

Multi-Objective Optimization and Learning

Outline

- Part 1: Multi-Objective Optimization (多目标优化)
 - Introduction
 - Multi-Objective Evolutionary Algorithms
- Part 2: Multi-Objective Learning
 - Introduction
 - Diverse and Accurate Ensemble Learning Algorithm
 - Class Imbalance Learning

Multi-Objective Optimization

- ❖ What is Multi-Objective Optimization
- ❖ Pareto Dominance

Multi-Objective Optimization (MOO)

- Compared to “optimisation” that we have seen previously:
 - More than one objective to be optimised, with or without constraints.

$$\min/\max F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$$

$$s.t. g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, J$$

$$h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \dots, n$$

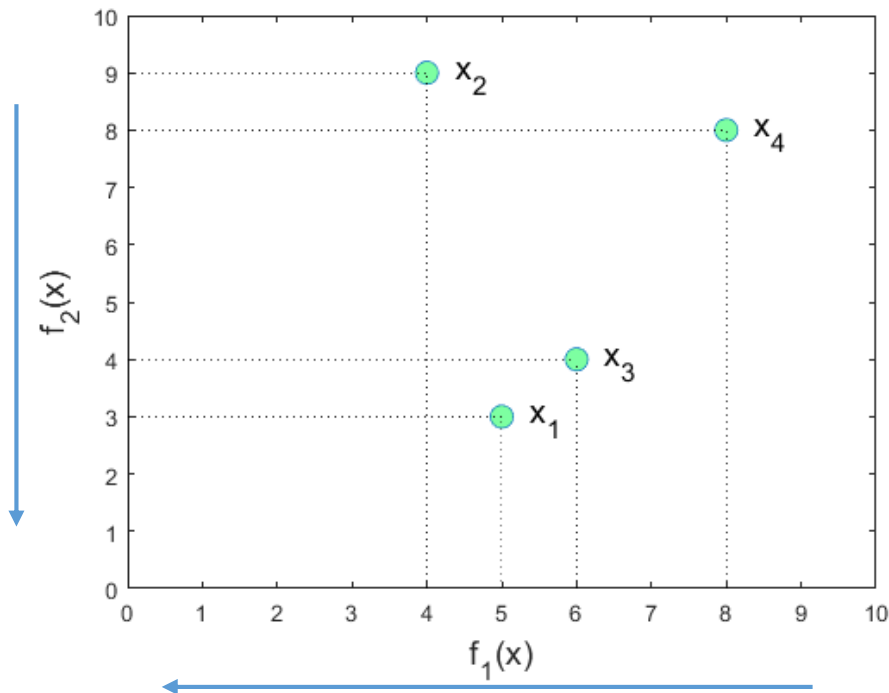
Lower
bound

Upper
bound

➤ \mathbf{x} is a vector of continuous, discrete or mixed variables.

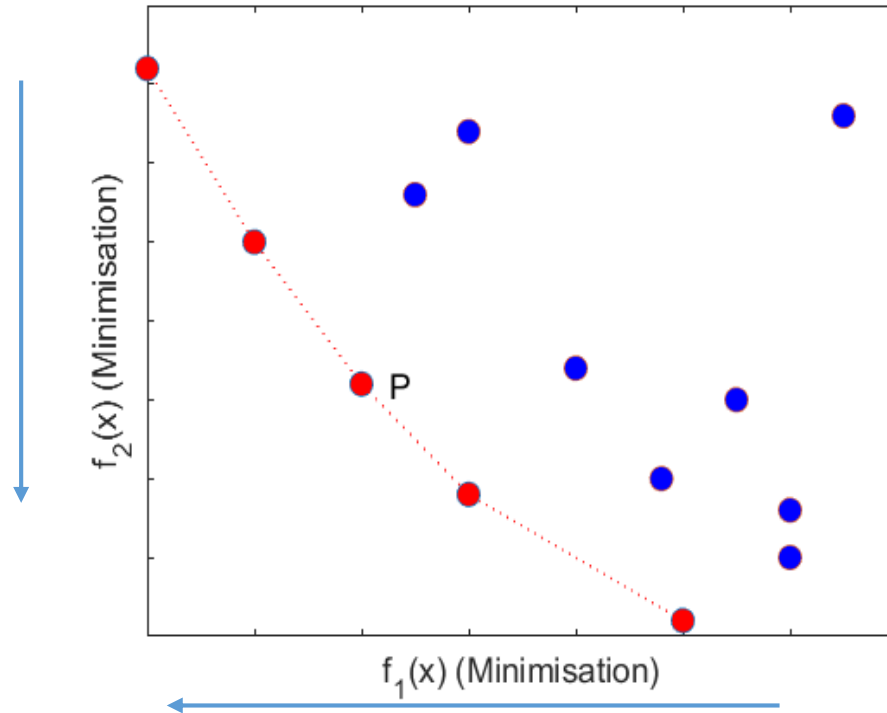
➤ “s.t.” stands for “subject to”.

