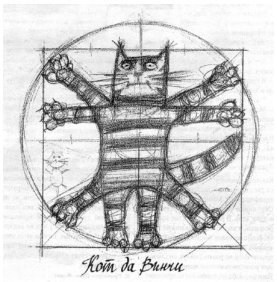
Нейроэволюционный метод псевдо-главных компонент

Юрий Цой

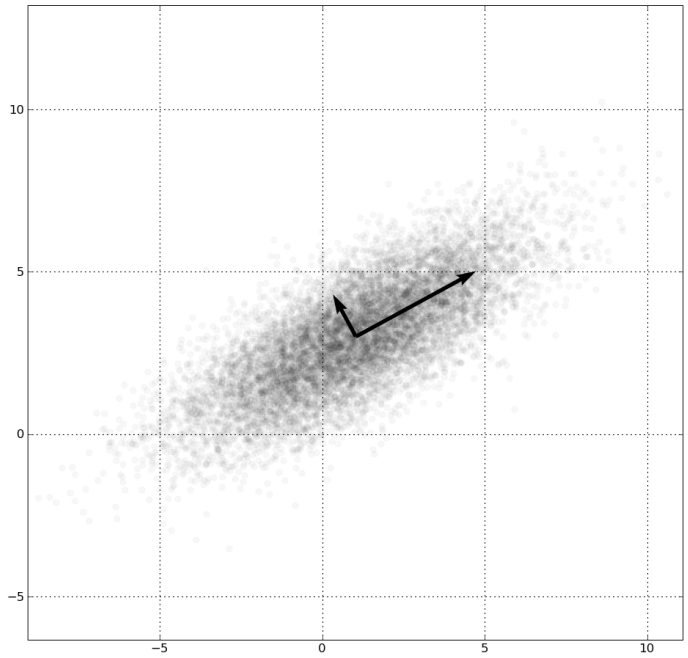
Томский политехнический университет
Томский государственный университет систем управления и радиоэлектроники
Лаборатория искусственного интеллекта A.G.I. Lab

23 января, 2012



Метод главных компонент

Метод главных компонент (МГК) – один из наиболее популярных методов сокращения размерности пространства признаков для решения задач распознавания образов и анализа данных.



Метод главных компонент

в МГК требуется найти собственные значения и векторы для матрицы автоковариации $\mathbf{C} = \mathbf{X}\mathbf{X}^T$, где $\mathbf{X}$ – матрица данных.

Плюсы

- $\mathbf{C}$ симметричная, существуют специальные хорошо исследованные численные методы.
- МГК имеет ясную геометрическую интерпретацию.

Минус

- МГК сложно параллелизовать.

Метод главных компонент

Существует нейросетевой метод для поиска главных компонент, использующий Хеббовский принцип, а именно правило Ойя.

Обобщенный алгоритм Хебба (Generalized Hebb Algorithm, GHA)

1. Инициализация весов линейной ИНС без скрытых нейронов. Количество выходов = требуемое количество компонент.
2. Для каждого вектора данных:

