Computational Data Analysis

Machine Learning

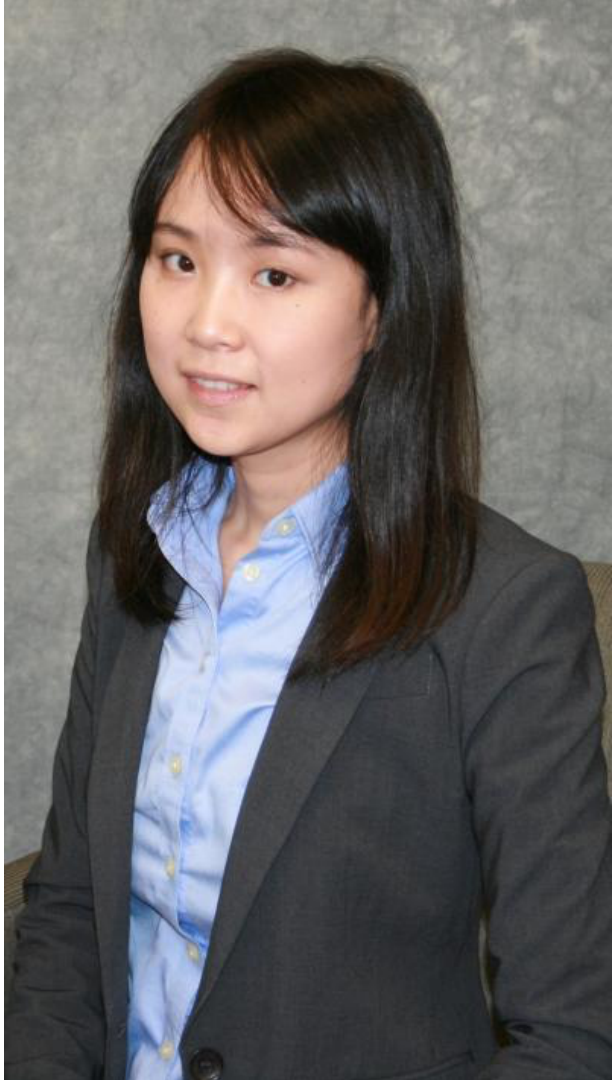
Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

Kernel methods



Revisit nonlinear regression

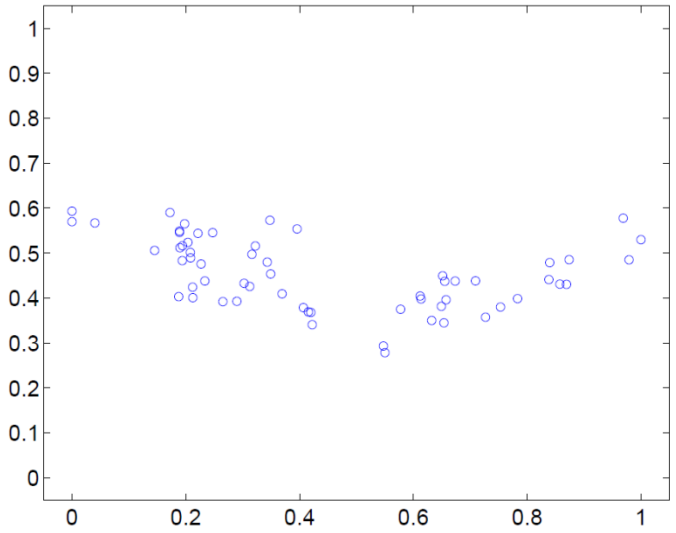
Want to fit a polynomial regression model

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^d + \epsilon$$

Let $\tilde{x} = \left(1, x, x^2, \ldots, x^d\right)^\top$

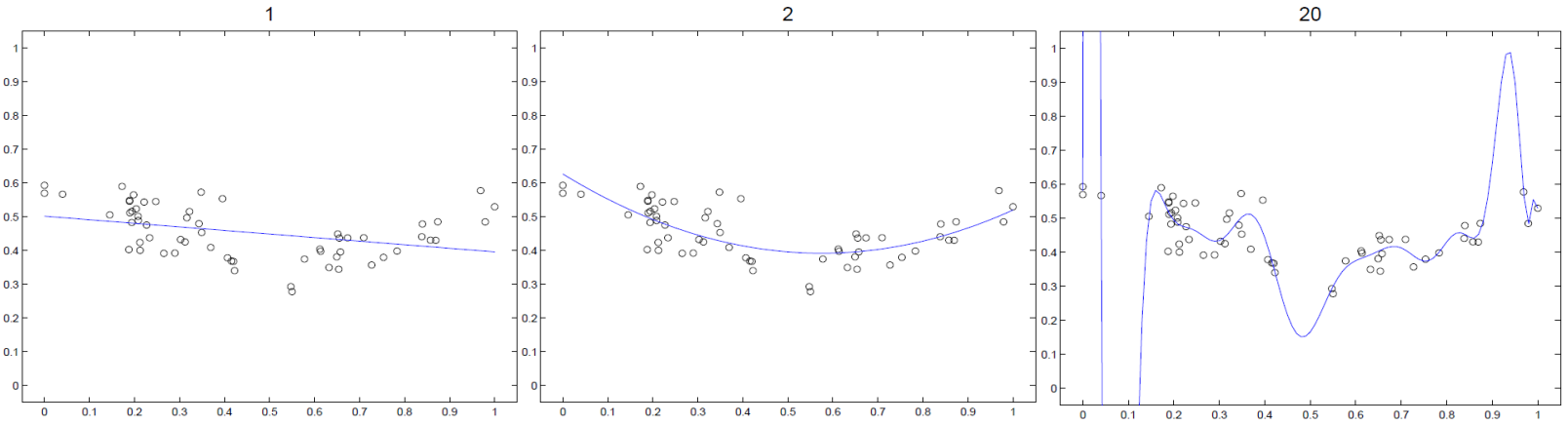
and $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_d)^\top$