Pareto (帕雷托) Dominance



- x_a **dominates** x_b if
 - Solution x_a is **no worse** than x_b in **all** objectives.
 - Solution x_a is **strictly better** than x_b in **at least one** objective.
 - Denote as $x_a \leq x_b$, if minimisation.
- x_a dominates $x_b \Leftrightarrow x_b$ is dominated by x_a
- **[Question]** Can you tell if
 - x_2 dominates x_1 ?
 - [Answer] Not comparable.
 - x_1 dominates x_3 ?
 - [Answer] Yes.
 - x_4 dominates x_2 ?
 - [Answer] Not comparable.

Pareto Front



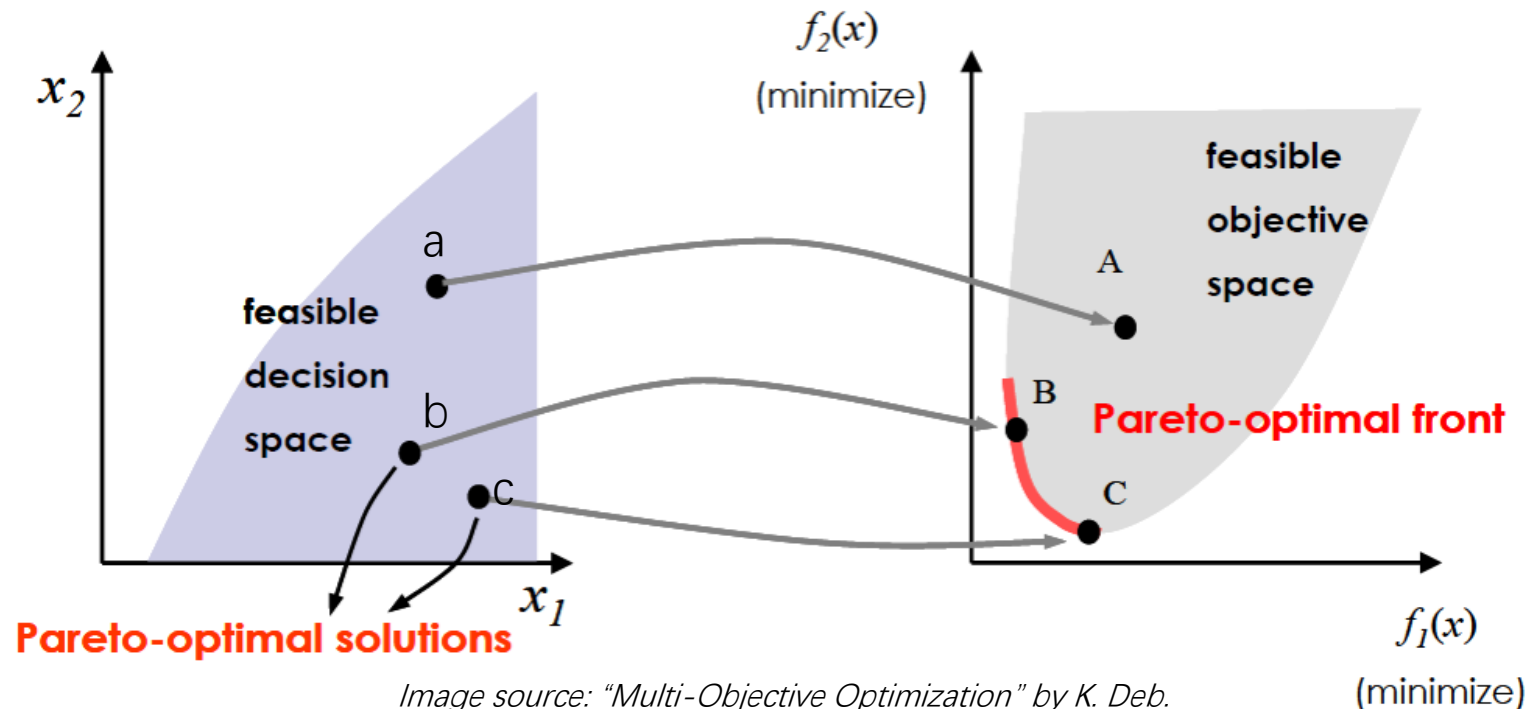
Pareto optimal: red points

Pareto optimal front: dashed red curve

- Among a set of solutions P , the **non-dominated solution set** is a set of solutions that are not dominated by any member of P .
- The non-dominated set of the entire feasible decision space is called the **Pareto-optimal set**.
- The boundary defined by the set of all point mapped from the Pareto optimal set is called the **Pareto optimal front**.

Pareto Optimal Solutions

- Pareto optimal **set** in the **decision** space (决策空间).
- Pareto optimal **front** in the **objective** space (目标空间).



Main Goals of MOO

1. To find a set of solutions as close as possible to the Pareto optimal front (convergence 收敛性).
2. To find a set of solutions as diverse as possible (diversity 多样性).

