2020 年 10 月 23 日（金）　ゼミ資料　　M１　林 美衣

Fashion Outfit Complementary Item Retrieval の読み込み（前まで使っていた先行研究の人が書いた最も新しい論文）

今後見直す時のために全文和訳しておきました．ネットワークの構造はわかったものの，所々に出てくる indexing(インデックス作成)の解釈がよくわからなかったり，検索用でない他のモデルとの比較方法も記載はされてるもののよくわからなかった…

互換性が AUC によって毎回評価されているが，ROC 曲線の下の面積ということまでしかわからず AUC がイマイチなんなのかわかっていない．調べてもよくわからない…．compatibility はこの値で比較されているからこの AUC についてわかったら，コーディネートの点数の表し方への道も開ける…?

今回は前みたいにこのコーデの点数は〇〇点！みたいなタスクはなかった．しかし前の研究も特別なことをして点数を出していたわけではなくロスか何かをそのまま利用してそれを正規化？していただけなので今回のモデルでもできるかな…

アブスト

補完的なファッションアイテムの推奨は、ファッション衣装の完成に不可欠です。 既存の方法は主に服装の互換性予測に焦点を当てていますが、検索設定には焦点を当てていません。 服装補完アイテム検索の新しいフレームワークを提案します。 具体的には、カテゴリベースの部分空間アテンションネットワークが提示されます。これは、部分空間アテンションを学習するためのスケーラブルなアプローチです。 さらに、衣装全体のアイテムの関係をより適切にモデル化する衣装ランキングの損失を導入します。 衣装の互換性、FITB、新しい検索タスクについてメソッドを評価します。 実験結果は、互換性の予測と補完的なアイテム検索の両方で、私たちのアプローチが最先端の方法よりも優れていることを示しています。

イントロ

服装補完アイテムの取得は、服装を完成させるための互換性のあるアイテムを見つけるタスクです。 たとえば、上、下、靴のある（部分的に構築された）服を考えて、それらとうまく合う（つまり互換性のある）バッグを見つけます。 これは、特にオンライン小売業者にとって重要な推奨問題です。 顧客は、以前に選択または購入したものによく合う衣料品を頻繁に購入します。 互換性のあるアイテムを適切なタイミングで推奨できると、ショッピング体験が向上します。 図 1 に、問題を示します。

Figure 1. An example of outfit complementary item retrieval. First row: partial outfit with a missing item (shoe). Second row: retrieved results using our method.

図 1.衣装の補完的なアイテム検索の例。 最初の行：アイテム（靴）が不足している部分的な衣装。 2 行目：メソッドを使用して取得した結果。

視覚的またはテキストの類似性検索（[22、19]など）とは異なり、補完的なアイテム検索の対象アイテム（欠落アイテムまたはターゲットアイテム）は、不完全なコーディネートのカテゴリとは異なるカテゴリからのものです。 そのため、コーディネートの他のアイテムとは視覚的に異なります。 代わりに、グローバルなコーディネートスタイルと一致しています。 [16、15]で説明されているように、補完的なファッションアイテムは、色、パターン、素材、機会などの複数の属性次元に沿って互換性があります。 この多様性に対処できることが不可欠です。 さらに、実用的な服装完成システムは、徹底的な比較が禁止されている大規模な検索をサポートする必要があります。

[18、11]などの以前の研究では、ペアワイズ補完検索を扱っていましたが、服全体の互換性を明示的に扱っていませんでした。 [5、15、2]などの最近のアプローチは、服装の互換性の予測に焦点を当てていますが、検索の設定には焦点を当てていません。 技術的には、分類スコアを使用してデータベース内のアイテムをランク付けして検索することは可能ですが、大規模に行うことは現実的ではありません。 これらのアプローチは、大規模な服装の補完的なアイテム検索を念頭に置いて設計されていません（詳細な分析と比較については、セクション 2 とセクション 4 を参照してください)。

この作業では、アイテムの互換性の概念を捉えるために、[17、16、15]と同様に、複数のスタイルのサブスペースを使用するアプローチを採用しています。 以前の作業とは対照的に、部分空間は共有モデルを使用して生成されますが、部分空間のアテンションの重みはアイテムのカテゴリに基づいて条件付けられます。 私たちのシステムは、大規模な検索用に設計されており、互換性予測、穴埋め（FITB)、および服装の補完的なアイテム検索に関して最先端のパフォーマンスを上回っています（セクション 4 を参照)。

要約すると、私たちの研究の主な技術的貢献は次のとおりです。

•アイテムのカテゴリのみに依存するカテゴリベースのアテンション選択メカニズムを提案します。 これにより、スケーラブルなインデックス作成と、服の不足しているアイテムの検索を実行できます。

•コーディネート全体で機能する新しいコーディネートランキング損失関数を提案します。これにより、互換性の予測と取得の精度が向上します。

残りの論文は次のように構成されています。 セクション 2 では、関連する作業を確認します。 セクション 3.1 と 3.2 は、それぞれカテゴリベースの部分空間アテンションネットワークとコーディネートランキングの損失について説明しています。セクション 3.3 では、インデックス作成と取得のパイプラインについて説明します。 実装の詳細と実験結果はセクション 4 に記載されています。セクション 5 は、今後のステップについての議論で論文を締めくくります。

