docfbe readme

角田 雅照, 大杉 直樹, 柿元 健, 門田 暁人, 松本 健一
奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア工学講座 CFbE 研究グループ

masate-t@is.naist.jp, naoki-o@is.naist.jp
Tel: 0743-72-5312
Fax: 0743-72-5319

1．概要

docfbe は, 協調フィルタリングを用いて見積もりを行うコマンドライン形式のアプリケーションです. CSV (Comma Separated Values)形式で記述された見積もり対象データと見積もりアルゴリズムを指定すると, 見積もり結果をCSV 形式で出力します.

2．見積もり方法

docfbe は以下の 3 つのステップによって指定したメトリクスを見積もります.

1. **正規化** :データに含まれるメトリクスの値域を (0.0～1.0 の値など)に正規化します. 正規化の方法はコマンドラインオプションにより指定します.
2. **類似度計算** :見積もり対象のプロジェクトと他のプロジェクトがどれぐらい似ているかを計算し, *n* 個 (*n* の値はコマンドラインオプションにより指定します)の類似プロジェクトを見つけます. 類似度計算の方法はコマンドラインオプションにより指定します.
3. **予測値計算** :類似度計算で見つけた *n* 個の類似ケースの値を平均するなどして, 予測値を計算します. 予測値計算の方法はコマンドラインオプションにより指定します.

見積もり精度を向上させるために, さらに拡張ステップが追加される場合もあります. docfbe 実行時に拡張ステップを追加する場合は, 対応するコマンドラインオプションを指定します.

3．チュートリアル

3.1. データの準備

docfbe を用いて現在進行中のソフトウェアプロジェクトの試験工数を見積もることを考え, 具体的な手順について説明します. まずは, 見積もりに用いるデータを準備します. どのようなメトリクスを用いて試験工数を見積もるかは, 組織で収集されているメトリクスに依存します. 例えば, 試験工数

の他に次のメトリクスが収集されている場合を考えます.

- 過去に実施された, 完了済プロジェクトのデータ
  - 設計工数
  - 製造工数
  - 試験工数
- 見積もり対象である, 現在進行中プロジェクトのデータ
  - 設計工数
  - 製造工数

例えば, 完了済プロジェクトのデータが以下の場合を考えます.

プロジェクト名	設計工数	製造工数	試験工数
A 社向け開発	50	30	40
B 社向け開発	45	25	35
C 社向け開発	40	50	60
C 社　第 2 事業部向け開発	45	55	65

データは CSV 形式である必要があります. 例えば, 前述の表を CSV 形式で表すと次のようになります.

プロジェクト名, 設計工数, 製造工数, 試験工数
A 社向け開発, 50, 30, 40
B 社向け開発, 45, 25, 35
C 社向け開発, 40, 50, 60
"C 社　第 2 事業部向け開発", 45, 55, 65

docfbe はファイル中に含まれている空白文字（スペース, タブ文字など）を無視します. データとして空白文字を含めたい場合は, 引用符（"）, あるいは, 2 重引用符（""）で値を囲って下さい（例えば, 上記 "C 社　第 2 事業部　向け開発" のように記述します）. データとして引用符と2 重引用符を含めることはできません. この CSV 形式のデータをファイル名 learning.csv として保存し, docfbe と同じフォルダに保存しておきます. このような完了済プロジェクトのデータを見積もりモデル構築のための学習データ（learning data）と呼びます.

同様に, 現在進行中のプロジェクトのデータが以下の場合を考えます.

プロジェクト名	設計工数	製造工数	試験工数
D 社向け開発	47	27	
E 社向け開発	42	52	

このデータ中では, 開発中で試験工数が不明であるので, これらのデータでは試験工数の値が空欄になっています. この空欄になっている試験工数の値を docfbe で見積もります. このデータを CSV 形式で表すと次のようになります.