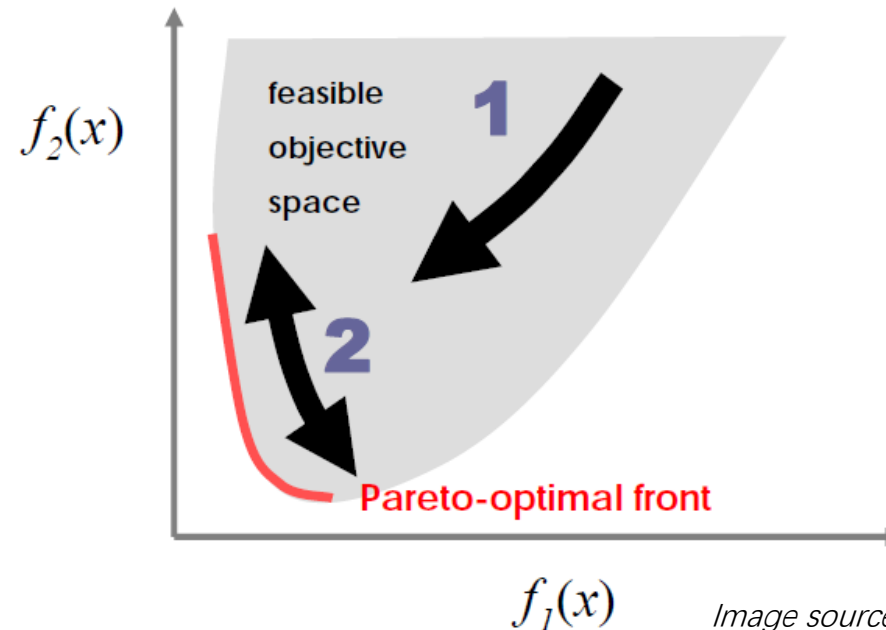


Image source: "Multi-Objective Optimization" by K. Deb.

Pareto Dominance Relation

- Reflexive?
 - No. Any solution x does not dominate itself.
- Symmetric?
 - No. $x_a \leq x_b \not\Rightarrow x_b \leq x_a$.
- Antisymmetric?
 - No.
- Transitive?
 - Yes. If $x_a \leq x_b$ and $x_b \leq x_c$ then $x_a \leq x_c$.
- $x_a \not\leq x_b \not\Rightarrow x_b \leq x_a$.

*Antisymmetric Relation: A binary relation R is antisymmetric iff:
If $R(a, b)$ and $R(b, a)$ then $a = b$.*

How to Solve a MOO Problem?

- Straightforward solution: Convert it to a single-objective problem.
 - The weighted sum approach.
- Provide several solutions, “approach” (逼近) the solutions to the Pareto front, then select a solution from the set.
 - Non-trivial, depends on the decision maker’s experience.
- Provide several solutions, a decision maker selects an area of solutions, then apply local search.
 - Non-trivial, depends on the decision maker’s experience.

Convert to Single Objective

- It's straightforward:

- Build a single objective using a weighted sum of objectives:

$$\textit{Combined Objective} = \alpha * f_1 + (1 - \alpha) * f_2$$

- It seems to be a very simple method!

- [Questions]

1. What the value α should be?
2. If you don't know the exact value, how to decide/compute the value of α ?
 - There exist various methods for setting α .

Weakness

- We don't know the exact weights in many cases. Though there are various methods for computing the weights, they also have weakness:
 - Rely on the assumption of convexity/differentiability.
 - Require knowledge of bounds of the objective values.
 - The solution highly depends on the choice of weights.

⇒ Search in the solution space involves search in the weight space.
- We get only one solution given a set of weights.
 - Unable to provide different trade-off to the decision maker.
 - We don't really know other possible trade-off among objectives.

Multi-Objective Evolutionary Algorithms (MOEAs)

- ❖ Introduction to MOEAs

- ❖ Non-dominated Sorting GA (NSGA II)

Advantages of MOEAs

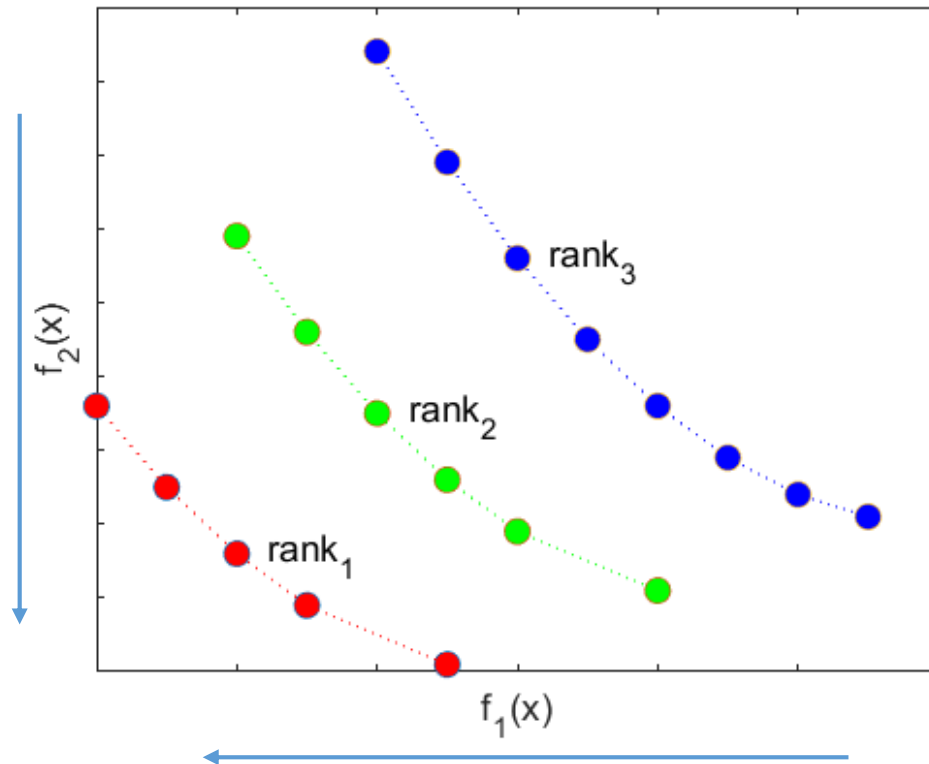
- They can provide a set of non-dominated solutions in a single run without requiring the set of weights.
- They do not require the objective functions to be convex, smooth, or even continuous (fewer assumptions).
- They can handle nonlinear constraints.
- They can deal with uncertainty and dynamics better than others.