関連研究

服装の互換性予測：互換性の予測は、ペアワイズのアイテム間レベルおよび服装全体のレベルでの以前の多くの作業で対処されています。 [18、11]では、異なるカテゴリのアイテムを埋め込むために、共通の機能スペース（スタイルスペース）が使用されていました。埋め込みモデルは、アイテムの共同購入と共同ビュー情報の大規模なデータベースを使用して学習されました。アイテムの互換性は、埋め込みスペースのペアワイズ特徴距離によって決定されます。これらの作品の焦点は、衣装全体ではなく、アイテムごとでした。 [5、16、15、10]などの後の研究では、Polyvore の衣装データを使用し、衣装レベルの互換性を最適化しています。 [5]では、衣装アイテムは、双方向 LSTM モデルへの入力トークンのシーケンスとして扱われていました。 LSTM は、さまざまなサイズの衣装を扱うことを可能にします。この作品では、([2、16、15]）などの最近の研究で評価に使用されている空欄記入タスク（FITB）も紹介しました。比較のために、このタスクでモデルのベンチマークも行い、結果をセクション 4 に示します。最近[2]で服装の互換性予測にグラフィカルコンボリューションネットワーク（GCN）[9]が使用されました。グラフは、衣装で一緒に発生するアイテムを接続することによって作成されました。 GCN は、ペアワイズエッジ予測に基づいてトレーニングされました。最高のパフォーマンスは、テスト時のグラフ接続を活用することで得られます。これは、特に新しいアイテムをカタログ（テストセット）に導入できる場合は実用的ではありません。 （動的）カタログに追加された新しいアイテムは、多くの場合、カタログ内の既存のアイテムとは何の関係もありません。したがって、新しいアイテムの埋め込みを計算する方法は不明です。一般に、これらのメソッド（[5、2、15]など）は、互換性分類のために設計およびトレーニングされており、検索用に最適化されていませ

んでした。

単一の空間で類似性を計算する代わりに、最近のいくつかのアプローチ[17、16、15]は、類似性のさまざまな概念をキャプチャするためにサブスペースの埋め込みを学習することを検討しました。 Vasileva etal。 [16]は、CSN [17]モデルを適応させて、服装の互換性をモデル化するためのタイプ対応の埋め込みを学習しました。 彼らは、それぞれがアイテムカテゴリのペア（たとえば、トップス-ボトムス、トップス-シューズ、ボトムス-シューズなど）について、合計 66 の条件付きサブスペースを学習しました。 タンら。 [15]は、共有部分空間と各部分空間の重要性を学習することにより、[16]のパフォーマンスをさらに改善しました。 彼らの方法では、テスト中に入力画像のペアが必要です。 また、共有部分空間の埋め込みを利用します。これにより、[16]のように多数のモデルを処理する必要がなくなります。 サブスペースの条件付き要素はアイテムカテゴリであり、[15]のようなターゲットアイテム画像ではありません。これにより、大規模なインデックス作成と検索のために個々のアイテムの特徴抽出を実行できます。

補完的なアイテム検索：上記の互換性予測アプローチのいくつか（[18、11、16]）を検索に使用できます。既存の Polyvore 衣装データセット（[5、16]）から抽出されたデータを取得するために、最新の研究（[16、15]）と比較しました。補完的なアイテム検索は、生成的敵対的ニューラルネットワーク（GAN）などの生成モデルを使用して実行できます（例：[21、7、14]）。入力画像が与えられると、生成モデルは、ターゲットの補完的なアイテムの表現（たとえば、画像）を生成するようにトレーニングされます。次に、この生成されたアイテムを使用して、インデックス付きデータベースから実際の補完アイテムを取得できます。これらのアプローチのほとんどは、ペアワイズ検索タスクに対応しており、上から下または下から上へのケースでのみ評価されています。トップスからネックレス、ネックレスから靴など、細かいディテールが重要な典型的な服装の他のカテゴリのペアでどれだけうまく機能するかは明らかではありません。最近の研究[8]では、シーン画像との互換性に基づいて補完的なアイテムが取得されました。グローバルおよびローカル（シーンパッチ）の互換性は、シーンと製品の互換性を測定します。代わりに、私たちのアプローチは、衣装の補完的なアイテム検索に焦点を当て、ターゲット画像と衣装内のすべてのアイテムとの間の互換性を考慮します。

距離計量学習：距離計量学習は検索に不可欠です。 クロスエントロピー損失（分類）、ヒンジ損失（[20]など）、トリプレット損失（[13、4]）、プロキシ（[12]）などを使用してモデルをトレーニングできます。[16、15]では、トリプレット 損失が使用されましたが、衣装の互換性の予測にのみ使用されました。 この作業では、衣装全体でアイテムの関係をより有効に活用するために、新しい衣装のランキング損失を提案します。

提案されたアプローチ

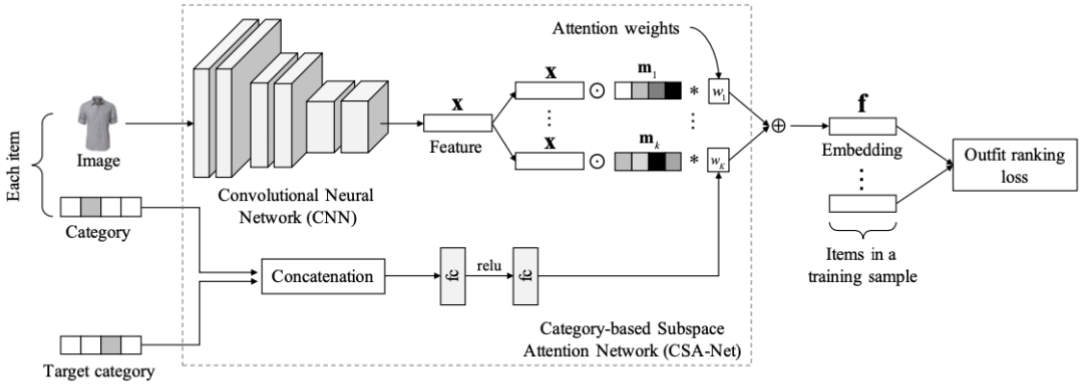

Figure 2. Overview of our framework. Our framework takes a source image, its category vector and a target category vector as inputs. The image is passed through a CNN to extract a visual feature vector, which is multiplied by a set of masks to obtain the subspace embeddings. The concatenation of two category vectors is used to predict the subspace attention weights, which select the proper subspace embeddings for the final embedding computation. Our network is trained by a ranking loss, which operates on the entire outfit.