$$\mathrm{y} = \theta^\top \tilde{x} + \epsilon$$



Overfitting / Underfitting by different polynomial degrees

- Blue points: training data points, Red points: test data points
- Choose the correct polynomial degree is not easy (needs cross-validation)



Under-fitting

Over-fitting

Problem of explicitly constructing features

- Explicitly construct feature map $\phi(x): R^n \mapsto F$, feature space can grow really large and really quickly.

- The if consider all polynomial feature of degree d
  - E.g. x_1^d, $x_1 x_2 \dots x_d$, $x_1^2 x_2 \dots x_{d-1}$
  - Total number of such feature is

$$\binom{d+n-1}{d} = \frac{(d+n-1)!}{d!\,(n-1)!}$$

  - $d = 6, n = 100$, there are 1.6 billion terms

Can we avoid expanding features: kernel trick

- Rather than consider features explicitly, let's consider their inner product

- Can we merge two steps using a clever function $k(x_i, x_j)$

  E.g. Polynomial $d = 2$

  - $\phi(x)^\top \phi(y) = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_2 x_1 \end{pmatrix}^\top \begin{pmatrix} y_1^2 \\ y_1 y_2 \\ y_2^2 \\ y_2 y_1 \end{pmatrix} = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2$
  - $= (x_1 y_1 + x_2 y_2)^2 = (x^\top y)^2$

$O(n)$ *computation*!

- Polynomial kernel degreee d, $k(x, y) = (x^\top y)^d = \phi(x)^\top \phi(y)$

Feature space is not unique

Eg. Polynomial $d = 2$

$$\phi(x)^\top \phi(y) = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_2 x_1 \end{pmatrix}^\top \begin{pmatrix} y_1^2 \\ y_1 y_2 \\ y_2^2 \\ y_2 y_1 \end{pmatrix} = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2$$
$$= (x_1 y_1 + x_2 y_2)^2 = (x^\top y)^2$$

$$\phi(x)^\top \phi(y) = \begin{pmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{pmatrix}^\top \begin{pmatrix} y_1^2 \\ \sqrt{2} y_1 y_2 \\ y_2^2 \end{pmatrix} = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2$$
$$= (x_1 y_1 + x_2 y_2)^2 = (x^\top y)^2$$

What $k(x, y)$ can be called a kernel function?

- $k(x, y)$ equivalent to compute inner product of features

$$k(x, y) = \phi(x)^{\top}\phi(y)$$

- Given a dataset $D = \{x^1, \ldots x^m\}$, compute pairwise kernel function $k\left(x^i, x^j\right)$ and form a $m \times m$ kernel matrix (Gram matrix)

$$K = \begin{pmatrix} k(x^1, x^1) & \ldots & k(x^1, x^m) \\ \vdots & \ddots & \vdots \\ k(x^m, x^1) & \ldots & k(x^m, x^m) \end{pmatrix}$$

- $k(x, y)$ is a kernel function, if and only if the Gram matrix K is positive semi-definite

$$\forall v \in R^m, v^{\top}Kv \geq 0$$

Typical kernel for vector data

- Polynomial of degree d
  - $k(x, y) = (x^\top y)^d$
- Polynomial of degree up to d
  - $k(x, y) = (x^\top y + c)^d$
- Exponential kernel (infinite degree polynomials)
  - $k(x, y) = \exp(s \cdot x^\top y)$
- Gaussian RBF kernel
  - $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$
- Laplace Kernel
  - $k(x, y) = \exp\left(-\frac{\|x-y\|}{2\sigma^2}\right)$
- Exponentiated distance
  - $k(x, y) = \exp\left(-\frac{d(x,y)^2}{s^2}\right)$