Multi-Objective Evolutionary Algorithms (MOEAs)

❖ Introduction to MOEAs

❖ Non-dominated Sorting GA (NSGA II)

Key Ingredient of NSGA II: *Non-dominated Sorting*



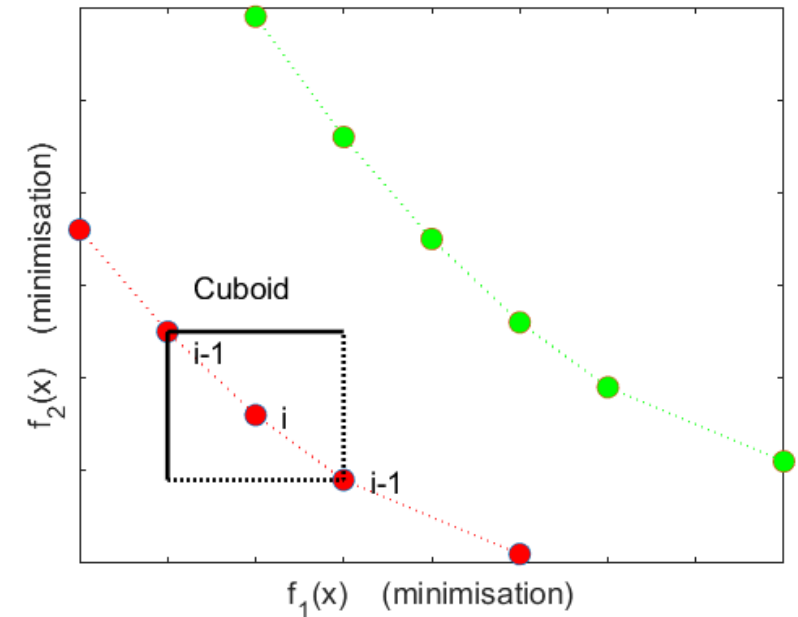
- Classify the solutions into a number of mutually exclusive non-dominated sets.

- $F = \bigcup_{i=1}^3 rank_i$

Comparing Solutions

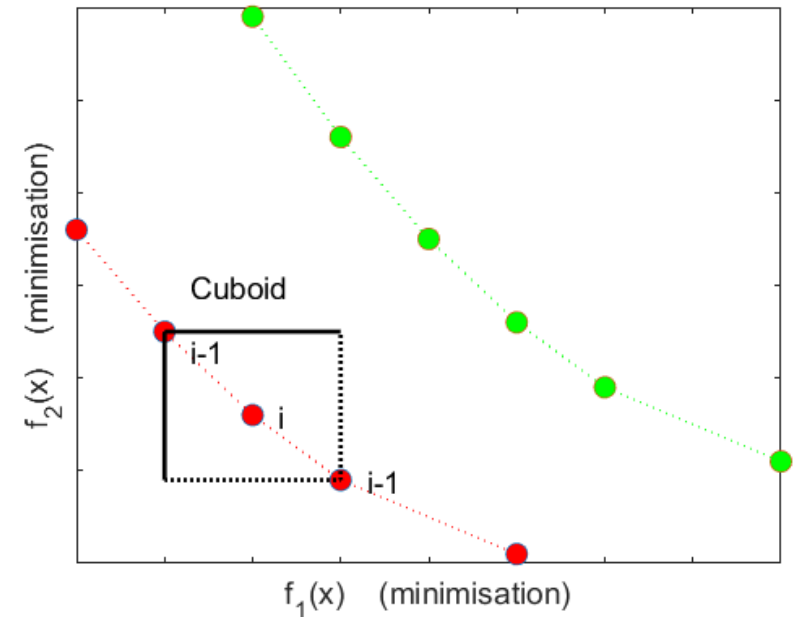
- **Crowding tournament selection**

- Assume that every solution has a non-domination rank and a local crowding distance.
- A solution x_a wins a tournament against another solution x_b
 - If the solution x_a has a **better rank**.
 - If they have the same rank but solution x_a has a **larger crowding distance** than solution x_b .



