Снижение размерности пространства в задачах моделирования законов физики с помощью методов машинного обучения *

П. А. Северилов[1], **В. В. Стрижов**[2]

Аннотация: В работе исследуется задача

Ключевые слова:

1 Введение

В данной работе решается задача

2 Постановка задачи

Пусть дана выборка $(\mathbf{X}, \mathbf{Y})$, где $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\mathsf{T} \in \mathbb{R}^{n \times m}$ — матрица независимых переменных, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]^\mathsf{T} \in \mathbb{R}^{n \times k}$ — матрица целевых переменных.

3 Обзор методов снижения размерности для задачи декодирования

Методы снижения размерности позволяют найти низкоразмерное представление исходных данных. Найденное представление используется для построения прогностической модели. При этом метод снижения размерности может учитывать как зависимости в исходной переменной $\mathbf{x}$, так и в целевой переменной $\mathbf{y}$.

---

*no

[1]Московский физико-технический институт, severilov.pa@phystech.edu

[2]Вычислительный центр имени А. А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, Московский физико-технический институт, strijov@phystech.edu

3.0.1 Метод главных компонент для задачи декодирования.

Для устранения линейной зависимости и снижения размерности исходного пространства широко используется метод главных компонент (principal component analysis, PCA). Метод PCA находит низкоразмерное представление матрицы $\mathbf{X} = \mathbf{TP}$, такое что новое представление $\mathbf{T} \in^{m \times l}$ содержит максимальную долю дисперсии исходной матрицы. При этом матрица отображения $\mathbf{P} \in^{l \times n}$ $(\mathbf{PP}^{\mathsf{T}} =)$ содержит правые собственные вектора матрицы ковариаций $\mathbf{X}^{\mathsf{T}}\mathbf{X}$.

Метод PCA является базовым методом снижения размерности пространства. Существует множество модификаций базового метода. Вероятностный PCA [**?**] рассматривает задачу снижения размерности в терминах вероятностной модели, решая задачу с помощью вариационного EM алгоритма. Разреженный PCA [**?**] вводит в постановку задачи lasso регуляризацию для того, чтобы сделать матрицу отображения $\mathbf{P}$ разреженной и более интерпретируемой. Нелинейный ядерный PCA [**?**] отображает исходные данные с помощью нелинейного отображения и использует RKHS для решения исходной задачи.

После нахождения матрицы отображения $\mathbf{P}$ задача (**??**) принимает вид

$$(\mathbf{B}, \mathbf{T}, \mathbf{Y}) = \Big\| \underset{m \times r}{\mathbf{Y}} - \underset{m \times l}{\mathbf{T}} \cdot \underset{l \times r}{\mathbf{B}} \Big\|_2^2 \to \min_{\mathbf{B}}. \tag{1}$$

Модель прогнозирования (**??**) в случае снижения размерности с помощью PCA принимает вид:

$$\mathbf{y} = \mathbf{B}^{\mathsf{T}}\mathbf{t} + \boldsymbol{\varepsilon} = \mathbf{B}^{\mathsf{T}}\mathbf{P}\mathbf{x} + \boldsymbol{\varepsilon} = \mathbf{x} + \boldsymbol{\varepsilon}, \quad \text{где } \; = \mathbf{B}^{\mathsf{T}}\mathbf{P}. \tag{2}$$

3.0.2 Метод частичных наименьших квадратов для задачи декодирования.

Основным недостатком метода PCA является отсутствие учёта взаимосвязи между исходными признаками $\boldsymbol{\chi}_j$ и целевыми столбцами $\boldsymbol{\nu}_j$. Метод частичных наименьших квадратов (partial least squares, PLS) проецирует исходную матрицу $\mathbf{X}$ и целевую матрицу в скрытое пространство малой размерностью l $(l < n)$. Метод PLS находит в скрытом пространстве матрицы $\mathbf{T}, \mathbf{U} \in^{m \times l}$, которые лучше всего описывают исходные матрицы $\mathbf{X}$ и $\mathbf{Y}$. При этом PLS максимизирует ковариацию между столбцами матриц $\mathbf{T}$ и $\mathbf{U}$ соответственно. Метод PLS соответствует следующей коммутативной диаграмме:

$$\begin{tikzcd}
\mathbf{x} \in^{n} \arrow[rr, "\mathbf{f}"] \arrow[dr, bend right, "\mathbf{W}"'] & & \mathbf{y} \in^{r} \arrow[dl, bend left, "\mathbf{C}"] \\
& \mathbf{t}, \mathbf{u} \in^{\ell} \arrow[ul, bend right, "\mathbf{P}"'] \arrow[ur, bend left, "\mathbf{Q}"] &
\end{tikzcd} \tag{3}$$

Метод PLS был впервые предложен в работах [? ? ?]. Подробное описание алгоритма приведено в работах [? ? ? ? ?]. В работах [? ?] приведен обзор обобщений базовой модели PLS. В работе [?] приведена модификация метода PLS для получения разреженного набора признаков.

