
Aprendizagem de Máquina

Algoritmos Evolutivos e Inteligência de Enxames

Telmo de Menezes e Silva Filho

tmfilho@gmail.com/telmo@de.ufpb.br

www.de.ufpb.br

UFPB



Departamento de
ESTATÍSTICA

Sumário

Introdução

Metaheurísticas

Algoritmos Evolutivos

Otimização por Enxames de Partículas (PSO)

Para Terminar



Introdução

- ▶ Comportamento racional = agir corretamente na hora certa
- ▶ Agir corretamente = fazer o que é esperado para atingir seus objetivos, dada a informação disponível
- ▶ Não necessariamente envolve pensamentos (raciocínios lógicos)
 - ▶ A ação pode ser resultado de um reflexo
 - ▶ Ex.: Tirar a mão de um objeto quente
 - ▶ O raciocínio lógico deve ser usado para alcançar um objetivo



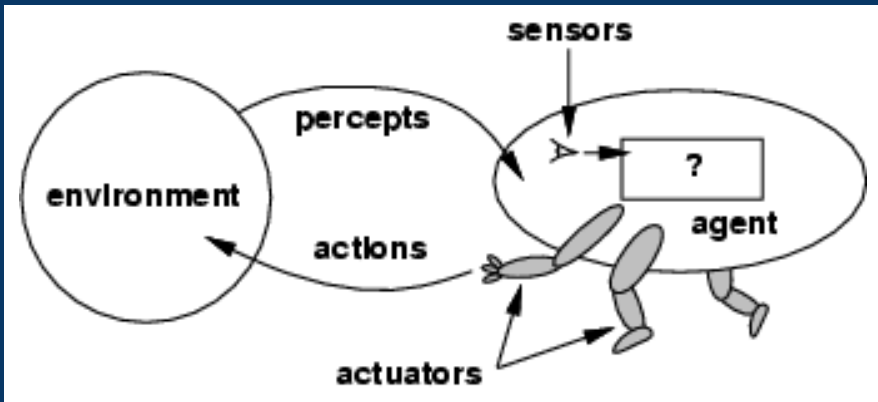
Agentes Racionais

- ▶ Um **agente** é algo que percebe e age
- ▶ De forma abstrata, um agente é uma função que mapeia uma sequência de percepções em uma ação
- ▶ Para cada tipo de ambiente e tarefa, buscamos o agente com a melhor performance
- ▶ Às vezes limitações computacionais impedem a racionalidade perfeita
 - ▶ Racionalidade limitada: fazer o melhor possível dentro das limitações computacionais



Agentes Racionais

- ▶ Um agente é algo capaz de perceber seu ambiente por meio de **sensores** e de agir sobre esse ambiente por meio de **atuadores**



Exemplos

- ▶ Agente humano
 - ▶ Sensores: Olhos, ouvidos e outros órgãos
 - ▶ Atuadores: Mãos, pernas, boca e outras partes do corpo
- ▶ Agente robótico
 - ▶ Sensores: câmeras e detectores de infravermelho
 - ▶ Atuadores: vários motores
- ▶ Agente de software
 - ▶ Sensores: entrada do teclado, conteúdo de arquivos e pacotes vindos da rede
 - ▶ Atuadores: tela, disco, envio de pacotes pela rede



Inteligência de Enxames

- ▶ Seria possível construir sistemas com comportamento inteligente inspirados no comportamento de insetos sociais?
 - ▶ Comportamento coletivo descentralizado, auto-organizado
 - ▶ População de agentes simples que interagem localmente entre si e com o ambiente
 - ▶ Normalmente não há controle centralizado
 - ▶ Interações locais produzem comportamento global



Inteligência de Enxames

Pesquisa e projeto de algoritmos ou dispositivos para solução de problemas distribuídos inspirados no comportamento coletivo de colônias de insetos sociais e outras sociedades de animais



Metaheurísticas



Heurística

- ▶ É um método ou processo criado com o objetivo de encontrar soluções para um problema. É um procedimento simplificador (embora não simplista) que, em face de questões difíceis envolve a substituição destas por outras de resolução mais fácil a fim de encontrar respostas viáveis, ainda que imperfeitas



