Трюки с текстом

Интеллектуальный анализ данных, 20!7

По материалам open data science и Карпатых

Малютин Евгений Алексеевич

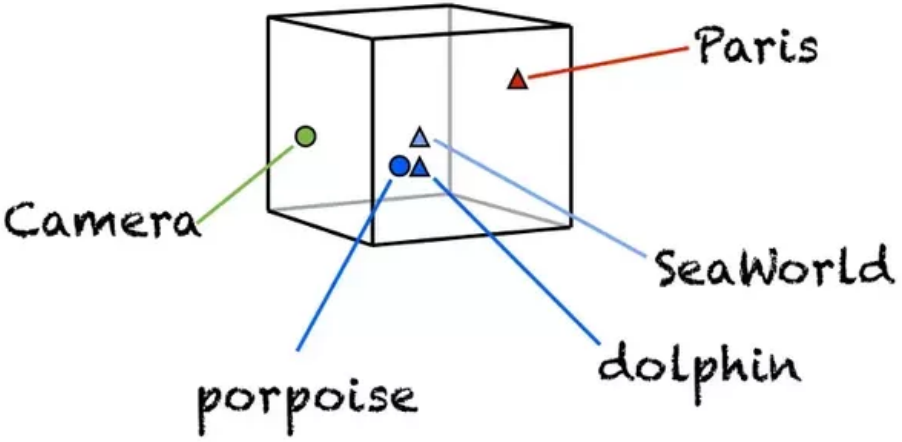
Планчик

Сегодня в программе

- Способы представления текста (которые пришли из древности)
- Способы представления здорового человека
- Генетика

Word embeddings

- Embedding — это сопоставление произвольной сущности (например, узла в графе или кусочка картинки) некоторому вектору.



Word Embedding

Как представить слова?

- Пронумеровать слова в тексте.

Word Embedding

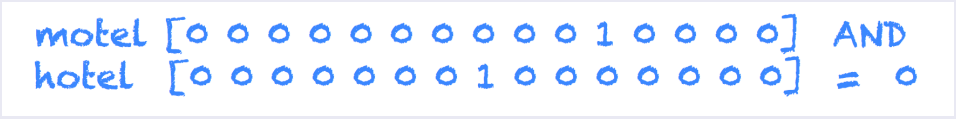
Как представить слова?

- Пронумеровать слова в тексте.
- Абсолютно не отражает семантику

Word Embedding

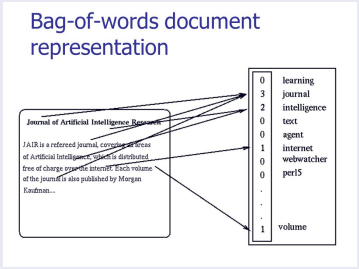
Как представить слова?

- Пронумеровать слова в тексте.
- Абсолютно не отражает семантику
- Составить векторов – One-Hot-Encoding (OHE)



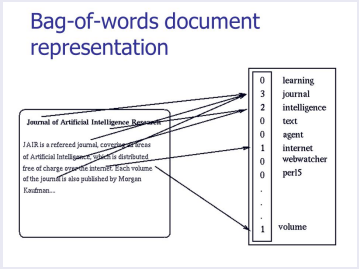
Как представить слова?

- Так, стоп, мы же изначально говорили о текстах. Bag Of Words



Как представить слова?

- Так, стоп, мы же изначально говорили о текстах. Bag Of Words

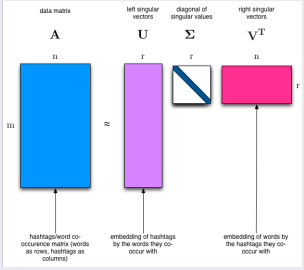


- И матрица терм-документ

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
computer	1	1	0	0	0	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0
user	0	1	1	0	1	0	0	0	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

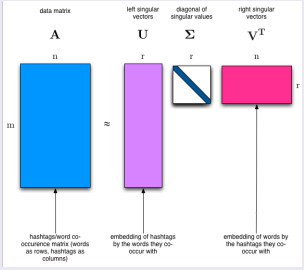
На путик к тематик-моделинг

- Матрицу "слово-документ"пытаются представить в виде произведения двух матриц "слово-тема"и "тема-документ".