Kernel are used to develop nonlinear methods

- Strategy: We do not directly construct feature map, but choose a kernel function (represent inner product of data features)
- Replacing inner product with kernels
- Examples
    - Kernel SVM
    - Kernel Ridge regression
    - Kernel PCA
    - Kernel method for comparing distributions (two-sample test)

Example: SVM dual problem and kernelize

- Dual problem for SVM

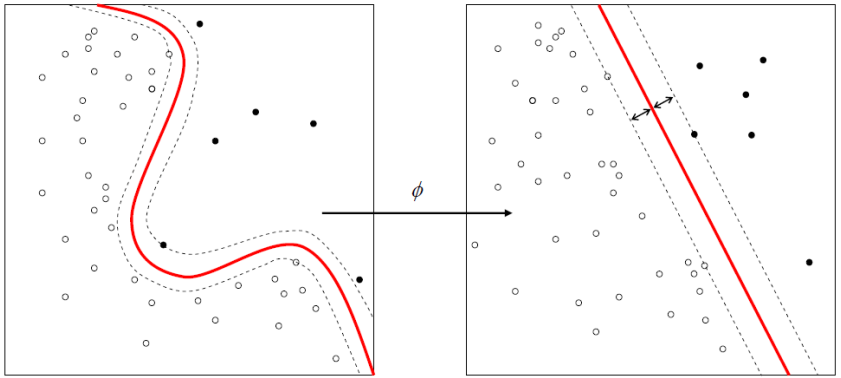
$$\text{Max}_\alpha \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^i y^j x^{i\top} x^j$$

$$s.t. \sum_i \alpha_i y^i = 0$$

$$0 \le \alpha_i \le C$$

Replace by $k(x^i, x^j)$

- Equivalent to finding a non-linear decision boundary
  - implicitly map data to a new nonlinear feature space
  - find linear decision boundary in the new space



Developing kernel ridge regression

- Matrix inversion lemma ($B \in R^{n \times m}$):

$$(BB^\top + \lambda I)^{-1} B = B(B^\top B + \lambda I)^{-1}$$

- Note that $X = (x^1, x^2, \ldots x^m)$

- Evaluate ridge regression solution: $\theta^r = (XX^\top + \lambda I)^{-1} Xy$ on a new test point x

$$\begin{aligned} x^\top \theta^r &= x^\top (XX^\top + \lambda I)^{-1} \, Xy \\ &= x^\top X (X^\top X + \lambda I)^{-1} \, y \end{aligned}$$

Kernel ridge regression

- Prediction

$$x^\top \theta^r = \theta^{rT} x = y^\top (X^\top X + \lambda I_n)^{-1} X^\top x$$

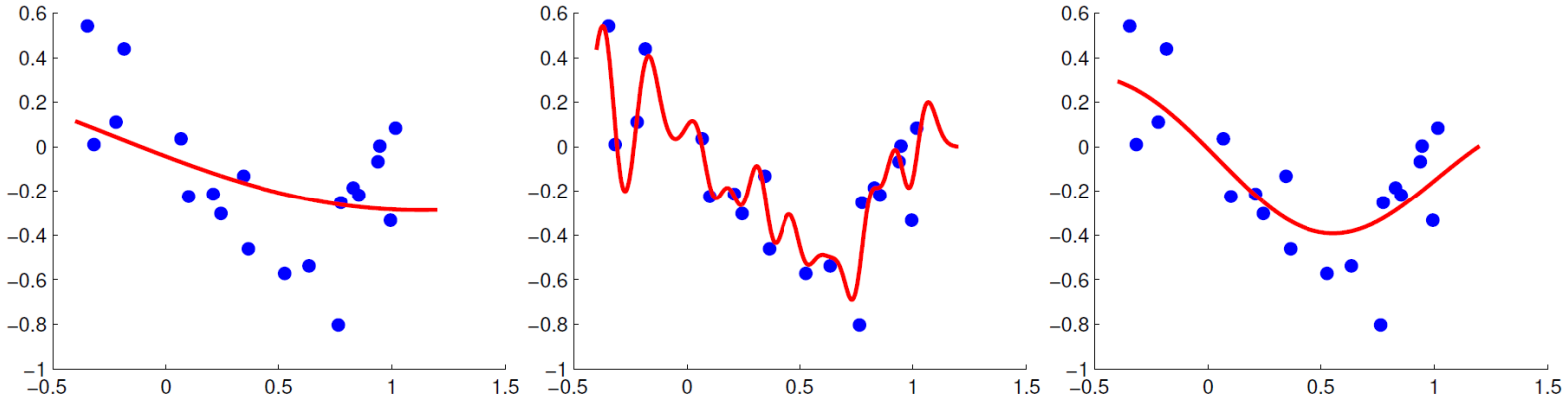
only depends on inner products!