Metaheurística

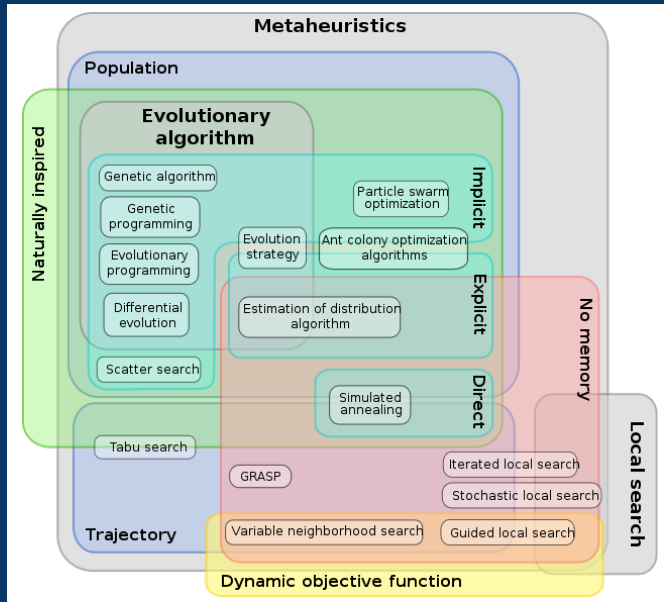
- ▶ É um método heurístico para resolver de forma genérica problemas de otimização (normalmente da área de otimização combinatória)
- ▶ Metaheurísticas são geralmente aplicadas a problemas para os quais não se conhecem algoritmos eficientes.
- ▶ Utilizam combinação de escolhas aleatórias e conhecimento histórico dos resultados anteriores adquiridos pelo método para se guiarem e realizar suas buscas pelo espaço de pesquisa em vizinhanças dentro do espaço de pesquisa, o que evita paradas prematuras em ótimos locais



Metaheurística

- ▶ Iterativamente melhorar um conjunto de soluções
- ▶ Pouco conhecimento do problema
- ▶ Precisa poder distinguir boas soluções
- ▶ Geralmente encontra boas soluções
- ▶ Não é garantido encontrar o ótimo global
- ▶ Adaptáveis: parâmetros ajustáveis





Quando aplicar

- ▶ Algoritmos usados em problemas nos quais existe pouca informação:
 - ▶ não se conhece a forma de uma solução ótima
 - ▶ não se sabe como encontrá-la
- ▶ Uma exploração completa é impossível devido ao tamanho do espaço
- ▶ Porém se você tem uma solução candidata, ela pode ser avaliada



Aplicações

- ▶ Evolução de redes neurais artificiais
- ▶ Agrupamento
- ▶ Escalonamento de tarefas (Multi-objective Job shop scheduling)
- ▶ Roteamento de veículos (Capacitated Vehicle Routing)
- ▶ Definição de formato e tamanho para componentes
- ▶ Caminho ótimo para operações de perfuração automatizadas
- ▶ Posicionamento de bases em computação móvel



Algoritmos Evolutivos



Algoritmos Evolutivos

- ▶ Uma população de indivíduos existe em um ambiente com fontes limitadas
- ▶ A competição por estas fontes causa a seleção dos indivíduos que melhor se adaptam ao ambiente
- ▶ Estes indivíduos atuam como sementes para as novas gerações através de recombinação e mutação
- ▶ Os novos indivíduos têm seus fitness avaliados e competem (inclusive com seus pais) pela sobrevivência
- ▶ A Seleção Natural causa um aumento no fitness da população.