На путик к тематик-моделинг

- Матрицу "слово-документ"пытаются представить в виде произведения двух матриц "слово-тема"и "тема-документ".

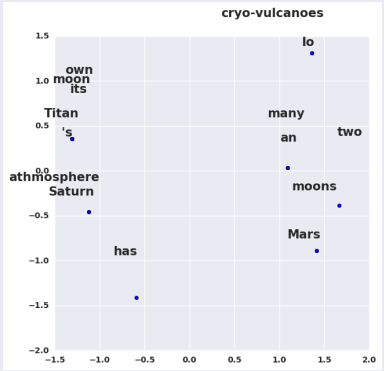


- Пример, вот есть у нас корпус:

```
s = ['Mars has an athmosphere', "Saturn 's moon Titan has its own athmosphere",
     'Mars has two moons', 'Saturn has many moons', 'Io has cryo-vulcanoes']
```

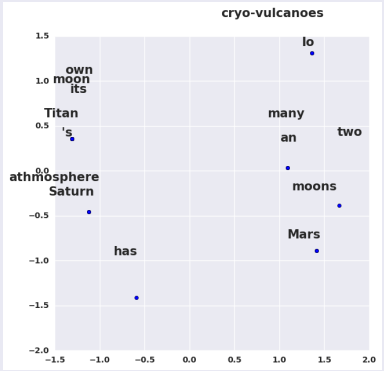
На путик к тематик-моделинг

- А получаем в итоге это:



На путик к тематик-моделинг

- А получаем в итоге это:

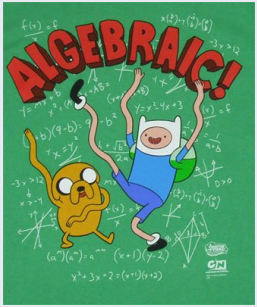


- и подумываем об этом:

$$TF - IDF(w, d, C) = \frac{count(w, d)}{count(d)} * log(\frac{\sum_{d' \in C} |w \in d'|}{|C|})$$

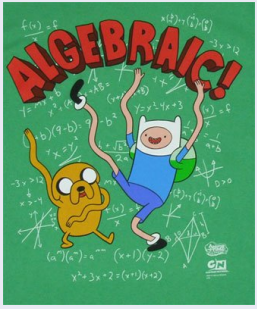
И тут пришел Томаш Миколов и всех спас

- Гипотеза локальности — “слова, которые встречаются в одинаковых окружениях, имеют близкие значения”.



И тут пришел Томаш Миколов и всех спас

- Гипотеза локальности — “слова, которые встречаются в одинаковых окружениях, имеют близкие значения”.



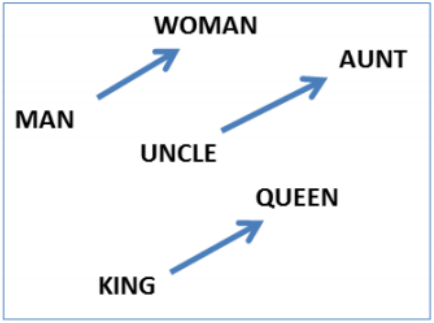
- Soft-max:

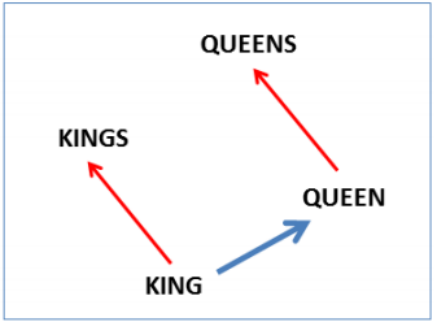
$$P(w_o|w_c) = \frac{e^{s(w_o,w_c)}}{\sum_{w_i \in V} e^{s(w_i,w_c)}};$$

w_o — вектор целевого слова, w_c – вектор контекста. А $s(w_1, w_2) : R^n \times R^n \to R$

И тут пришел Томаш Миколов и всех спас

- И ваще:





(Mikolov et al., NAACL HLT, 2013)

Negative Sampling

- В случае CBOW функционалом в задаче минимизации выступает дивергенция Кульбаха-Лейблера

$$KL(p||q) = \int p(x) log \frac{p(x)}{q(x)} dx;$$

где $p(x)$ – распределение вероятностей слов из корпуса, q(x) — распределение, которое порождает наша модель.