- Kernel ridge regression: replace inner product by a kernel function
  - $X^\top X \to K = \left(k\left(x^i, x^j\right)\right)_{m \times m}$
  - $X^\top x \to k_x = \left(k\left(x^i, x\right)\right)_{m \times 1}$
  - Prediction $f(x) = y^\top (K + \lambda I_n)^{-1} k_x$

Kernel ridge regression

Use Gaussian rbf kernel

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$



$large\ \sigma, large\ \lambda$ | $small\ \sigma, small\ \lambda$ | $small\ \sigma, large\ \lambda$

Use cross-validation to choose parameters

Principal component analysis (PCA)

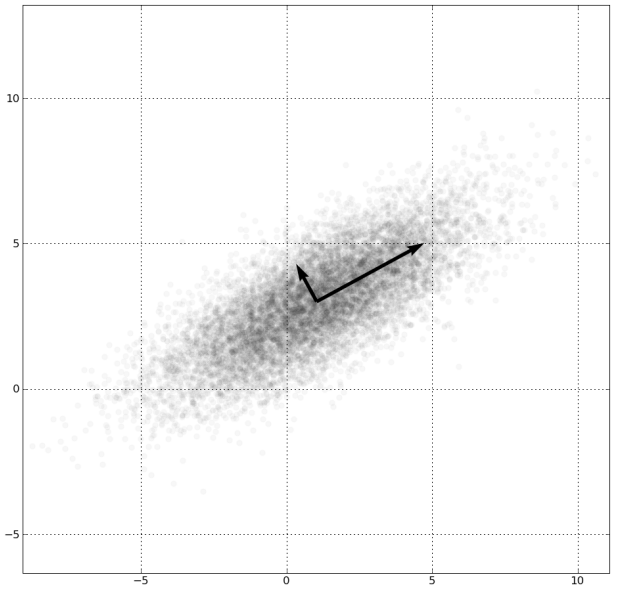
- Given a set of m centered observations $x^i \in R^d$, PCA finds the direction that maximizes the variance

$$w^* = argmax_{\|w\|\leq 1} \frac{1}{m} \sum_i \left(w^\top x^i\right)^2$$

$$= argmax_{\|w\|\leq 1} \frac{1}{m} w^\top X X^\top w$$

- $X = (x^1, x^2, \ldots, x^m)$
- $C = \frac{1}{m} X X^\top$
- w^* can be found by solving the following eigenvalue problem

$$Cw = \lambda w$$



Alternative expression for PCA

- We can show the principal component lies in the span of the data

$$w = \sum_{i=1}^{m} \alpha_i x^i = X\alpha$$

- Plug this in we have
  - $Cw = \frac{1}{m} XX^\top X\alpha = \lambda\, X\alpha = \lambda w$

- Furthermore, for each data point x^i, the following relation holds
  - ${x^i}^\top Cw = \frac{1}{m} {x^i}^\top XX^\top X\alpha = \lambda {x^i}^\top X\alpha, \forall i$
  - In matrix form, $\frac{1}{m} X^\top XX^\top X\alpha = \lambda X^\top X\alpha$

Only depends on inner product matrix

Kernel PCA

- Key Idea: Replace inner product matrix by kernel matrix
  - PCA: $\frac{1}{M} X^\top X X^\top X \alpha = \lambda X^\top X \alpha$
- Kernel PCA:
  - $\frac{1}{m} K K \alpha = \lambda K \alpha$, equivalent to

$$\frac{1}{m} K \alpha = \lambda \alpha$$

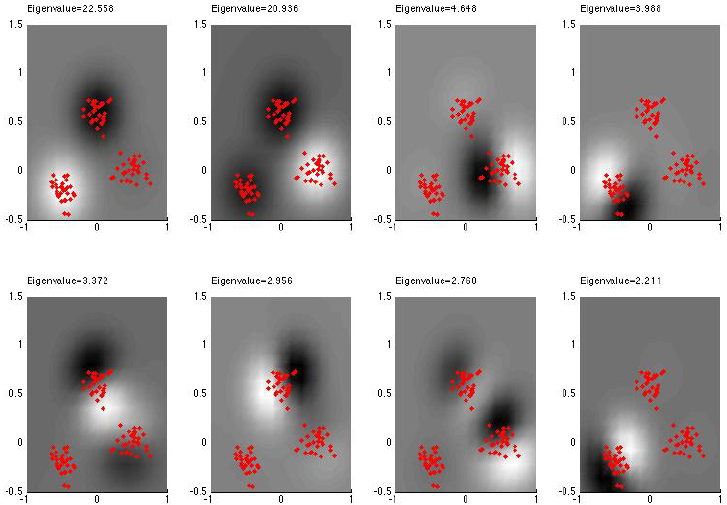
assuming Gram matrix is invertible

form an $m \times m$ kernel matrix K, and then perform eigen-decomposition on K

Kernel PCA

- Gaussian RBF kernel $\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$ over 2-dimensional space
- Eigenvector evaluated at a test point x is a function