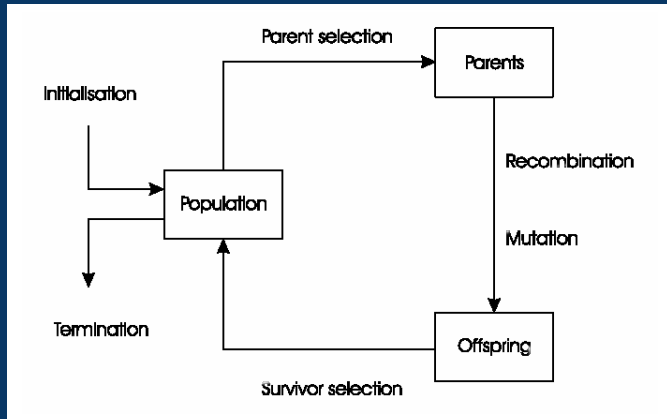


Algoritmos Evolutivos

- ▶ Os AEs estão na categoria dos algoritmos de geração e teste
- ▶ Os AEs são estocásticos e baseados em população
- ▶ Os Operadores de Variação (ou operadores genéticos – recombinação e mutação) criam a diversidade da população
- ▶ A Seleção reduz a diversidade e atua com a força propulsora para a qualidade



Esquema AE



Pseudocódigo Típico

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```



Tipos de AEs

- ▶ Historicamente, diferentes representações de soluções têm sido associadas com vários AEs, sendo os principais:
 - ▶ Strings Binárias: Algoritmos Genéticos
 - ▶ Vetores de Reais: Estratégias Evolutivas



Tipos de AEs

- ▶ Conceitualmente, as diferenças são bastante irrelevantes
- ▶ Já para a técnica de implementação, as diferenças são bem relevantes
- ▶ Cada uma das técnicas tem uma representação distinta que deve se adequar ao problema
- ▶ A escolha dos operadores de variação deve se adequar às representações escolhidas
- ▶ Os operadores de seleção em todos os casos usam apenas informações do fitness e são independentes das representações



Representações

- ▶ Dado um problema que será abordado por um AE
 - ▶ Uma solução candidata (indivíduo) existe em um espaço fenotípico (ou espaço de indivíduos)
 - ▶ Os indivíduos são codificados em cromossomos, que geram um espaço genotípico
 - ▶ Os cromossomos contêm genes, que são posições (usualmente fixadas) chamadas de locus, tendo um determinado valor
 - ▶ De forma a garantir um ótimo global, toda possível solução deve ser representada no espaço genotípico



Função de Avaliação - Fitness

- ▶ Representa as condições as quais a população deve se adaptar
- ▶ Assinala um valor real para o fitness (ou adaptação) de cada fenótipo, formando o critério base para a seleção
- ▶ Quanto maior o poder de discriminação melhor



Função de Avaliação - Fitness

- ▶ Tipicamente, quando se fala em fitness deseja-se uma maximização
 - ▶ Ex.: Imagine que deseja-se medir a qualidade de uma solução através de um ERRO. Assim,

$$\text{fitness} = \frac{1}{1 + \text{ERRO}}$$

- ▶ Quanto maior a função fitness, melhor a solução
- ▶ Porém pode-se ter também $\text{fitness} = \text{ERRO}$
 - ▶ Quanto menor melhor (menos intuitivo quando se pensa no termo fitness)



População

- ▶ É um conjunto de possíveis soluções (estáticas)
- ▶ Usualmente tem um tamanho fixo e é um multi-conjunto de genótipos
- ▶ Alguns AEs sofisticados também consideram estruturas espaciais sobre a população, e.g., uma grade
 - ▶ Este tipo de característica é muito comum quando o AE está sobre uma arquitetura paralela de processamento
 - ▶ Também é usado para se referir a questões como distância e vizinhança



População

- ▶ Os operadores de seleção usualmente utilizam unicamente informações da população
 - ▶ As probabilidades de seleção são referentes à geração atual
- ▶ A diversidade de uma população refere-se ao diferente número de fitness, fenótipos e genótipos presentes
- ▶ Geralmente, a população é fixa e se renova com base no fitness dos indivíduos



Mecanismo de Seleção dos Pais

- ▶ Assinala a probabilidade dos indivíduos atuarem como pais, o que depende dos seus respectivos valores de Fitness
 - ▶ Quanto mais apto, maior o poder de reprodução
- ▶ O mecanismo de seleção é usualmente probabilístico
 - ▶ As melhores soluções tem maiores chances de se tornarem pais do que as soluções de baixa qualidade
 - ▶ Porém, nenhum indivíduo tem probabilidade zero de seleção!
- ▶ A natureza estocástica deste processo auxilia na fuga de máximos locais