図 2.フレームワークの概要。私たちのフレームワークは、ソース画像、そのカテゴリベクトル、およびターゲットカテゴリベクトルを入力として受け取ります。画像は CNN を通過して視覚的特徴ベクトルを抽出し、これにマスクのセットを掛けて部分空間の埋め込みを取得します。 2 つのカテゴリベクトルの連結は、部分空間の注意の重みを予測するために使用されます。これにより、最終的な埋め込み計算に適切な部分空間の埋め込みが選択されます。私たちのネットワークは、衣装全体で機能するランキングの損失によって訓練されています。

図 2 は、フレームワークのシステム概要を示しています。私たちのフレームワークには、ソース画像、そのカテゴリ、ターゲットカテゴリの 3 つの入力があります。入力画像は CNN を通過して、画像の特徴ベクトルを抽出します。以前の研究[17、15]と同様に、マスクのセットが画像特徴ベクトルに適用され、複数の部分空間の埋め込みが生成されます。それらは類似性のさまざまな側面に対応します。 2 つのカテゴリベクトルの連結は、アテンションの重みを予測するためにサブネットワークに送られます。その目的は、最終的な埋め込み計算に適切な部分空間埋め込みを選択することです。カテゴリベースのアテンション部分空間ネットワークの詳細は、セクション 3.1 に示されています。

私たちのネットワークは、衣装全体で機能するランキング損失を使用してトレーニングされています。 衣装 O、ポジティブアイテム p、およびネガティブアイテムのセット N を含むトレーニングサンプルが与えられます。ここで、ポジティブアイテムは衣装と互換性のあるアイテムであり、ネガティブアイテムは互換性のないアイテムです。 損失は、正の距離 d（O、p）と負の距離 d（O、N）に基づいて計算されます。 フレームワーク全体がエンドツーエンドで最適化されています。 衣装ランキングの損失の詳細はセクション 3.2 に示されています。

私たちのシステムは、大規模な検索用に設計されています。 セクション 3.3 で、インデッ

クス作成と衣装アイテムの取得のフレームワークを示します。

カテゴリベースの部分空間アテンションネットワーク

このセクションでは、カテゴリベースのサブスペースアテンションネットワーク（CSA-Net）について説明します。 単一の空間で類似性を計算する代わりに、スタイルの部分空間（[17、16、15]）を利用して、類似性の複数の次元をキャプチャします。 これらは衣装の他のアイテムと視覚的に異なるため、これは補完的なアイテムにとって重要です。 タンら[15]は共有部分空間を学習し、独立部分空間[16]よりも優れたパフォーマンスを実現しました。 ただし、彼らの方法では、推論中に部分空間を選択するために入力画像のペアが必要であり、これは検索には実用的ではありません。 対照的に、私たちの部分空間アテンションメカニズムは、アイテムのカテゴリにのみ依存します。 私たちのモデルは 1 つの画像と 2 つのカテゴリベクトルしか必要としないため、実用的なインデックス作成アプローチを構築できます。

ネットワークは、ソース画像 Is、そのカテゴリベクトル cs、およびターゲットカテゴリベクトル ct を入力として受け取り、特徴の埋め込み f を生成する非線形関数 ψ（Is、cs、ct）を学習します。 画像は最初に CNN に渡され、視覚的特徴ベクトル（x で示される）が抽出されます。 2 つのカテゴリベクトルの連結（セマンティックカテゴリのワンホットエンコーディング）は、サブネットワークを使用して部分空間のアテンションの重み（w1 … wk）（k は部分空間の数）を予測するために使用されます。 これは 2 つの全結合層と最後にソフトマックス層が含まれています。 そして、画像特徴ベクトルと同じ次元を持つ学習可能なマスクのセット（m1、...、mk）が、アダマール積を介して画像特徴ベクトルに適用されます。 これらのマスクは、異なる類似性部分構造をエンコードする部分空間のセットに特徴を投影するように学習されます。 ネットワークによって構築される最終的な埋め込みは、部分空間の埋め込みの加重和です。

$$f = \sum_{i=1}^{k} (x \odot \boldsymbol{m_i}) * w_i$$

ここで、k は部分空間の数、x はベース CNN の後の画像特徴ベクトル、mi は学習可能なマスク、wi はアテンションの重み、f は最終的な埋め込みです。

衣装ランキングの損失

既存のアプローチ[16、15]は、トリプレットロスを使用して、服装の互換性を予測するための特徴埋め込みを学習します。 トリプレットを形成するために、最初にランダムなペアの画像（アンカー画像とポジティブ画像）が一つのコーディネートから選択され、ポジティブ画像と同じセマンティックカテゴリを持つネガティブ画像がランダムにサンプリングされます。トリプレットロスは、正のサンプルと比較したときのマージンよりも大きい距離だけ

負のサンプルをアンカー画像から遠ざけることによって最適化されます。 対照的に、補完的なアイテムと服の互換性を判断するときは、単一のアイテムだけでなく、服の既存のすべてのアイテムの類似性を考慮します。

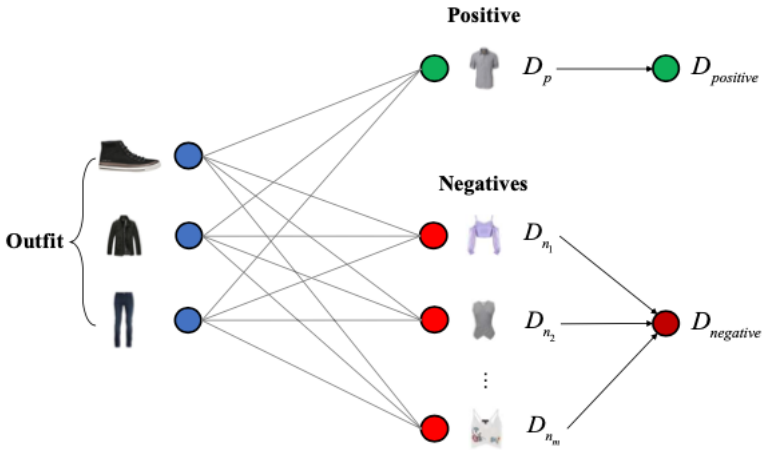

Figure 3. Illustration of the proposed outfit ranking loss that considers the item relationship of the entire outfit.