Crowding Distance

- Determine **Crowding Distance**
 - Denotes half of the perimeter of the enclosing cuboid with the nearest neighbouring solutions in the same rank.
 - Estimation of the largest cuboid enclosing a particular solution (**density estimation**).
 - *Example (Figure on right):*
 - The crowding distance of the i^{th} solution in its front (red) is the average side-length of the cuboid (box).



Non-dominated Sorting GA [1]

- Step 1 $\left\{ \begin{array}{l} R_t = P_t \cup Q_t \\ \mathcal{F} = \text{fast-nondominated-sort}(R_t) \end{array} \right.$ combine parent and children population
 $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all non-dominated fronts of R_t
- Step 2 $\left\{ \begin{array}{l} \text{until } |P_{t+1}| < N \\ \quad \text{crowding-distance-assignment}(\mathcal{F}_i) \\ \quad P_{t+1} = P_{t+1} \cup \mathcal{F}_i \end{array} \right.$ till the parent population is filled
 calculate crowding distance in \mathcal{F}_i
 include i -th non-dominated front in the parent pop
- Step 3 $\left\{ \begin{array}{l} \text{Sort}(P_{t+1}, \geq_n) \\ P_{t+1} = P_{t+1}[0 : N] \end{array} \right.$ sort in descending order using \geq_n
 choose the first N elements of P_{t+1}
- Step 4 $\left\{ \begin{array}{l} Q_{t+1} = \text{make-new-pop}(P_{t+1}) \\ t = t + 1 \end{array} \right.$ use selection, crossover and mutation to create
 a new population Q_{t+1}

Non-dominated Sorting GA [1]

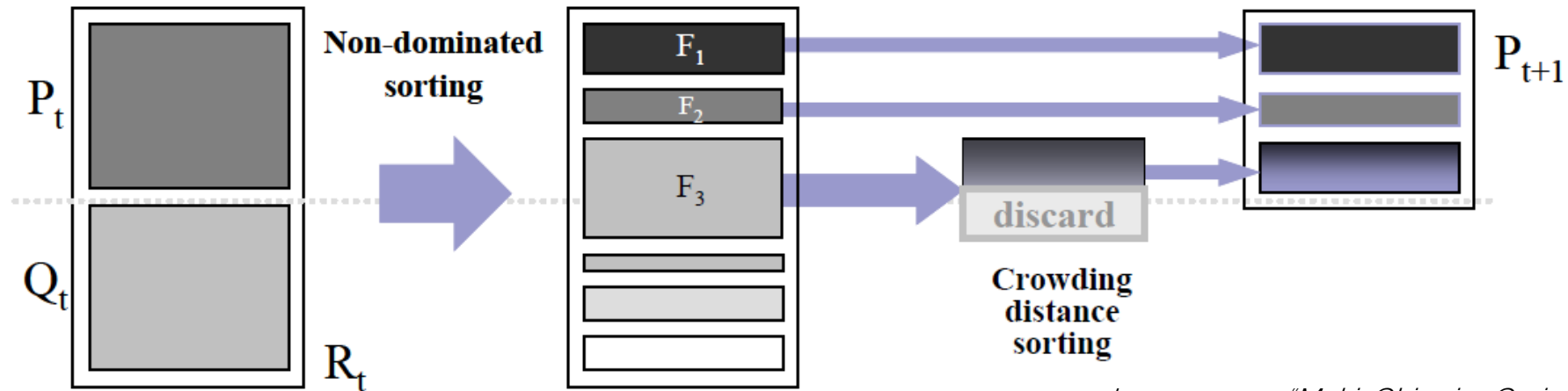


Image source: "Multi-Objective Optimization" by K. Deb.

Non-dominated Sorting GA

- **Advantages**

- The diversity among non-dominated solutions is maintained using the crowding procedure: No extra diversity control is needed.
- Elitism protects an already found Pareto-optimal solution from being deleted.

- **Disadvantages**

- When there are more than N members in the first nondominated set, some Pareto-optimal solutions may be discarded.

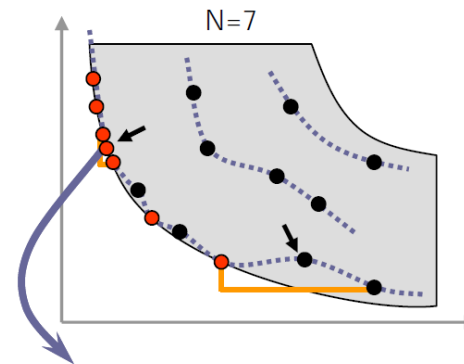


Image source: "Multi-Objective Optimization" by K. Deb.