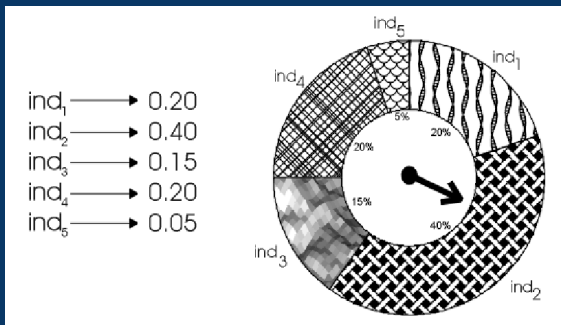


Método Da Roleta

- Assinala para cada indivíduo uma fatia da roleta proporcional ao seu fitness

$$p(ind_i) = \frac{fitness(ind_i)}{\sum_{pop} fitness(ind)}$$

- Gira-se a roleta para selecionar um indivíduo



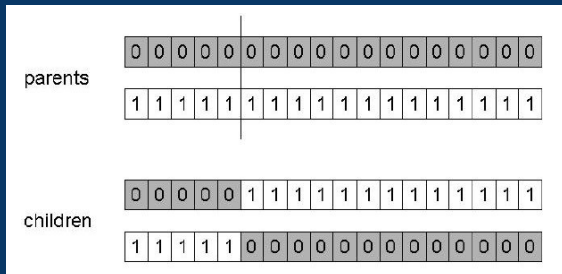
Recombinação

- ▶ Mistura informações: Pais \rightarrow Prole
- ▶ O processo de mistura (partes dos pais) é estocástico
- ▶ A maior parte da prole é esperada ser pior, ou de mesma qualidade dos pais
- ▶ Entretanto, este processo também garante que alguns filhos serão melhores que os pais devido a combinação de elementos de genótipos que conduzam a boas características
- ▶ Este princípio tem sido utilizado pela Natureza por milhões de anos



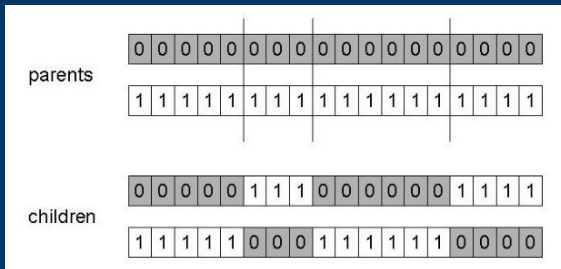
Cruzamento (Crossover) de 1 Ponto

- ▶ Escolhe-se aleatoriamente um ponto sobre os dois pais
- ▶ Divide-se os pais nesse ponto sorteado
- ▶ Cria-se os filhos através da troca dos “pedaços” dos pais
- ▶ Crossover geralmente é feito com uma probabilidade p_c , tipicamente escolhida no intervalo $[0.6, 0.9]$



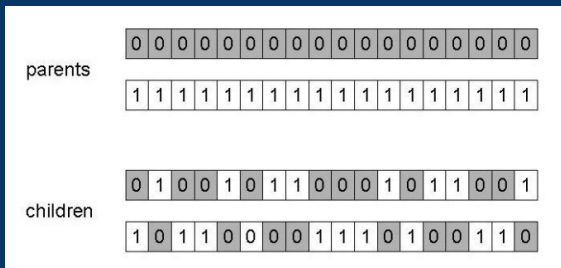
Cruzamento (Crossover) de n Pontos

- ▶ Escolha n pontos aleatórios de corte
- ▶ Divida os cromossomos ao longo destes pontos
- ▶ Una as partes, alternando as contribuições dos pais



Cruzamento (Crossover) Uniforme

- ▶ Assinale “CARA” para um pai, e “COROA” para o outro
- ▶ Arremesse uma moeda para a escolha de cada gene do filho
- ▶ Faça uma cópia inversa para o segundo filho
- ▶ A herança dos genes fica independente da posição