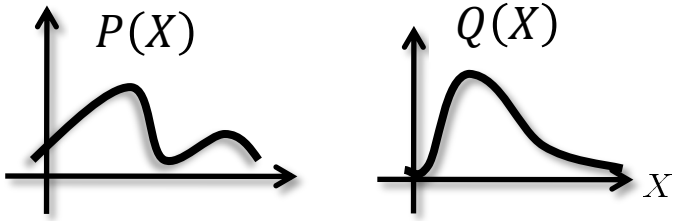
$$w^\top \phi(x) = \sum_i \alpha_i \langle \phi(x^i), \phi(x) \rangle = \sum_i \alpha_i k(x^i, x)$$



Comparing two distributions (two-sample test)

- Two-sample test for general distributions,
  - $H_0: P(x) = Q(x)?$
- We can use KL-divergence

$$KL(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$



- Given a set of samples

$$(x^1, \dots, x^m) \sim P(X), \left(\tilde{x}^1, \dots, \tilde{x}^{m'}\right) \sim Q(X)$$

$$\int P(x) \log \frac{P(x)}{Q(x)} dx \approx \int \hat{P}(x) \log \frac{\hat{P}(x)}{\hat{Q}(x)} dx$$

Need to estimate the density function first, and they can be noisy

Embedding distributions into feature space

- Summary statistics for distributions

$E_{X\sim P}[X]$ Mean

$E_{X\sim P}[XX^T]$ Covariance

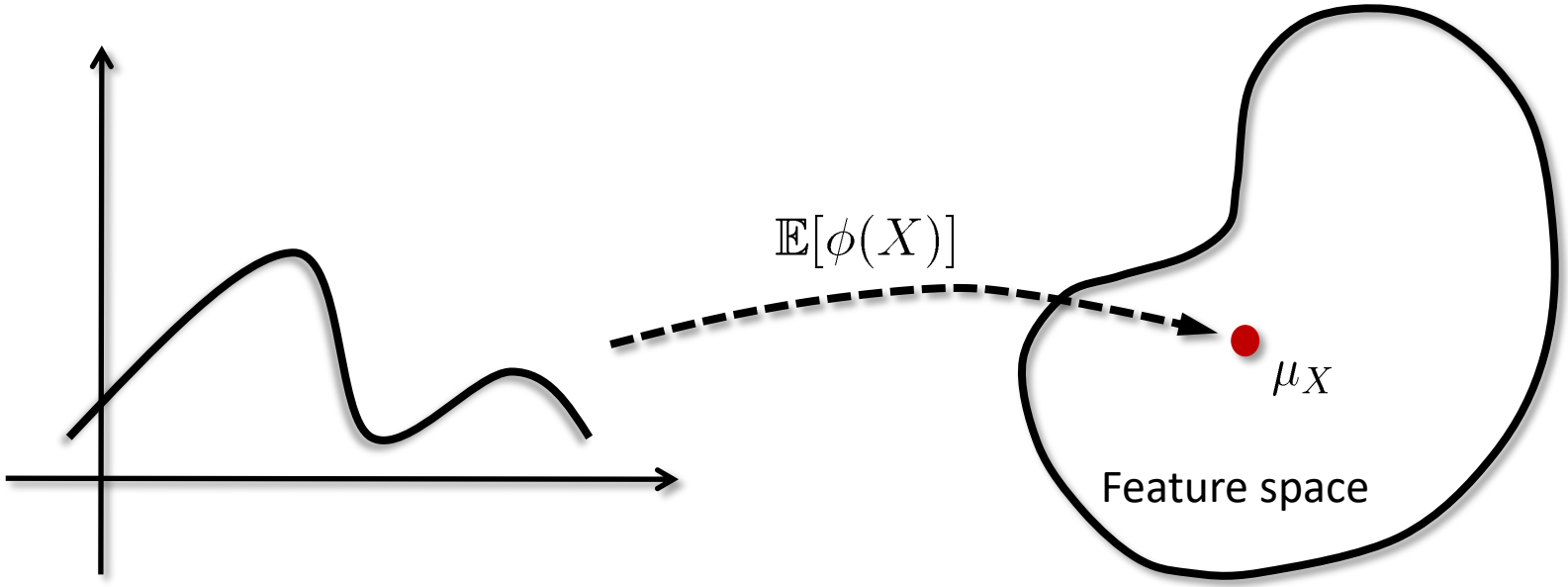
$E_{X\sim P}[\phi(X)]$ expected features