A Pareto-optimal solution is discarded

Multi-Objective Learning

- ❖ Introduction to Multi-Objective Learning
- ❖ Diverse and Accurate Ensemble Learning Algorithm
- ❖ Class Imbalance Learning
- ❖ Multi-Objective Ensemble Learning
- ❖ Conclusion

An Example Error Function

- Negative correlation learning defines a simple error function for each network i as follows (N is the size of training set):

$$\varepsilon_i = \frac{1}{N} \sum_{n=1}^N \left(\frac{1}{2} (F_i(x_n) - y_n)^2 + \lambda p_i(x_n) \right)$$

where $p_i(x_n) = (F_i(x_n) - F(x_n)) \sum_{j \neq i} (F_j(x_n) - F(x_n))$.

Where are Multiple Objectives?

- There are many methods for learning diverse and accurate ensembles, e.g., boosting, bagging, negative correlation learning, etc.
- In general: Accuracy + λ Diversity
- We would like to maximize both the accuracy of each individual learners and the diversity among individuals.
- These are in essence two separate criteria/objectives.

Multi-objective Learning

- Multi-objective learning treats **accuracy** and **diversity** as two separate but key objectives in learning.
- Multi-objective optimization algorithms, such as multi-objective evolutionary algorithms (MOEAs), are used as learning algorithms.
- The result from such an MOEA is a **non-dominated set of solutions** (i.e., learners), which ideally form the ensemble we are interested.
 - A Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," Journal of Mathematical Modelling and Algorithms, 5(4):417-445, December 2006.

Flexibility and Generality

- Multi-objective learning offers a highly flexible and general framework for considering different requirements in learning.
- For example, we can include an additional regularization term, as an additional objective [6]. Thus, three objectives are optimised:
 1. objective of performance;
 2. objective of diversity;
 3. objective of regularisation.

$$e_i = \frac{1}{M} \sum_{n=1}^N (f_i(\mathbf{x}_n) - y_n)^2 - \frac{\lambda}{M} \sum_{n=1}^N (f_i(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n))^2 + \alpha_i \mathbf{w}_i^T \mathbf{w}_i.$$

Multi-Objective Learning

- ❖ Introduction to Multi-Objective Learning
- ❖ Diverse and Accurate Ensemble Learning Algorithm
- ❖ Class Imbalance Learning
- ❖ Conclusion

Diverse and Accurate Ensemble Learning Algorithm (DIVACE)

- Two objectives:

- **Accuracy:**

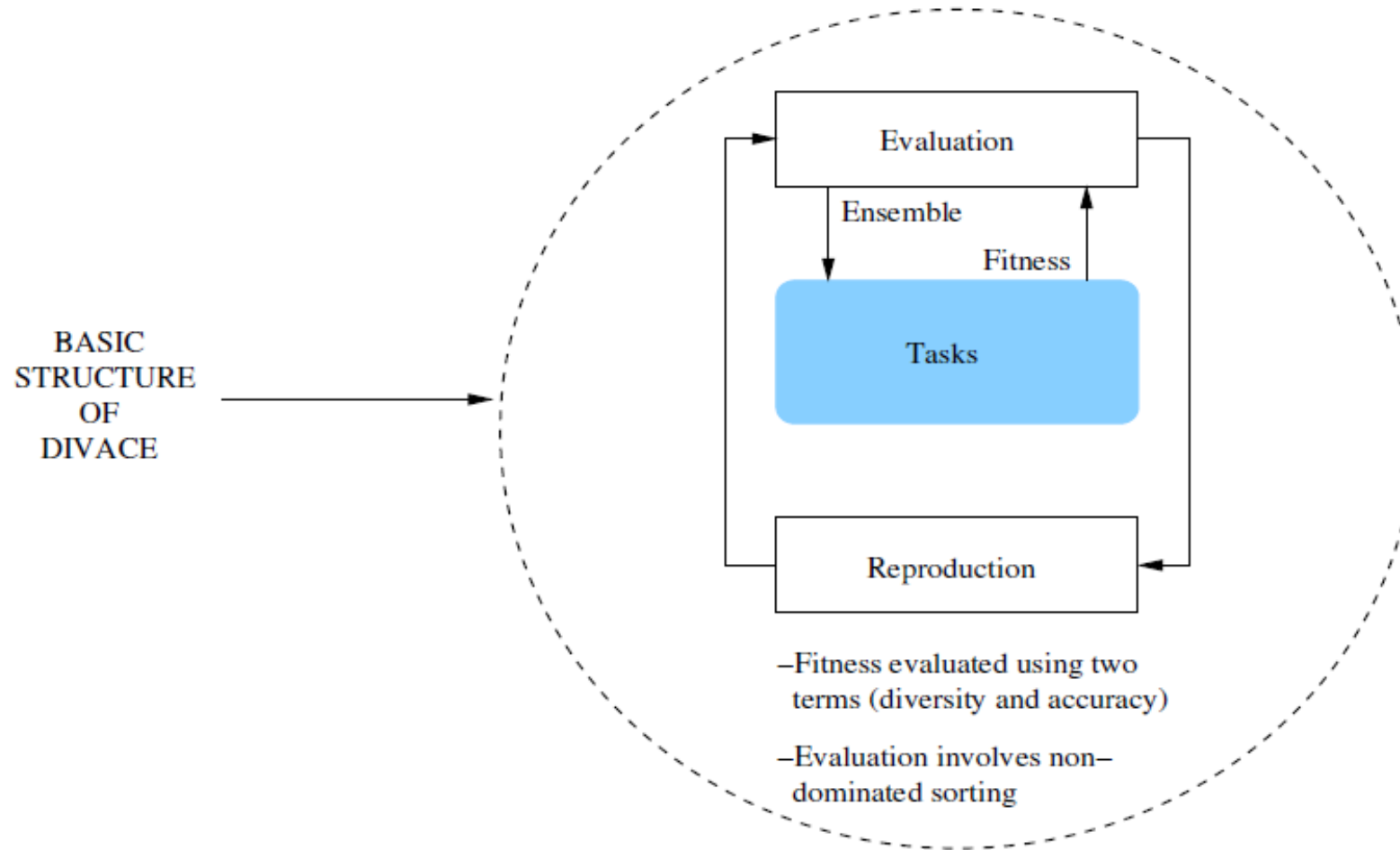
$$\text{min err}_k = \frac{1}{N} \sum_{i=1}^N (f_k^i - o^i)^2$$