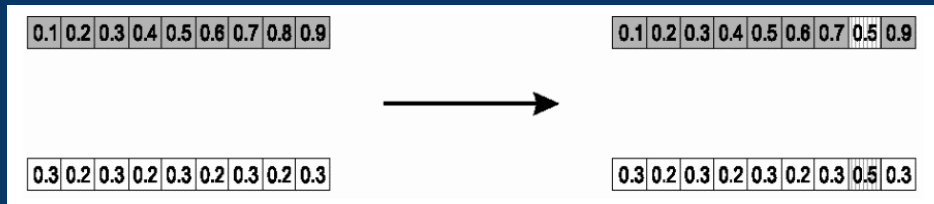
Para Outras Representações

- ▶ Para representações com números reais ou inteiros, pode-se usar as mesmas formas de cruzamento que vimos até agora
- ▶ Para o caso real, podemos também realizar combinações aritméticas entre os pais $z_i = \alpha x_i + (1 - \alpha)y_i$, onde $0 \leq \alpha \leq 1$
- ▶ α pode ser
 - ▶ Constante: Cruzamento Aritmético Uniforme
 - ▶ Variável (varia com a geração)
 - ▶ Aleatório, sendo um diferente valor para cada utilização



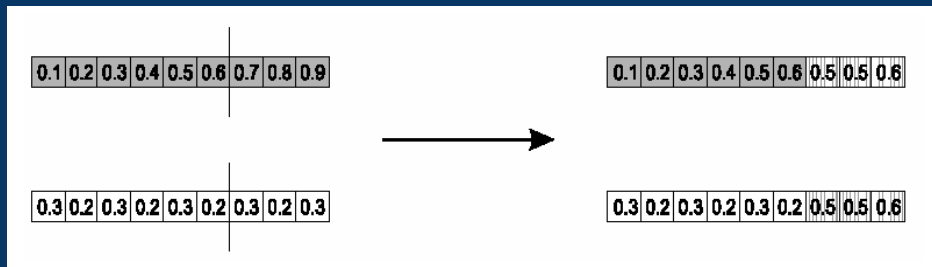
Cruzamento Aritmético Simples

- ▶ Selecciona-se aleatoriamente apenas um gene para aplicar o cruzamento aritmético
- ▶ O outro filho fica com o resultado contrário, ou seja $z_i = (1 - \alpha)x_i + \alpha y_i$
- ▶ Com $\alpha = 0.5$, teríamos:



Cruzamento Aritmético por Ponto

- ▶ Aplique o cruzamento aritmético a partir de um ponto aleatório dos cromossomos pais
- ▶ Com $\alpha = 0.5$, teríamos:



Cruzamento Aritmético Completo

- ▶ Mais comumente utilizado
- ▶ Com $\alpha = 0.5$, teríamos:

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----



0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----



Mutação

- ▶ Atua sobre um genótipo e gera outro genótipo
- ▶ Elemento essencial de aleatoriedade (diversidade)
- ▶ Garante conectividade ao espaço de busca
 - ▶ Dado um número infinito de gerações, todas posições do espaço de busca são alcançáveis



Mutação

- ▶ Altere cada gene independentemente com probabilidade p_m
- ▶ p_m é tipicamente escolhida entre $1 / (\text{tamanho_pop})$ e $1 / (\text{tamanho_crom})$

parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1



Mutação em Números Reais

- ▶ Para cromossomos de números reais, podemos selecionar o valor resultante da mutação uniformemente de $[LB_i, UB_i]$, onde LB_i e UB_i são o valor mínimo e máximo possíveis para a i -ésima posição do cromossomo
- ▶ Porém, o método mais comum é somar um número aleatório para cada variável separadamente, a partir de uma distribuição Gaussiana $\mathcal{N}(0, \sigma^2)$, garantindo o intervalo de validade de cada variável