- Pick a kernel, and generate a different summary statistic

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

Illustration: embedding of distribution

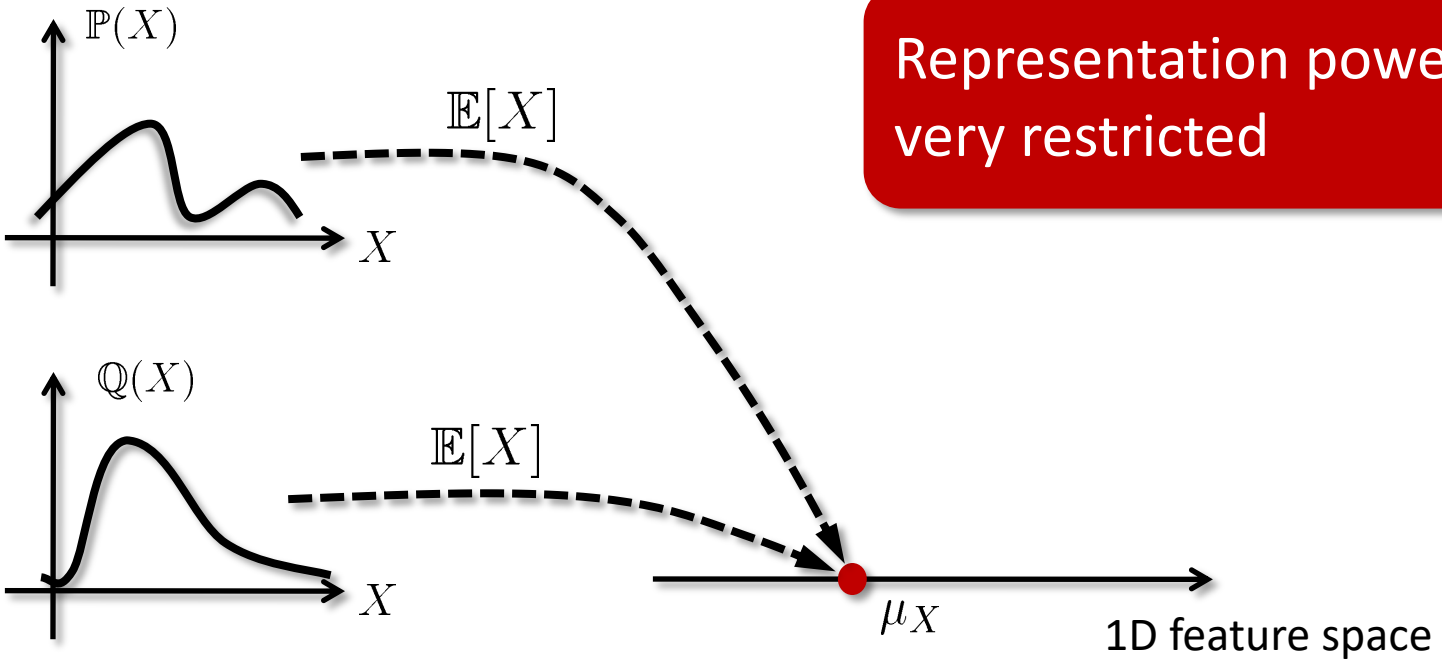
Transform the ***entire*** distribution to expected features



Feature map: $\phi(X)$

Embedding distributions: Mean

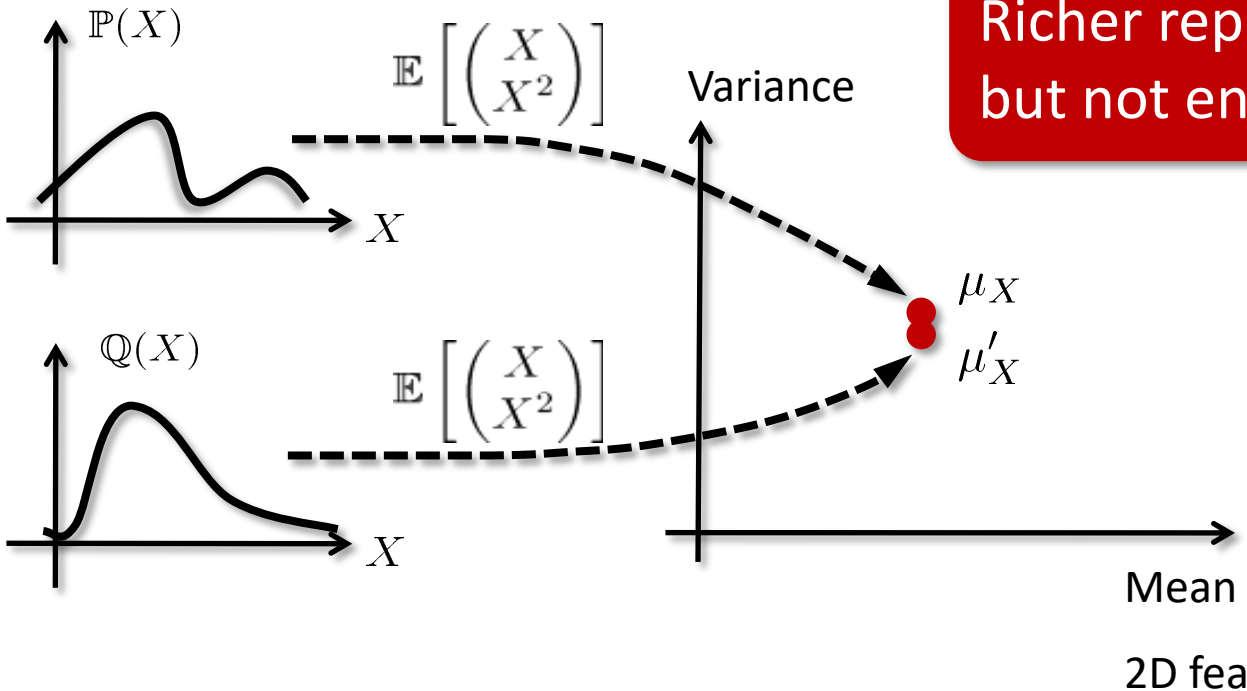
Mean reduces the ***entire*** distribution to a single number