図 3.衣装全体のアイテムの関係を考慮した提案された衣装のランキング損失の図。

図 3 は、衣装のランキングの損失を示しています。 トレーニングの反復ごとに、トリプルのミニバッチをサンプリングします。各トリプルの Y = {O、p、N}は、画像のセット O = $\{I_1^o, \ldots, I_n^o\}$を含む衣装で構成されます。 服によく合うポジティブ画像 p = $\{I^p\}$、および服と一致しないネガティブ画像のセット N = $\{I_1^n, \cdots, I_m^n\}$。 衣装 O の各画像の特徴の埋め込みは、次のように計算されます。

$$f_i^o = \psi\left(I_i^o, c\left(I_i^o\right), c(I^p)\right); i = 1 \to n$$

ここで、I_i^oは O の画像、ψ（・）は私たちが提案するカテゴリベースのサブスペースアテンションネットワーク、c（.）は入力画像をワンホットカテゴリベクトルにマッピングする関数であり、c（I_i^o）は画像I_i^oのソースカテゴリベクトルで，c（I^p）は、ポジティブ画像I^pからのターゲットカテゴリベクトルです。

同様に、ポジ画像の特徴の埋め込みは次のように計算されます。

$$f_i^p = \psi\left(I^p, c\left(I_i^o\right), c(I^p)\right); i = 1 \to n$$

ポジティブ画像には複数の埋め込み（i = 1→n）があります。これは、異なるカテゴリベクトル c（I_i^o）が異なる部分空間のアテンションを生成し、その結果、異なる埋め込みが発生するためです。 これらの埋め込みは、ペアワイズ距離計算で使用されます（式 5 を参照)。 また、入力カテゴリ（c（I_i^o）や c（I^p）など）の順序がシステムのパフォーマンスに影響を与えないことも確認しました。 セクション 4.4 の順序反転の説明を参照してく

ださい。

各ネガ画像の特徴の埋め込みは、次の方法で計算されます。

$$f_i^{n_j} = \psi\left(I_j^n, c\left(I_i^o\right), c\left(I_j^n\right)\right), j = 1 \to m, i = 1 \to n \tag{4}$$

ポジティブ画像と同様に、各ネガティブ画像には、異なるカテゴリベクトル c（I_i^o）を使用した複数の埋め込みがあります。

各アイテム s（ポジティブまたはネガティブアイテム）から衣装全体までの距離を次のように定義します。

$$D_{outfit}(O, s) = \frac{1}{n}\sum_{i=1}^{n} d\left(f_i^o, f_i^s\right) \tag{5}$$

ここで、i は衣装 O の画像、f_i^o と f_i^s はそれぞれ画像 i と s の特徴埋め込みであり、$d\left(f_i^o, f_i^s\right)$ は 2 つの画像間のペアワイズ距離です。 式 5 を使用して、正のアイテム Dp = D outfit（O、p）と j = 1 から m までの負のアイテム Dnj = D outfit（O、nj）の距離を計算できます。 次に、ネガティブセットのすべてのネガティブアイテムを単一の距離に集約する集約関数 ϕ（最小または平均など）を使用して Dni に結合します。

$$D_N = \varphi\left(D_{n_1}, \ldots, D_{n_m}\right) \tag{6}$$

この服装ランキングの損失により、ネットワークは、服装と負のサンプルの間の距離が、服装と正のサンプルの間の距離よりも距離マージン m だけ大きい埋め込みを見つけるようになります。

$$l(O, p, N) = \max\left(0, D_p - D_N + m\right) \tag{7}$$

ここで、D_p と D_N はそれぞれ正と負の距離です。 以前のトリプレット損失とは対照的に、私たちの服のランキング損失は、単一のアイテムではなく、服全体に基づいて距離を計算します[16、15]。

コーディネート補完アイテム検索

コーディネート補完アイテム検索は、コーデの既存のアイテムとまとまりがあって一致するような互換性のあるアイテムのセットを検索するタスクです。 効率的な検索には、特徴抽出とインデックス作成が含まれます。これにより、テスト中にデータセット全体を線形にスキャンする必要がなくなります。 クエリ画像（コーデ）とターゲットカテゴリが与えられると、システムはその埋め込みを抽出し、それを使用してインデックス付きデータセット画像をクエリします。

互換性予測のための以前のアプローチは、検索には適していません。 [15]は画像のペアを利用します。これは、テスト中に徹底的なペアワイズ比較を必要とし、インデックス作成には実用的ではありません（インデックス作成中にターゲット（クエリ）画像を生成することは実用的ではありません）。 Cucurull ら[2]は服装の互換性のためにグラフィカル畳