Seleção por Sobrevivência

- ▶ O mesmo que recolocação
- ▶ A maior parte dos AEs usam uma população de tamanho fixo, necessitando de uma forma para garantir as novas gerações
- ▶ Geralmente determinística
 - ▶ Baseada em fitness: e.g., descartar o menos apto
 - ▶ Baseado em Geração: extingue os pais para a sobrevivência dos filhos



Inicialização

- ▶ A inicialização do AE geralmente é aleatória
- ▶ Necessita que seja garantido a possibilidade da varredura e mistura de todos os possíveis valores dos genes
- ▶ É possível a utilização e inclusão de soluções existentes, ou heurísticas específicas ao problema para “semear” a população



Condição de Término

- ▶ A condição de término deve ser checada a cada geração
 - ▶ Busca por um fitness mínimo
 - ▶ Quantidade máxima de gerações permitida
 - ▶ Queda a um nível mínimo de diversidade
 - ▶ Quantidade máxima de gerações sem aumento do fitness
 - ▶ Busca por alguma característica específica do problema



Exemplo

- ▶ Encontrar o máximo da função x^2 no intervalo inteiro $[0, 1, \dots, 31]$
 - ▶ Representação binária: $01101_2 = 13_{10}$
 - ▶ Tamanho da população: 4
 - ▶ Cruzamento de 1 ponto, mutação de 1 bit



Exemplo

► População inicial:

$$01101 = 13_{10} \rightarrow x^2 = 169$$

$$11000 = 24_{10} \rightarrow x^2 = 576$$

$$01000 = 8_{10} \rightarrow x^2 = 64$$

$$10011 = 19_{10} \rightarrow x^2 = 361$$



Exemplo – Evolução

Segunda
Geração

		X	f(x)
1	11011	27	729
2	11000	24	576
3	10111	23	529
4	10101	21	441

Terceira
Geração

		X	f(x)
1	11011	27	729
2	10111	23	529
3	01111	15	225
4	00111	7	49



Exemplo – Evolução

Quarta
Geração

		X	f(x)
1	11111	31	961
2	11011	27	729
3	10111	23	529
4	10111	23	529

Quinta
Geração

		X	f(x)
1	11111	31	961
2	11111	31	961
3	11111	31	961
4	10111	23	529



Otimização por Enxames de Partículas (PSO)



PSO - Particle Swarm Optimization

- ▶ Desenvolvida por James Kennedy (psicólogo) e Russell Eberhart (engenheiro), com base no comportamento de pássaros em revoadas modelado pelo biólogo Frank Heppner
- ▶ Inspirado no comportamento e na dinâmica dos movimentos dos pássaros, insetos e peixes
- ▶ Originalmente desenvolvido para problemas de otimização com variáveis contínuas
- ▶ Desempenho similar ao dos Algoritmos Genéticos



PSO - Particle Swarm Optimization



- ▶ PSO é um método baseado em população, como o Algoritmo Genético
 - ▶ Entretanto, o conceito básico é cooperação em vez da rivalidade
- ▶ Apesar da semelhança com AG, esta técnica não usa operadores genéticos (crossover, mutação, etc)
- ▶ Uma partícula movimenta-se com velocidade
 - ▶ Usando a própria experiência
 - ▶ Além da experiência de todas as partículas



Elementos do PSO

- ▶ A população de agentes.
- ▶ \mathbf{x}_i : posição do agente a_i no espaço de soluções
- ▶ f : função de avaliação
- ▶ \mathbf{v}_i : velocidade do agente a_i .
- ▶ $V(a_i)$: conjunto fixo de vizinhos do agente a_i .
 - ▶ Todos os agentes estão conectados, direta ou indiretamente



Elementos do PSO

- ▶ \mathbf{p}_i : $pbest_i$, a melhor posição encontrada pela partícula i
- ▶ \mathbf{g} : $gbest$, melhor posição encontrada por todas as partículas
- ▶ c_1 e c_2 : parâmetros cognitivo e social
- ▶ w e c_2 : parâmetro de inércia
- ▶ r_{1j} e r_{2j} : números aleatórios entre $[0, 1]$