Representation power very restricted

Embedding distributions: Mean + Variance

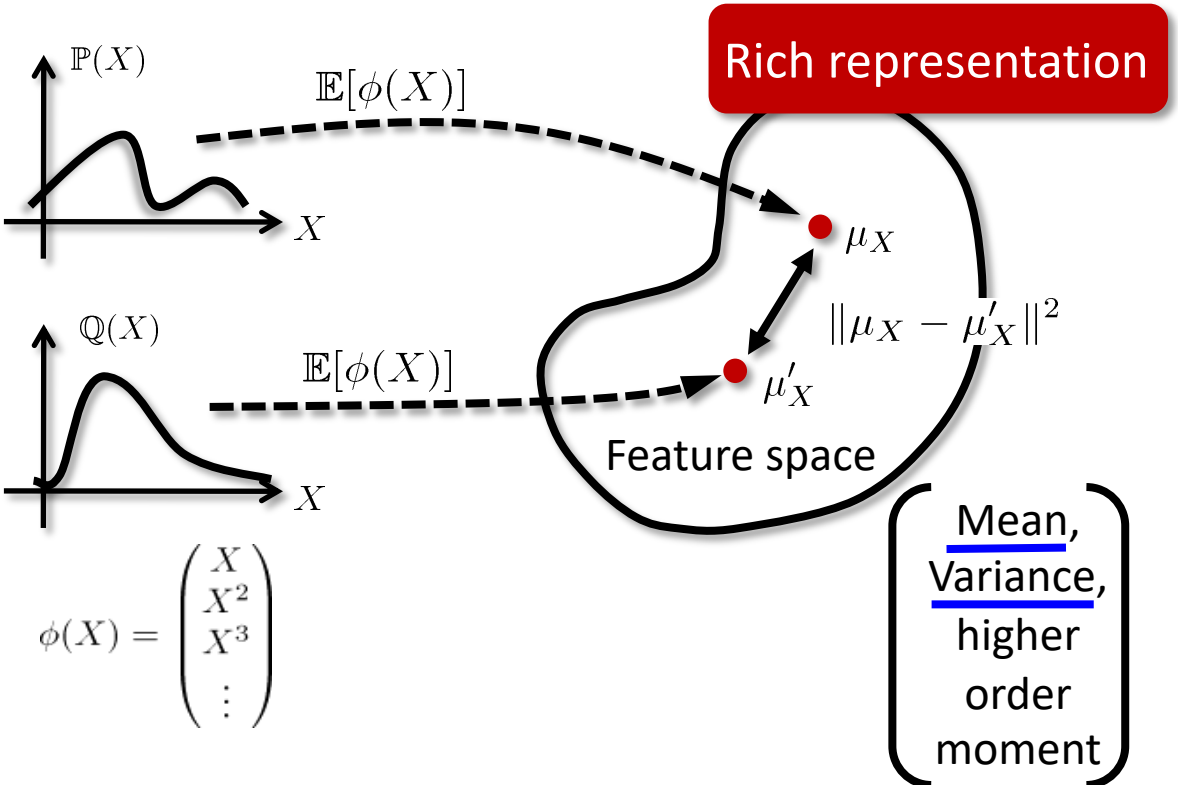
Mean and variance reduces the ***entire*** distribution to two numbers