プロジェクト名, 設計工数, 製造工数, 試験工数
D 社向け開発, 47, 27,
E 社向け開発, 42, 52,

この CSV 形式のデータをファイル名 estimating.csv として保存し, これも docfbe と同じフォルダに保存しておきます. このような現在進行中のプロジェクトのデータを見積データ (estimating data) と呼びます.

3.2. 見積もりを行う

docfbe の実行は windows コンソール (コマンドライン入力) を用いて行います. コンソールは, [スタート]-[ファイル名を指定して実行(R)]をクリックし, [名前(O)]に cmd と入力することで起動できます. コンソールを用いて, カレントディレクトリを docfbe があるディレクトリに設定します (例えば, docfbe が "c:¥bin" にある場合は, コンソールで "cd c:¥bin" と入力します). 最後に, 下記のようにコマンドを入力して docfbe を起動します.

```
docfbe -l=learning.csv -e=estimating.csv -t= 試 験 工 数 -ns=5 -n=normalize
-s=CosineSimilarity -p=WeightedSum -o=result.csv
```

上記のコマンド, 並びに, コマンドラインオプションの意味は次のとおりです.

- docfbe ..... docfbe の実行ファイルを起動するコマンド (ファイル名) です.
- -l ..... 学習データを指定します. docfbe があるディレクトリからの相対パス, あるいは, 絶対パスで指定します.
- -e ..... 見積データを指定します. docfbe があるディレクトリからの相対パス, あるいは, 絶対パスで指定します.
- -t ..... 見積もりたいメトリクスの名前を指定します. ここでは '試験工数 "を指定しています.
- -ns ..... 見積もりに使用する類似プロジェクトの数. ここでは 5 を指定しています.
- -n ..... 各メトリクスの正規化方法. ここでは normalize を指定しています.
- -s ..... 類似度計算方法. ここでは CosineSimilarity を指定しています.
- -p ..... 予測値計算方法. ここでは WeightedSum を指定しています.
- -o ..... 結果を出力するファイル名を指定します. docfbe があるディレクトリからの相対パス, あるいは, 絶対パスで指定します.

画面に CF based Estimation is successfully completed. と表示されれば, 見積もり完了です. docfbe と同じフォルダに result.csv というファイルが下記のような CSV 形式で出力されます.

```
プロジェクト名, 試験工数
D 社向け開発, 37.5025610864467
E 社向け開発, 62.4922359499621
```

上記の CSV 形式は, 以下のような見積もり結果を示しています.

プロジェクト名	試験工数
D 社向け開発	37.5025610864467
E 社向け開発	62.4922359499621

4. コマンドラインオプション

4.1. ヘルプオプション

4.1.1. --help / 省略形 :-h

操作方法のヘルプ（英語）を表示します.

4.2. 入出力データに関するオプション

4.2.1. --learning=ファイル名{,ファイル名} / 省略形 :-l=ファイル名{,ファイル名}

学習データとして与えるファイルを docfbe からの相対パス, あるいは, 絶対パスで指定します. 学習データには, 値が空欄になっている場所があってもかまいません. しかし, 見積もり対象のメトリクスが, 少なくともいくつかのプロジェクトで既に記録されていなければなりません.

学習データはCSV形式の表として指定します. 表の最上段の行にはメトリクスの名前をコンマで区切って格納します. また, 表の最左端の列には, プロジェクトの名前が格納されます. 表の各要素には, 各プロジェクトの各メトリクスの値が格納されます. 次に適切なデータと, 不適切なデータの例を挙げます.

- 適切なデータの例

```
プロジェクトID, 設計工数, 製造工数, 試験工数
ID-000, 50, 30, 40
ID-001, 45, 25, 35
```

- 不適切なデータの例（最上段にメトリクス名が入っていない）

```
ID-000, 50, 30, 40
ID-001, 45, 25, 35
プロジェクトID, 設計工数, 製造工数, 試験工数
```

- 不適切なデータの例（最左端にプロジェクト名が入っていない）

```
設計工数, 製造工数, 試験工数, プロジェクトID
50, 30, 40, ID-000
45, 25, 35, ID-001
```

また, コンマで区切って複数のファイル名を指定することで, ファイルをマージして処理することができます. 指定された複数のファイルに異なる名前の行（プロジェクト）や列（メトリクス）がある場合, 新しい行や列が追加されます. 指定した複数のファイルに同じ名前の行や列がある場合, 先に指定されたファイルの行や列の値を, 後で指定されたファイルの行や列の値で上書きします.