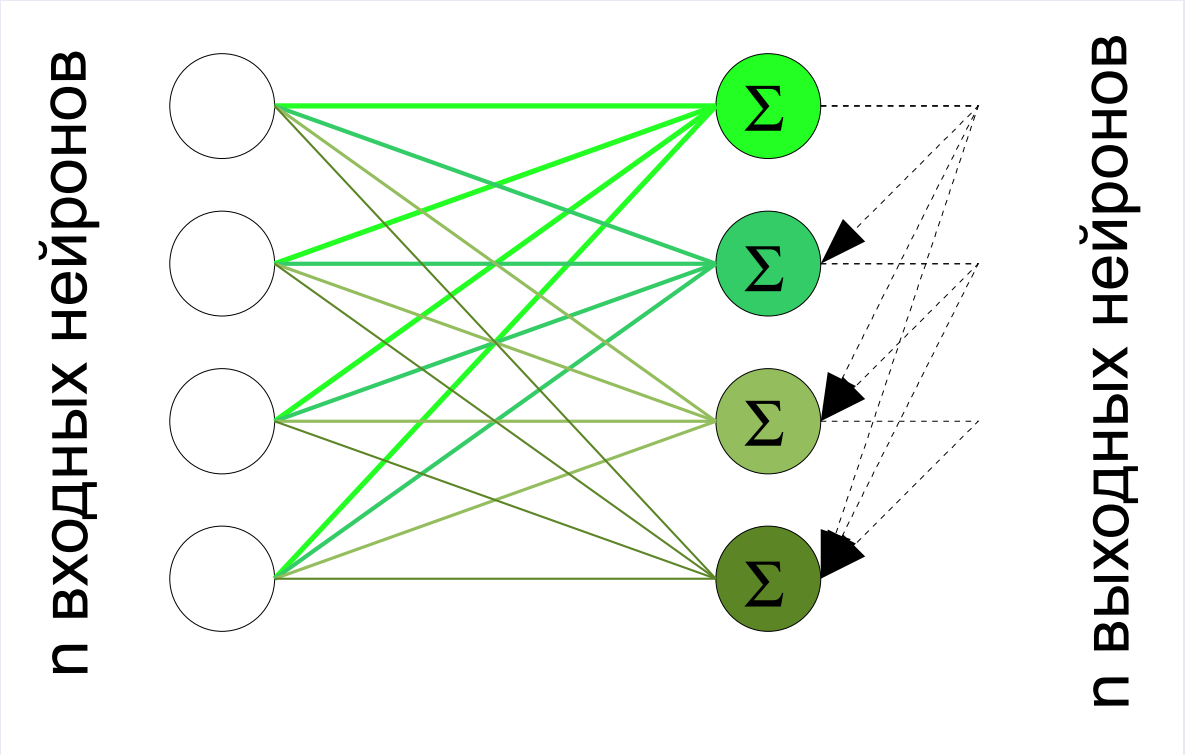
$$y_j(t) = \sum_{i=1}^{m} w_{ji}(t)x_i(t),$$

$$\Delta w_{ji}(t) = \eta\left[y_j(t)x_i(t) - y_j(t)\sum_{k=1}^{j} w_{ki}(t)y_k(t)\right],$$

3. Проверка критерия останова.

Метод главных компонент

Обобщенный алгоритм Хебба (Generalized Hebb Algorithm, GHA)



Что если . . .

Метод главных компонент

Необходимо найти **наиболее** значимые факторы (линейные подпространства) для имеющихся данных.
Другими словами, нужно избавиться от **наименее** незначимых факторов.

?

Что если одно из малозначимых подпространств найдено, можем ли мы его сразу же отбросить? Вдруг мы ошибемся и на последующих этапах оно станет более значимым?

Утверждение

Можно ли отбрасывать «приблизительные» собственные векторы?
Да! Благодаря следующему

Утверждение

Пусть $\mathbf{X} = \{\mathbf{X}_i, i = 1, \ldots, N\}, \mathbf{X}_i \in \mathcal{R}^n$ набор точек данных и $\mathbf{Q} = \{\mathbf{q}_i, i = 1, \ldots, n\}$ ортогональный базис в $\mathcal{R}^n$. Обозначим проекцию точек данных из $\mathbf{X}$ на координатный вектор $\mathbf{q}_i$ как $proj_{\mathbf{q}_i}(\mathbf{X})$, а дисперсию проекций на этот вектор через $Var(proj_{\mathbf{q}_i}(\mathbf{X}))$. Тогда сумма по всем координатам

$$\sum_{i=1,\ldots,n} Var(proj_{\mathbf{q}_i}(\mathbf{X}))$$

постоянна и не зависит от выбора $\mathbf{Q}$.

Другими словами . . .

Утверждение можно трактовать так, что сумма дисперсий проекций на координатные векторы является подобием конечного ресурса, который «распределяется» по координатным векторам.

Пусть столбцы $\hat{\mathbf{Q}}$ являются оценками собственных векторов. Малоинформативные столбцы из $\hat{\mathbf{Q}}$ будут обладать еще меньшим вкладом в описание данных, когда координаты оценок «главных» собственных векторов будут уточнены.

Можно отбросить собственные векторы, если соответствующие им собственные значения не будут удовлетворять некоторому критерию, без ущерба для уточнения координат более «важных» оценок собственных векторов. Эффективность!

Критерий?

Пусть оценки собственные векторы отсортированы в порядке убывания дисперсии проекций (т.е. значимости).

Критерий для отбрасывания оценок «плохих» собственных векторов:

$$\frac{Var(proj_{\hat{\mathbf{q}}_1}(\mathbf{X}))}{Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X}))} < \tau \tag{1}$$

где $\hat{\mathbf{q}}_i$ – оценка i-го собственного вектора, τ – порог. Типичные (эвристические) значения для τ: 5, 10, 15, ...