Negative Sampling

- В случае CBOW функционалом в задаче минимизации выступает дивергенция Кульбаха-Лейблера

$$KL(p||q) = \int p(x) log \frac{p(x)}{q(x)} dx;$$

где $p(x)$ – распределение вероятностей слов из корпуса, q(x) — распределение, которое порождает наша модель.

- $$KL(p||q) = \sum_{x \in V} p(x) log \frac{p(x)}{q(x)}$$

– в нашем, дискретном случае

Negative Sampling

- В случае CBOW функционалом в задаче минимизации выступает дивергенция Кульбаха-Лейблера

$$KL(p||q) = \int p(x) log \frac{p(x)}{q(x)} dx;$$

где $p(x)$ – распределение вероятностей слов из корпуса, q(x) — распределение, которое порождает наша модель.

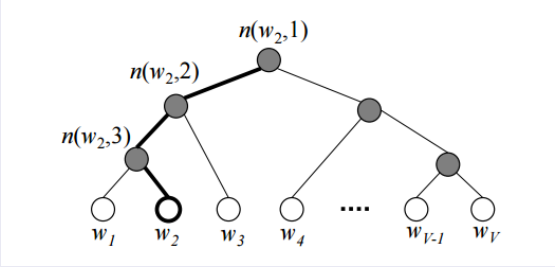
- $$KL(p||q) = \sum_{x \in V} p(x) log \frac{p(x)}{q(x)}$$

– в нашем, дискретном случае

- $$NegS(w_o) = \sum_{i=1, x_i \sim D}^{i=k} -log(1 + e^{s(x_i, w_o)}) + \sum_{j=1, x_j \sim D'}^{j=l} -log(1 + e^{-s(x_j, w_o)});$$

Hierarchical SoftMax

- Построим дерево Хаффмана со словами в узлах



- $$p(w = w_o) = \prod_{j=1}^{L(w)-1} \sigma([n(w, j+1) = lch(n(w, j))] v_{n(w,j)}^T u)$$

– где $\sigma(x)$ — функция softmax; $[true] = 1, [false] = -1$; $lch(n)$ — левый сын вершины n; $u = v_{w_I}$, если используется метод skip-gram, $u = \frac{1}{h} \sum_{k=1}^{h} v_{w_{I,k}}$ то есть, усредненный вектор контекста, если используется CBOW.

Hierarchical SoftMax

Ну вообще-то:

- На каждом шаге мы можем:
  $p(n, left) = \sigma(v_n^T u)$
  $p(n, right) = 1 - p(n, left) = 1 - \sigma(v_n^T u) = \sigma(-v_n^T u)$
- Затем на каждом шаге вероятности перемножаются ($L(w) - 1$ шагов) и получается искомая формула.
- $O(V) \to O(log(V))$

От слова – к тексту

Что делают:

- Усреднение
- TF-IDF усреднение
- Модель GloVe
  - SVD и word2vec вместе
  - требует больше текста
  - не смог найти как работает =(
- Часто отдельно коддируют *POS*, например в случае

На практике

Задача регрессии.

-

Мотивация

А может стащить идею у природы?

- Организмы эволюционируют со временем, изменяя свой генотип
- Механизм дарвиновской эволюции:
  - Родилось новое поколение.
  - Из него часть особей выросла и дала потомство, часть погибла.
  - Погибают неприспособленные, выживают приспособленные, у потомков остаются лучшие черты.

Основные компоненты

- Пространство гипотез, из которых мы должны выбрать лучшую

Основные компоненты

- Пространство гипотез, из которых мы должны выбрать лучшую
- Функция приспособленности *Fitness*

Основные компоненты

- Пространство гипотез, из которых мы должны выбрать лучшую
- Функция приспособленности *Fitness*
- Набор генетических операций, которые можно применять:

Основные компоненты

- Пространство гипотез, из которых мы должны выбрать лучшую
- Функция приспособленности *Fitness*
- Набор генетических операций, которые можно применять:
  - Операции скрещивания (кроссовер) размножение особей.