み込みニューラル（GCN）ネットワークを利用します。 ただし、分類モデルを検索に適応させる方法は不明です。

対照的に、我々は、カテゴリーベースの部分空間アテンションネットワークに基づく新しいコーデ補完アイテム検索システムを提案します。

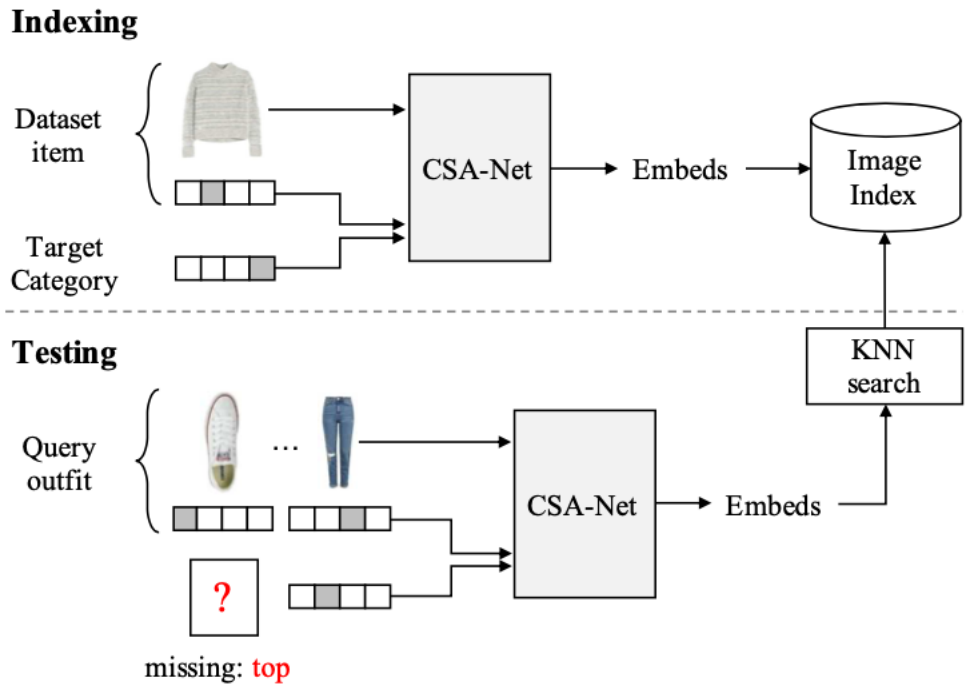

Figure 4. Our framework for outfit complementary item retrieval. For indexing, multiple embeddings of each item are computed by enumerating the target categories. More detailed explanations of category enumeration are in Section. 3.3. For testing, given a query item in an outfit, we compute its embedding by using its image, category vector and the target category vector into our model. KNN search is used to retrieve compatible items from the indexed dataset. The final ranking is obtained by fusing the scores from different query items.

図 4.コーデの補完的なアイテム検索のためのフレームワーク。 インデックス付けの場合、各アイテムの複数の埋め込みは、ターゲットカテゴリを列挙することによって計算されます。 カテゴリ列挙のより詳細な説明はセクション 3.3 にあります。テストでは、服のクエリアイテムを指定して、その画像、カテゴリベクトル、およびターゲットカテゴリベクトルをモデルに使用して、その埋め込みを計算します。 KNN 検索は、インデックス付きデータセットから互換性のあるアイテムを取得するために使用されます。 最終的なランキングは、さまざまなクエリアイテムのスコアを融合することによって取得されます。

図 4 は、検索のフレームワークを示しています。特徴抽出の場合、各アイテムの複数の埋め込みは、ターゲットカテゴリを列挙することによって計算されます。これは、カテゴリの組み合わせが異なれば部分空間のアテンションが異なり、埋め込みが異なるためです。クエリ画像は、テスト中に任意のカテゴリから取得できます。したがって、インデックス作成中に、さまざまなターゲットカテゴリからの埋め込みを計算する必要があります。たとえば、靴のアイテムのカテゴリベクトルは、トップス、ボトムスなどのカテゴリベクトルと連結され、インデックス作成中に複数の埋め込みが生成されます。テスト中にトップスアイテムが与えられた場合、次に、トップスカテゴリによって事前に計算された靴の埋め込みを検索します。各カテゴリの画像は、インデックス作成のために異なるターゲットカテゴリとペアになっているため、埋め込みサイズは高レベルのカテゴリの数に比例します（たとえば、Polyvore-Outfit データセットの 11 のセマンティックカテゴリ）。インデックス作成と検索には、既成の近似最近傍検索パッケージ（[1]など）を使用します。

テスト中に、コーデを指定して、ターゲットカテゴリの互換性のある画像のセットを取得します。 クエリ衣装の各アイテムについて、その画像、カテゴリベクトル、およびカテゴリベースの部分空間アテンションネットワークのターゲットカテゴリベクトルを使用して、埋め込みを抽出します。 次に、埋め込みを使用して、k 最近傍（KNN）検索を介してデータセット内の互換性のあるアイテムを検索します。 クエリ衣装の各アイテムに対して同様の手順を実行し、集計関数（平均融合など）を使用して、さまざまなクエリアイテムのランキングスコアを融合し、最終的なランキングを取得します。

実験

互換性の予測、空欄を埋める（FITB）、および服装の補完的なアイテム検索タスクについて、この方法を評価します。

評価

服装の互換性と FITB：互換性予測と FITB については、互換性予測の最大のデータセットである Polyvore Outfit [16]データセットの最先端の方法と比較します。 データセットには、バラバラなセットとバラバラでないセットが含まれています。 バラバラでないセットの場合、トレーニングとテストの両方の分割で一部のアイテム（完全な衣装ではない）が見られる場合があります。 バラバラなセットの場合、テスト/検証セットの衣装は、トレーニングセットの衣装と共通のアイテムを共有しません。 バラバラでないセットはバラバラなセットよりもはるかに大きく、53,306 のトレーニング服と 10,000 のテスト服を含む一方、バラバラなセットは、16,995 のトレーニング服と 15,145 のテスト服が含まれます。