Algoritmo do PSO

```
inicialize a nuvem de partículas
repita
  para  $i = 1$  até  $m$ 
    se  $f(\mathbf{x}_i) < f(\mathbf{p}_i)$  então
       $\mathbf{p}_i = \mathbf{x}_i$ 
      se  $f(\mathbf{x}_i) < f(\mathbf{g})$  então
         $\mathbf{g} = \mathbf{x}_i$ 
      fim se
    fim se
  para  $j = 1$  até  $n$ 
     $r_{1j} = \text{rand}()$  ,  $r_{2j} = \text{rand}()$ 
     $v_{ij} = wv_{ij} + c_1 r_{1j}(p_{ij} - x_{ij}) + c_2 r_{2j}(g_j - x_{ij})$ 
  fim para
   $\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i$ 
fim para
até satisfazer o critério de parada
```

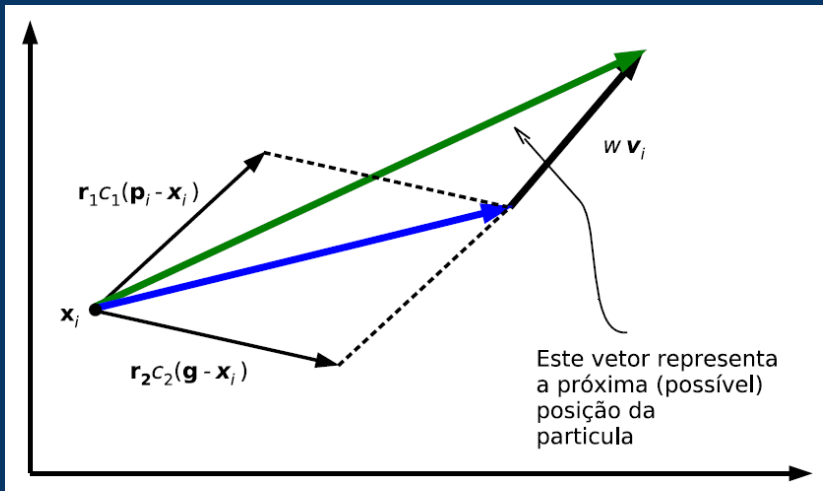


Termos da Velocidade

- ▶ Termo de **inércia**:
 - ▶ Força a partícula a mover-se na mesma direção
 - ▶ Tendência para seguir a própria direção com a mesma velocidade
- ▶ Termo **cognitivo**:
 - ▶ Melhora o indivíduo
 - ▶ Força a partícula a voltar a uma posição anterior que seja melhor do que a atual
 - ▶ Tendência conservativa
- ▶ Termo **social**:
 - ▶ Força a partícula a seguir a direção de seus melhores vizinhos
 - ▶ Como em todo rebanho, mas seguindo os melhores



Interpretação Geométrica



Prós e Contras do PSO

▶ **Prós:**

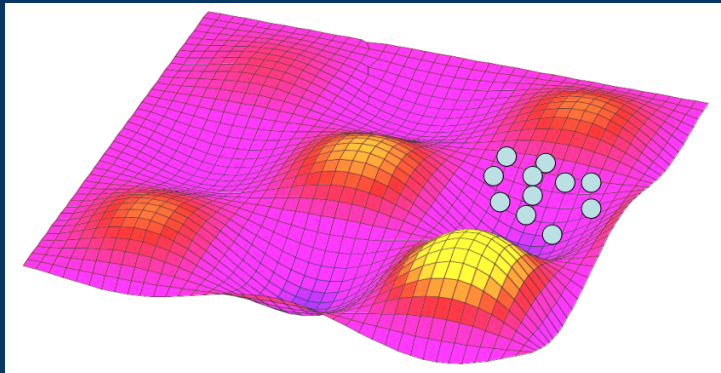
- ▶ Insensível a mudança de escala das variáveis
- ▶ Implementação simples
- ▶ Adaptável a computadores paralelos
- ▶ Não requer cálculo de derivadas
- ▶ Poucos parâmetros para serem definidos pelo usuário
- ▶ Bom para encontrar o mínimo global