Основные компоненты

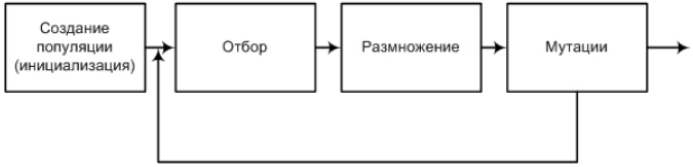
- Пространство гипотез, из которых мы должны выбрать лучшую
- Функция приспособленности *Fitness*
- Набор генетических операций, которые можно применять:
  - Операции скрещивания (кроссовер) размножение особей.
  - Мутации редкие изменения отдельных особей.

Основные компоненты

- Пространство гипотез, из которых мы должны выбрать лучшую
- Функция приспособленности *Fitness*
- Набор генетических операций, которые можно применять:
  - Операции скрещивания (кроссовер) размножение особей.
  - Мутации редкие изменения отдельных особей.
- Целевое значение $Fitness_{max}$, к которому мы стремимся (??)

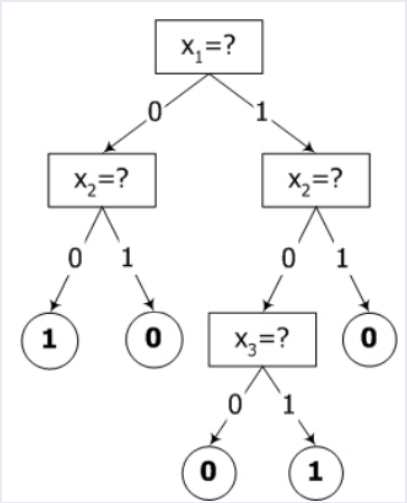
Общая схема алгоритма

- Сгенерировать начальную популяцию.
- Пока не достигнуто значение, большее $Fitness_{max}$(??):
  - Выбрать часть существующей популяции (отдавая предпочтение более приспособленным особям).
  - Применить к этой части генетические операции, породив потомков.
  - Подсчитать *Fitness* для особей новой популяции.



Представление гипотез

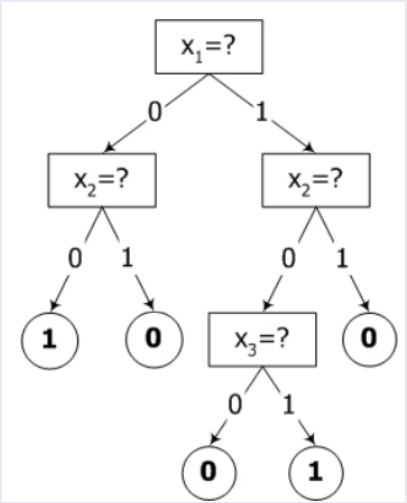
Пример(!) Чтобы успешно применять генетические алгоритмы – гипотезы необходимо преобразовать в бинарную строку



- Бинарная строка для каждой переменной: $(x_1 = 1) \wedge (x_2 = 0) \wedge (x_3 = 1) \rightarrow (f = 1) \rightarrow 1\ 0\ 1\ 1$

Представление гипотез

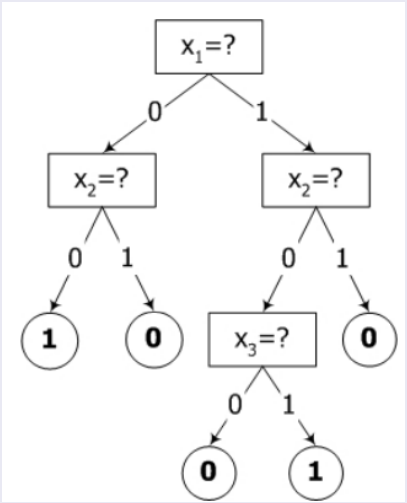
Пример(!) Чтобы успешно применять генетические алгоритмы – гипотезы необходимо преобразовать в бинарную строку



- Бинарная строка для каждой переменной: $(x_1 = 1) \wedge (x_2 = 0) \wedge (x_3 = 1) \rightarrow (f = 1) \rightarrow 1\ 0\ 1\ 1$
- А что если: $(x1 = 0) \wedge (x2 = 1) \rightarrow (f = 0)$