Polyvore Outfit データセットの FITB タスクの各質問での誤った選択は、正しい選択と同じカテゴリを持つアイテムからサンプリングされますが、Maryland Polyvore [5]の

FITB タスクは、カテゴリの制約なしにランダムにサンプリングされます。 Polyvore Outfit データセットは、きめ細かいアイテムタイプの注釈も提供します。

服の互換性を予測するためのタスクは、服のファッションアイテムのセットの互換性を予測することです。 標準メトリック AUC [5]を使用して、互換性予測のパフォーマンスを評価します。これは，受信者動作特性曲線の下の面積を測定しています．FITB の場合、衣装のアイテムのサブセットと候補アイテムのセット（4 つのアイテム、1 つのポジティブと 3 つのネガティブ）が与えられた場合、タスクは最も互換性のある候補を選択することです。 性能は全体的な精度に基づいて評価されます[5]。

服装補完アイテム検索：現在のデータセットにはグラウンドトゥルース検索アノテーションはありません。 この目的のために、Polyvore Outfit データセットに基づいて新しいデータセットを作成しました。 FITB テストセットの衣装をクエリ衣装として使用し、ポジティブイメージのランク（recall @ top k）を測定することにより、アルゴリズムの有効性を評価します。 これは定量的な評価指標を提供するため、結果は再現性があり、さまざまなアルゴリズム間で比較できます。

ポジティブイメージのランクは、システムの有用性を完全に示すものではありません。データベースにはクエリの衣装を補完する多くの項目が含まれている可能性があり、その一部は実際、人間の専門家によって次のように判断される可能性があります。 ポジティブよりも衣装によく一致しますが、グラウンドトゥルースとしてアノテーションが付けられる画像は 1 つだけです（検索リスト全体にアノテーションを付けることは難しいため、ポジティブ画像のランクを評価指標として使用します）。 ポジティブ画像と同様のスタイルの画像はランキングが進むため、より良い検索アルゴリズムではより高い想起が得られるため、相対ランクはさまざまなアルゴリズムの比較に役立ちます。

画像のランキング結果は、ポジティブ画像と同じカテゴリに制限されます。 テストセットには検索実験に十分な細かなカテゴリの画像がないため、トレーニング画像をディストラクタとして含めてテスト画像を補強し、3000 を超える画像があるきめの細かいカテゴリを選択します。 3000 を超える画像があるカテゴリには 3000 の画像のみを使用するため、すべてのカテゴリの画像の数は同じです。 検索実験にはバラバラでないセットには 27/153 のカテゴリを、バラバラなセットに 16/153 のカテゴリをトータルで選択します。

実装の詳細

衣装ランキングの損失を使用して、カテゴリベースの部分空間アテンションネットワークをトレーニングします。 ResNet18 [6]をバックボーン CNN モデルとして使用し、公正な比較のために最先端の方法[16、15]と同様にサイズ 64 を埋め込みます。これは Imagenet で事前にトレーニングされています。 [15]のように、部分空間の数を 5 に設定します。彼らの元の論文は、Polyvore Outfit データセットの部分空間番号のアブレーション研究を提供しています。 [13]と同様のオンラインマイニング方法でランキング損失を計算するた

め、各画像の部分空間埋め込みは 1 回だけ計算され、異なるペアワイズ距離に再利用されます。具体的には、トレーニングトリプルが与えられた場合、最初にすべての画像を CNN およびカテゴリベースの部分空間アテンションネットワークに渡して、部分空間の埋め込みを抽出します。アイテムの各ペアについて、2 つのカテゴリベクトルを使用して部分空間のアテンションの重みを計算し、それらに事前に計算された部分空間の埋め込みを掛けて、最終的な埋め込みを取得します。各衣装にはグラウンドトゥルースのネガティブ画像がないため、[16、15]と同様に、ポジティブ画像と同じカテゴリのネガティブ画像のセットをランダムにサンプリングします。セミハードネガ画像を選択して、衣装のランキングの損失をトレーニングします。ミニバッチサイズ 96 でネットワークをトレーニングし、ADAM を使用して最適化します。ランキング損失のマージンを 0.3 に設定し、初期学習率を$5e^{-5}$に設定しました。学習率をゼロに直線的に減少させる学習率スケジュールを採用しますが、初期学習率がすでに小さいため、ウォームアップ率をゼロに設定します（[3]などを参照)。

ベースライン

私たちの方法を、互換性予測、穴埋め（FITB)、および服装の補完的なアイテム検索タスクに関するいくつかのベースラインアプローチと比較します。 服装補完アイテム検索の既存のアプローチがないため、検索用に既存の互換性予測方法[16、15]を変更します。

Siamese-Net [16]：ResNet18 を使用して単一のスペースへの埋め込みを学習する Veit ら[18]のアプローチ。

タイプ対応[16]：著者からの最新のコードを使用し、Polyvore Outfit データセットでモデルを再トレーニングします。これは、このペーパーで最初に報告されたものよりも優れたパフォーマンスを示しています。 インデックス付けでは、各画像を 66 のタイプの部分空間に投影して、特徴を生成します。 テスト中、衣装の各テストアイテムについて、(画像のペアからの）タイプを使用して、特徴インデックスを取得し、そのスペース内の画像をランク付けします。 平均融合は、さまざまなアイテムのスコアを融合するために使用されます。

SCE-Net [15]：著者のコードを使用して、Polyvore Outfit データセットでモデルを再トレーニングします。 彼らの元のモデルは画像検索を実行できないため、検索実験のベースラインとして彼らの平均モデルを比較します。 平均モデルは、類似性条件マスクからの出力を平均することによって特徴を計算します。 平均融合は、さまざまなアイテムのスコアを融合するために使用されます。