Так можно удалять мелоинформативные линейные подпространства, не зная точных координат более значимых собственных векторов $\Rightarrow$ метод псевдоглавных компонент, МпГК. Работает и для GHA!

Алгоритм (линейная ИНС без скрытых слоев)

1. **Инициализация** случайной популяции, каждая особь является потенциальным решением для МпГК.
2. **Оценка** каждой особи:

$$f = \alpha * \sum_{i=1,\dots,n} Var(proj_{\hat{\mathbf{q}}_i}(\mathbf{X})) \rightarrow max,$$

$$\alpha = (\hat{\mathbf{q}}_1^T \mathbf{r})^2, \mathbf{r} = \mathbf{C}\hat{\mathbf{q}}_1 / \|\mathbf{C}\hat{\mathbf{q}}_1\|.$$

   и удаление узлов ИНС, для которых выполняется критерий (1).
3. **Селекция**
4. **Скрещивание** и **Мутация**.
5. Если выполнены условия останова алгоритма, то перейти на **Шаг 6**, иначе перейти на **Шаг 2**.
6. **Вернуть** лучшее найденное решение.

Вычисление приспособленности

1. Задать веса ИНС с использованием генов особи.
2. Применить ортогонализацию Грама-Шмидта к векторам весов выходных нейронов ИНС.
3. Вычислить выходные сигналы ИНС для каждого примера из обучающего множества.
4. Вычислить дисперсию выходных сигналов ИНС.
5. Отсортировать выходные нейроны ИНС по убыванию дисперсии.
6. Скопировать полученный вектор весов ИНС обратно в хромосому.

Оператор кроссинговера

Два родителя $\rightarrow$ один потомок.
Скрещивание производится «понейронно» с использованием формулы (для k-го выходного нейрона; i-я особь полагается лучше, чем j-я):

$$\mathbf{c}^{(k)} = \mathbf{w}_i^{(k)} + \frac{|v_i^{(k)} - v_j^{(k)}|}{\|\mathbf{w}_i^{(k)} - \mathbf{w}_j^{(k)}\|}(\mathbf{w}_i^{(k)} - \mathbf{w}_j^{(k)}) \quad (2)$$

где $\mathbf{w}_i^{(k)}$ – вектор весов k-го нейрона i-й особи, $v_i^{(k)}$ – дисперсия сигналов («вес») k-го нейрона; $\|\cdot\|$ – Евклидова норма.

Выражение (2) можно использовать как аппроксимацию градиента «качества» k-го нейрона, при движении с началом в точке $\mathbf{w}_i^{(k)}$.

Цели & Тестовые задачи

Цели

1. Важно определить, возможно ли эффективное сокращение размерности пространства признаков.
2. Поскольку МпГК не направлен на вычисление точных координат главных компонент, то необходимо выяснить, как этот факт влияет на точность классификации.

Тестовые задачи (Proben1)

cancer1 (9), *card1* (51), *diabetes1* (8), *glass1* (9), *heart1* (35), *horse1* (58), *soybean1* (82), *thyroid1* (21).

Настройки

50 поколений 1000 эпох

Ранняя остановка с 10 % порогом.

Сравнение

Задача	$\tau = 5$	$\tau = 10$	$\tau = 15$	$\tau = 20$
cancer1 (9)	2,30 (1)	2,82 (1,2)	1,78 (4,6)	1,84 (6,3)
card1 (51)	16,28 (28,5)	15,41 (50,7)	15,64 (51)	15,76 (51)
diabetes1 (8)	24,95 (7,6)	25,00 (8)	25,00 (8)	25,00 (8)
glass1 (9)	36,23 (5,5)	33,02 (6,7)	32,07 (7,9)	32,26 (8,4)
heart1 (35)	21,13 (22,3)	19,91 (31,5)	20,00 (34,2)	20,04 (35)
horse1 (58)	28,79 (35,3)	29,23 (57,7)	30,66 (58)	29,56 (58)
soybean1 (82)	50,65 (3,3)	20,47 (11,7)	11,94 (26,1)	10,47 (37,3)
thyroid1 (21)	7,19 (8,9)	6,03 (16,3)	5,87 (18)	5,92 (19,8)