Представление гипотез

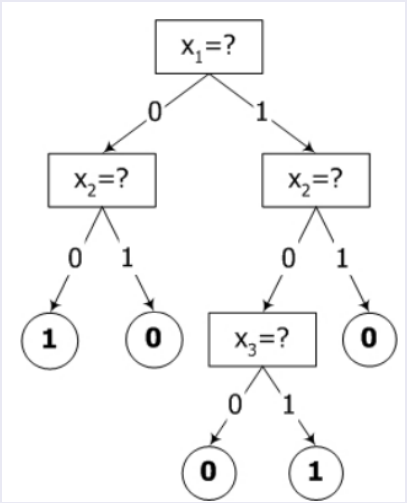
Пример(!) Чтобы успешно применять генетические алгоритмы – гипотезы необходимо преобразовать в бинарную строку



- Бинарная строка для каждой переменной: $(x_1 = 1) \wedge (x_2 = 0) \wedge (x_3 = 1) \rightarrow (f = 1) \rightarrow 1\ 0\ 1\ 1$
- А что если: $(x1 = 0) \wedge (x2 = 1) \rightarrow (f = 0)$
- Тогда кодируем как 10 01 11 1

Представление гипотез

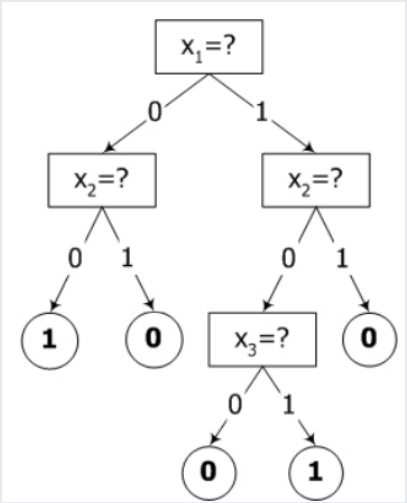
Пример(!) Чтобы успешно применять генетические алгоритмы – гипотезы необходимо преобразовать в бинарную строку



- Бинарная строка для каждой переменной: $(x_1 = 1) \wedge (x_2 = 0) \wedge (x_3 = 1) \rightarrow (f = 1) \rightarrow 1\ 0\ 1\ 1$
- А что если: $(x1 = 0) \wedge (x2 = 1) \rightarrow (f = 0)$
- Тогда кодируем как 10 01 11 1
- ...и кодируем всё дерево конкатенацией

Представление гипотез

Пример(!) Чтобы успешно применять генетические алгоритмы – гипотезы необходимо преобразовать в бинарную строку



- Бинарная строка для каждой переменной: $(x_1 = 1) \wedge (x_2 = 0) \wedge (x_3 = 1) \rightarrow (f = 1) \rightarrow 1\ 0\ 1\ 1$
- А что если: $(x1 = 0) \wedge (x2 = 1) \rightarrow (f = 0)$
- Тогда кодируем как 10 01 11 1
- ...и кодируем всё дерево конкатенацией
- или по коду Грея.

Выбор родителй

- Панмиксия (свободное скрещивание)

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность
  - критичен к численности популяции

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность
  - критичен к численности популяции
- Инбридинг

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность
  - критичен к численности популяции
- Инбридинг
- Аутбридинг

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность
  - критичен к численности популяции
- Инбридинг
- Аутбридинг
- Выбор с селекцией

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность
  - критичен к численности популяции
- Инбридинг
- Аутбридинг
- Выбор с селекцией
  - быстро сходится

Выбор родителй

- Панмиксия (свободное скрещивание)
  - универсальность
  - критичен к численности популяции
- Инбридинг
- Аутбридинг
- Выбор с селекцией
  - быстро сходится
  - попадает в локальные экстремумы

Дискретная рекомбинация

- Дискретная рекомбинация: случайным образом производится обмен хромосомами
- Промежуточная рекомбинация: Потомок = P1 + α(P2 – P1)
- Линейная рекомбинация: то же, что промежуточная, то множитель выбирается 1 раз

Кроссовер (кроссинговер)