Richer representation, but not enough

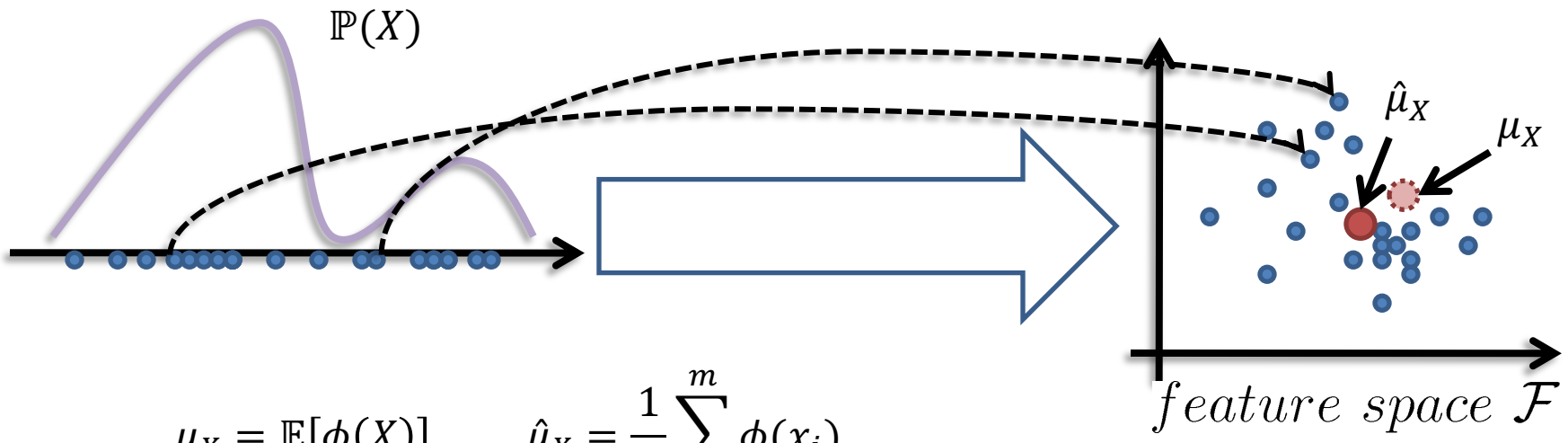
Embedding with kernel features

Transform distribution to infinite dimensional vector



Empirical estimation using data

- One-to-one mapping from $\mathbb{P}(X)$ to $\mu(X)$ for given kernels
- Sample average converges to true mean at $O_p(m^{-1/2})$



$$\mu_X = \mathbb{E}[\phi(X)], \qquad \hat{\mu}_X = \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)$$

Estimating embedding distances

- Given samples $(x^1, \ldots, x^m) \sim P(X), \left(\tilde{x}^1, \ldots, \tilde{x}^{m'}\right) \sim Q(X)$
- Distance between distributions can be represented as inner products

$$||\mu_X - \mu'_X||^2 = \langle \mu_X, \mu_X \rangle - 2\langle \mu_X, \mu'_X \rangle + \langle \mu'_X, \mu'_X \rangle$$

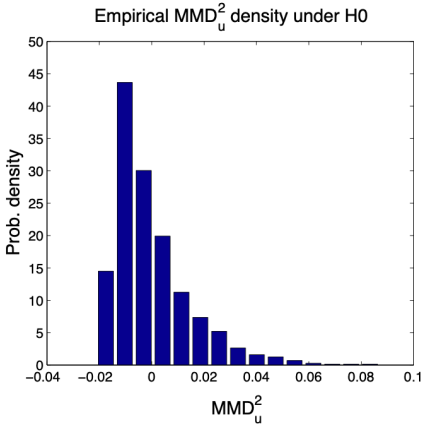
$$\begin{aligned} \langle \mu_X, \mu'_X \rangle &= \left\langle \mathbb{E}_{X \sim P}[\phi(X)], \mathbb{E}_{X' \sim Q}[\phi(X')] \right\rangle \\ &= \mathbb{E}_{X \sim P, X' \sim Q}[\underbrace{\langle \phi(X), \phi(X') \rangle}_{k(X, X')}] \end{aligned}$$

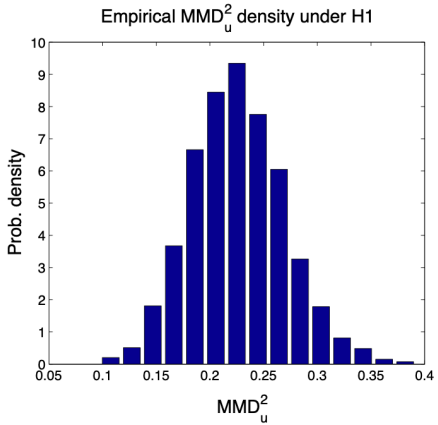
$$\approx \frac{1}{mm'} \sum_{i=1}^{m} \sum_{j=1}^{m'} k\left(x^i, \tilde{x}^j\right)$$

Empirical estimates

Kernel two-sample test

- Given two sets of samples $(x^1, \ldots, x^m) \sim P(X), \left(\tilde{x}^1, \ldots, \tilde{x}^{m'}\right) \sim Q(X)$
- Decide the distributions of the two sets of samples are different when the empirical estimate of $||\mu_X - \mu'_X||^2$ is greater than a threshold
- Application: anomaly detection, clinical trial - deciding whether or not a drug is effective, etc.





Gretton, Arthur, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. "A kernel two-sample test." *The Journal of Machine Learning Research* 13, no. 1 (2012): 723-773.


SUMMARY