Таблица: Ошибки классификации (%) для различных значений τ.
Усреднено по 10 запускам

Сравнение

Задача	Proben1	GA	Prunning	$\tau = 15$
cancer1 (9)	1.38	1.24	1.1	1.78 (4.6)
card1 (51)	14.05	14.27	13.7	15.64 (51)
diabetes1 (8)	24.10	23.70	20.8	25.00 (8)
glass1 (9)	32.7	47.62	30.2	32.07 (7.9)
heart1 (35)	19.72	21.87	18.5	20.00 (34.2)
horse1 (58)	29.19	26.44	26.9	30.66 (58)
soybean1 (82)	9.06	8.47	N/A	11.94 (26.1)
thyroid1 (21)	2.32	6.12	5.7	5.87 (18)

Таблица: Значения ошибок классификации (%), полученные для других подходов на наборе Proben1

Изменение размерности

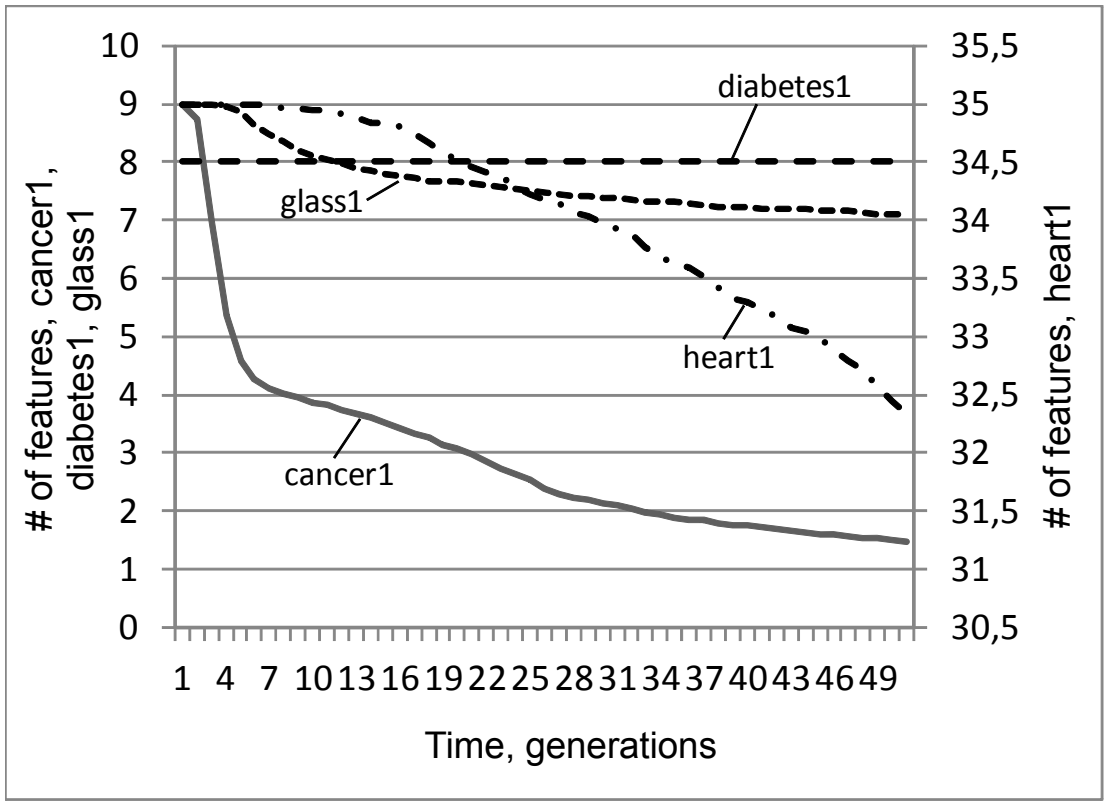


Рис.: Изменение средней размерности пространства признаков для задач *cancer1*, *diabetes1*, *glass1* и *heart1* при $\tau = 10$. Усреднено по 100 запускам.