▶ **Contras:**

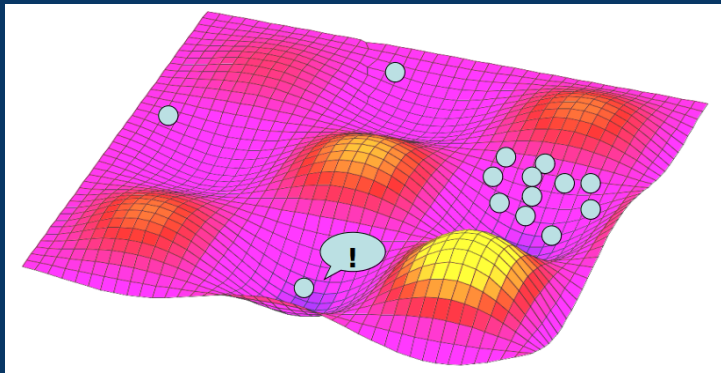
- ▶ Rápido para localizar a bacia de atração das boas soluções, mas lento no ajuste fino da solução (como nos algoritmos genéticos)
- ▶ As partículas tendem a se agrupar, ou seja, devido a uma convergência rápida demais, a solução fica presa em um ponto ótimo local



- ▶ Problema: Partículas tendem a se agrupar, reduzindo a capacidade de movimentos da nuvem para soluções melhores



- Solução: reiniciar algumas partículas em novas posições, podendo resultar em soluções melhores. As demais partículas podem mover-se para estas áreas



Alguns Detalhes de Implementação

- ▶ Limites superior e inferior: $x_{ij} \in [LB_j, UB_j]$, caso x_{ij} saia desse intervalo, fazer $x_{ij} = LB_j$ ou $x_{ij} = UB_j$, conforme o caso, além de $v_{ij} = 0$
- ▶ Velocidade máxima: $-v_{max} \leq v_{ij} \leq v_{max}$
- ▶ Em geral, não é necessário armazenar o \mathbf{g}
 - ▶ Basta armazenar o índice i tal que $\mathbf{p}_i = \mathbf{g}$



Para Terminar

- ▶ Algoritmos Genéticos/Evolutivos e de enxames **não são** em si métodos **preditivos**, mas podem ser usados para treiná-los
 - ▶ Exemplo: ao invés de ajustar os coeficientes e intercepto de uma regressão logística para um problema com p variáveis usando métodos tradicionais, podemos:
 - ▶ criar uma população aleatória de possíveis soluções como vetores de $p + 1$ posições
 - ▶ usar os algoritmos que vimos hoje para encontrar a melhor solução, tendo como alvo alguma medida de desempenho, como acurácia, precisão ou cobertura



Para Terminar

- ▶ Também é muito comum usá-los para otimizar os hiperparâmetros (tuning) dos modelos de AM
- ▶ Inicializa-se uma população aleatória com combinações de valores para os hiperparâmetros desejados
- ▶ A função de avaliação é uma medida de desempenho do modelo ajustado tendo como base a combinação de hiperparâmetros contida na partícula
 - ▶ Exemplo: para uma RNA cada partícula/cromossomo pode representar a quantidade de neurônios em cada camada escondida: [15, 5, 0, 30, 12]
- ▶ Naturalmente isso pode ser muito **custoso**, mas é um método muito utilizado pelas gigantes da computação (Google, Facebook, Amazon) para encontrar as arquiteturas de redes que melhor resolvem os problemas



Sugestão de Atividade

- ▶ Implemente o PSO e o Algoritmo Genético para dados reais
- ▶ Teste suas implementações usando funções clássicas de benchmark para algoritmos populacionais:
https://en.wikipedia.org/wiki/Test_functions_for_optimization
- ▶ Trabalhe no projeto :D





Aprendizagem de Máquina

Algoritmos Evolutivos e Inteligência de Enxames

Telmo de Menezes e Silva Filho

tmfilho@gmail.com/telmo@de.ufpb.br

www.de.ufpb.br

UFPB



Departamento de
ESTATÍSTICA