Cucurull ら[2]は、ファッションの互換性予測にグラフ畳み込みネットワーク（GCN）を利用していることに注意してください。 ただし、彼らの方法では、互換性予測に分類モデルを使用します。これは、検索タスクには直接適用できません。

		Polyvore Outfits-D		Polyvore Outfits	
Method	Feature	FITB Accuracy	Compat. AUC	FITB Accuracy	Compat. AUC
Siamese-Net [16]	ResNet18	51.80	0.81	52.90	0.81
Type-aware [16]	ResNet18 + Text	55.65	0.84	57.83	0.87
SCE-Net average [15]	ResNet18 + Text	53.67	0.82	59.07	0.88
CSA-Net + outfit ranking loss (ours)	ResNet18	**59.26**	**0.87**	**63.73**	**0.91**

Table 1. Comparison of different methods on the Polyvore-Outfit dataset (where -D denotes for disjoint set). We report the results of our method and state-of-the-art retrieval methods: Type-aware [16] and SCE-Net average [15] (Note that the original SCE-Net reports 61.6 FITB accuracy and 0.91 Compat. AUC on the non-disjoint set (Polyvore Outfits), which requires pairs of input images and is not designed for retrieval). All the methods use ResNet18 and embedding size 64 for fair comparison. Note that our method does not use text feature. Our category-based subspace attention network and outfit ranking loss shows superior performance compared to the baseline methods.

表 1.Polyvore-Outfit データセットのさまざまな方法の比較（-D はバラバラなセットを示します)。私たちの方法と最先端の検索方法の結果を報告します：タイプ認識[16]と SCE-Net 平均[15]（元の SCE-Net はバラバラでないセット（Polyvore Outfits）において、61.6FITB 精度と 0.91 互換性 AUC を報告していることに注意してください。これは入力画像のペアを必要とし、検索用には設計されていません)。すべてのメソッドは、公正な比較のために ResNet18 と埋め込みサイズ 64 を使用します。私たちの方法はテキスト機能を使用しないことに注意してください。私たちのカテゴリーベースの部分空間アテンションネットワークと服装ランキングの損失は、ベースラインの方法と比較して優れたパフォーマンスを示しています。

互換性と FITB 実験

私たちのメソッドは検索用に設計されていますが、空欄を埋める（FITB）タスクと装備の互換性タスクに関する結果も報告します（表 1 を参照)。 SCE-Net 平均[15]の場合、再トレーニングされたモデルは、元のバラバラなセットのパフォーマンスを報告しないため、私たちは自分の実行に基づいてパフォーマンスを報告します。また、タイプ認識方式のより優れたバージョン[16]と比較すると、再トレーニングされたモデルは、元のモデルよりも優れたパフォーマンスを実現します。

SCE-Net 平均モデルよりも私たちのモデルを使用すると顕著な改善が見られ、FITB が 4〜5%改善され、服装の互換性が 3〜5%改善されました。 特に、私たちの結果（63.73FITB 精度とバラバラでない集合で 0.91AUC）は、元の SCE-Net よりもさらに優れたパフォーマンスを達成しています。これは、このデータセットで最先端のパフォーマンス（61.6FITB 精度と 0.91AUC）を取得します。また、私たちの方法では、ベースライン方法としてテキスト機能を使用していません[16、15]。この方法では、テキストを使用して視覚とテキストの共同埋め込みを学習します。 テキスト機能（BERT [3]など）を利用して、パフォーマンスをさらに向上させることができます。 おそらくトレーニングセットがバラバラでないセットよりもはるかに小さく、サブスペースの埋め込みを学習するにはより多くのトレーニングサンプルが必要なため、すべての方法でバラバラなセットのパフォーマンスが低いことがわかります。

Loss function	FITB Accuracy	Compat. AUC
Triplet loss	56.17 / 60.91	0.85 / 0.90
Outfit ranking loss	**59.26 / 63.73**	**0.87 / 0.91**

Table 2. Comparison of triplet loss and outfit ranking loss of our model in disjoint / non-disjoint set.

表 2.バラバラ/非バラバラなセットでのモデルのトリプレット損失と衣装ランキング損失の比較。

トリプレット損失と服装ランキング損失：トリプレット損失と服装ランキング損失のパフォーマンスを比較します（表 2 を参照)。 モデルが元のトリプレット損失と衣装ランキング損失でトレーニングされているさまざまな設定で評価します。 結果は、私たちの服装ランキングの損失が、FITB タスクで約 3%のトリプレット損失のパフォーマンスをさらに改善することを示しています。 服装ランキングの損失は、単一のアイテムのペアではなく、服装全体のアイテムの関係を考慮します。これにより、新しいアイテムが服装のすべてのアイテムと互換性を持つようになります。

Aggregation function	FITB Accuracy	Compat. AUC
Average	56.19 / 60	0.84 / 0.89
Min	**59.26 / 63.73**	**0.87 / 0.91**

Table 3. Comparison of different aggregation functions of our outfit ranking loss in disjoint / non-disjoint set.

表 3.バラバラ/非バラバラなセットでのアウトフィットランキング損失のさまざまな集計関数の比較。

最小集計関数と平均集計関数：服装ランキングの損失でさまざまな集計関数を比較します（表 3 を参照)。 最小集計関数は、ランキング損失トレーニングにより多くのハードネガティブな例を選択するため、平均関数よりも優れたパフォーマンスを達成することがわかります。

順序の反転：また、フレームワークをトレーニングするときに入力カテゴリの順序を反転する、服のランキング損失での順序の反転を実験しました。 ただし、おそらくネットワークが十分なペアを確認したため、学習された特徴の埋め込みがさまざまな順序に一般化されたため、フリッピングによる改善はあまり見られませんでした。

検索実験

	Polyvore Outfits-D			Polyvore Outfits		
Method	Top 10	Top 30	Top 50	Top 10	Top 30	Top 50
Type-aware [16]	3.66%	8.26%	11.98%	3.50%	8.56%	12.66%
SCE-Net average [15]	4.41%	9.85%	13.87%	5.10%	11.20%	15.93%
CSA-Net + outfit ranking loss (ours)	**5.93%**	**12.31%**	**17.85%**	**8.27%**	**15.67%**	**20.91%**

Table 4. Comparison of different methods in complementary outfit item retrieval task (recall@top k). Our method shows a consistent improvement over the baseline approaches for different k.

