in $\widetilde{\mathcal{T}}_1$ and $\widetilde{\mathcal{T}}_2$

Initialize $\widetilde{\nu}_{ij}^{(0)} = \mathbf{0}$, for all $(i, j) \in \mathcal{E}$

for $t = 1$ **to** T **do**

  for $(i, j) \in \mathcal{E}$ **do**

    $\widetilde{\nu}_{ij}^{t} = \sigma(W_1 x_i + W_2 \sum_{k \in \mathcal{N}(i) \setminus j} \widetilde{\nu}_{ki}^{(t-1)})$

  end for

end for

for $i \in \mathcal{V}$ **do**

  $\widetilde{\mu}_i = \sigma(W_3 x_i + W_4 \sum_{k \in \mathcal{N}(i) \setminus j} \widetilde{\nu}_{ki}^{(T)})$

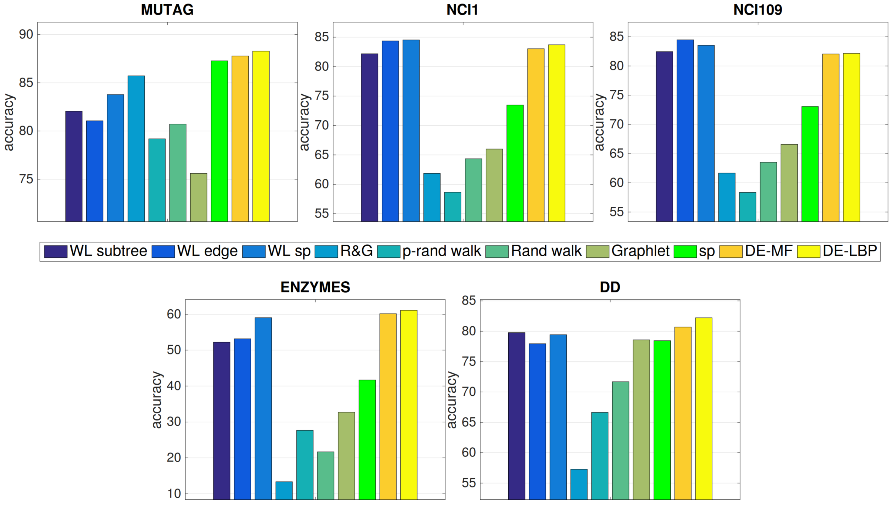
end for

return $\{\widetilde{\mu}_i\}_{i \in \mathcal{V}}$

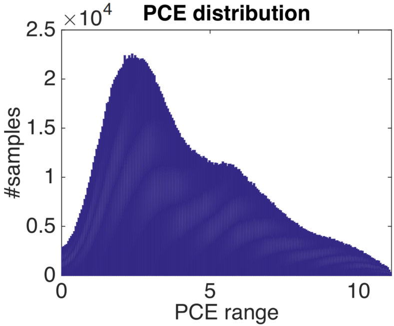
Algorithm 3 Discriminative Embedding

Input: Dataset $\mathcal{D} = \{\chi_n, y_n\}_{n=1}^N$, loss function $l(f(\chi), y)$.
Initialize $\mathbf{U}^0 = \{\mathbf{W}^0, \mathbf{u}^0\}$ randomly.
for $t = 1$ **to** T **do**
  Sample $\{\chi_t, y_t\}$ uniform randomly from $\mathcal{D}$.
  Construct latent variable model $p(\{H_i^t\}|\chi_n)$ as ($\boxed{5}$).
  Embed $p(\{H_i^t\}|\chi_n)$ as $\{\widetilde{\mu}_i^n\}_{i \in \mathcal{V}_n}$ by Algorithm $\boxed{1}$ or $\boxed{2}$ with $\mathbf{W}^{t-1}$.
  Update $\mathbf{U}^t = \mathbf{U}^{t-1} + \lambda_t \nabla_{\mathbf{U}^{t-1}} l(f(\widetilde{\mu}^n; \mathbf{U}^{t-1}), y_n)$.
end for
return $\mathbf{U}^T = \{\mathbf{W}^T, \mathbf{u}^T\}$

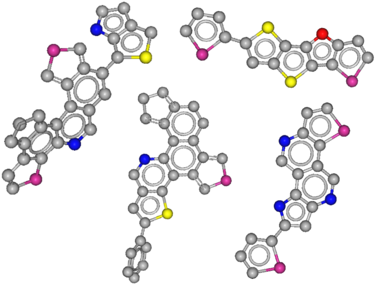
Graph Dataset Results



Harvard Clean Energy Project (CEP)

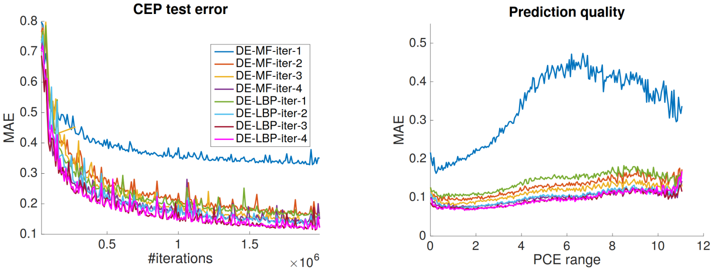


(a) PCE distribution



(b) Sample molecules

CEP Results



(a) Test error vs iterations

(b) Prediction quality

Figure 4: Details of training and prediction results for DE-MF and DE-LBP with different number of fixed point iterations.

	test MAE	test RMSE	# params
Mean Predictor	1.9864	2.4062	1
WL lv-3	0.1431	0.2040	1.6m
WL lv-6	0.0962	0.1367	1378m
DE-MF	0.0914	0.1250	0.1m
DE-LBP	**0.0850**	**0.1174**	0.1m

Table 3: Test prediction performance on CEP dataset. WL lv-k stands for Weisfeiler-lehman with degree k.

Resources

- Dai et al., *Discriminative Embeddings of Latent Variable Models*
- Cristianini and Shawe-Taylor, *Kernel Methods for Pattern Analysis*
- Kriege et al., *Survey on Graph Kernels*
- Panangaden, *Notes on Metric Spaces*
- Fasshauer, *Positive Definite Kernels: Past, Present and Future*
- Cuturi, *Positive Definite Kernels in Machine Learning*
- Gormley and Eisner, *Structured Belief Propagation for NLP*
- Forsyth, *Mean Field Inference*
- Tseng, *Probabilistic Graphical Models*