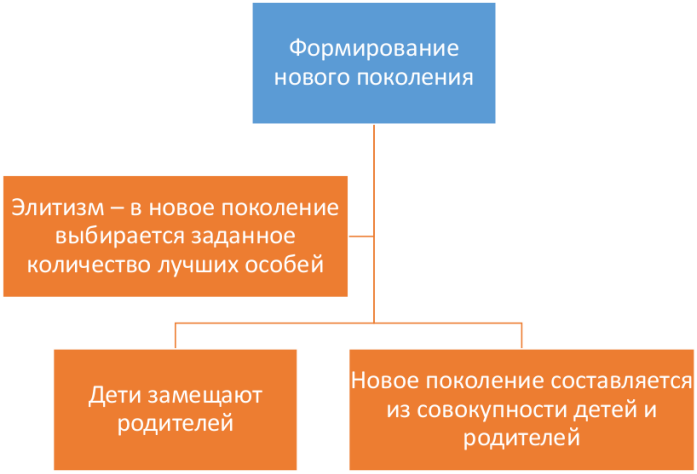
- Одноточечный: внутри хромосомы случайным образом выбирается точка, относительно которой родители обмениваются частями
- Двухточечный: выбираются 2 точки, относительно которой родители обмениваются сегментами

Основной алгоритм

Мутация

- Цель – выбивание популяции из локального экстремума
- Случайным образом меняется случайно выбранный ген в хромосоме
  $x_1, x_2, ..., x_{n-1}, x_n, x_{n+1}, ..., x_m \to x_1, x_2, ..., x_{n-1}, \widehat{x_n}, x_{n+1}, ..., x_m$
- Мутации могут происходить не только в одной точке

Основной алгоритм



Популярные модели ГА

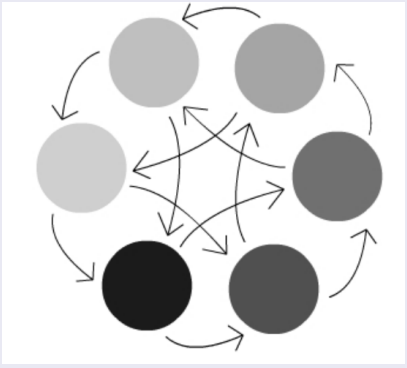
Genitor

- На каждом шаге одна пара случайных родителей создает только одного ребенка.
- Ребенок заменяет одну из худших особей популяции.

CHC

- Для нового поколения выбираются NN лучших различных особей среди родителей и детей.
- Для скрещивания все особи разбиваются на пары, но скрещиваются только те, между которыми расстояние Хэмминга больше некоторого порогового.
- При скрещивании используется так называемый HUX-оператор (Half Uniform Crossover) – каждому потомку переходит ровно половина битов каждого родителя.
- «Катастрофическая мутация» (Cataclysmic Mutation): все строки, кроме самой приспособленной, подвергаются сильной мутации (изменяется около трети битов).

Островная модель (Island Model)



- Популяция разбивается на несколько подпопуляций.
- Каждая развивается отдельно с помощью ГА.
- Изредка (например, каждые 5 поколений) происходит миграция – острова обмениваются несколькими хорошими особями.

Популярные модели ГА

Кооперативная коэволюция

- Применяется для построения композиции алгоритмов.
- В каждом поколении строится не одна, а множество популяций (на некотором подмножестве объектов и некотором подмножестве признаков). Каждому индивиду будет поставлен в соответствие некоторый базовый алгоритм.
- В ходе эволюции базовые алгоритмы обучаются кооперировать друг с другом с целью поиска наилучшего решения. При этом каждая популяция специализируется в своей области объектов и в своём подпространстве признаков.
- Функция адаптивности оценивает не качество алгоритма в отдельности, а его полезность для композиции.

Популярные модели ГА

Cooperative Coevolution Ensemble Learner

- Плюсы:
- Сравним с бустингом или бэггингом по качеству классификации
- Строит короткие композиции
- Применим к любым базовым алгоритмам
- Автоматически отбирает информативные объекты и признаки
- Минусы
- Сложен в реализации
- Большое число параметров
- Долго работает (решается с помощью распараллеливания)

Генетические алгоритмы

Плюсы

- Большое число свободных параметров, позволяющим эффективно встраивать эвристики;
- Эффективное распараллеливание;
- Работает заведомо не хуже абсолютно случайного поиска;
- Связь с биологией, дающая некоторую надежду на исключительную эффективность ГА в природе.

Минусы

- Большое количество свободных параметров, которое превращает "работу с ГА"в "игру с ГА";
- Недоказанность сходимости;
- В простых целевых функциях (гладкие, один экстремум и т.п.) генетика всегда проигрывает по скорости простым алгоритмам поиска.