Исходная матрица $\mathbf{X}$ и целевая матрица $\mathbf{Y}$ проецируются на скрытое пространство следующим образом:

$$\underset{m\times n}{\mathbf{X}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times n}{\mathbf{P}} + \underset{m\times n}{\mathbf{E}_{\mathbf{x}}} = \sum_{k=1}^{l} \underset{m\times 1}{\boldsymbol{\tau}_k} \cdot \underset{1\times n}{\mathbf{p}_k^{\mathsf{T}}} + \underset{m\times n}{\mathbf{E}_{\mathbf{x}}}, \tag{4}$$

$$\underset{m\times r}{\mathbf{Y}} = \underset{m\times l}{\mathbf{U}} \cdot \underset{l\times r}{\mathbf{Q}} + \underset{m\times r}{\mathbf{E}_{\mathbf{y}}} = \sum_{k=1}^{l} \underset{m\times 1}{\boldsymbol{\nu}_k} \cdot \underset{1\times r}{\mathbf{q}_k^{\mathsf{T}}} + \underset{m\times r}{\mathbf{E}_{\mathbf{y}}}. \tag{5}$$

Здесь $\mathbf{T}$ и $\mathbf{U}$ — образы исходных матриц в скрытом пространстве, причём столбцы матрицы $\mathbf{T}$ ортогональны; $\mathbf{P}$ и $\mathbf{Q}$ — матрицы перехода; $\mathbf{E}_{\mathbf{x}}$ и $\mathbf{E}_{\mathbf{y}}$ — матрицы остатков. Метод PLS восстанавливает линейную зависимость между столбцами матриц $\mathbf{T}$ и $\mathbf{U}$

$$\mathbf{U} \approx \mathbf{T}\mathbf{B}, \quad \mathbf{B} = \mathrm{diag}(\beta_k), \quad \beta_k = \boldsymbol{\nu}_k^{\mathsf{T}}\boldsymbol{\tau}_k/(\boldsymbol{\tau}_k^{\mathsf{T}}\boldsymbol{\tau}_k),$$

где $\{\boldsymbol{\tau}_k\}_{k=1}^{l}$, $\{\boldsymbol{\nu}_k\}_{k=1}^{l}$ — столбцы матриц $\mathbf{T}$ и $\mathbf{U}$ соответственно.

Метод решает следующую оптимизационную задачу:

$$\max_{\|\mathbf{p}\|_2=\|\mathbf{q}\|_2=1} [\mathrm{cov}(\mathbf{X}\mathbf{p}, \mathbf{Y}\mathbf{q})^2] = \max_{\mathbf{p},\mathbf{q}} \frac{\mathbf{p}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{Y}\mathbf{q}}{\sqrt{\mathbf{p}^{\mathsf{T}}\mathbf{p}}\sqrt{\mathbf{q}^{\mathsf{T}}\mathbf{q}}}. \tag{6}$$

4 Обзор методов моделирования законов физики

4.1 Лагранжева механика

- **Проблемы HNN**: гамильтонов формализм требует, чтобы координаты системы были «каноническими»(($\mathbf{q}, \mathbf{p}$) должны подчиняться соотношениям, заданным скобками Пуассона)

- **Решение проблемы**: использовать лагранжианы систем (обеспечивают сохранение полной энергии, могут делать это с использованием произвольных координат)

Моделирование динамики системы с помощью лагранжиана

1. Найти аналитические выражения для кинетической и потенциальной энергии (T, V)

2. Записать лагранжиан $\mathcal{L} = T - V$

3. Применить ограничение Эйлера-Лагранжа $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_j} = \frac{\partial \mathcal{L}}{\partial q_j}$