このオプションは必ず指定する必要があります.

4.2.2. --estimating=ファイル名{,ファイル名} / 省略形 :-e=ファイル名{,ファイル名}

見積データとして与えるファイルを docfbe からの相対パス, あるいは, 絶対パスで指定します. データの指定方法や形式は, 学習データの場合 (-l オプションで指定する) と同じです.

学習データと同一のファイルを見積データとして指定することもした可能です. 学習データと同一のファイルを見積データとして指定した場合, 同一ファイルに含まれる他のプロジェクトのデータを用いて, 各プロジェクトのデータを見積もります.

このオプションは必ず指定する必要があります.

4.2.3. --output=ファイル名{,ファイル名} / 省略形 :-o=ファイル名{,ファイル名}

見積もり結果を出力するファイルを docfbe からの相対パス, あるいは, 絶対パスで指定します. 指定された名前のファイルが既に存在する場合, 結果は上書きされます.

このオプションを指定しない場合, 標準出力に結果が出力されます. なお, 処理の途中経過は, 標準エラー出力に出力されるため, パイプを用いて他のプログラムに出力を渡すことができます.

4.2.4. --target=メトリクス名{,メトリクス名} / 省略形 :-t=メトリクス名{,メトリクス名}

見積もり対象のメトリクス名を指定します. コンマで区切って複数のメトリクスを同時に指定することもできます. また, このオプションにアスタリスク (*) を指定すると, 全てのメトリクスを (その値が空欄かどうかに関わらず) 見積もります.

このオプションを指定しない場合, 見積データ中の空欄になっている部分が全て見積もり対象となります.

4.2.5. --merge=マージ方法 /省略形 :-m=マージ方法

学習データと見積データが異なるメトリクスを含んでいる場合に 2 つのデータをマージする方法を指定します. 学習データと見積データが異なるメトリクスを含んでいる場合, 一般には協調フィルタリングは実行できません. このため, 2 つのデータをマージして同じメトリクスを含む表を作成する必要があります. 以下の文字列の中からいずれかを指定します.

- none ..... マージ機能を使用しない. 学習データと見積データが異なるメトリクスを含んでいる場合はエラーを表示して処理を停止します.
- both ..... 学習データと見積データ両方のメトリクスを残します.
- leaning ..... 学習データのメトリクスだけを残し, 見積データのみに含まれるメトリクスを無視します.
- estimating ..... 見積データのメトリクスを残し, 学習データのみに含まれるメトリクスは無視されます.

このオプションを指定しない場合, デフォルト値 “both” が使用されます.

4.2.6. --default=デフォルト値 / 省略形 :-d=デフォルト値

空欄に指定した値を代入します. 一般に, プロジェクトの工数見積もりには用いません. 協調フィルタリングに用いるデータが 0 か 1 の 2 値の場合に用いる場合があります.

4.3. アルゴリズムに関するオプション

4.3.1. --neighbors-size=ネイバーフッドサイズ / 省略形 :-ns=ネイバーフッドサイズ

見積もりに使用する類似プロジェクトの数を指定します. 通常は 2〜10 程度の値を指定します. 類似プロジェクト数を 1〜10 に変化させた場合, 見積もり精度 (見積もりの正確さ) は大きく影響を受けます. しかし, 11 以上で値を変化させても, 見積もり制度はあまり変化しません.

このオプションを指定しない場合, あるいは, 0 を指定した場合, 学習データに含まれる全てのプロジェクトの値を用いて見積もりを行います.