表 4.補完的な衣装アイテム検索タスク（recall @ top k）のさまざまな方法の比較。 私たちの方法は、さまざまな k のベースラインアプローチに対して一貫した改善を示しています。

表 4 は、検索結果を示しています。ここでは、27/153 および 16/153 の細かいカテゴリでそれぞれ recall @ top k（k = 10、30、50）を使用して、非バラバラなセットとバラバラなセットのパフォーマンスを評価しています。 きめの細かい各カテゴリには同じ数の画像（3000 など）があるため、画像が多い特定のカテゴリに想起が偏ることはありません。報告された数値は、すべてのカテゴリの平均想起の平均です。 オリジナルの SCE-Net [15]は、セクション 3.3 で説明したように検索を実行できないため、検索実験の平均モデルと比較します。

私たちの方法は、さまざまな k のベースラインアプローチに対して一貫した改善を示しています。 バラバラな集合と非バラバラな集合の両方で、recall @ top 10、top 30、top 50 の SCE-Net 平均モデルに比べてそれぞれ約 2〜5%の改善が得られます。 セクション 4.1 で説明したように、テストカタログには互換性のある項目が多数ある可能性があるため、メトリックはシステムの実際のパフォーマンスを反映していない可能性があります。フレームワークの検索結果の例を図 6 に示します。

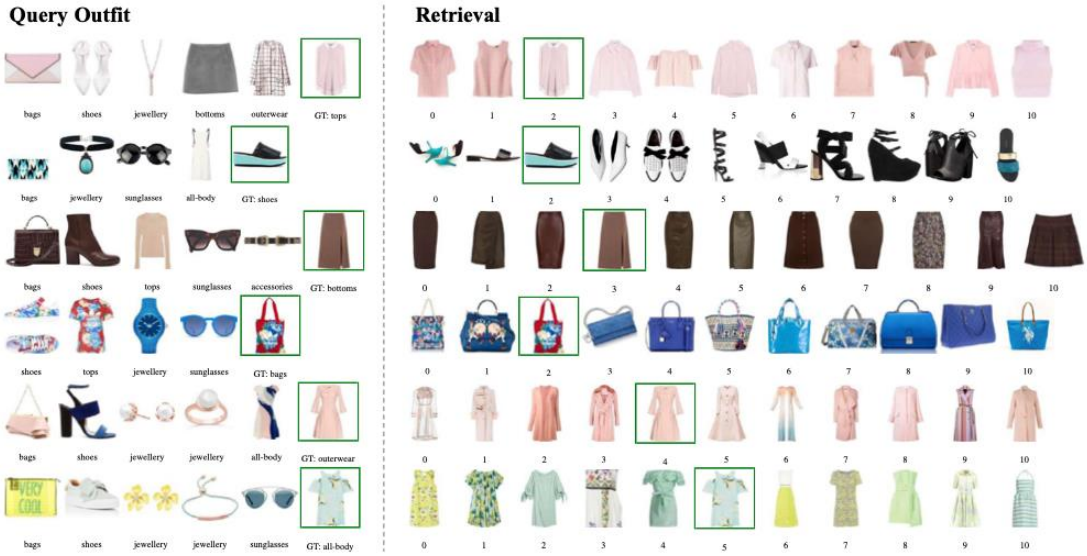

Figure 5. Example retrieval results of our method. Left column: query (incomplete) outfits and (missing) target items. Right column: top 10 retrieval results over 3000 images, where ground truth items are in green boxes. Our approach retrieves a list of compatible items that match to the style of the entire outfit.

Figure 6. Some failure cases of our method. The failure cases are mainly caused by similar colors, textures and styles to the target item (row 1 and row 2), and multiple compatible items (row 3 and row 4) that also match to the query outfit. Note that these items may not be necessarily incompatible with the query outfits, see Section 4.1 for the discussions about the dataset limitations.

図 5.メソッドの検索結果の例。 左の列：(不完全な）衣装と（不足している）ターゲットアイテムをクエリします。 右の列：3000 を超える画像の上位 10 件の検索結果。グラウンドトゥルースアイテムは緑色のボックスで囲まれています。 私たちのアプローチは、服全体のスタイルに一致する互換性のあるアイテムのリストを取得します。

図 6.メソッドのいくつかの失敗ケース。失敗ケースは主に、ターゲットアイテム（行 1 と行 2）やクエリの衣装に一致する複数の互換性のあるアイテム（行 3 と行 4）に類似した色、テクスチャ、スタイルが原因で発生します。これらのアイテムは必ずしもクエリの衣装と互換性がないということではないことに注意してください。データセットの制限に関する説明については、セクション 4.1 を参照してください。

結論

衣装補完アイテム検索のための新しいアプローチを提示しました。 既存のアプローチのインデックス作成の問題を克服するために、単一の画像と 2 つのカテゴリベクトルを取得してアテンションメカニズムを形成するカテゴリベースの部分空間アテンションネットワークを設計しました。これにより、フレームワークのインデックス作成と取得がスケーラブルになります。 さらに、服全体のアイテムの関係を考慮した服のランキング損失を導入し、服の互換性を向上させます。 実験結果は、私たちのモデルが、服装の互換性、FITB、および検索タスクにおいて、いくつかの最先端のアプローチよりも優れていることを示しています。