- **Diversity:**

$$\text{min corr}_k = \sum_{i=1}^N (f_k^i - f^i) \left[\sum_{j \neq k, j=1}^M (f_j^i - f^i) \right]$$

- f^i is the ensemble's output for a training sample i ;
- f_k^i is the k^{th} base learner's output for a training sample i ;
- o^i is the desired output (true value) for a training sample i .

DIVACE: Basic Structure



DIVACE: Main Steps [5]

1. Initialize a random population of M networks, initialize the weights to uniformly distributed random values in the range of (0,1).
2. Apply **Back-Propagation (BP)** to all individuals in the population.
3. Repeat the following until stopping condition(s) is(are) met:
 - 1) Evaluate the population in accordance with the **two objective functions** and label the $S_{NonDominated}$ using the **non-dominate sorting algorithm**.
 - 2) If there $|S_{NonDominated}| < 3$, then **a repair rule** is used [5].
 - 3) Delete all dominated individuals from the population.
 - 4) Repeat the following until population size is M :
 - i. *Self-adaptive crossover operator updates itself.*
 - ii. *Select 3 parents uniformly at random from the population.*
 - iii. *Perform crossover (similar to Differential Evolution).*
 - iv. *Perform mutation (additive Gaussian noise).*
 - v. *Apply BP to child and add it to the population.*

Multi-Objective Learning

- ❖ Introduction to Multi-Objective Learning
- ❖ Diverse and Accurate Ensemble Learning Algorithm
- ❖ Class Imbalance Learning
- ❖ Multi-Objective Ensemble Learning
- ❖ Conclusion

Class Imbalance Learning

- **Class imbalance learning** refers to learning from imbalanced data sets, in which some classes of examples (minority) are highly under-represented comparing to other classes (majority).
- **Learning difficulty**: poor generalization on the minority class.
- **Learning objective**: obtaining a classifier that will provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class.

Multi-class Imbalance Learning

- Multi-class imbalance: there are more than two classes with uneven class distributions.
 - E.g. In software defect prediction: there are different types of defects.
 - Most existing imbalance learning techniques are **only designed for and tested in two-class scenarios**.
 - Existing methods are not effective or even cause a negative effect when there is more than one minority/majority class.
-
- S. Wang and X. Yao, "Multi-Class Imbalance Problems: Analysis and Potential Solutions," IEEE Transactions on Systems, Man and Cybernetics, Part B, 42(4):1119-1130, August 2012.

Multi-objective Class Imbalance Learning [5]

- Multi-objective learning treats **single-class performance as separate objectives**.
- Multi-objective optimization algorithms, such as multi-objective evolutionary algorithms (MOEAs), are used as learning algorithms.
- The result from such an MOEA is a non-dominated set of solutions (i.e., learners), which ideally form an ensemble we are interested.

Multi-Objective Learning

- ❖ Introduction to Multi-Objective Learning
- ❖ Diverse and Accurate Ensemble Learning Algorithm
- ❖ Class Imbalance Learning
- ❖ Multi-Objective Ensemble Learning
- ❖ Conclusion

Multi-objective Ensemble Learning

- Multi-objective ensemble learning does improve the performance of single-objective learning.
- The use of different measures as separate objectives helped to increase the diversity in the ensemble and improve ensemble learning performance.
- The ensembles did well even on those performance measures that were **not** used in multi-objective learning, which provides an evidence demonstrating the robustness of the result.

Software Effort Estimation (SEE)

- Problem description:
 - Estimation of the effort required to develop a software project (e.g., in person-hours). <- Very useful to you for succeeding in CSE!
 - Based on features such as
 - functional size (numerical),
 - required reliability (ordinal),
 - programming language (categorical),
 - development type (categorical),
 - team expertise (ordinal), etc.
- Importance:
 - Main factor influencing project cost.
 - Overestimation vs. underestimation.