4.3.2. --normalize=正規化方法 / 省略形 :-n=正規化方法

メトリクスの値を正規化する方法を指定します. 以下の文字列の中からいずれかを指定します.

- value ..... 正規化を行わない.
- normalize ..... 各メトリクスの最大値が 1.0, 最小値が 0.0 になるように各列を正規化する.
- standardize ..... 各メトリクスの平均値と標準偏差を用いて各列を標準化する.
- order ..... 各メトリクスの値を順位に変換して各列を正規化する.
- d-normalize ..... normalize を行方向と列方向に行う.
- d-standardize ..... standardize を行方向と列方向に行う.
- d-order ..... order を行方向と列方向に行う.

多くの場合, normalize, standardize, order のいずれかの見積もり精度が最も高くなります. データの特性に応じ, 最適な正規化方法は異なります. 最適な方法を決定するためには, 各正規化方法によって見積もりを行い, 見積もり値を実測値と比較することで各方法の精度を探索的に調査する必要があります.

value は全てのメトリクスのとる値域が同じで, 値を変換する必要がないときに用います. 例えば, 全てのメトリクスのとる値域が 1〜5 の範囲である場合, value を指定します. d-normalize, d-standardize, d-order の見積もり精度は, 一般にはあまり高くありません, この場合, 通常は使う必要はありません.

4.3.3. --similarity=類似度計算方法 / 省略形 :-s=類似度計算方法

類似度計算方法を指定します. 以下の文字列の中からいずれかを指定します.

- CosineSimilarity ..... コサイン計算を用いて類似度を計算します.
- AdjustedCosineSimilarityWithAverage ..... コサイン計算と各メトリクスの平均値を用いて類似度を計算します.
- AdjustedCosineSimilarityWithMedian ..... コサイン計算と各メトリクスの中央値を用いて類似度を計算します.
- CorrelationCoefficientWithAverage ..... 相関係数と各メトリクスの平均値を用いて類似度を計算します.
- CorrelationCoefficientWithMedian ..... 相関係数と各メトリクスの中央値を用いて類似度を計算します.

- RankCorrelation ..... 順位相関係数を用いて類似度を計算します.
- DistanceSimilarityWithAverage ..... ユークリッド距離と各メトリクスの平均値を用いて類似度を計算します.
- DistanceSimilarityWithMedian ..... ユークリッド距離と各メトリクスの中央値を用いて類似度を計算します.

多くの場合, CosineSimilarity とCorrelationCoefficientWithAverage のいずれかの見積もり精度が最も高くなります. データの特性に応じ, 最適な類似度計算方法は異なります. 最適な方法を決定するためには, 各類似度計算方法によって見積もりを行い, 見積もり値を実測値と比較することで各方法の精度を探索的に調査する必要があります. なお, 上記類似度計算方法の内, median と average が異なるだけの方法は, 精度が大きく変化しません. どちらか一方 (例えば median)を試してみて見積もり精度が高かった場合だけ, もう一方 (例えば average)を試すと良いでしょう.

4.3.4. --prediction=予測値計算方法 / 省略形 :-p=予測値計算方法

予測値計算方法を指定します. 以下の文字列の中からいずれかを指定します.