Изменение дисперсии проекций

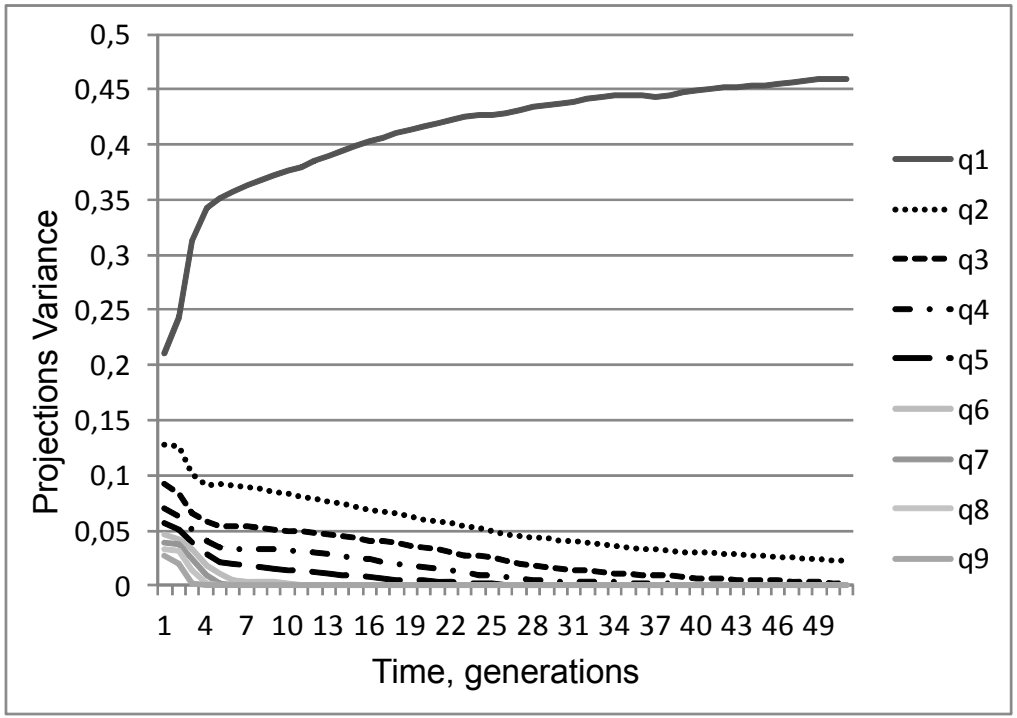


Рис.: Изменение средней дисперсии проекции данных на оценки собственных векторов для задачи *cancer1*, $\tau = 10$. Усреднено по 100 запускам.

Вычислительная сложность

Нейроэволюционный МпГК

Вычислительная сложность: $O(tKNn^2)$,
где t – количество поколений, K – размер популяции, N – количество обучающих примеров, n – исходная размерность пространства признаков.

Типичный случай 1 (*cancer1*): $K \sim N \sim t \sim n^2 : O(n^8)$
Типичный случай 2 (*card1* или *horse1*): $K \sim N \sim t \sim n : O(n^5)$
Идеальное распараллеливание: $K \sim N \sim O(1), t \sim n :\approx O(n^{3.5})$ (начинают влиять другие этапы)

Обобщенный алгоритм Хебба (GHA)

Вычислительная сложность: $O(tNn^2)$.

Типичный случай 1 (*cancer1*): $N \sim n^2, t \sim n^2 : O(n^6)$
Типичный случай 2 (*card1* или *horse1*): $N \sim n, t \sim n\sqrt{n} : O(n^{4.5})$
Идеальное распараллеливание: $N \sim O(1), t \sim n\sqrt{n} :\approx O(n^{3.5})$

Обсуждение

Вывод

Существую случаи, когда нет необходимости определять точные координаты собственных векторов матрицы ковариации обучающих данных для снижения размерности пространства признаков в задаче классификации.

Открытый вопрос

Какова цена за использование приближенных главных компонент?

Заключение

По пунктам:

- Оригинальный метод сокращения размерности с использованием нейроэволюционного МпГК.
- В методе используются специальные процедуры оценки приспособленности и скрещивания.
- Несмотря на получаемые неточные координаты собственных векторов точность классификации *сравнима* с обучением ИНС с использованием более традиционных методов и при этом обеспечивается *сокращение размерности*.

Будущие исследования:

1. Параллелизация МпГК. Наибольшие временные затраты идут на вычисление приспособленности ($\sim 75 - 80\%$).
2. Определение оценки, вызванной неточностью определения собственных векторов.

Спасибо за внимание!

Работа выполнена при поддержке РФФИ,
проекты № 09-08-00309-а, 11-07-00027-а.