4. Решить получившуюся систему дифференциальных уравнений

4.2 Лагранжевы нейронные сети

- **Ключевая идея**: параметризовать нейронной сетью лагранжиан $\mathcal{L}$, получить выражение ограничения Эйлера-Лагранжа, обратно распространить ошибку через полученные ограничения

- Получение
$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_j} = \frac{\partial \mathcal{L}}{\partial q_j} \Rightarrow \frac{d}{dt}\nabla_{\dot{q}}\mathcal{L} = \nabla_q \mathcal{L}$$
$$\left(\nabla_{\dot{q}}\nabla_{\dot{q}}^\top \mathcal{L}\right)\ddot{q} + \left(\nabla_q \nabla_{\dot{q}}^\top \mathcal{L}\right)\dot{q} = \nabla_q \mathcal{L}$$
$$\ddot{q} = \left(\nabla_{\dot{q}}\nabla_{\dot{q}}^\top \mathcal{L}\right)^{-1}\left[\nabla_q \mathcal{L} - \left(\nabla_q \nabla_{\dot{q}}^\top \mathcal{L}\right)\dot{q}\right]$$

- Для заданного набора координат $x_t = (q_t, \dot{q}_t)$ получили метод вычисления $\dot{x}_t = (\dot{q}_t, \ddot{q}_t)$ из параметризованного лагранжиана.

- **Функция ошибки**:
$$\mathcal{L} = \left\| \dot{x}_t^{\mathcal{L}_\theta} - \dot{x}_t^{true} \right\|_2$$

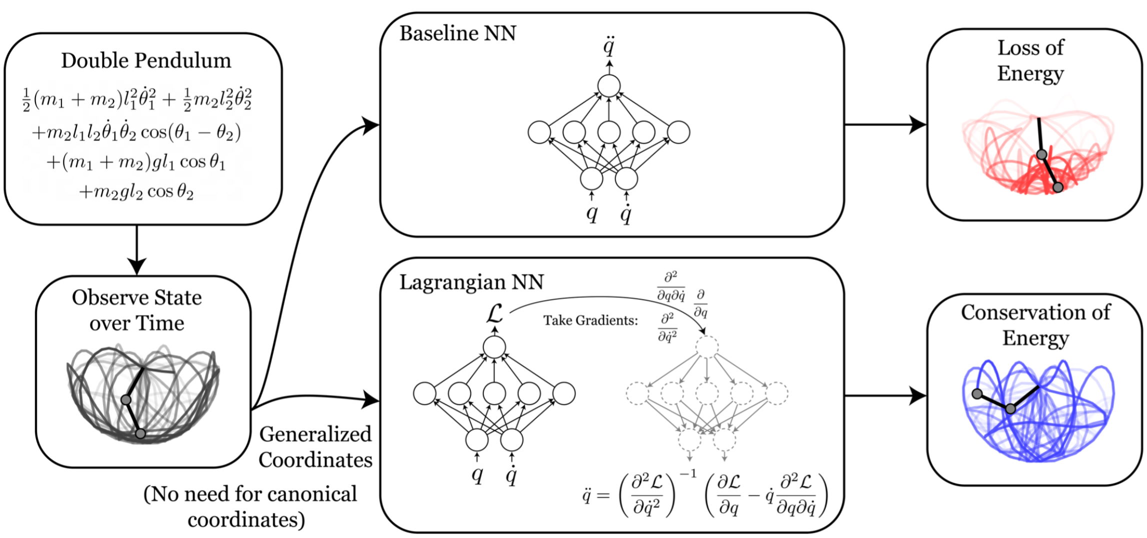

Рис. 1: Схема работы Lagrangian Neural Networks (LNN) моделирования динамики двойного маятника в сравнении с базовым решением (Baseline NN)

Сравнение

5 Вычислительный эксперимент

Целью вычислительного эксперимента является В рамках вычислительного эксперимента написан программный комплекс для решения поставленных задач [**?**].

	Neural net	Neural ODE	HNN	DeLaN	LNN (ours)
Can model dynamical systems	✓	✓	✓	✓	✓
Learns differential equations		✓	✓	✓	✓
Learns exact conservation laws			✓	✓	✓
Learns from arbitrary coords.	✓	✓		✓	✓
Learns arbitrary Lagrangians					✓

Рис. 2: Сравнение подходов моделирования физических систем нейронными сетями

6 Заключение

В работе рассмотрена задача