- WeightedSum ..... 類似プロジェクトの加重平均を用いて予測値を計算します.
- AdjustedWeightedSumWithAverageOfColumn ..... 類似プロジェクトの加重平均と見積もり対象メトリクスの平均値を用いて予測値を計算します.
- AdjustedWeightedSumWithMedianOfColumn ..... 類似プロジェクトの加重平均と見積もり対象メトリクスの中央値を用いて予測値を計算します.
- AdjustedWeightedSumWithAverageOfNeighbors ..... 類似プロジェクトの加重平均と類似プロジェクトにおける見積もり対象メトリクスの平均値を用いて予測値を計算します.
- AdjustedWeightedSumWithMedianOfNeighbors ..... 類似プロジェクトの加重平均と類似プロジェクトにおける見積もり対象メトリクスの中央値を用いて予測値を計算します.
- AdjustedWeightedSumWithAverageOfRow ..... 類似プロジェクトの加重平均と各プロジェクトにおけるメトリクスの平均値を用いて予測値を計算します.
- AdjustedWeightedSumWithMedianOfRow ..... 類似プロジェクトの加重平均と各プロジェクトにおけるメトリクスの中央値を用いて予測値を計算します.
- AmplifiedWeightedSumWithAveragedMultiplier ..... 類似プロジェクトの加重平均と倍率修正値の平均値を用いて予測値を計算します.
- AmplifiedWeightedSumWithMedianOfMultiplier ..... 類似プロジェクトの加重平均と倍率修正値の中央値を用いて予測値を計算します.
- AmplifiedWeightedSumWithWeightedMultiplier ..... 類似プロジェクトの加重平均と倍率修正値の加重平均値を用いて予測値を計算します.

多くの場合, WeightedSum の見積もり精度が最も高くなります. データの特性に応じ, 最適な予測値計算方法は異なります. 最適な方法を決定するためには, 各予測値計算方法によって見積もりを行い, 見積もり値を実測値と比較することで各方法の精度を探索的に調査する必要があります. なお, 上記予測値計算方法の内, median とaverage が異なるだけの方法は, 精度の変化が大きく変化しません. どちらか一方 (例えば median)を試してみて見積もり精度が高かった場合だけ, もう一方 (例えば average)を試すと良いでしょう.

4.3.5. --inverse-frequency / 省略形 :-if

類似度計算を行う場合に, 多くのプロジェクトで値が記録されているメトリクスはプロジェクトの特徴を表しにくいと考え, 類似度計算における結果への寄与を弱める拡張計算です. 協調フィルタリングを用いてオススメの書籍や映画などを予測する場合に, このオプションを指定すると予測精度が上がる可能性があります. 一般に, プロジェクトの工数見積もりには用いません.

4.3.6. --case-amplifier / 省略形 :-ca

指定した数値で類似度をべき乗します. 類似度が高いプロジェクトが見積もりにより強く影響するようになり, 類似度が低いプロジェクトは見積もりにより影響しにくくなります.

4.4. 出力結果の書式に関するオプション

4.4.1. --disp-neighbors / 省略形 :-dn

類似プロジェクトの名前を出力結果に加えて表示します. どのプロジェクトが見積もり結果に影響を与えたかを確認できます.

4.4.2. --disp-similarities / 省略形 :-ds

見積もり対象のプロジェクトに対する類似プロジェクトの類似度を出力結果に加えて表示します. 各類似プロジェクトが, どの程度似ているのかを確認できます.

4.4.3. --disp-values / 省略形 :-dv

類似プロジェクトのメトリクスの値を出力結果に加えて表示します. どのような値が見積もり結果に影響を与えたかを確認できます.

4.4.4. --disp-distribution / 省略形 :-dd

見積もり対象のプロジェクトに対する他のプロジェクトの類似度の分布を出力結果に加えて表示します. 類似度の偏りを確認できます.

4.4.5. --recommendation / 省略形 :-r

協調フィルタリングを用いてオススメの書籍や映画を予測した結果に適した出力形式で出力します. 各ユーザに対するオススメの書籍や映画などを, オススメの程度が高い順にソートし, オススメの程度を表すスコアと共に出力します.
一般に, ソフトウェアの開発工数を見積もる場合, このオプションは使用しません.

Contract to the Author

Masateru Tsunoda, Naoki Ohsugi

Software Engineering Laboratory,
Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST),
630-0192, JAPAN

E-mail: masate-t@is.naist.jp, naoki-o@is.naist.jp
Web: http://se.naist.jp/~masate-t/, http://se.naist.jp/~naoki-o/
Voice: +81-(0)743-72-5312
Fax: +81-(0)743-72-5319