Machine Learning in SEE

- Uses completed projects as training examples for creating SEE models, e.g.,
 - Multilayer Perceptrons (MLPs).
 - Radial Basis Function networks (RBFs).
 - Regression Trees (RTs).
- Can be used as decision support tools.

Different Performance Measures in SEE

- Several different performance measures are used in SEE.

Mean Magnitude of the Relative Error:

$$MMRE = \frac{1}{T} \sum_{i=1}^T MRE_i,$$

where $MRE_i = |\hat{y}_i - y_i|/y_i$; \hat{y}_i is the predicted effort; and y_i is the actual effort.

Percentage of estimations within 25% of the actual values:

$$PRED(25) = \frac{1}{T} \sum_{i=1}^T \begin{cases} 1, & \text{if } MRE_i \leq \frac{25}{100} \\ 0, & \text{otherwise} \end{cases}.$$

Logarithmic Standard Deviation:

$$LSD = \sqrt{\frac{\sum_{i=1}^T \left(e_i + \frac{s^2}{2} \right)^2}{T - 1}},$$

where s^2 is an estimator of the variance of the residual e_i and $e_i = \ln y_i - \ln \hat{y}_i$.

Current Situation

- There is no universally agreed single performance measure.
- The relationship among different measures in SEE is not well understood.
- Existing SEE approaches use **at most one measure** during the learning procedure. It is unclear whether a model/learner trained using one measure would still perform well under a different measure.
- Many papers did not even report the measure they used in training!

SEE by Multi-objective Learning

- How about viewing SEE as a multi-objective learning problem?
- Each performance measure is considered explicitly as a separate objective in learning.

But Why?

- A multi-objective algorithm can be used to create SEE models that are generally good in terms of all objective measures, and present different trade-offs among these measures.
- These different trade-offs can help us to understand to what extent different measures behave differently and what the relationship among these measures is.
- They help to enhance the **robustness** of the models.

Some Research Questions

1. What is the relationship among different performance measures for SEE?
2. Can we use different performance measures as a source of diversity to create SEE ensembles? In particular, can that improve on the performance measures used as objectives with respect to a standard learning algorithm for the same type of base model?
3. Is it possible to outperform the state-of-the-art?

Ensemble Member Selection

- Sometimes it is unnecessary to include the entire set of classifiers found by MOEAs in an ensemble. A subset would be sufficient, or even better.
 - X. Yao and Y. Liu, "Making use of population information in evolutionary artificial neural networks," IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 28(3):417-425, June 1998.
- There are various methods in the literature for selecting a diverse subset of classifiers from a large set
 - U. Bhowan, M. Johnston, M. Zhang and X. Yao, "Reusing Genetic Programming for Ensemble Selection in Classification of Unbalanced Data," IEEE Transactions on Evolutionary Computation, 18(6):893-908, December 2014.

Multi-Objective Learning

- ❖ Introduction to Multi-Objective Learning
- ❖ Diverse and Accurate Ensemble Learning Algorithm
- ❖ Class Imbalance Learning
- ❖ Multi-Objective Ensemble Learning
- ❖ Conclusion

Concluding Remarks for Multi-Objective Learning

- Multi-objective learning fits naturally with ensembles.
- There are different forms of multi-objective learning, e.g., different objectives.

Reading Materials for This Lecture

- [1] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2), 181-197.
- [2] H. Wang and X. Yao, Objective Reduction Based on Nonlinear Correlation Information Entropy, *Soft Computing*, June 2016, Volume 20, Issue 6, pp 2393–2407.
- [3] Y. Yuan, H. Xu, B. Wang and X. Yao, A New Dominance Relation Based Evolutionary Algorithm for Many-Objective Optimization, *IEEE Transactions on Evolutionary Computation*, 20(1):16-37, February 2016.
- [4] K. Praditwong and X. Yao, A New Multi-objective Evolutionary Optimisation Algorithm: The Two-Archive Algorithm, *Proc. of the 2006 International Conference on Computational Intelligence and Security (CIS'2006)*, 3-6/11/2006, Ramada Pearl Hotel, Guangzhou, China. IEEE Press, Volume 1, pp.286-291.
- [5] A Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," *Journal of Mathematical Modelling and Algorithms*, 5(4):417-445, December 2006.
- [6] H. Chen and X. Yao, "Multiobjective Neural Network Ensembles based on Regularized Negative Correlation Learning," *IEEE Transactions on Knowledge and Data Engineering*, 22(12):1738-1751, December 2010.
- [7] L. L. Minku and X. Yao, "Software Effort Estimation as a Multi-objective Learning Problem," *ACM Transactions on Software Engineering and Methodology*, 22(4), Article No. 35, October 2013, 32 pages.
- [8] Z. Gong, H. Chen, B. Yuan and X. Yao, "Multiobjective Learning in the Model Space for Time Series Classification," *IEEE Transactions on Cybernetics*, online on 23 January 2018.
- [9] Minku, L. L., White, A. P., & Yao, X. (2010). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 22(5), 730-742.
- [10] Oza, N. C., & Russell, S. (2001, August). Experimental comparisons of online and batch versions of bagging and boosting. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 359-364). ACM.